Time-Space Tradeoffs, Multiparty Communication Complexity, and Nearest-Neighbor Problems

Paul Beame*
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
beame@cs.washington.edu

Erik Vee*
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
env@cs.washington.edu

ABSTRACT

We extend recent techniques for time-space tradeoff lower bounds using multiparty communication complexity ideas. Using these arguments, for inputs from large domains we prove larger tradeoff lower bounds than previously known for general branching programs, yielding time lower bounds of the form $T = \Omega(n \log^2 n)$ when space $S = n^{1-\epsilon}$, up from $T = \Omega(n \log n)$ for the best previous results. We also prove the first unrestricted separation of the power of general and oblivious branching programs by proving that $1GAP$, which is trivial on general branching programs, has a time-space tradeoff of the form $T = \Omega(n \log^2(n/S))$ on oblivious branching programs.

Finally, using time-space tradeoffs for branching programs, we improve the lower bounds on query time of data structures for nearest neighbor problems in d dimensions from $\Omega(d/\log n)$, proved in the cell-probe model [8, 5], to $\Omega(d)$ or $\Omega(d\sqrt{\log d/\log\log d})$ or even $\Omega(d \log d)$ (depending on the metric space involved) in slightly less general but more reasonable data structure models.

1. INTRODUCTION

Recently, the first non-trivial time-space tradeoff lower bounds have been shown for decision problems in P [7, 1, 2, 6]. These results are the culmination of almost two decades of analysis of branching programs, natural generalizations of decision trees to directed graphs that provide elegant models of both non-uniform time T and space S simultaneously. The key ideas in these recent papers extend notions from 2-party communication complexity previously used in the study of restricted branching programs, such as *oblivious* branching programs [3] or *read-k* branching programs [9], to general branching programs.

In this paper we extend and improve these results in several directions. We develop a new lower bound criterion, based on extending 2-party communication complexity ideas to multiparty communication complexity, that applies to general branching programs. We show that if a function is not constant on large *embedded cylinder intersections* then it has a large time-space tradeoff lower bound. This generalizes a lower bound technique from [7, 6] based on analyzing functions on large embedded rectangles. Applying this criterion to an explicit Boolean function based on a multilinear form over $\mathbb{F}_{2^s}$ for suitable s, we show lower bounds that yield $T = \Omega(n \log^2 n)$ when $S \leq n^{1-\epsilon} \log |D|$ for large input domain D. This improves the best lower bounds for general branching programs and matches the best lower bounds known even for oblivious branching programs.

As a warm up for our argument we give an alternative, conceptually simple proof, based on the ideas in the recent lower bounds for general branching programs, of the relationship between multiparty communication complexity and time-space tradeoffs for oblivious branching programs shown by Babai, Nisan, and Szegedy [4].

Using this we obtain time-space tradeoff lower bounds of the form $T = \Omega(n \log^2(n/S))$ for $1GAP$ on oblivious branching programs. Since $1GAP$, the canonical complete problem for L, has a trivial general branching program of time n and width n (and therefore space $O(\log n)$), this provides the first separation between general and oblivious branching program computation. (Separations have previously been shown between oblivious read-once branching programs and read-once branching programs, but not in the general case.)

Finally, extending the observation of Miltersen, Nisan, Safra, and Wigderson [13] that small space branching programs are natural choices for static data structures and, thus, that query lower bounds are related to time-space tradeoff lower bounds, we develop lower bounds for nearest-neighbor problems in a variety of metric spaces (U^d, Δ). In the nearest neighbor problem we are given a database of n points in U^d and a query point $x \in U^d$ and must determine the closest element to x in the database. We analyze the λ-near neighbor problem, a decision version of this problem, that as observed in [8], is at least as easy as the nearest-neighbor problem.

The obvious algorithms for either problem are simply to store the elements of the database themselves and compare the query x in turn to each element of the database, using $O(dn)$ words and $O(dn)$ query time, or, if $|U|$ is small, to pre-compute the answers to all possible queries using $O(|U|^d)$ space and constant query time. More complex algorithms achieve query time polynomial in d and $\log n$ in spaces with larger $|U|$ but they still require $n^{\Omega(d)}$ storage in the worst case (see [8] for an overview of these algorithms). In general, for large dimensions d, no better algorithms are known.

It is generally conjectured that there is no nearest neighbor data structure having $(nd)^{O(1)}$ memory cells of size $(\log n)^{O(1)}$ and query-time $(d \log n)^{O(1)}$. However, even small lower bounds are of interest since in certain applications, such as semantic indexing, d is the number of terms, on the order of tens of thousands, and n is the number of texts, on the order of millions to billions.

* Research supported by NSF grants CCR-9800124 and CCR-0098066.

STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.

Borodin, Ostrovsky, and Rabani [8], show that even for computation in Yao's strong cell-probe model with cells that may contain up to $(d \log n)^{O(1)}$ bits, solving the λ-near neighbor problem on the Hamming cube $\{0,1\}^d$ requires query time $\Omega(d/\log n)$ if the number of words is polynomial in dn. Barkol and Rabani [5], have even extended these results to randomized computation but the results are still far from the upper bounds.

While lower bounds in the cell probe model apply to the broadest possible class of data structure algorithms, it is not reasonable to expect that data structure algorithms can implement all the features of the cell probe model which includes, at zero cost, the ability to have a completely different access algorithm for each instantiation of the query and the ability to remember the entire history of the computation given the query.

We consider slightly more restricted data structure algorithms with word size $O(\log n)$ that are allowed to make decisions based on one coordinate $x_i \in U$ of the query at a time (at unit cost) and which use only $(dn)^{o(1)}$ bits of extra space during their execution. In this model, for certain d-dimensional metric spaces, we prove query time lower bounds of the form $\Omega(d \log d)$, or $\Omega(d\sqrt{\log d/\log\log d})$, depending on the space. These results follow easily from time-space tradeoff lower bounds for general branching programs.

The set U in each of the metric spaces in these lower bounds is of size d^c for some constant c and it would be interesting if we could extend these bounds to the Hamming space $\{0,1\}^d$ considered in [8, 5]. We are able to do this in the special case that the data structure corresponds to an oblivious branching program; that is, the bits of the query x are accessed in the same order, no matter what the values of these bits are. In this case we obtain a query time lower bound of the form $\Omega(d \log d)$. This lower bound is based on a careful combinatorial construction of a suitable database.

2. DEFINITIONS

Throughout this paper, D will denote a finite set and n a positive integer. We use $[n]$ to denote the set $\{1,\ldots,n\}$ and $[n]-1$ to denote $\{0,\ldots n-1\}$. We view the input space, D^N, as the set of maps from N to D; we normally take N to be $[n]$, and write D^n for $D^{[n]}$. If $A \subset N$, a point $\sigma \in D^A$ is a *partial input on* A. For a partial input σ, $vars(\sigma)$ denotes the set of indices A for which σ makes an assignment, and $unset(\sigma)$ denotes the set $N - vars(\sigma)$. If σ and π are partial inputs with $vars(\sigma) \cap vars(\pi) = \emptyset$, then $\sigma\pi$ denotes the partial input on $vars(\sigma) \cup vars(\pi)$ that agrees with σ and $vars(\sigma)$ and with π on $vars(\pi)$.

For $x \in D^N$ and $A \subset N$, the *projection* x_A *of* x *onto* A is the partial input on A that agrees with x. For $S \subset D^N$, $S_A = \{x_A : x \in S\}$. If f is a function with domain D^N and ρ is a partial input, the *restriction* of f by ρ, denoted f^ρ, is the function with domain $D^{unset(\rho)}$ defined by the rule $f^\rho(\sigma) = f(\sigma\rho)$ for $\sigma \in D^{unset(\rho)}$.

We adopt the usual definitions of deterministic branching programs. A D-way branching program is a directed acyclic graph satisfying the following: There is a unique vertex with in-degree 0, called the *start node*. The sink nodes are labeled with an output value. Every non-sink node is labeled with a variable name. The out-degree of every non-sink node, v, is precisely $|D|$ (we allow multi-edges), and every value from D is assigned to precisely one out-edge from v.

A branching program computes a function in the same way a decision tree does. *Time*, T, is the length of the longest consistently labeled path from the start node to a sink node. The *size* of a branching program is the number of its nodes. *Space*, S, is the base 2 logarithm of size. Any lower bound proven for a branching program also implies a lower bound for other computational models, such as Turing machines and Random Access Machines.

A *leveled branching program* is a branching program in which the underlying graph is leveled. By a result of Pippenger [14], making a branching program leveled does not change T and adds at most $\log T$ to S. An *oblivious branching program* is a leveled branching program in which all the nodes on each level are labeled with the same variable. Call the sequence of variables reached at each level the *query sequence* of the oblivious branching program.

Multiparty communication complexity, introduced by Chandra, Furst and Lipton [10] in order to study oblivious branching programs, is an extension of the usual 2-party communication complexity. Suppose that p parties, each with unlimited computational power, wish to exchange information to compute the value of $f : D^n \to \{0,1\}$, whose input has been divided according $\mathcal{P} = \{P_1,\ldots,P_p\}$, a p-partition of the n variables $\{x_1,\ldots,x_n\}$. The i-th party receives the value of every x_j *except* for those in P_i. The parties exchange information by writing bits one at a time on a common blackboard according to a protocol which specifies, based on the bits on the blackboard, who is to write next. The *multiparty communication complexity of* f with respect to the fixed partition $\mathcal{P}$, $C_{\mathcal{P}}(f)$, is the minimum number of bits required to be written. We also define the *best partition p-party communication complexity of* f, $C_p^{best}(f)$, to be the minimum fixed-partition communication complexity of f, taken over all p-partitions of the inputs into *equal size* sets.

Several lower bounds on the multiparty communication complexity of Boolean functions have been shown in [10, 4, 11, 12, 15] in the fixed partition model. The lower bound techniques for multiparty communication complexity developed following [4] are an extension of the lower bound techniques for 2-party communication complexity which rely on analyzing the properties of functions on large combinatorial rectangles. In the case of p-party communication complexity, lower bounds are proven by analyzing properties of functions on large *cylinder intersections*, which are sets of the form $C_1 \cap C_2 \cap \cdots \cap C_p$ where each $C_i = C'_i \times D^{P_i} \subseteq D^N$ is a *cylinder* in that it only depends on the variables read by party i.

3. MULTIPARTY COMMUNICATION COMPLEXITY AND BRANCHING PROGRAMS

3.1 Oblivious Branching Programs

There is a close relationship between multiparty communication complexity and oblivious branching program complexity.

DEFINITION 3.1. *Given a sequence s of values from a finite set N and a partition $\mathcal{P}$ of $N' \subseteq N$, the number of* alternations *of s with respect to $\mathcal{P}$ is the minimal r such that s can be written as $s = s_1 s_2 \cdots s_r$ and each s_i contains no elements from at least one class in $\mathcal{P}$. Each s_i is called an* alternation *of s with respect to $\mathcal{P}$.*

PROPOSITION 3.1. *[10, 3, 4] Let $f : D^N \to \{0,1\}$, let ρ be a partial assignment to the variables of f, and let $\mathcal{P}$ be a partition of $unset(\rho)$. If there is an oblivious branching program $\mathcal{B}$ computing f that has width W and whose query sequence has at most r alternations with respect to $\mathcal{P}$, then $(r-1)\log W + 1 \geq C_{\mathcal{P}}(f^\rho)$.*

PROOF. Associate one party j with each $P_j \in \mathcal{P}$. Each party has access to ρ and $\mathcal{B}$. Suppose that the query sequence of $\mathcal{B}$ is $s_1 \ldots s_r$ where each s_i is an alternation with respect to $\mathcal{P}$ and does not contain any element from the class $P_{j_i} \in \mathcal{P}$. The parties all follow the computation beginning with the start node of $\mathcal{B}$. For $i = 1, \ldots, r$, party j_i follows the path in $\mathcal{B}$ until the end of the query sequence s_i and writes the name of the node of $\mathcal{B}$ reached at the end of s_i on the blackboard. This is possible since any query to $vars(\rho)$ can be answered by any party and any query to $unset(\rho)$ avoids P_{j_i} and so can be answered by party j_i. For $i < r$ the name of the node requires at most $\log W$ bits since $\mathcal{B}$ has width at most W. For $i = r$ this requires only 1 bit to yield the value of f^ρ. $\square$

The following lemma, an extension of one found in [7, 6], is an alternative to the approach based on generalized meanders in [4].

LEMMA 3.2. *Let s be a sequence of kn elements from $[n]$. Divide s into r equal segments $s_1 \ldots s_r$ each of length kn/r. Independently assign each s_ℓ to one of p sets $B_1, \ldots, B_p$ uniformly at random with $\Pr[s_\ell \in B_j] = 1/p$. Define a random variable ν_j to be the number of elements of $[n]$ not appearing in any segment in B_j and let $\mu = E(\nu_j)$. Then $\mu > n4^{-k/p}$ and $Pr[|\nu_j - \mu| > \frac{1}{2}\mu] < 4^{k/p+1}k^2/r$.*

PROOF. We first calculate the expectation. For any $i \in [n]$, the probability that element i never appears in any segment in B_j is $(1 - 1/p)^{t(i)}$, where $t(i)$ is the number of segments in which i appears. Since $p \geq 2$, this probability is at least $4^{-t(i)/p}$. Thus

$$E(\nu_j) \geq \sum_{i=1}^{n} 4^{-t(i)/p} \geq n4^{-\sum_{i=1}^{n} t(i)/(pn)} = n4^{-k/p}.$$

The second inequality follows from the arithmetic-geometric mean inequality, and the last equality follows from the fact that the sequence s is of length kn, hence $\sum_{i=1}^{n} t(i) = kn$.

We next bound the variance. Let G_i be the event that variable $i \in [n] - \bigcup_{s_\ell \in B_j} s_\ell$ where here we identify each segment with the set of elements it contains. Further, for $1 \leq i, i' \leq n$, write $i \sim i'$ if i and i' appear in the same segment at least once. Then

$$Var(\nu_j) = \sum_{i,i'} (\Pr[G_i \wedge G_{i'}] - \Pr[G_i] \cdot \Pr[G_{i'}])$$

If i and i' never appear in a segment together, then the events G_i and $G_{i'}$ are independent, and the term $(\Pr[G_i \wedge G_{i'}] - \Pr[G_i] \cdot \Pr[G_{i'}])$ is 0. If i and i' do appear together in at least one segment, we upper bound the corresponding term by $\Pr[G_i] = (1-1/p)^{t(i)}$. Since each segment contains at most kn/r elements of $[n]$, the number of i' such that $i \sim i'$ is bounded above by $t(i)kn/r$. So

$$\begin{aligned} Var(\nu_j) &= \sum_{i=1}^{n} \sum_{i \sim i'} \Pr[G_i] \leq \frac{k}{r} n \sum_{i=1}^{n} t(i)(1 - 1/p)^{t(i)} \\ &\leq \frac{k}{r} \sum_{i=1}^{n} t(i) \sum_{i'=1}^{n} (1 - 1/p)^{t(i)} \leq k^2 n\mu/r. \end{aligned}$$

where the third inequality follows since $t(i)$ and $(1 - 1/p)^{t(i)}$ are positive and anti-correlated. Applying Chebyshev's inequality we have

$$Pr[|\nu_j - \mu| > \frac{1}{2}\mu] \leq \frac{Var(\nu_j)}{(\frac{1}{2}\mu)^2} < 4^{k/p+1}k^2/r$$

as required. $\square$

Combining Lemma 3.2 with Proposition 3.1, we have

THEOREM 3.3. *Let $f : D^n \to \{0, 1\}$ and let $\mathcal{B}$ be an oblivious branching program that computes f in time $T = kn$ and width W. Then there is a subset of the variables $N' \subset [n]$ of size $n - \frac{1}{2}4^{-k/p}n$ such that for any partial input ρ on N', $4^{k/p+1}k^2p \cdot \log W \geq C_p^{best}(f^\rho)$.*

PROOF. Apply Lemma 3.2 with $r = 4^{k/p+1}k^2p$ to the query sequence s of $\mathcal{B}$. This divides $\mathcal{B}$ into r blocks of height kn/r corresponding to the segments of s. Given a set of such blocks B, let $unseen(B)$ be the set of variables not queried at any level in B. By Lemma 3.2, for $j = 1, \ldots, p$, the probability that a random assignment of these blocks to sets $B_1, \ldots, B_p$ has $unseen(B_j) < 4^{-k/p}n/2$ is less than $4^{k/p+1}k^2/r \leq 1/p$. Therefore by the probabilistic method there exists an assignment of blocks to the sets such that all p sets B_j have $unseen(B_j) \geq 4^{-k/p}n/2$.

Under this assignment, the $unseen(B_j)$ do not necessarily form a partition, since the sets may overlap. However, it is straightforward to choose p *disjoint* sets, $U_1, U_2, \ldots, U_p$, with the property that $U_j \subset unseen(B_j)$ and $|U_j| = \frac{1}{2p}4^{-k/p}n$ for all parties, j. Now, set $N' = [n] - \bigcup_{i=1}^{n} U_i$. Then the U_i form an equal-size p-partition, $\mathcal{P}$ of $[n] - N'$. Also, by construction, the query sequence of $\mathcal{B}$ has at most r alternations with respect to $\mathcal{P}$.

Let ρ be any partial assignment on N'. Proposition 3.1 immediately yields $4^{k/p+1}k^2p \cdot \log W \geq C_{\mathcal{P}}(f^\rho) \geq C_p^{best}(f^\rho)$ as required. $\square$

3.2 Lower Bounds for *1GAP*

Recall that *1GAP* is simply the problem of st-connectivity for directed graphs that have out-degree one. More formally, for $x \in [n]^n$, we define $1GAP_n : [n]^n \to \{0, 1\}$ by $1GAP_n(x) = 1$ if and only if there is a sequence of indices, $i_1, i_2, \ldots, i_\ell$ such that $i_1 = 1$, $i_\ell = n$ and for all $j = 1, \ldots, \ell - 1$, we have $x_{i_j} = i_{j+1}$. (Notice that under this encoding, the vertices s and t correspond to indices 1 and n, respectively, and that the value of x_n does not affect the value of $1GAP_n(x)$.)

To prove lower bounds for *1GAP* we will use p-party communication complexity lower bounds previously shown for *generalized inner product* (*GIP*), which is given by $GIP_{m,p}(z_1, z_2, \ldots z_p) = \bigoplus_{j=1}^{m} \bigwedge_{i=1}^{p} z_{i,j}$ where each $z_i \in \{0, 1\}^m$ and $z_{i,j}$ is the j-th bit of z_i.

PROPOSITION 3.4. *[4] Let $\mathcal{P}$ be the 'uniform' partition in which each $\{z_{i,1}, \ldots, z_{i,m}\}$ is a single class. Then $C_{\mathcal{P}}(GIP_{m,p})$ = $\Omega(m/4^p)$.*

THEOREM 3.5. *Let $N' \subseteq [n]$ with $|N'| = n - (2m + 1)p$. Then there is a partial assignment, $\rho \in [n]^{N'}$, such that $C_p^{best}(1GAP_n^\rho) = \Omega(m/4^p)$.*

PROOF. We give a reduction from $GIP_{m,p}$ based on the simple ordered binary decision diagram (OBDD) (a variant of an oblivious read-once branching program in which edges may skip over levels, see e.g. [16]), $\mathcal{B}$, of width 2 with $2pm + 2$ nodes that computes $GIP_{m,p}$.

Set ρ to be the partial assignment that puts a self-loop at all vertices corresponding to indices in $N' - \{1\}$. If $1 \notin N'$, let node $s = 1 \in [n] - N'$; otherwise let s be some node in $[n] - N'$ and set $\rho(1) = s$. Given a best-partition p-party communication complexity protocol for $1GAP_n^\rho$, let $\mathcal{P} = \{P_1, \ldots, P_p\}$ be the partition of $[n] - N'$ into p equal-sized classes used by this protocol and assume w.l.o.g. that $s \in P_1$. We embed the nodes of $\mathcal{B}$ in $([n] - N') \cup \{n\}$ by mapping the start node of $\mathcal{B}$ to s, mapping the remaining nodes in $\mathcal{B}$ that read each z_j to the nodes of P_j, and mapping the 1-sink node of $\mathcal{B}$ to n and the 0-sink to 1. Given an input z, we fix the out-edges of the embedded $\mathcal{B}$ to obtain a graph G_z for which $1GAP^\rho(G_z) = GIP_{m,p}(z)$. Since each party can construct the part of G_z it needs, we have a p-party communication complexity protocol solving $GIP_{m,p}$ under the uniform partition $\mathcal{P}'$. Hence, $C_p^{best}(1GAP_n^\rho) \geq C_{\mathcal{P}'}(GIP_{m,p}) = \Omega(m/4^p)$ by Proposition 3.4. $\square$

THEOREM 3.6. *If an oblivious branching program with time T and space S solves $1GAP_n$, then $T = \Omega(n \log^2(n/S))$.*

PROOF. Let $k = T/n$. From Theorem 3.3, for any $p \geq 2$, there is a N' of size $n - \frac{1}{2}4^{-k/p}n$ such that, for any partial assignment ρ on N', $4^{k/p+1}k^2 p \log W \geq C_p^{best}(1GAP_n^\rho)$. Further, from Theorem 3.5, we know that for the uniform partition $\mathcal{P}$, $C_p^{best}(1GAP_n^\rho) \geq C_\mathcal{P}(GIP_{m,p})$ for $m = 4^{-k/p}n/(4p)$. Combining this with Proposition 3.4, we see that $4^{k/p+1}k^2 p \log W \geq C_\mathcal{P}(GIP_{m,p}) \geq C'm/4^p = C'4^{-k/p-p-1}n/p$ for some constant $C' > 0$. Rearranging and using $p \geq \log p$, setting $p = \sqrt{k}$ in order to minimize $k/p + p$, using $S \geq \log W$, and taking logarithms we obtain $\log(n/S) \leq C''\sqrt{k}$ for some constant $C'' > 0$. Therefore $T = kn \geq Cn \log^2(n/S)$ for some $C > 0$ as required. $\square$

3.3 General Branching Programs

We say that a subset $E \subseteq D^N$ is an *embedded p-cylinder intersection* iff there exist p disjoint subsets $A_1, \ldots, A_p \subset N$, a partial assignment ρ to $N - \bigcup_{j=1}^p A_j$, and sets $C_j \subseteq D^{unset(\rho)-A_j}$ for $j = 1, \ldots, p$ such that $E = \bigcap_{j=1}^p (C_j \times D^{A_j} \times \rho)$. Following [6], we call the A_j the *feet* of E, ρ the *spine* of E, and the C_j the *legs* of E. $(\rho, A_1, \ldots, A_p)$ is called a *footprint* of E; it is *balanced* if $|A_1| = |A_2| = \ldots = |A_p|$, and *ordered* if $A_1 < A_2 < \ldots < A_p$, where $A < B$ for sets A and B if every element of A is smaller than every element of B. The *foot-size* $m(E)$ is $\min_j |A_j|$ and the *density* $\delta(E)$ is $|E|/|D|^{unset(\rho)}$. Embedded rectangles are embedded 2-cylinder intersections.

Here, we extend one of the approaches to obtaining time-space tradeoff lower bounds in [7, 6] from one based on embedded rectangles to one based on embedded p-cylinder intersections.

THEOREM 3.7. *Let $k, p, r \geq 2$ be integers such that $n \geq r \geq 4^{k/p+2}k^2 p$ and let $m \leq \lceil 4^{-k/p}n/(2p^2) \rceil$. Let $\mathcal{B}$ be a D-way branching program of length at most $(k-1)n$ and size at most 2^S and let $I \subseteq \mathcal{B}^{-1}(1)$. There is a set $\mathcal{E}$ of embedded p-cylinder intersections with balanced, ordered footprints whose union covers a subset $I' \subseteq I$ with $|I'| \geq |I|/2$ such that each $E \in \mathcal{E}$ satisfies $E \subseteq \mathcal{B}^{-1}(1)$, $m(E) = m$ and $\delta(E) \geq 2^{-2mp\log_2(n/m)-Sr-2}|I|/|D^n|$.*

Let $\mathcal{B}$ be a leveled branching program on D^n of length kn. Given $L \subseteq [kn] - 1$ and an input $x \in D^n$ define $\texttt{reads}_\mathcal{B}(x, L) \subseteq [n]$ to be the set of indices of variables queried in $\mathcal{B}$ on input x at levels in L and let $\texttt{unseen}_\mathcal{B}(x, L) = [n] - \texttt{reads}_\mathcal{B}(x, L)$.

Given a partition $\mathcal{L} = (L_1, \ldots, L_p)$ of $[kn]-1$ and an input $x \in D^n$, define $\texttt{nodes}_\mathcal{B}(x; L_1, \ldots, L_p)$ to be the sequence of nodes of $\mathcal{B}$ that x reaches at levels $\ell \in [kn-1]$ such that $\ell \in L_j$ for some $j \leq p$ for which $\ell - 1 \notin L_j$.

LEMMA 3.8. *Let $\mathcal{B}$ be a leveled branching program of length kn and $(L_1, \ldots, L_p)$ be a partition of $[kn] - 1$. Let $A_0, A_1, \ldots, A_p$ be a partition of $[n]$, $\sigma \in D^{A_0}$, and let $(v_1, \ldots, v_r)$ be a set of nodes of $\mathcal{B}$. Define $E \subseteq \mathcal{B}^{-1}(1)$ to be the set of all inputs x such that $\texttt{nodes}_\mathcal{B}(x; L_1, \ldots, L_p) = (v_1, \ldots, v_r)$, $A_j \subseteq \texttt{unseen}_\mathcal{B}(x, L_j)$ for all $j = 1, \ldots, p$, and $x_{A_0} = \sigma$. Then E is an embedded p-cylinder intersection with footprint $(\sigma, A_1, \ldots, A_p)$.*

PROOF. Let $C_1 = E_{[n]-A_0-A_1}, \ldots, C_p = E_{[n]-A_0-A_p}$ and let F be the embedded p-cylinder intersection defined by σ, $A_1 \ldots, A_p$ and $C_1, \ldots, C_p$. Clearly, $E \subseteq F$, and it suffices to show that $F \subseteq E$.

Let $z \in F$. By definition of F, for each $j = 1, \ldots, p$ there is a $y^j \in E$ such that $z_{[n]-A_0-A_j} = y^j_{[n]-A_0-A_j}$. Furthermore $z_{A_0} = y^j_{A_0} = \sigma$ so $z_{[n]-A_j} = y^j_{[n]-A_j}$ for $j = 1, \ldots, p$.

For any j, on input y^j, levels of $\mathcal{B}$ in L_j only access variables in $[n] - A_j$ and y^j and z agree on those variables, so the outcomes of queries on z at levels in L_j are the same those as on input y^j, although it is not immediate that they both follow the same path since they *a priori* might arrive at different nodes in these levels to begin with. However, since $z, y^1, \ldots, y^p$ all begin at the same start node in L_j we see that by induction on the length of the path that for each j, z follows precisely the same path in L_j as y^j. This implies that $\texttt{nodes}_\mathcal{B}(z; L_1, \ldots, L_p) = (v_1, \ldots, v_r)$ and that $\texttt{unseen}_\mathcal{B}(z, L_j) = \texttt{unseen}_\mathcal{B}(y^j, L_j) \supseteq A_j$ for $j = 1, \ldots, p$ and $\mathcal{B}(z) = 1$. Therefore $z \in E$ as required. $\square$

The follow proposition will be useful for satisfying the ordered footprint condition.

PROPOSITION 3.9. *Let $B'_1, \ldots, B'_p \subseteq [n]$ be disjoint sets. Then there exist sets $B_j \subseteq B'_j$ for each $j = 1, \ldots, p$ and a permutation $\pi : [p] \to [p]$ such that $B_{\pi(1)} < B_{\pi(2)} < \cdots < B_{\pi(p)}$ and $|B_j| \geq |B'_j|/p$ for each j.*

PROOF. We prove this by induction on p. The statement is clearly true for $p = 1$. Suppose it is true for $p - 1 > 0$. Let $i \in [n]$ be minimum such that there is some k with $|B'_k \cap [i]| \geq |B'_k|/p$. Let $B_k = B'_k \cap [i]$, $\pi(1) = k$, and define $B''_j = B'_j \setminus [i]$ for $j \neq k$. By construction, $|B'_j \setminus [i]| \geq (p-1)|B'_j|/p$ for $j \neq k$. We now apply the inductive hypothesis to the $p - 1$ sets B''_j for $j \neq k$ to define the remainder of π and sets $B_j \subset B''_j \subseteq B'_j$ with $|B_j| \geq |B''_j|/(p-1) \geq |B'_j|/p$ for $j \neq k$. Clearly $B_1, \ldots, B_p$ are disjoint. $\square$

PROOF OF THEOREM 3.7. Assume that r divides kn. By Lemma 3.2, if we choose $L_1, \ldots, L_p \in [kn] - 1$ by dividing the levels of $\mathcal{B}$ into r blocks of height kn/r and randomly, uniformly, and independently include each block in one of the L_j, then for any $x \in D^n$, for each $j \in [p]$, $\Pr[|\texttt{unseen}_\mathcal{B}(x, L_j)| < 4^{-k/p}n/2] < 4^{k/p+1}k^2/r \leq 1/(4p)$. Therefore, there exists a choice of this assignment such that for some $I'' \subseteq I$ with $|I''| \geq 3|I|/4$, for all inputs $x \in I''$, $|\texttt{unseen}_\mathcal{B}(x, L_j)| \geq 4^{-k/p}n/2$ for all $j = 1, \ldots, p$. Fix this choice.

For the choice of $L_1, \ldots, L_p$ and all x, there are $r' < r$ elements in $\texttt{nodes}_\mathcal{B}(x; L_1, \ldots, L_p)$, at most one each from levels ikn/r of $\mathcal{B}$, for $i = 1, \ldots, r - 1$. For any input $x \in I''$, we find disjoint $B'_1, \ldots, B'_p$ with each $|B'_j| = 4^{-k/p}n/(2p) = m'$, $B'_j \subseteq \texttt{unseen}_\mathcal{B}(x, L_j)$. We then apply Proposition 3.9 to find sets $B_j \in B'_j$ with $|B_j| = m \leq \lceil m'/p \rceil$ and a permutation $\pi : [p] \to [p]$ such that $B_{\pi(1)} < B_{\pi(2)} < \cdots < B_{\pi(p)}$.

Let $B_0 = [n] - B_1 \cdots - B_p$. Using the construction given by Lemma 3.8, let E be the embedded p-cylinder intersection containing x on which $\mathcal{B}$ outputs 1 defined by $B_0, B_1 \ldots, B_p$, node sequence $(v_1, \ldots, v_{r'}) = \texttt{nodes}_\mathcal{B}(x; L_1, \ldots, L_p)$, and $\sigma = x_{B_0}$. Finally, we define $A_0 = B_0$ and $A_j = B_{\pi(j)}$ for $j = 1, \ldots, p$ to yield an ordered footprint $(\sigma, A_1, \ldots, A_p)$ for E.

We count the total number of such embedded p-cylinder intersections over all elements in I''. To specify one such cylinder intersection, it suffices to describe its feet, spine, and its associated node sequence. There are fewer than 2^{Sr} choices of its node sequence, $|D|^{n-pm}$ choices of its spine, and at most $\binom{n}{m}^p \leq 2^{pH_2(m/n)n} = 2^{2pm\log_2(n/m)}$ choices of disjoint $A_1, \ldots, A_p$ each of size m.

Thus in total we obtain $2^{2pm\log_2(n/m)+Sr}|D|^{n-pm}$ such cylinder intersections E that partition a set containing I''. Therefore, by Markov's inequality, the portion of I'' covered by such intersections that have density $\delta(E) < 2^{-2pm\log_2(n/m)-Sr-2}|I|/|D^n|$ is at most $1/3$ of I''. Therefore there is an $I' \subseteq I''$ with $|I'| \geq |I|/2$ covered by embedded p-cylinder intersections with feet of size m and with density at least $2^{-2pm\log_2(n/m)-Sr-2}|I|/|D^n|$. $\square$

To use Theorem 3.7, we need a function with high p-party communication complexity when $p > 2$. We use p-tensor analogues of the quadratic forms considered in [7, 2, 6], although our tensors do not generalize either the modified Sylvester or random Hankel matrices used in those bounds.

We define our function $\text{TENS}_{p,t,q}$ in several steps. For simplicity, Let $q = 2^s > n$ and $e_1, \ldots e_n$ be distinct elements of field $\mathbb{F}_q$. Over $\mathbb{F}_q$ define vectors $v_1, \ldots, v_t \in \mathbb{F}_q^n$ by $v_i = (e_1^{i-1}, \ldots, e_n^{i-1})$. Let $\mathcal{T} = \sum_{i=1}^t \bigotimes_{j=1}^p v_i$; that is, for $y_1, \ldots y_p \in \mathbb{F}_q^n$,

$$\mathcal{T}(y_1, \ldots, y_p) = \sum_{i=1}^t \prod_{j=1}^p (v_i \cdot y_j) = \sum_{k_1,\ldots,k_p \in [n]} a_{k_1,\ldots,k_p} \prod_{j=1}^p y_{j,k_j}$$

where

$$a_{k_1,\ldots,k_p} = \sum_{i=1}^t v_{i,k_1} \cdots v_{i,k_p} = \sum_{i=1}^t e_{k_1}^{i-1} \cdots e_{k_p}^{i-1}.$$

Thus we can identify $\mathcal{T}$ by the $[n]^p$ array of coefficients $a_{k_1,\ldots,k_p}$. Then in analogy with the argument in [2] we modify $\mathcal{T}$ by setting most of the entries in this array to 0, defining $L(\mathcal{T})(y_1, \ldots, y_p) = \sum_{1 \leq k_1 < k_2 < \ldots < k_p \leq n} a_{k_1,\ldots,k_p} \prod_{j=1}^p y_{j,k_j}$. Finally, let $\phi : \mathbb{F}_{2^s} \to \mathbb{F}_2$ be any linear map and define $\text{TENS}_{p,t,q} : \mathbb{F}_{2^s}^n \to \mathbb{F}_2$ by $\text{TENS}_{p,t,q}(x) = \phi(L(\mathcal{T})(x, \ldots, x))$; for definiteness, we identify the elements of $\mathbb{F}_{2^s}$ with the residues of polynomials in $\mathbb{F}_2[z]$ modulo some irreducible degree s polynomial and take $\phi(h) = h(0)$.

THEOREM 3.10. *Let $1 > \epsilon > 0$ be arbitrary, let $p \geq 2$ be an integer with $p < (\epsilon/8)\log n$, let $s \geq \epsilon p 4^p \log n$, let $q = 2^s$, $D = \mathbb{F}_q$ and let $t = \lceil n^{1-\epsilon/4} \rceil$. then any D-way branching program computing $\text{TENS}_{p,t,q} : D^n \to \{0,1\}$ in time $T \leq (\epsilon/16)np\log n$ requires space $S \geq n^{1-\epsilon}\log|D|$.*

In order to derive our results for $\text{TENS}_{p,t,q}$ we must show that it is not constant on any embedded p-cylinder intersection with a balanced, ordered footprint that has large feet and density.

For any disjoint $A_1, \ldots, A_p \subseteq [n]$, and a tensor T define $T_{A_1,\ldots,A_p}$ to be the tensor on $\mathbb{F}_q^{A_1} \times \ldots \times \mathbb{F}_q^{A_p}$ given by the $A_1 \times \ldots A_p$ subarray of the array of T. Observe that for $\mathcal{T}$, $\mathcal{T}_{A_1,\ldots,A_p} = \sum_{i=1}^t \bigotimes_{j=1}^p u_{ji}$ where $u_{ji} = (v_i)_{A_j}$.

LEMMA 3.11. *If E is an embedded p-cylinder intersection with balanced, ordered footprint $(\rho, A_1, \ldots, A_p)$ on which $\text{TENS}_{p,t,q}$ is constant then there is an embedded p-cylinder intersection $E' \subseteq E$ with the same footprint and $\delta(E') \geq \delta(E)/2^p$ on which $\phi \circ \mathcal{T}_{A_1,\ldots,A_p}$ is constant.*

PROOF. Let $A_0 = [n] - A_1 - \ldots - A_p$. Observe that, since tensors are multilinear, $L(\mathcal{T})(x, \ldots, x)$ equals

$$\sum_{j_1,\ldots,j_p \in \{0,1,\ldots,p\}} L(\mathcal{T})_{A_{j_1},\ldots A_{j_p}}(x_{A_{j_1}}, \ldots, x_{A_{j_p}}).$$

On E, note that $x_{A_0} = \rho$. By construction of the map L and the fact that the footprint is ordered, for a permutation π, $L(\mathcal{T})_{A_{\pi(1)},\ldots,A_{\pi(p)}} = 0$ unless π is the identity. Further, any term that has an index containing A_0 must not have an index for at least one of the A_j for $j \geq 1$. For $j = 1, \ldots, p$, collect all terms in the sum that have indices A_i for all $i < j$ but do not have index A_j and call the function given by that sum f_j. Then

$$\begin{aligned} &\phi(L(\mathcal{T})(x, \ldots, x)) \\ &\quad= \phi(L(\mathcal{T})_{A_1,\ldots A_p}(x_{A_1}, \ldots, x_{A_p}) + f_1 + \ldots + f_p) \\ &\quad= \phi(\mathcal{T}_{A_1,\ldots A_p}(x_{A_1}, \ldots, x_{A_p})) + \phi(f_1) + \ldots + \phi(f_p) \end{aligned}$$

by the order condition on the footprint and the linearity of ϕ.

Let $C_1, \ldots, C_p$ be the legs of E. For $b \in \mathbb{F}_2$ let Let $C_j^b = \{z \in C_j \mid \phi(f_j^\rho(z)) = b\}$ and for $b_1, \ldots, b_p \in \mathbb{F}_q$ let $E^{b_1,\ldots,b_p} \subseteq E$ be the embedded p-cylinder intersection defined by ρ and $C_1^{b_1}, \ldots, C_p^{b_p}$. Choose $E' = E^{b_1,\ldots,b_p}$ for the $(b_1, \ldots, b_p)$ that maximizes $|E^{b_1,\ldots,b_p}|$. Clearly $|E'| \geq |E|/2^p$ and $\phi \circ f_1 + \ldots + \phi \circ f_p$ is constant on E'. $\square$

If $\mathcal{A}$ is a footprint of an embedded p-cylinder intersection, then $C(\mathcal{A})$ denotes the set of all embedded p-cylinder intersections whose footprint is $\mathcal{A}$; we extend this to p-partitions of $\mathcal{P}$ by identifying $\mathcal{P}$ with a footprint with an empty spine.

For any function $f : \mathbb{F}_q^n \to \{0,1\}$, we define its *discrepancy with respect to $\mathcal{A}$* by

$$\begin{aligned} \Gamma_{\mathcal{A}}(f) = \max_{C \in C(\mathcal{A})} |&Pr[f(x) = 1 \text{ and } x \in C] \\ &-Pr[f(x) = 0 \text{ and } x \in C]|. \end{aligned}$$

Following Raz [15], for $f : (\mathbb{F}_q^m)^p \to \mathbb{F}_2$ define $\Delta(f) = |E_{y_1,\ldots,y_p}[(-1)^{f(y_1,\ldots,y_p)}]|$. We say that f is *additive in its j-th component* if for all $a_1, \ldots, a_p, b_j \in \mathbb{F}_q^m$,

$$\begin{aligned} f(a_1, \ldots, a_j + b_j, \ldots, a_p) &= f(a_1, \ldots, a_j, \ldots, a_p) \\ &\quad + f(a_1, \ldots, b_j, \ldots, a_p). \end{aligned}$$

The following proposition is implicit in [15] and extends ideas from [11, 12].

PROPOSITION 3.12 ([15]). *Let $\mathcal{P}$ be the p-partition of $[mp]$ into $[m], [m] + m, \ldots, [m] + (p-1)m$. If $f : (\mathbb{F}_q^m)^p \to \mathbb{F}_2$ is additive on each component then $\Gamma_{\mathcal{P}}(f) \leq \Delta(f)^{1/2^p}$.*

Since the function $f : \mathbb{F}_q^{A_1 \cup \ldots \cup A_p} \to \mathbb{F}_2$ given by $f = \phi \circ \mathcal{T}_{A_1,\ldots,A_p}$ is additive in each component, by Proposition 3.12 we need only bound $\Delta(f)$ to bound $\Gamma_{(A_1,\ldots,A_p)}(f)$. The key property we use in this analysis is that if $t \leq |A_1|, \ldots, |A_p|$, the defining vectors for $\mathcal{T}_{A_1,\ldots,A_p}$ have the property that $u_{j1}, \ldots, u_{jt}$ are linearly independent for each $j = 1, \ldots, p$. This is immediate from the following lemma.

LEMMA 3.13. *Let $A \subseteq [n]$ with $|A| \geq t$ and for $i = 1, \ldots, t$. For $v_1, \ldots, v_t$ defined as above $(v_1)_A, \ldots, (v_t)_A$ are linearly independent.*

PROOF. Suppose that $\alpha_1(v_1)_A + \cdots + \alpha_t(v_t)_A = 0$. Define the polynomial $f(z) = \alpha_1 + \alpha_2 z + \cdots + \alpha_t z^{t-1}$. Then our assumption implies that $f(e_\ell) = 0$ for all $\ell \in A$. Therefore f has at least $|A| \geq t$ distinct roots and since it has degree at most $t-1$, it must be identically 0, implying that all the α_j are 0. $\square$

We now analyze the behavior of $\mathcal{T}_{A_1,\ldots,A_p}$, finding it convenient to do this analysis in a more general fashion by analyzing arbitrary tensors T with similar properties.

LEMMA 3.14. *If $T = \sum_{i=1}^t \alpha_i \bigotimes_{j=1}^p u_{ji}$ and $\beta, \beta' \in \mathbb{F}_q - \{0\}$ then $\Pr_{y_1,\ldots,y_p}[T(y_1, \ldots, y_p) = \beta] = \Pr_{y_1,\ldots,y_p}[T(y_1, \ldots, y_p) = \beta']$ and $\Pr_{y_1,\ldots,y_p}[T(y_1, \ldots, y_p) = 0] \geq 1/q$*

PROOF. Fix $a_2, \ldots, a_p \in \mathbb{F}_q^n$. Then

$$T(y_1, a_2, \ldots, a_p) = \sum_{i=1}^{t} \alpha_i (\prod_{j=2}^{p} (u_{ji} \cdot a_j))(u_{1i} \cdot y_1) = v_{a_2,\ldots,a_p} \cdot y_1$$

where $v_{a_2,\ldots,a_p} = \sum_{i=1}^t \alpha_i (\prod_{j=2}^p (u_{ji} \cdot a_j)) u_{1i}$. If $v_{a_2,\ldots,a_p} = 0$ then $\Pr_{y_1}[v_{a_2,\ldots,a_p} \cdot y_1 = \beta] = 0 = \Pr_{y_1}[v_{a_2,\ldots,a_p} \cdot y_1 = \beta']$ and $\Pr_{y_1}[v_{a_2,\ldots,a_p} \cdot y_1 = 0] = 1$. Otherwise, for any $\gamma \in \mathbb{F}_q$, $\Pr_{y_1}[v_{a_2,\ldots,a_p} \cdot y_1 = \gamma] = 1/q$. The lemma follows. □

LEMMA 3.15. *Suppose that $T = \sum_{i=1}^t \alpha_i \bigotimes_{j=1}^p u_{ji}$, that for each $j = 1, \ldots, p$, $u_{j1}, \ldots, u_{jr}$ are linearly independent over $\mathbb{F}_q$, and that at least $w > 0$ of the α_i are nonzero. Then $\Pr_{y_1,\ldots,y_p}[T(y_1, \ldots, y_p) = 0] \leq \sum_{k=1}^{p-1} (4/q)^{w/2^k} + 1/q$.*

PROOF. We prove this by induction on p for all such T. In the case $p = 1$,

$$T(y_1) = \sum_{i=1}^t \alpha_i (u_{1i} \cdot y_1) = (\sum_{i=1}^t \alpha_i u_{1i}) \cdot y_1.$$

Since $\sum_{i=1}^t \alpha_i u_{1i} \neq 0$ by the linear independence of $u_{11}, \ldots, u_{1t}$, $\Pr_{y_1}[T(y_1) = 0] = 1/q$ as required.

Now assume the statement for all $(p-1)$-tensors T' of this form and all $w > 0$ for $p > 1$. For $a \in \mathbb{F}_q^n$ let $N_p(a) = \#\{i \mid u_{pi} \cdot a \neq 0 \text{ and } \alpha_i \neq 0\}$. Let $S_p = \{a \in \mathbb{F}_q^n \mid N_p(a) \geq w/2\}$.

$$\begin{aligned} \Pr_{y_1,\ldots,y_p}[T(y_1, \ldots, y_p) = 0] &\leq \Pr_{y_p}[y_p \notin S_p] \\ &+ \max_{a_p \in S_p} \Pr_{y_1,\ldots,y_{p-1}}[T(y_1, \ldots, y_{p-1}, a_p) = 0] \end{aligned} \qquad (1)$$

Now for $a \in S_p$,

$$T(y_1, \ldots, y_{p-1}, a) = \sum_{i=1}^t \alpha_i (u_{pi} \cdot a) \bigotimes_{j=1}^{p-1} u_{ji} = \sum_{i=1} \alpha'_i \bigotimes_{j=1}^{p-1} u_{ji}$$

where $\alpha'_i = \alpha_i (u_{pi} \cdot a)$ and, since $a \in S_p$, $\alpha'_i \neq 0$ for at least $w' = w/2$ values of i. Therefore we can apply the inductive hypothesis to the $(p-1)$-tensor $T(y_1, \ldots, y_{p-1}, a)$ and $w' = w/2$ to bound the second term in (1). The first term in (1) is bounded above by $\sum_{k \geq w/2} \binom{w}{k} q^{-k} < 2^w q^{-w/2} = (4/q)^{w/2}$. Adding both bounds yields the desired result. □

COROLLARY 3.16. *If $m \geq t$ then $\Delta(\phi \circ \mathcal{T}_{A_1,\ldots,A_p}) \leq 2 \sum_{k=1}^{p-1} (4/q)^{t/2^k} \leq (4/q)^{t/2^p}$ and $\Gamma_{(A_1,\ldots,A_p)}(\phi \circ \mathcal{T}_{A_1,\ldots,A_p}) \leq (4/q)^{t/4^p}$.*

PROOF OF THEOREM 3.10. Let $m = t$, $k = \lfloor (p/2) \log(n/(2p^2 m)) \rfloor \geq (\epsilon/16) p \log n$ and observe that $m \leq 4^{-k/p} n/(2p^2) \leq 4^{1/p} m$. Let $r = \lceil 4^{k/p+2} k^2 p \rceil$ and observe that $r \leq (4np/m) \log^2(n/(2p^2 m)) \leq n$. Since $|\text{TENS}_{p,t,q}^{-1}(1)| > |D^n|/4$ we apply Theorem 3.7 to any branching program $\mathcal{B}$ computing $\text{TENS}_{p,t,q}$ to find an embedded p-cylinder intersection E with balanced, ordered footprint $(\rho, A_1, \ldots, A_p)$ with $m(E) = m$ and $\delta(E) \geq 2^{-2mp \log_2(n/m) - Sr - 4}$ on which $\text{TENS}_{p,t,q}$ is 1. Applying Lemma 3.11 we get an embedded p-cylinder intersection E' on the same footprint and $\delta(E') \geq \delta(E)/2^p$ on which $\phi \circ \mathcal{T}_{A_1,\ldots,A_p}$ is constant. By corollary 3.16 we must have $2^{-2mp \log_2(n/m) - Sr - p - 4} \leq (4/q)^{m/4^p}$. Solving for S, plugging in the upper bound on r, and bounding tiny terms yields $S \geq m^2[4^{-p} \log q - 2p \log(n/m)]/(16np \log^2(n/(2p^2 m)))$. Since $s = \log q \geq \epsilon p 4^p \log n$, $S \geq m^2 (\log q)/(32np4^p \log^2(n/(2p^2 m))) \geq n^{1-\epsilon} \log q$ since $p \leq (\epsilon/8) \log n$. □

Although we would like to derive improved bounds for the Boolean case as well, the prospects are not good for obtaining such bounds based on using multiple parties in the subtler argument for this case introduced by Ajtai [1, 2]. The key argument there uses a property that is true in the 2-party case but whose analogue is false in the p-party case for $p \geq 3$. Furthermore, in this argument a distribution of layers is chosen so that most layers are not assigned to any party. Values read in these layers would become part of the spine of the embedded cylinder intersection and the main multi-party advantage, larger feet for these cylinder intersections, would evaporate.

4. NEAREST NEIGHBOR DATA STRUCTURES AND TIME-SPACE TRADEOFFS

The *λ-near neighbor problem* $\lambda\mathbf{NN}$ over a metric space defined on U^d with metric Δ, has as input a query $x \in U^d$, a database $\mathcal{D} = \{y_1, \ldots, y_n\} \subset U^d$ as well as a fixed real number $\lambda = \lambda(n, d)$ and accepts iff there is some $y \in \mathcal{D}$ such that $\Delta(x, y) \leq \lambda$. The static data structure version of the problem allows arbitrary preprocessing based on $\mathcal{D}$, Δ, and λ and allows one to store information in some number of cells of memory, each of limited size so that given the query x as input, one can compute $\lambda\mathbf{NN}(x, \mathcal{D})$ efficiently. Thus three natural complexity measures are the amount that can be stored in each memory cell, the number of memory cells, and the query time.

The following is an extension of an observation of Miltersen, Nisan, Safra, and Wigderson [13] on the relationship between static data structure problems in the cell-probe model and time-space tradeoffs.

THEOREM 4.1. *Let $f : Q^m \times D^n \to O$ be a static data structure problem; i.e., given a query $x \in D^m$ and a database $\mathcal{D} \in D^n$, compute an output $f(x, \mathcal{D}) \in O$. If for every $\mathcal{D}_0 \in D^n$ there is a Q-way branching program $\mathcal{B}_{\mathcal{D}_0}$ computing the function $f^{\mathcal{D}_0}$ in time T and space S, then for any $k \geq 1$ there is a static cell-probe data structure using 2^S memory cells of $b = |Q|^k (S + \log m)$ bits each to store any database so that the query time to solve f is at most $\lceil T/k \rceil$.*

PROOF. The memory cells of the data structure will store the 2^S nodes of the Q-way branching program $\mathcal{B}_{\mathcal{D}_0}$. Each memory cell corresponding to a node $v \in \mathcal{B}_{\mathcal{D}_0}$ contains the names of the variables queried and pointers to the nodes reachable by each of the $|Q|^k$ paths of length k starting at v in $\mathcal{B}_{\mathcal{D}_0}$. □

Thus cell-probe lower bounds require time-space tradeoff lower bounds. We prove a converse under somewhat more restrictive assumptions about the data structure. We assume that, given a query, a data structure algorithm initially reads some portion of the input query and then chooses a memory cell from which to read. At each subsequent step, based on the contents of the memory cell identified in the previous step, it reads some more from the input and determines which memory cell to read from next. Such an algorithm will always have enough storage to name a memory location in the data structure and it may have some additional work space.

THEOREM 4.2. *Let $f : Q^m \times D^n \to O$ be a static data structure problem. If there is a data structure having at most 2^S cells of memory that reads at most k consecutive components of the query in a single time step and solves f using query time at most T and additional work space at most S, then for every $\mathcal{D}_0 \subset D^n$ there is a Q^k-way branching program running in time $O(T)$ and space $O(S + \log T)$ that computes $f(x, \mathcal{D}_0)$.*

PROOF. A node in the branching program will correspond to a pair consisting of the name of a memory cell and a configuration of the additional work space of the data structure (including the program counter of the algorithm). With the database fixed to $\mathcal{D}_0$, the contents of each memory cell are determined by its name. Given this configuration, the k components of the query to be accessed in a single time-step are determined by the algorithm, the configuration of the additional work space and the memory cell just accessed from the previous step. The new values of these quantities are determined by the value of these query components. $\square$

Thus we can obtain lower bounds for data structure problems for natural classes of such algorithms by proving time-space tradeoff lower bounds. In this section we do this for several λ-near neighbor problems. Although the memory cell size does not appear in the above statement explicitly, the bound on the number of query components that can be read in a single step is usually a function of the number that can fit in a single memory cell.

4.1 λ-Near Neighbor Lower bounds in Large Spaces

Given a metric Δ_0 on U we can derive a metric Δ on U^d as the ℓ_1 *composition of* Δ_0, defining $\Delta(x,y) = \sum_{i=1}^d \Delta_0(x_i, y_i)$. Thus the usual Hamming metric on U^d is just the ℓ_1 composition of the inequality metric. Similarly if $U = \{0,1\}^k$ and Δ_0 is the Hamming metric on U then its ℓ_1 composition is just the Hamming metric on $\{0,1\}^{kd}$. Call this the *compositional Hamming metric* on U^d.

For $a \in \{0,1\}^k$, define $\overline{a}$ to be a with each bit complemented and $double(a) = a\overline{a} \in \{0,1\}^{2k}$. For $x = (x_1, \ldots, x_d) \in (\{0,1\}^k)^d$ define $double^*(x) = (double(x_1), \ldots, double(x_d))$. Define $\mathcal{D}^k \subset (\{0,1\}^{2k})^d$ to be the set of all $\binom{d}{2}2^k$ vectors $y^k_{i,j,a}$ for $1 \le i < j \le d$, $a \in \{0,1\}^k$ where $y^k_{i,j,a}$ is $double(a)$ in its i-th and j-th coordinate and 0^{2k} in all other coordinates.

THEOREM 4.3. *Let (U^d, Δ) be the metric space where $U = [d^4] - 1$, and $\Delta(x,y)$ is the usual Hamming metric on U^d, the number of coordinates on which x and y differ. Then there exists a database $\mathcal{D}_0 \subset U^d$ of size $n = \Theta(d^4)$ and λ such that any U-way branching program (or RAM algorithm) computing $\lambda\mathbf{NN}(x, \mathcal{D}_0)$ on (U^d, Δ) in time T and space S requires $T = \Omega(d\sqrt{\log(d/S)/\log\log(d/S)})$.*

PROOF. Assume without loss of generality that $d = 2^k$ for some integer k. In [6], it is shown that the element distinctness problem, $ED_d : ([d^2]-1)^d \to \{0,1\}$ has a time-space tradeoff lower bound of the form $T = \Omega(d\sqrt{\log(d/S)/\log\log(d/S)})$. For suitable λ and $\mathcal{D}_0$ we give a reduction from ED_d to $\lambda\mathbf{NN}(x, \mathcal{D}_0)$.

We identify elements of $[d^2] - 1$ with elements of $\{0,1\}^{2k}$ and elements of $[d^4] - 1$ with elements of $\{0,1\}^{4k}$. Set $\mathcal{D}_0 = \mathcal{D}^{2k}$ and $\lambda = d - 2$.

Observe that for all $a \in \{0,1\}^{2k}$, $double(a) \ne 0^{4k}$. If $ED_d(x) = 1$ then there is an $a \in [d^2] - 1$ and $i \ne j$ such that $x_i = a = x_j$ and it is easy to see that $\Delta(double^*(x), y^{2k}_{i,j,a}) = d - 2 = \lambda$. Furthermore, if $ED_d(x) = 0$ then for all $y \in \mathcal{D}^{2k}$, $\Delta(double^*(x), y) \ge d - 1 > \lambda$. $\square$

COROLLARY 4.4. *Any data structure algorithm solving $\lambda\mathbf{NN}$ over (U^d, Δ) where $U = [d^4] - 1$, and $\Delta(x,y)$ is the usual Hamming metric on U^d, that uses $2^{(nd)^{o(1)}}$ memory cells and at most $(nd)^{o(1)}$ additional space and reads one component of the query per time step requires query time $\Omega(d\sqrt{\log d/\log\log d})$.*

THEOREM 4.5. *Let (U^d, Δ) be the metric space where $U = [d^6] - 1$, and $\Delta(x,y)$ is the compositional Hamming metric on U^d. Then there exists a database $\mathcal{D}_0 \subset U^d$ of size $n = \Theta(d^5)$ and λ such that any U-way branching program (or RAM algorithm) computing $\lambda\mathbf{NN}(x, \mathcal{D}_0)$ on (U^d, Δ) in time T and space S requires $T = \Omega(d\log((d\log d)/S))$.*

PROOF. Assume without loss of generality that $d = 2^k$ is an integral power of 2. For $\gamma < 1/2$, define the Hamming closeness problem $HAM_\gamma : (\{0,1\}^k)^d \to \{0,1\}$ to be one on input $(x_1, \ldots, x_d)$ if and only if there is some $i \ne j$ such that the Hamming distance between x_i and x_j is at most γk. In [6], it is shown that the Hamming closeness problem, $HAM_\gamma : ([d^\ell] - 1)^d \to \{0,1\}$ where $\ell > 2/(1 - H_2(\gamma))$ and $H_2(\gamma) = -\gamma\log_2\gamma - (1-\gamma)\log_2(1-\gamma)$ has a time-space tradeoff lower bound of the form $T = \Omega(d\log((d\log d)/S))$. We will consider $\gamma = 1/20$ for which $2/(1 - H_2(\gamma)) < 3$. For a suitable $\mathcal{D}_0$ and λ we give a reduction from HAM_γ to $\lambda\mathbf{NN}(x, \mathcal{D}_0)$.

We identify elements of $[d^3] - 1$ with elements of $\{0,1\}^{3k}$ and elements of $[d^6] - 1$ with elements of $\{0,1\}^{6k}$. Set $\mathcal{D}_0 = \mathcal{D}^{3k}$ and $\lambda = 3(d-2)k + 6\gamma k = (3d - 5.7)\log d$.

Observe that for $a \in \{0,1\}^{3k}$, $\Delta_0(double(a), 0^{6k}) = 3k$ and that for any $a, b, c \in \{0,1\}^{3k}$

$$\begin{aligned} &\Delta_0(double(a), double(c)) + \Delta_0(double(b), double(c)) \\ &\quad = \; 2(\Delta_0(a,c) + \Delta_0(b,c)) \ge 2\Delta_0(a,b) \end{aligned}$$

by the triangle inequality. Therefore, if $HAM_\gamma(x) = 1$ then there are $a, b \in [d^2] - 1$ and $i \ne j$ such that $x_i = a$, $x_j = b$, and $\Delta_0(a,b) \le 3\gamma k$. For this value observe that $\Delta(double^*(x), y^{3k}_{i,j,a}) \le \lambda$. If $HAM_\gamma(x) = 0$ then by the above observations for every $y^{3k}_{i,j,c} \in \mathcal{D}^{3k}$, $\Delta(double^*(x), y^{3k}_{i,j,c}) \ge \lambda + 2$. $\square$

COROLLARY 4.6. *Any data structure algorithm solving $\lambda\mathbf{NN}$ over (U^d, Δ) where $U = [d^6] - 1$, and $\Delta(x,y)$ is the compositional Hamming metric on U^d, that uses $2^{(nd)^{o(1)}}$ memory cells and at most $(nd)^{o(1)}$ additional space and reads one component of the query per time step requires query time $\Omega(d\log d)$.*

4.2 λ-Near Neighbor Lower Bounds in $\{0,1\}^d$

Note that by setting $d = 6d'\log d'$ and apply Corollary 4.6 with d' instead of d, we obtain

COROLLARY 4.7. *Any data structure algorithm solving $\lambda\mathbf{NN}$ on the Hamming space over $\{0,1\}^d$ with $2^{(nd)^{o(1)}}$ memory cells and $(nd)^{o(1)}$ additional space that reads $O(\log d) = O(\log n)$ consecutive bits of the query per step requires $\Omega(d)$ query time.*

This lower bound is larger than those of [8, 5] by a $\Theta(\log n)$ factor. (Note that, in the communication game model used in those papers, a model even stronger and less reasonable than the cell-probe model, query time $\Theta(d/\log n)$ is optimal.)

In this section we prove a query-time lower bound for the $\{0,1\}^d$ Hamming model that is a factor $\Omega(\log d)$ larger still but under the restriction that we can access only one bit of the query per step and that this corresponds to an oblivious rather than a general branching program. Note that this is incomparable with Corollary 4.7.

THEOREM 4.8. *Any data structure algorithm solving $\lambda\mathbf{NN}$ on the Hamming space over $\{0,1\}^d$ using $2^{(nd)^{o(1)}}$ memory cells and $(nd)^{o(1)}$ additional space that accesses one query bit per time step and in a fixed order requires query time $\Omega(d\log d)$.*

The rest of this section is devoted to the proof of the time-space tradeoff lower bound for $\lambda\mathbf{NN}$ over $\{0,1\}^d$ on oblivious branching programs that implies this theorem.

It will be useful to consider elements of $\{0,1\}^d$ as vectors from $\mathbb{F}_2^d$ whose indices are from $[d]-1$ rather than $[d]$. Furthermore, we assume that d is a power of two, set $k=\log_2 d$, and view those indices as themselves elements of $\mathbb{F}_2^k$. Thus, if $i\in\mathbb{F}_2^k$ and $v\in\mathbb{F}_2^d$ is a vector, then v_i will denote the i-th coordinate of v and we extend this notation to the partial vectors v_J for subsets $J\subseteq\mathbb{F}_2^k$ defined as the projection of v on J in accordance with our usual notation. Given indices i,j for coordinates of $v\in\mathbb{F}_2^d$, the expressions $i\cdot j$ and $i+j$ are well-defined, taking place in $\mathbb{F}_2^k$ rather than over the integers. We will use $\mathbf{0}$ and $\mathbf{1}$ to denote the all 0's and all 1's vectors respectively.

DEFINITION 4.1. *Let $V\subseteq\mathbb{F}_q^N$ be a vector space, $v\in V$, and $S\subseteq V$. Define $\mathbf{zeroes}(v)=\{i\in N: v_i=0\}$, let $\mathbf{ones}(v)$ denote the complement of $\mathbf{zeroes}(v)$, and define $\mathbf{zeroes}(S)=\bigcap_{s\in S}\mathbf{zeroes}(s)$.*

Throughout this section, Δ will denote the usual Hamming metric on $\mathbb{F}_2^d$. For any subset of the coordinates, $S\subset\mathbb{F}_2^k$, we extend this metric to pairs of partial vectors u_S,v_S defined on S in the obvious way and define $\Delta_S(u,v)=\Delta(u_S,v_S)$. Finally, for $u,v\in\mathbb{F}_2^d$, let $u\wedge v$ denote the bitwise AND of u and v so that $(u\wedge v)_i=u_iv_i$ for all $i\in\mathbb{F}_2^k$. Notice that $\Delta(u\wedge v,u\wedge w)=\Delta_{\mathbf{ones}(u)}(v,w)$ for any $u,v,w\in\mathbb{F}_2^d$.

We create a database $\mathcal{D}_0$ with $\Theta(d^2)$ elements over $\mathbb{F}_2^d$ for which computing $\lambda\mathbf{NN}_{\mathcal{D}_0}(x)=\lambda\mathbf{NN}(x,\mathcal{D}_0)$ with $\lambda=d/4$ has a large time-space tradeoff lower bound on oblivious branching programs. We will derive the lower bound using Theorem 3.3 with $p=2$ and a reduction of the 2-party fixed-partition communication problem $\mathbf{EQUALITY}$ to the 2-party best-partition communication problem $\lambda\mathbf{NN}^{\rho}_{\mathcal{D}_0}$ for a suitable ρ.

The database $\mathcal{D}_0$ is based on extensions of Hadamard codes. Given $v\in\mathbb{F}_2^k$ and $\alpha\in\mathbb{F}_2$, define $\Phi:\mathbb{F}_2^{k+1}\to\mathbb{F}_2^d$ by $\Phi(v,\alpha)_i=v\cdot i+\alpha$ for each $i\in\mathbb{F}_2^k$. Define the database

$$\mathcal{D}_0=\{\Phi(u,\alpha_0)\wedge\Phi(v,\alpha_1):\alpha_0,\alpha_1\in\mathbb{F}_2,u,v\in\mathbb{F}_2^d,\overline{u}\cdot\overline{v}=1\}.$$

The elements of our database are based on pairs of outputs of Φ. If we only needed to prove lower bounds with respect to the 2-party fixed-partition as opposed to the best-partition model then a simpler construction based on the following Lemma 4.9 would suffice. It shows that, if a set of coordinates S has a natural pairing and a vector ω is far away on S from every member of a family of outputs of Φ, then the values of ω on those paired coordinates satisfy $\mathbf{EQUALITY}$ or its dual. Lemma 4.10 shows that whenever ω satisfies these properties then a simple encoding of ω is in fact far from all vectors in $\mathcal{D}_0$, and conversely. This is the key property that allows us to prove the reduction from $\mathbf{EQUALITY}$ even in the best-partition model.

LEMMA 4.9. *Let $c\in\mathbb{F}_2^k-\{0\}$ and $\beta\in\mathbb{F}_2$. Let $S\subseteq\mathbb{F}_2^d$ be such that for all $i\in S$, $i+c\in S$. Suppose that for all $(v,\alpha)\in\mathbb{F}_2^{k+1}$ with $v\cdot c=\beta$, $\Delta_S(\Phi(v,\alpha),\omega)\ge\frac{1}{2}|S|$.*
1. If $\beta=0$, then $\omega_{i+c}=\overline{\omega}_i$ for all $i\in S$.
2. If $\beta=1$, then $\omega_{i+c}=\omega_i$ for all $i\in S$.

LEMMA 4.10. *Let $\omega\in\mathbb{F}_2^d$. Let $(u,\alpha)\in\mathbb{F}_2^{k+1}$, with $u\ne\mathbf{0}$. Let $\lambda\in\{0,\dots,d\}$. Then $\Delta((\Phi(u,\alpha)\wedge\omega),z)\ge\lambda$ for all $z\in\mathcal{D}_0$ if and only if*
1. $\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega,\mathbf{0})\ge\lambda$ and
2. $\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega,\Phi(v,\beta))\ge\lambda$ for all $(v,\beta)\in\mathbb{F}_2^{k+1}$ such that $\overline{u}\cdot\overline{v}=1$.

We give the proofs of these lemmas after some preliminaries. Given a vector space W with subspace U and $u\in W$, the set $u+U=\{w\in W: w=u+u' \text{ for some } u'\in U\}$ is called an *affine subspace of* W, or simply an *affine space*; U is called the *underlying space* of $u+U$. If V is an affine subspace, we denote its underlying space by $\widehat{V}$. Observe that

PROPERTY 4.11. *(1) if V,V' are affine spaces then $V\cap V'$ is an affine space and $\widehat{V\cap V'}=\widehat{V}\cap\widehat{V'}$ unless $V\cap V'=\emptyset$. (2) If $V\not\subseteq V'$, then $|V\cap V'|\le|\overline{V}\cap V'|$.*

We use two key properties of Φ.

PROPERTY 4.12. *(1) $\Phi:\mathbb{F}_2^{k+1}\to\mathbb{F}_2^d$ is linear and (2) For any $(v,\alpha)\in\mathbb{F}_2^{k+1}$, the sets $\mathbf{zeroes}(\Phi(v,\alpha))$ and $\mathbf{ones}(\Phi(v,\alpha))$ are affine subspaces of $\mathbb{F}_2^k$.*

We rely on the linearity of Φ for our reduction from $\mathbf{EQUALITY}$. Since some of our properties apply to all linear functions over larger fields we state them in this form. Let $\mathbb{F}_q$ be the field of q elements, and let n',k' be integers such that $n'=q^{k'}$. We will think of the indices of vectors from $\mathbb{F}_q^{n'}$ as elements of $\mathbb{F}_q^{k'}$.

LEMMA 4.13. *Let V be a vector space, and suppose $\phi:V\to\mathbb{F}_q^{n'}$ is a linear function. Let $x\in\mathbb{F}_q^{n'}$, let $u,v\in V$, and let $S\subseteq\mathbf{ones}(\phi(v))$. Then $\sum_{\alpha\in\mathbb{F}_q}\Delta_S(x,\phi(u+\alpha v))=(q-1)|S|$.*

PROOF. Let $i\in S\subseteq\mathbf{ones}(\phi(v))$. Clearly, there is exactly one $\alpha\in\mathbb{F}_q$ such that $x_i=\phi(u_i)+\alpha\phi(v_i)$ since $\phi(v_i)\ne0$. So $\sum_{\alpha\in\mathbb{F}_q}\Delta_i(x,\phi(u+\alpha v))=(q-1)$. Our claim follows. □

LEMMA 4.14. *Let V be an affine subspace of $\mathbb{F}_q^{k''}$ with underlying space $\widehat{V}$. Let $\phi:\mathbb{F}_q^{k''}\to\mathbb{F}_q^{n'}$ be a linear function. Let $S\subseteq\mathbb{F}_q^{k'}$, $T\subseteq\widehat{V}$, and $x\in\mathbb{F}_q^{n'}$. If for all $v\in V$, $\Delta_S(\phi(v),x)\ge(1-\frac{1}{q})|S|$, then $\Delta_{S\cap\mathbf{zeroes}(\phi(T))}(\phi(v),x)\ge(1-\frac{1}{q})|S\cap\mathbf{zeroes}(\phi(T))|$ for all $v\in V$.*

PROOF. Suppose $\Delta_S(\phi(v),x)\ge(1-\frac{1}{q})|S|$, and let T be any subset of $\widehat{V}$. We will show by induction on the size of T that $\Delta_{S\cap\mathbf{zeroes}(\phi(T))}(\phi(v),x)\ge(1-\frac{1}{q})|S\cap\mathbf{zeroes}(\phi(T))|$.

The base case is trivial, so suppose that for all $v\in V$, $\Delta_{S'}(\phi(v),x)\ge(1-\frac{1}{q})|S'|$ for $S'=S\cap\mathbf{zeroes}(\phi(T'))$ and T' a subset of T with $|T|=|T'|+1$.

Let $u\in T-T'$. Notice that for $\alpha\in\mathbb{F}_q$ that $v+\alpha u\in V$ since $u\in\widehat{V}$. Hence, for any $\alpha\in\mathbb{F}_q$, $\Delta_{S'}(\phi(v+\alpha u),x)\ge(1-\frac{1}{q})|S'|$. Further, since $\phi(v+\alpha u)$ and $\phi(v)$ agree on all coordinates in $\mathbf{zeroes}(\phi(u))$,

$$\begin{aligned}&\Delta_{S'\cap\mathbf{zeroes}(\phi(u))}(\phi(v),x)+\Delta_{S'\cap\mathbf{ones}(\phi(u))}(\phi(v+\alpha u),x)\\ \ge\;&(1-\frac{1}{q})|S'|.\end{aligned}$$

Summing the above inequalities over all $\alpha\in\mathbb{F}_q$ and applying Lemma 4.13, we see that

$$\Delta_{S'\cap\mathbf{zeroes}(\phi(u))}(\phi(v),x)\ge(1-\frac{1}{q})|S'\cap\mathbf{zeroes}(\phi(u))|.$$

Since $S'\cap\mathbf{zeroes}(\phi(u))=S\cap\mathbf{zeroes}(\phi(T))$, this proves our claim. □

We can now apply this $q=2$ to derive Lemma 4.9 as a corollary.

PROOF OF LEMMA 4.9. Fix $i \in S$. Let $U_i^0 = \{(v,\alpha) \in \mathbb{F}_2^{k+1} : v \cdot i + \alpha = 0\}$ and let $V_c^\beta = \{(v,\alpha) \in \mathbb{F}_2^{k+1} : v \cdot c = \beta\}$. Clearly, $U_i^0 \cap V_c^0 \subseteq \widehat{V}_c^\beta$. Furthermore, $\mathbf{zeroes}(\Phi(U_i^0 \cap V_c^0)) = \{i, i+c\} \subseteq S$. Applying Lemma 4.14 with $T = U_i^0 \cap V_c^0$ we see that $\Delta_{\{i,i+c\}}(\Phi(v,\alpha),\omega) \geq 1$ for all $(v,\alpha) \in V_c^\beta$.

(i) If $\beta = 0$, then $(\mathbf{0},0),(\mathbf{0},1) \in V_c^\beta$ and so

$$\Delta_{\{i,i+c\}}(\mathbf{0},\omega) \geq 1 \text{ and } \Delta_{\{i,i+c\}}(\mathbf{1},\omega) \geq 1$$

which together imply that $\omega_{i+c} = \overline{\omega}_i$.

(ii) If $\beta = 1$, then choose a u such that $u \cdot c = 1$. We see that the i-th coordinate and the $(i+c)$-th coordinate of $\Phi(u,\alpha)$ must differ for any $\alpha \in \mathbb{F}_2$. Further, $\Phi(u,\overline{\alpha}) = \overline{\Phi}(u,\alpha)$. Hence,

$$\begin{aligned} \Delta_{\{i,i+c\}}((0,1),\omega_{\{i,i+c\}}) &\geq 1 \\ \Delta_{\{i,i+c\}}((1,0),\omega_{\{i,i+c\}}) &\geq 1 \end{aligned}$$

and thus $\omega_{i+c} = \omega_i$. □

In order to prove Lemma 4.10, we use the following technical lemma based on the fact that Hadamard codewords are characteristic vectors of affine subspaces.

LEMMA 4.15. *Let $z \in \mathcal{D}_0$ with $z \neq \mathbf{0}$. Let $(u,\alpha) \in \mathbb{F}_2^{k+1}$ with $u \neq \mathbf{0}$. If $\mathbf{ones}(\Phi(u,\alpha)) \supseteq \mathbf{ones}(z)$, then there exists a $(v,\beta) \in \mathbb{F}_2^{k+1}$ such that $z = \Phi(u,\alpha) \wedge \Phi(v,\beta)$.*

PROOF. Since $z \in \mathcal{D}_0$, there are $(s,\beta_0),(t,\beta_1) \in \mathbb{F}_2^{k+1}$ with $\overline{s} \cdot \overline{t} = 1$ such that $z = \Phi(s,\beta_0) \wedge \Phi(t,\beta_1)$. For convenience, let $U = \mathbf{ones}(\Phi(u,\alpha))$ and let $V = \mathbf{ones}(z) = \mathbf{ones}(\Phi(s,\beta_0)) \cap \mathbf{ones}(\Phi(t,\beta_1))$. Note that both U and V are affine spaces. Since $z \neq \mathbf{0}$, $V \neq \emptyset$. Hence, $\widehat{V} = \{i \in \mathbb{F}_2^k : s \cdot i = 0 = t \cdot i\}$, so that $\widehat{V}^\perp = \{\mathbf{0}, s, t, s+t\}$. Similarly, we see that $\widehat{U}^\perp = \{\mathbf{0}, u\}$. By hypothesis, $U \supseteq V$, so $\widehat{U} \supseteq \widehat{V}$ and thus $\widehat{U}^\perp \subseteq \widehat{V}^\perp$. Hence, $u \in \{\mathbf{0}, s, t, s+t\}$. Since $u \neq \mathbf{0}$ by assumption, we have three cases to consider.

(i) If $u = s$, then $z = \Phi(u,\beta_0) \wedge \Phi(t,\beta_1)$ with $\overline{u} \cdot \overline{t} = 1$.

(ii) If $u = t$, then $z = \Phi(s,\beta_0) \wedge \Phi(u,\beta_1)$ with $\overline{s} \cdot \overline{u} = 1$.

(iii) If $u = s+t$, then note that $z = \Phi(s,\beta_0) \wedge \Phi(t,\beta_1) = (\Phi(s,\beta_0) + \Phi(t,\overline{\beta_1})) \wedge \Phi(t,\beta_1) = \Phi(u,\beta_0 + \overline{\beta_1}) \wedge \Phi(t,\beta_1)$ by linearity of Φ. Furthermore, $\overline{u} \cdot \overline{t} = (\overline{s} + t) \cdot \overline{t} = \overline{s} \cdot \overline{t} + t \cdot \overline{t} = 1$.

So in all cases, $z = \Phi(u,\alpha') \wedge \Phi(v,\beta)$ for some $\alpha' \in \mathbb{F}_2$ and $(v,\beta) \in \mathbb{F}_2^{k+1}$ with $\overline{u} \cdot \overline{v} = 1$.

If $\alpha' = \alpha$, we are done. Otherwise, $\alpha' = \overline{\alpha}$, so that $\mathbf{ones}(z) \subseteq \mathbf{ones}(\Phi(v,\overline{\alpha})) = \mathbf{zeroes}(\Phi(v,\alpha))$. But $\mathbf{ones}(z) \subseteq \mathbf{ones}(\Phi(v,\alpha))$ by assumption. So $\mathbf{ones}(z) = \emptyset$. That is, $z = \mathbf{0}$, a contradiction. □

PROOF OF LEMMA 4.10. For the first direction, suppose that $\Delta((\Phi(u,\alpha) \wedge \omega), z) \geq \lambda$ for all $z \in \mathcal{D}_0$. Then in particular for any $(v,\beta) \in \mathbb{F}_2^{k+1}$ with $\overline{u} \cdot \overline{v} = 1$, $\Delta((\Phi(u,\alpha) \wedge \omega), (\Phi(u,\alpha) \wedge \Phi(v,\beta))) \geq \lambda$. This implies that $\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, \Phi(v,\beta)) \geq \lambda$ for all such (v,β). Furthermore, $\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, \mathbf{0}) = \Delta((\Phi(u,\alpha) \wedge \omega), \mathbf{0}) \geq \lambda$ since $\mathbf{0} \in \mathcal{D}_0$. This proves the first direction.

For the other direction, suppose that $\Delta((\Phi(u,\alpha) \wedge \omega), z) < \lambda$ for some $z \in \mathcal{D}_0$. If $z = \mathbf{0}$, then $\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, \mathbf{0}) = \Delta((\Phi(u,\alpha) \wedge \omega), \mathbf{0}) < \lambda$. So we may assume that $z \neq \mathbf{0}$. We have two cases to consider:

(i) If $\mathbf{ones}(\Phi(u,\alpha)) \supseteq \mathbf{ones}(z)$, then by Lemma 4.15, $z = \Phi(u,\alpha) \wedge \Phi(v,\beta)$ for some $(v,\beta) \in \mathbb{F}_2^{k+1}$ such that $\overline{u} \cdot \overline{v} = 1$. Hence,

$$\begin{aligned} &\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, \Phi(v,\beta)) \\ &\quad = \Delta((\Phi(u,\alpha) \wedge \omega), (\Phi(u,\alpha) \wedge \Phi(v,\beta))) \\ &\quad = \Delta(\Phi(u,\alpha) \wedge \omega, z) < \lambda. \end{aligned}$$

(ii) If $\mathbf{ones}(\Phi(u,\alpha)) \not\supseteq \mathbf{ones}(z)$, then $|\mathbf{ones}(\Phi(u,\alpha)) \cap \mathbf{ones}(z)| \leq |\mathbf{zeroes}(\Phi(u,\alpha)) \cap \mathbf{ones}(z)|$ by Property 4.11 since $\mathbf{ones}(\Phi(u,\alpha))$ and $\mathbf{ones}(z)$ are both affine spaces. That is, $\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\mathbf{0}, z) \leq \Delta_{\mathbf{zeroes}(\Phi(u,\alpha))}(\mathbf{0}, z)$. Hence,

$$\begin{aligned} &\Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, \mathbf{0}) \\ &\quad \leq \Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, z) + \Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\mathbf{0}, z) \\ &\quad \leq \Delta_{\mathbf{ones}(\Phi(u,\alpha))}(\omega, z) + \Delta_{\mathbf{zeroes}(\Phi(u,\alpha))}(\mathbf{0}, z) \\ &\quad = \Delta((\Phi(u,\alpha) \wedge \omega), z) < \lambda \end{aligned}$$

This completes the proof. □

In order to define the encoding of the input to **EQUALITY** we use the following simple extension of the usual arguments for best-partition communication complexity with additional twists to handle our particular needs.

LEMMA 4.16. *Let $A, B \subseteq \mathbb{F}_2^k$ with $|A| = |B| = m > 1$ and $A \cap B = \emptyset$. Then there exist $c \in \mathbb{F}_2^k - \{\mathbf{0}, \mathbf{1}\}$, $\alpha_c \in \mathbb{F}_2$, and sets $A' \subseteq A \cap \mathbf{ones}(\Phi(\overline{c}, \alpha_c))$ and $B' \subseteq B \cap \mathbf{ones}(\Phi(\overline{c}, \alpha_c))$ such that $B' = c + A'$ and $|A'| = |B'| = m^2/(4d)$.*

PROOF. We first find an appropriate c. Notice that

$$\sum_{u \in \mathbb{F}_2^d, u \neq \mathbf{1}} |A \cap (u+B)| \geq (\sum_{a \in A} |B|) - |A \cap (\mathbf{1}+B)| \geq m^2 - m$$

where the first inequality stems from the fact that for every $a \in A$, there are at least $|B|$ values for u such that $a \in u+B$. Hence, by the pigeonhole principle, there is a $c \neq \mathbf{1}$ such that $|A \cap (c+B)| \geq (m^2 - m)/d \geq \frac{m^2}{2d}$ for $m > 1$. Also, $c \neq \mathbf{0}$ since $A \cap B = \emptyset$. Let $A'' = A \cap (c+B)$, and let $B'' = c + A''$. Notice that $A'' \subseteq A$ and $B'' \subseteq B$.

Since $\mathbf{ones}(\Phi(\overline{c}, 0))$ and $\mathbf{ones}(\Phi(\overline{c}, 1))$ are disjoint and together cover all of $\mathbb{F}_2^k$, there is an $\alpha_c \in \mathbb{F}_2$ such that $|A'' \cap \mathbf{ones}(\Phi(\overline{c}, \alpha_c))| \geq \frac{1}{2}|A''| = m^2/(4d)$. Choose A' so that $A' \subseteq A'' \cap \mathbf{ones}(\Phi(\overline{c}, \alpha_c))$ and $|A'| = m^2/(4d)$. Let $B' = c + A'$. Observe that since, trivially, $\overline{c} \cdot c = 0$ we have $\mathbf{ones}(\Phi(\overline{c}, \alpha_c)) = c + \mathbf{ones}(\Phi(\overline{c}, \alpha_c))$. Therefore,

$$\begin{aligned} B' &= c + A' \subseteq (c + A'') \cap (c + \mathbf{ones}(\Phi(\overline{c}, \alpha_c))) \\ &= B'' \cap \mathbf{ones}(\Phi(\overline{c}, \alpha_c)). \end{aligned}$$

This completes the proof. □

Given A, B as in Lemma 4.16, fix some A', B', c and α_c guaranteed by the lemma. Write A' as $A' = \{a_1, a_2, \ldots, a_{m^2/(4d)}\}$. Since $B' = c + A'$, we may also write B' as $B' = \{b_1, b_2, \ldots, b_{m^2/(4d)}\}$ with $b_i = c + a_i$ for all $i = 1, 2, \ldots, m^2/(4d)$. For $x, y \in \{0,1\}^{m^2/(4d)}$, define a vector $\gamma_{A,B}(x,y) \in \mathbb{F}_2^d$ as follows: If $c \cdot c = 0$, define $\gamma_{A,B}(x,y)$ to be the vector which is 0 outside $\mathbf{ones}(\Phi(\overline{c}, \alpha_c)) \supset A' \cup B'$, and has x_j in location a_j, y_j in location b_j for $j = 1, \ldots, m^2/(4d)$, and 1 elsewhere. If $c \cdot c = 1$, define $\gamma_{A,B}(x,y)$ to be the vector which is 0 outside $A' \cup B'$, and has x_j in location a_j and $\overline{y}_j$ in location b_j for $j = 1, \ldots, m^2/(4d)$.

Observe that for $c \cdot c = 0$, $\gamma_{A,B}(x,y)_i = \gamma_{A,B}(x,y)_{i+c}$ for all $i \in \mathbb{F}_2^k$ if and only if $x = y$. Further, we note that for $c \cdot c = 1$, $\overline{\gamma_{A,B}(x,y)_i} = \gamma_{A,B}(x,y)_{i+c}$ for all $i \in \mathbb{F}_2^k$ if and only if $x = y$. Also observe that $\Delta(\gamma_{A,B}(x,y), \mathbf{0}) \geq d/4$ whenever $x = y$.

Notice that $\gamma_{A,B}(x,y)$ is 0 for all coordinates in $\mathbf{ones}(\Phi(\overline{c}, \alpha_c))$. Also notice that given the coordinate sets A and B, it is possible to compute the value of $\gamma_{A,B}(x,y)$ on all coordinates of $\overline{B}$ given only the value of x. Similarly, we can compute the value of $\gamma_{A,B}(x,y)$ on all coordinates of $\overline{A}$ given the value of y.

LEMMA 4.17. *Given disjoint $A, B \subseteq \mathbb{F}_2^k$ with $|A| = |B| = m$, and $x, y \in \{0,1\}^{m^2/(4d)}$, let $c \in \mathbb{F}_2^k$ and $\gamma_{A,B}(x,y)$ be defined as above. Then $x = y$ if and only if for all $(v,\beta) \in \mathbb{F}_2^{k+1}$ such that $\overline{v} \cdot c = 1$, $\Delta_{\mathbf{ones}(\Phi(\overline{c},\alpha_c))}(\gamma_{A,B}(x,y), \Phi(v,\beta)) \geq d/4$.*

PROOF. Either $c \cdot c = 0$ or $c \cdot c = 1$. (i) Suppose $c \cdot c = 0$. To prove one direction, suppose that

$$\Delta_{\mathbf{ones}(\Phi(\overline{c},\alpha_c))}(\gamma_{A,B}(x,y), \Phi(v,\beta)) \geq d/4$$

for all $(v,\beta) \in \mathbb{F}_2^{k+1}$ such that $\overline{v} \cdot c = 1$. Note that $\overline{v} \cdot c = 1$ if and only if $v \cdot c = 1$ since $c \cdot c = 0$. So we may apply Lemma 4.9 with $S = \mathbf{ones}(\Phi(\overline{c},\alpha_c))$ to see that $\gamma_{A,B}(x,y)_i = \gamma_{A,B}(x,y)_{i+c}$ for all $i \in \mathbf{ones}(\Phi(\overline{c},\alpha_c))$. That is, $x = y$ as required.

Now consider the other direction. Suppose that $x = y$. Then $\gamma_{A,B}(x,y)_i = \gamma_{A,B}(x,y)_{i+c}$ for all $i \in \mathbf{ones}(\Phi(\overline{c},\alpha_c))$.

For any $(v,\beta) \in \mathbb{F}_2^{k+1}$ with $v \cdot c = 1$, we see that $\Phi(v,\beta)_i = \overline{\Phi}(v,\beta)_{i+c}$ for all $i \in \mathbb{F}_2^d$. Hence, $\Delta_{\{i,i+c\}}(\gamma_{A,B}(x,y), \Phi(v,\beta)) = 1$ for any $i \in \mathbf{ones}(\Phi(\overline{c},\alpha_c))$. Thus, $\Delta_{\mathbf{ones}(\Phi(\overline{c},\alpha_c))}(\gamma_{A,B}(x,y), \Phi(v,\beta)) = |\mathbf{ones}(\Phi(\overline{c},\alpha_c))|/2 = d/4$.

(ii) The case when $c \cdot c = 1$ follows analogously. $\square$

THEOREM 4.18. *Let $\lambda = 1/4$, $d = 2^k$, $N = \mathbb{F}_2^k$, and $N' \subseteq N$ with $|N'| = d - 2m$. Then over $\{0,1\}^N$, there is a partial assignment, $\rho \in \{0,1\}^{N'}$, such that $C_2^{best}(\lambda\mathbf{NN}^{\rho}_{\mathcal{D}_0}) \geq m^2/(4d)$*

PROOF. We do this by reduction from $\mathbf{EQUALITY}_{m^2/4d}$. Let A, B be disjoint subsets of $N - N'$ with $|A| = |B| = m$. Given $x, y \in \{0,1\}^{m^2/(4d)}$, define $\gamma_{A,B}(x,y)$ based on A and B as above. By Lemma 4.17, $x = y$ if and only if $\Delta_{\mathbf{ones}(\Phi(\overline{c},\alpha_c))}(\gamma_{A,B}(x,y), \Phi(v,\beta)) \geq d/4$ for all $(v,\beta) \in \mathbb{F}_2^{k+1}$ such that $\overline{v} \cdot c = 1$. For $x = y$, $\Delta_{\mathbf{ones}(\Phi(\overline{c},\alpha_c))}(\gamma_{A,B}(x,y), \mathbf{0}) \geq d/4$, so we may apply Lemma 4.10 to find that $x = y$ if and only if for all $z \in \mathcal{D}_0$, $\Delta(\gamma_{A,B}(x,y), z) \geq d/4$

Set ρ to be the partial assignment that is equal to $\gamma_{A,B}(x,y)$ for coordinates in $\overline{A} \cap \overline{B}$, and unassigned elsewhere. Note that ρ is constant with respect to x and y. Then given a two-party best-partition communication complexity protocol solving $\lambda\mathbf{NN}^{\rho}_{\mathcal{D}_0}$, we can construct a two-party communication complexity protocol solving $\mathbf{EQUALITY}$ on $m^2/4d$ bits. Hence, $C_2^{best}(\lambda\mathbf{NN}^{\rho}_{\mathcal{D}_0}) \geq C_2(\mathbf{EQUALITY}_{m^2/4d}) = m^2/(4d)$. $\square$

THEOREM 4.19. *If an oblivious branching program with time T and space S solves $\lambda\mathbf{NN}_{\mathcal{D}_0}$ over $\{0,1\}^d$, then $T = \Omega(d\log(d/S))$.*

PROOF. Let $\ell = T/d$. From Theorem 3.3 with $p = 2$, there is an N' of size $|N'| = d - 2^{-\ell-1}d$ along with a partial assignment, ρ on N' such that $C_2^{best}(\lambda\mathbf{NN}^{\rho}_{\mathcal{D}_0}) \leq 2^{\ell+3}\ell^2 \log W$ and from Corollary 4.18 with $m = 2^{-\ell-2}d$, we have $C_2^{best}(\lambda\mathbf{NN}^{\rho}_{\mathcal{D}_0}) \geq 2^{-2\ell-6}d$. Combining this, we obtain $2^{3\ell+9}\ell^2 \geq d/\log W$. Since $S \geq \log W$ we obtain $\ell \geq C_0 \log(d/S)$ for some constant $C_0 > 0$ and thus $T = \ell d \geq C_0 d \log(d/S)$. $\square$

Acknowledgements

Thanks to Yuval Rabani for suggesting the investigation of time-space tradeoffs for the nearest neighbor problem and to Matt Cary for helpful discussions.

5. REFERENCES

[1] M. Ajtai. Determinism versus non-determinism for linear time RAMs with memory restrictions. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 632–641, Atlanta, GA, May 1999.

[2] M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proceedings 40th Annual Symposium on Foundations of Computer Science*, pages 60–70, New York,NY, October 1999. IEEE.

[3] Noga Alon and Wolfgang Maass. Meanders and their applications in lower bounds arguments. *Journal of Computer and System Sciences*, 37:118–129, 1988.

[4] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, October 1992.

[5] O. Barkol and Y. Rabani. Tighter lower bounds for nearest neighbor search and related problems in the cell probe model. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 388–396, Portland,OR, May 2000.

[6] Paul Beame, Michael Saks, Xiaodong Sun, and Erik Vee. Super-linear time-space tradeoff lower bounds for randomized computation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 169–179, Redondo Beach, CA, November 2000. IEEE.

[7] Paul W. Beame, Michael Saks, and Jayram S. Thathachar. Time-space tradeoffs for branching programs. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*, pages 254–263, Palo Alto, CA, November 1998. IEEE.

[8] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neight search and related problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 312–321, Atlanta, GA, May 1999.

[9] Allan Borodin, A. A. Razborov, and Roman Smolensky. On lower bounds for read-k times branching programs. *Computational Complexity*, 3:1–18, October 1993.

[10] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 94–99, Boston, MA, April 1983.

[11] F. R. K. Chung. Quasi-random classes of hypergraphs. *Random Structures and Algorithms*, 1(4):363–382, 1990.

[12] F. R. K. Chung and P. Tetali. Communication complexity and quasi-randomness. *SIAM Journal on Discrete Mathematics*, 6(1):110–123, 1993.

[13] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

[14] Nicholas J. Pippenger. On simultaneous resource bounds. In *20th Annual Symposium on Foundations of Computer Science*, pages 307–311, San Juan, Puerto Rico, October 1979. IEEE.

[15] R. Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9:113–122, 2000.

[16] Ingo Wegener. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. Society for Industrial and Applied Mathematics, 2000.