Dartmouth College
Dartmouth Digital Commons

Dartmouth College Ph.D Dissertations | Theses and Dissertations

11-1-2010

Some Communication Complexity Results and their Applications

Joshua E. Brody
Dartmouth College

Follow this and additional works at: https://digitalcommons.dartmouth.edu/dissertations

 Part of the Computer Sciences Commons

Recommended Citation

Brody, Joshua E., "Some Communication Complexity Results and their Applications" (2010). *Dartmouth College Ph.D Dissertations*. 34.
https://digitalcommons.dartmouth.edu/dissertations/34

This Thesis (Ph.D.) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Ph.D Dissertations by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.

Some Communication Complexity Results and their Applications

Dartmouth Computer Science Technical Report TR2011-699

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Joshua Brody

DARTMOUTH COLLEGE
Hanover, New Hampshire
November, 2010

Examining Committee:

(chair) Amit Chakrabarti, Ph.D.

Peter Winkler, Ph.D.

Prasad Jayanti, Ph.D.

Emanuele Viola, Ph.D.

Brian W. Pogue, PhD
Dean of Graduate Studies

Abstract

Communication Complexity represents one of the premier techniques for proving lower bounds in theoretical computer science. Lower bounds on communication problems can be leveraged to prove lower bounds in several different areas.

In this work, we study three different communication complexity problems. The lower bounds for these problems have applications in circuit complexity, wireless sensor networks, and streaming algorithms.

First, we study the multiparty pointer jumping problem. We present the first nontrivial upper bound for this problem. We also provide a suite of strong lower bounds under several restricted classes of protocols.

Next, we initiate the study of several non-monotone functions in the distributed functional monitoring setting and provide several lower bounds. In particular, we give a generic adversarial technique and show that when deletions are allowed, no nontrivial protocol is possible.

Finally, we study the Gap-Hamming-Distance problem and give tight lower bounds for protocols that use a constant number of messages. As a result, we take a well-known lower bound for one-pass streaming algorithms for a host of problems and extend it so it applies to streaming algorithms that use a constant number of passes.

Acknowledgements

I am extremely indebted to my advisor, Amit Chakrabarti, who spent the last five years patiently transforming me into a (hopefully) competent researcher. I thank him for the research guidance, support, and mentorship.

Peter Winkler has been an excellent source of problems, puzzles, ideas, and inspiration. I am greatful for the positive influence he's been on my academic development. I'd also like to thank my other thesis committee members, Prasad Jayanti and Emanuele Viola, for their time, consideration, and helpful comments.

The results in this thesis are the result of several collaborations. I thank my coauthors for the works presented in this thesis: Chrisil Arackaparambil, Amit Chakrabarti, Oded Regev, Thomas Vidick, and Ronald de Wolf. I'm also greatful to Sergey Bratus, David Kotz, Anna Shubina, and Elad Verbin for successful collaborations on other research.

I thank past and current members of the Dartmouth Theory Reading Group, including (but not limited to) Umang Bhaskar, Vibhor Bhatt, Scott Drysdale, Lisa Fleischer, Chien-Chung Huang, Ranganath Kondapally, and Afra Zomorodian. I am proud to be a part of such a reading group and excited to see the research it sparks in the future.

I'm also greatful to several fellow graduate students, including Brandon Kerr, Peter Johnson, Elena Davidson Strange, Sandra Van Ginhoven, Christopher Masone, Jon Denning, Joe Cooley, Danny Milisavljevic, and Meghan Mella, who made life as a graduate student more than just writing papers and taking classes.

Finally, I would like to thank my parents Ronald and Suzy Brody for their encouragement and support and my most wonderful best friend and fiancée Scout Sinclair for her everlasting love.

Contents

Chapter 1

Introduction

Computational complexity theory studies the powers and limitations of computation and seeks to classify problems in terms of the amount of resources needed to solve them. In short, it is the study of what computers can and *cannot* do. The analysis of algortihms considers particular solutions for problems and analyzes how much of a particular resource (e.g. space, time, etc.) the solution uses. For particular kinds of problems, there are often general techniques for solving a problem or analyzing the algorithm. Such techniques include greedy algorithms, dynamic programming, divide and conquer, and linear programming. In contrast, relatively few general techniques exist for proving *lower bounds* in complexity theory.

Communication complexity represents one of the most general and useful techniques for proving lower bounds. In a communication complexity problem, input is split between multiple players $\text{PLR}_1, \text{PLR}_2, \ldots$, who wish to compute some function of the input. As no single player sees the entire input, players must communicate together to solve the problem. A *protocol* describes how players communicate to determine the correct output. One wishes to evaluate how much communication must be exchanged to jointly compute the function. The cost of a protocol is the worst-case amount of communication sent. The communication complexity is the minimum cost of any protocol that correctly computes the function. Usually, the primary goal

is to prove lower bounds on the communication that must be sent. By itself, communication complexity is natural and mathematically interesting. However, its true power and beauty lie in its abstraction—lower bounds on communication problems often capture the innate hardness of other problems in computer science, including problems which do not involve communication. Indeed, researchers have used reductions from commmunication complexity to prove lower bounds in several areas, including circuit complexity [CFL83, KW90, HG91, BNS92, BT94], data structures [Ajt88, MNSW98, PT06], and streaming algorithms [AMS99, IW03]. It is this flexibility and wide application to other areas of computer science that makes communication complexity such a rich and important subfield.

1.1 Our Contributions

We focus our work on three communication complexity problems.

Multiparty Pointer Jumping. In the multiparty pointer jumping problem (MPJ_k), there are k players, and the input is divided into k pieces, one piece per player. The first piece is an index $i \in [n]$; the kth piece is an n-bit string x, and the rest of the pieces are functions $f_2, \ldots, f_{k-1} : [n] \to [n]$. The goal in this problem is to compute $\text{MPJ}_k(i, f_2, \ldots, f_{k-1}, x) := x[f_{k-1} \circ \cdots \circ f_2(i)]$. Of particular interest here are one-way, number-on-the-forehead (NOF) protocols, where PLR_i, the ith player, sees all inputs *except* the ith piece, and where each player sends a single message, and the messages are sent in the order $\text{PLR}_1, \text{PLR}_2, \ldots, \text{PLR}_{k-1}$. PLR_k outputs the answer. A sufficiently strong lower bound for the communication complexity of deterministic protocols for MPJ_k would have major implications in circuit complexity. Specifically, it would place MPJ_k outside of ACC^0.

The complexity class ACC^0 comprises all polynomial-size, constant-depth circuits that use only AND and OR gates. It is easy to show via a counting argument that there exist boolean functions that lie outside this class; however, proving that an explicit function lies outside

ACC^0 remains a tantalizing open problem. A line of research in the early nineties [Yao90, HG91, BT94] showed that if $f \in \mathsf{ACC}^0$ then f has a polylogarithmic NOF communication protocol for a polylogarithmic number of players. Thus, showing a $n^{\Omega(1)}$ lower bound on MPJ_k for any $k = \text{ polylog}(n)$ players is enough to place MPJ_k outside ACC^0.

In Chapter 2, we give a surprising sublinear $o(n)$ *upper bound* for MPJ_k for all $k \geq 3$. Our protocol builds on the work of Pudlák, Rödl, and Sgall [PRS97], who give an $o(n)$ protocol for $\text{MPJ}_3^{\text{perm}}$, a restricted version of the problem where there are only three players and f_2, the middle piece of the input is restricted to be a permutation. This contradicts a long-standing $\Omega(n)$ lower bound conjecture. Intuitively, an $\Omega(n)$ lower bound seemed reasonable, because while each player knows about most of the input, he doesn't know which piece of input is important. From any player's point of view, there are n different pieces of information he could send information about. It was reasonable to expect that to make any progress, each player would need to send $\Omega(1)$ information about each piece, and hence the communication would be $\Omega(n)$. Indeed, in the case of two players, this $\Omega(n)$ bound was proved in [Abl96]. Our upper bound refutes the intuition that this lower bound extends to MPJ_k for $k > 2$ players. In Section 2.4, we consider the communication complexity of several restricted protocols that compute MPJ_k. In a *myopic* protocol, each PLR_i is allowed to see all behind him, but only a single layer ahead of him; that is, he sees only inputs $1, \ldots, i-1$, and layer $i+1$. In a collapsing protocol, PLR_i sees only the composition of layers of inputs, i.e., PLR_j sees $f_{j-1} \circ \cdots \circ f_2(i)$ and $x \circ f_{k-1} \circ \cdots \circ f_{j+1}$. In these restricted models of communication, we show very strong lower bounds. Specifically, we show that in any myopic protocol, some player must send $n/2$ bits, and that in any collapsing protocol, some player must send at least $n - O(\log n)$ bits. A portion of these results is joint work with Amit Chakrabarti.

Distributed Functional Monitoring. In this problem, each of a series of k sensors receives a stream of data and communicates with a central coordinator, who wants to continuously monitor a function of the global state of the system. The goal is to minimize the amount

of communication sent between the coordinator and the sensors. This problem captures the main cost bottleneck in modern wireless sensor networks—sensors are now strong enough to perform significant computation locally; but power is costly, and communicating with a central server drains the battery. Cormode, Muthukrishnan, and Yi [CMY08] formalized the model and gave upper and lower bounds for monitoring frequency moments. In Chapter 3, we extend their work, by providing a suite of lower bounds, and by considering nonmonotonic functions. In particular, we provide a general adversarial lower bound technique and use it to show that for many nonmonotonic functions, the sensors can essentially do nothing better than send all input to the coordinator whenever it arrives. In [ABC09], we give an efficient protocol for monitoring empirical entropy, thus showing that for other nonmonotonic functions, good upper bounds are possible.

Gap Hamming Distance. In the Gap Hamming Distance problem (GHD), Alice and Bob each have n-bit strings, and they wish to determine whether the Hamming distance between their inputs is large (i.e., more than $n/2 + \sqrt{n}$) or small (i.e., less than $n/2 - \sqrt{n}$). The problem is interesting because of its connection to a host of data stream problems. Indyk and Woodruff [Woo04, IW03] defined the problem, made the connection to streaming algorithms, and gave an $\Omega(n)$ lower bound for randomized one-way protocols. As a result, Indyk and Woodruff achieved tight space lower bounds for one-pass streaming algorithms that approximate the kth frequency moment F_k of a data stream. Unfortunately, because the lower bounds for GHD held only for one-way protocols, the lower bound said nothing about multiple-pass streaming algorithms. Proving lower bounds for multiple-round protocols that computed GHD was a long-standing open problem, both in communication complexity and in streaming algorithms. In Chapter 4, we show an $\Omega(n)$ lower bound for GHD for any $O(1)$-round protocol. As a result, we extend the one-pass streaming lower bound to a lower bound for algorithms that use a constant number of passes. Later, in Chapter 5, we give a new lower bound, exponentially improving the lower bound's dependence on the number of rounds. The proof of the new lower

bound is also much simpler. These results are joint work with Amit Chakrabarti. Chapter 5 is also joint work with Oded Regev, Thomas Vidick, and Ronald de Wolf.

1.2 Main Technical Contribution

From a technical perspective, an important contribution of this thesis is to prove strong Round Elimination Lemmas (RELs). The Round Elimination Lemma has been an important and well known technique in the literature for proving lower bounds in communication complexity.

The classic round elimination lemma was implicit in [Ajt88, Xia92, Mil94] and formalized in the work of Miltersen, Nisan, Safra, and Wigderson [MNSW98]. Their formulation is for randomized protocols and works for *any* boolean function $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$. The lemma works by generalizing f to a direct-sum style problem $P_m(f)$, where Alice's input consists of m strings $x_1, \ldots, x_m \in \mathcal{X}$ and Bob's input consists of $y \in \mathcal{Y}$, an index $i \in [m]$, and $x_1, \ldots, x_{i-1}$, and the goal is to compute $f(x_i, y)$.

Miltersen et al. showed that if there is a k-round protocol for $P_m(f)$ in which Alice speaks first and sends much fewer than m bits, then there is a $(k-1)$-round protocol for f in which Bob speaks first and the error is not *too* much more than in $P_m(f)$. Intuitively, since Alice sends $\ll m$ bits, she must not reveal much information about x_j for *some* j. By fixing Alice's first message and this j, the problem $f(x_j, y)$ remains difficult. The error increase in the elimination lemma of Miltersen et al. was multiplicative—the error increased from δ to $O(\sqrt{\delta})$. Later, Sen and Venkatesh [SV08] gave an improved round elimination lemma that increased the error by a small additive term.

Round elimination lemmas have been particularly effective in proving space/query lower bounds for static data structures and space/pass tradeoffs in streaming algorithms. Ajtai [Ajt88] used a round elimination argument to get cell probe lower bounds for the static predecessor problem. Subsequent authors ([Xia92, MNSW98, BF99, SV08]) gave improved bounds.

Pătraşcu and Thorup [PT06] used a *cell probe elimination* lemma to obtain optimal query/space tradeoffs for static data structures that support predecessor queries. Chakrabarti [Cha07] and Viola and Wigderson [VW07] each used *party elimination* lemmas to prove lower bounds on the one-way complexity of MPJ_k. Guha and McGregor [GM08] employ *pass* elimination to attain pass/space tradeoffs for streaming algorithms. While many of these results go by different names (probe elimination, party elimination, pass elimination, ...) all of them fall under the rubric of *elimination lemmas*. Other examples include [CCGL03, Gol09, PV10].

Our Contributions. Our round elimination lemmas depart from the direct-sum structure of the classical elimination lemma. Instead, we focus on a more general notion that if there is a "good" k-round protocol, then there is a "good" $(k-1)$-round protocol. Then, we reduce to a base case that has a few number (often zero) of rounds and derive a contradiction by proving that there are no "good" protocol with that many rounds. What it means for a protocol to be "good" depends on the problem, and identifing the nature of the goodness is half the battle and often requires generalizing the problem. We note our contributions below.

In Chapter 2, we generalize the multiparty pointer jumping problem to work on settings where $i \in [m]$ for some $m \leq n$, and the second piece of the input is a function $f_2 : [m] \to [n]$. In our round elimination step, we decrease m as we reduce the number of players. This result stands out because the communication complexity we achieve is tight up to second-order terms, in contrast to the classic round elimination lemmas, which lose constant factors (or more) each time the lemma is invoked.

In Chapter 4, we use a round elimination lemma to solve the Gap Hamming Distance problem, solving as a result a long-standing open problem in data stream algortihms. While we do not employ a direct-sum style elimination argument, we fail to avoid the exponential dependence on k that commonly occurs with such arguments. In fact, our dependence on k is actually much *worse* than exponential—we show that any k-round protocol must have communication at least $n/2^{O(k^2)}$. In Chapter 5, we prove a new, simpler, round elimination

lemma that yields an $\tilde{\Omega}(n/k^2)$ lower bound. This improves on previous bounds for all $k = O(n^{1/4}/\log n)$ and is the first round elimination argument for general protocols where the lower bound is nontrivial for $k = \omega(\log n)$ rounds. This result departs from the standard round elimination lemma in another important aspect. Typical round elimination arguments construct a $(k-1)$-round protocol by starting with a k-round protocol and fixing the first message to maximize the size of the "message set", i.e., to maximize the number of inputs on which Alice sends this message. Then, a $(k-1)$-round protocol is created by leveraging the size of this first set. Our lemma crucially exploits the *geometry* of the input space in addition to the size of the first set.

Chapter 2

Multiparty Pointer Jumping

We study the one-way number-on-the-forehead (NOF) communication complexity of the k-layer pointer jumping problem. This classic problem has connections to many areas of complexity theory. Perhaps most importantly, a sufficiently strong lower bound on the communication complexity would place multi-party pointer jumping outside ACC^0. A burst of recent research suggested an $\Omega(n)$ lower bound for constant k. Our first result is a surprising $o(n)$ upper bound for the problem that holds for all $k \geq 3$, dashing hopes for such a lower bound.

On the lower bound side, we first consider *myopic* protocols, where players see only one layer ahead, but can still see arbitrarily well behind. We show an exact lower bound of n bits in this model. Furthermore, when a protocol is charged for the *maximum* communication sent by a single player rather than the *total* communication, we show a $n/2$-bit lower bound, independent of the number of players, as well as a $n \log^{(k-1)} n(1 - o(1))$ lower bound for a (non-Boolean) version of the problem. Both of these bounds are tight up to $(1 \pm o(1))$ factors.

Finally, a closer look at the protocol that achieves our upper bound shows that all but one of the players are *collapsing*, i.e. their messages depend only on the composition of the layers ahead of them. We consider protocols where *all* players are collapsing and show a strong $n - O(\log n)$ maximum communication lower bound holds in this case.

2.1 Introduction

Multi-party communication complexity in general, and the *pointer jumping* problem (also known as the *pointer chasing* problem) in particular, has been the subject of plenty of recent research. This is because the model, and sometimes the specific problem, bears on several aspects of computational complexity: among them, circuit complexity [Yao90, HG91, BT94], proof size lower bounds [BPS05], space lower bounds for streaming algorithms [AMS99, GM07, CJP08], and space/time tradeoffs for dynamic data structures [Păt10]. The most impressive known consequence of a strong multi-party communication lower bound would be to exhibit non-membership in the complexity class ACC^0; details can be found in Beigel and Tarui [BT94] or in the textbook by Arora and Barak [AB07]. Vexingly, it is not even known whether or not $\mathsf{ACC}^0 \supseteq \mathsf{NEXP}$.

The setting of multi-party communication is as follows. There are k players (for some $k \geq 2$), whom we shall call $\text{PLR}_1, \text{PLR}_2, \ldots, \text{PLR}_k$, who share an input k-tuple $(x_1, x_2, \ldots, x_k)$. The goal of the players is to compute some function $f(x_1, x_2, \ldots, x_k)$. There are two well-studied sharing models: the *number-in-hand* model, where PLR_i sees x_i, and the *number-on-the-forehead* (NOF) model, where PLR_i sees all x_j such that $j \neq i$. Our focus in this chapter will be on the latter model, which was first introduced by Chandra, Furst and Lipton [CFL83]. It is in this model that communication lower bounds imply lower bounds against ACC^0. We shall use $C(f)$ to denote the deterministic communication complexity of f in this model. Also of interest are randomized protocols that only compute $f(x)$ correctly with high probability: we let $R_\varepsilon(f)$ denote the ε-error randomized communication complexity of f. Most of our results hold for deterministic protocols, which is a strength for our upper bounds. Moreover, it is not a serious weakness for our lower bounds, because the ACC^0 connection only calls for a deterministic lower bound.

Notice that the NOF model has a feature not seen elsewhere in communication complexity:

the players *share* plenty of information. In fact, for large k, each individual player already has "almost" all of the input. This intuitively makes lower bounds especially hard to prove and indeed, to this day, no nontrivial lower bound is known in the NOF model for any explicit function with $k = \omega(\log n)$ players, where n is the total input size. The pointer jumping problem is widely considered to be a good candidate for such a lower bound. As noted by Damm, Jukna and Sgall [DJS98], it has many natural special cases, such as shifting, addressing, multiplication and convolution. This motivates our study.

As a further motivation, Pătraşcu [Păt10] recently showed that sufficiently strong lower bounds for a 3-person NOF protocol for a version of set disjointness would imply polynomial query lower bounds for a large range of dynamic data structures. Only $\Omega(\log n)$ bounds are currently known.

2.1.1 The Pointer Jumping Problem and Previous Results

There are a number of variants of the pointer jumping problem. Here we study two variants: a Boolean problem, MPJ_k^n, and a non-Boolean problem, $\widehat{\text{MPJ}}_k^n$ (henceforth, we shall drop the superscript n). In both variants, the input is a subgraph of a fixed layered graph that has $k+1$ layers of vertices, with layer 0 consisting of a single vertex, v_0, and layers 1 through $k-1$ consisting of n vertices each (we assume $k \geq 2$). Layer k consists of n vertices in the case of $\widehat{\text{MPJ}}_k$ and 2 vertices in the case of MPJ_k. The input graph is a subgraph of the fixed layered graph in which every vertex (except those in layer k) has outdegree 1. The desired output is the name of the unique vertex in layer k reachable from v_0, i.e., the final result of "following the pointers" starting at v_0. The output is therefore a single bit in the case of MPJ_k or a $\lceil \log n \rceil$-bit string in the case of $\widehat{\text{MPJ}}_k$.[1]

The functions MPJ_k and $\widehat{\text{MPJ}}_k$ are made into NOF communication problems as follows: for each $i \in [k]$, a description of the ith layer of edges (i.e., the edges pointing into the ith layer of

[1] Throughout this chapter we use "log" to denote logarithm to the base 2.

vertices) is written on PLR_i's forehead. In other words, PLR_i sees every layer of edges except the ith. The players are allowed to write one message each on a public *blackboard* and must do so in the fixed order $\text{PLR}_1, \text{PLR}_2, \ldots, \text{PLR}_k$. The final player's message must be the desired output. Notice that the specific order of speaking—$\text{PLR}_1, \text{PLR}_2, \ldots, \text{PLR}_k$—is important to make the problem nontrivial. Any other order of speaking allows an easy deterministic protocol with only $O(\log n)$ communication.

Consider the case $k = 2$. The problem MPJ_2 is equivalent to the two-party communication problem INDEX, where Alice holds a bit-vector $x \in \{0,1\}^n$, Bob holds an index $i \in [n]$, and Alice must send Bob a message that enables him to output x_i. It is easy to show that $C(\text{MPJ}_2) = n$. In fact, Ablayev [Abl96] shows the tight tradeoff $R_\varepsilon(\text{MPJ}_2) = (1 - H(\varepsilon))n$, where H is the binary entropy function. It is tempting to conjecture that this lower bound generalizes as follows.

Conjecture 1. *There is a nondecreasing function $\xi : \mathbb{Z} \to \mathbb{R}^+$ such that, $\forall\, k :\ C(\text{MPJ}_k) = \Omega(n/\xi(k))$.*

Note that, by the results of Beigel and Tarui [BT94], in order to show that $\text{MPJ}_k \notin \mathsf{ACC}^0$ it would suffice, for instance, to prove the following (possibly weaker) conjecture.

Conjecture 2. *There exist constants $\alpha, \beta > 0$ such that, for $k = n^\alpha$, $C(\text{MPJ}_k) = \Omega(n^\beta)$.*

Conjecture 1 is consistent with (and to an extent motivated by) research prior to this work. In weaker models of information sharing than the NOF model, an equivalent statement is known to be true, even for randomized protocols. For instance, Damm, Jukna and Sgall [DJS98] show an $\Omega(n/k^2)$ communication lower bound in the so-called *conservative* model, where PLR_i has only a limited view of the layers of the graph behind her: she only sees the result of following the first $i-1$ pointers. Chakrabarti [Cha07] extends this bound to randomized protocols and also shows an $\Omega(n/k)$ lower bound in the so-called *myopic* model, where PLR_i has only a limited view of the layers ahead of her: she cannot see layers $i+2, \ldots, k$.

This greatly improved a lower bound of Gronemeier [Gro06], who defined the myopic model and gave a weak lower bound for myopic MPJ_k protocols.

For the full NOF model, Wigderson, building on the work of Nisan and Wigderson [NW93], showed that $C(\text{MPJ}_3) = \Omega(\sqrt{n})$. This result is unpublished, but an exposition can be found in Babai, Hayes and Kimmel [BHK01]. Recently, Viola and Wigderson [VW07] generalized this result and extended it to randomized protocols, showing that $R_{1/3}(\text{MPJ}_k) = \Omega(n^{1/(k-1)}/k^{O(k)})$. Of course, this bound falls far short of that in Conjecture 1 and does nothing for Conjecture 2. However, it is worth noting that the Viola-Wigderson bound in fact applies to the much smaller subproblem of *tree pointer jumping* (denoted TPJ_k), where the underlying layered graph is a height-k tree, with every vertex in layers 0 through $k-2$ having $n^{1/(k-1)}$ children and every vertex in layer $k-1$ having two children. It is easy to see that $C(\text{TPJ}_k) = O(n^{1/(k-1)})$. Thus, one might hope that the more general problem MPJ_k has a much stronger lower bound, as in Conjecture 1.

On the upper bound side, Damm et al. [DJS98] have shown that $C(\widehat{\text{MPJ}}_k) = O(n \log^{(k-1)} n)$, where $\log^{(i)} n$ is the ith iterated logarithm of n. This improves on the trivial upper bound of $O(n \log n)$. Their technique does not yield anything nontrivial for the Boolean problem MPJ_k, though. However, Pudlák, Rödl and Sgall [PRS97] obtain an amazing sublinear upper bound of $O(n \log\log n / \log n)$ for a special case of MPJ_3. Their protocol works only when every vertex in layer 2 has *indegree* 1, or equivalently, when the middle layer of edges in the input describes a *permutation* of $[n]$. It is remarkable that even this is possible. The proof of the $O(n \log\log n / \log n)$ upper bound is deep, and our upper bound builds upon it. For this reason, we include a development of the protocol and proof of it's correctness in Section 2.3.1.

2.1.2 Our Results

The protocol of Pudlák et al. [PRS97] did not rule out Conjecture 1, but it did suggest caution. Our first result is the following upper bound—in fact the first nontrivial upper bound on

$C(\text{MPJ}_k)$—that falsifies the conjecture.

Theorem 3. *For $k \geq 3$, we have*

$$C(\text{MPJ}_k) \;=\; O\left(n\left(\frac{k\log\log n}{\log n}\right)^{(k-2)/(k-1)}\right).$$

In particular, $C(\text{MPJ}_3) = O(n\sqrt{\log\log n/\log n})$.

Next, we use a protocol for MPJ_{k-1} to obtain a sublinear upper bound for $\widehat{\text{MPJ}}_k$, at a cost of an extra $\log\log n$ factor in communication.

Theorem 4. *For $k \geq 4$, we have*

$$C(\widehat{\text{MPJ}}_k) \;=\; O\left(n\log\log n\left(\frac{k\log\log n}{\log n}\right)^{(k-3)/(k-2)}\right).$$

In particular, $C(\widehat{\text{MPJ}}_4) = O(n\log\log n\sqrt{\log\log n/\log n})$.

We next provide a suite of lower bounds for myopic protocols. Our first lower bound shows that in terms of total communication, no nontrivial myopic protocol is possible.

Theorem 5. *In a myopic protocol for MPJ_k, at least n bits must be communicated in total.*

This stands in contrast to the state of affairs for $\widehat{\text{MPJ}}_k$. The only nontrivial protocol for $\widehat{\text{MPJ}}_k$ prior to our work was the aforementioned protocol of Damm et al. [DJS98] (see Section 2.1.1), which was both conservative *and* myopic.

We also provide a strong lower bound on the maximum communication of MPJ_k.

Theorem 6. *In a myopic protocol for MPJ_k, some player must communicate at least $n/2$ bits.*

We prove the above lower bounds on a generalized version of MPJ_k where there are m vertices in the first layer instead of n.

A closer look at of the proof of Theorem 6 shows that there exists a decreasing function $\phi : \mathbb{Z} \to [1/2, 1]$ such that any myopic protocol has maximum communication $n\phi(k)$.[2] Our next result shows that this is essentially tight.

Theorem 7. *For all $k \geq 3$, there exists a myopic protocol for MPJ_k in which each player sends $n\phi(k)(1 + o(1))$ bits.*

We also apply our lower bound technique to myopic protocols for $\widehat{\text{MPJ}}_k$, and prove a lower bound which nearly matches the upper bound of Damm et al.

Theorem 8. *In any deterministic myopic protocol for $\widehat{\text{MPJ}}_k$, some player must communicate at least $n\left(\log^{(k-1)} n - \log^{(k)} n\right)$ bits.*

Next, we show a lower bound for randomized protocols.

Theorem 9. *In any randomized myopic protocol for MPJ_k, some player must communicate at least $\Omega(n/(k \log n))$ bits.*

Chakrabarti's $\Omega(n/k)$ lower bound on the total communication immediately yields an $\Omega(n/k^2)$ maximum communication bound. Our bound is therefore an improvement for all $k \geq \log n$.

A closer look at the protocol that achieves the upper bound from Theorem 3 reveals that all players except for PLR_1 behave in the following way: the message sent by PLR_i depends only on layers 1 through $i - 1$ and the composition of layers $i + 1$ through k. We say that PLR_i is *collapsing*. This notion is akin to that of the aforementioned conservative protocols considered by Damm et al. Whereas a conservative player composes the layers behind hers, a collapsing player does so for layers ahead of hers.

Our final results consider what happens if we require *all* players in the protocol to be collapsing. We prove a strong linear lower bound, showing that even a single non-collapsing player makes an asymptotic difference in the communication complexity.

[2]The precise definition of $\phi(k)$ is somewhat technical; we defer it until Section 2.4.

Theorem 10. *In a protocol for* MPJ_k *where every player is collapsing, some player must communicate at least* $n - \frac{1}{2}\log n - 2 = n - O(\log n)$ *bits.*

One might wonder whether the collapsing requirement is so strong that nothing nontrivial is possible anyway. We show an upper bound similar to that of Damm et al. for $\widehat{\text{MPJ}}_k$, but for protocols where every player is collapsing.

Theorem 11. *For* $k \geq 3$*, there is an* $O(n \log^{(k-1)} n)$*-communication protocol for* $\widehat{\text{MPJ}}_k^{\text{perm}}$ *in which every player is collapsing. Here* $\widehat{\text{MPJ}}_k^{\text{perm}}$ *denotes the subproblem of* $\widehat{\text{MPJ}}_k$ *in which layers* 2 *through* k *of the input graph are permutations of* $[n]$*.*

The requirement that layers be permutations is a natural one and is not new. The protocol of Pudlák et al. also had this requirement; i.e., it gave an upper bound on $C(\text{MPJ}_3^{\text{perm}})$. Theorem 11 can in fact be strengthened slightly by allowing one of the layers from 2 through k to be arbitrary; we formulate and prove this stronger version in Section 2.8.

2.1.3 Relation to Dynamic Data Structures

Recently, Pătraşcu [Păt10] gave a three player NOF communication problem whose conjectured lower bound implies polynomial lower bounds for a host of dynamic data structure problems, including subraph connectivity, dynamic shortest paths, and dynamic reachability.

The following conjecutre captures the essence of the communication problem.

Conjecture 12 ([Păt10, Conjecture 9**]).** *Consider a* 3*-party number-on-the-forehead game in which Alice holds* $i \in [k]$*, Bob holds* $y_1, \ldots, y_k \in \mathcal{Y}$*, and Carol holds* $x \in \mathcal{X}$*. The goal is to compute* $g(x, y_i)$*, for some arbitrary* $g : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$*. If there is a protocol in which Alice begins with a private message to Bob of* $o(k)$ *bits, followed by* M *bits of bidirectional communication between Bob and Carol, then the* 2*-party communication complexity of* g *is* $O(M)$*.*

The intuition captured in this conjecture comes from the direct-sum style round elimination lemmas of [MNSW98] and [SV08] and proceeds as follows. Since Alice transmits $o(k)$ bits, there must be some i such that she reveals $o(1)$ information about y_i. Fixing Alice's message and this i leaves a nontrivial problem for g. If Alice's message depended only on $y_1, \ldots, y_k$, then a straightforward elimination result would hold. Fixing Alice's message and i, we can create a two-party protocol for g between Bob (holding x) and Carol (holding y) as follows: Carol sets $y_i := y$ and chooses values for $y_j \, (j \neq i)$ such that $y_1, \ldots, y_k$ is consistent with Alice's fixed message. Then, Carol and Bob can compute $g(x, y)$ by following the remainder of the three-party protocol.

Unfortunately, this conjecture does not immediately follow, because Alice's message depends on x as well as $y_1, \ldots, y_k$. As a result, neither Bob nor Carol has enough information to extend the inputs $(x, y) \Rightarrow (x, y_1, \ldots, y_k)$ in a way that is consistent with the fixed message.

Pătraşcu notes this but states "the information theoretic intuition of the (round elimination) lemma holds, and it is conceivable that the message of Alice can be eliminated in a black-box fashion for any communication problem of the appropriate direct sum structure."

Our upper bound for MPJ_3 refutes this statement. Specifically, we set $k := n$ and $\mathcal{Y} := [n]$ and view $\{y_i\}$ as the layer of edges (so y_i maps the ith vertex in layer 1 to the y_ith vertex in layer 2). Then, computing MPJ_3 amounts to computing $g(x, y_i) := x[y_i]$, and therefore the resulting two player problem is merely INDEX.

If the direct-sum intuition were to hold *in a black box fashion*, then our protocol for MPJ_3 in which each player sends $M := o(n)$ bits would result in a protocol for INDEX with communication $O(M) = o(n)$, contradicting the $\Omega(n)$ lower bound of Ablayev [Abl96]. Conjecture 12, as stated, might still hold. However, proving the conjecture will require exploiting the bidirectional nature of the underlying 2-party protocol.

2.1.4 Organization

The rest of the chapter is organized as follows. Section 2.2 introduces some notation that is used in subsequent sections. Theorems 3 and 4 are proven in Section 2.3. Theorems 5, 6, and 8 are proven in Section 2.4. Theorems 7, 9, 10, and 11 are proven in Sections 2.4, 2.6, 2.7, and 2.8 respectively. Finally, Section 2.9 concludes the chapter and discusses open problems.

2.2 Preliminaries and Notation

For the rest of the chapter, "protocols" will be assumed to be deterministic one-way NOF protocols unless otherwise qualified. Let $\mathcal{P}$ be a k-player protocol in which player i's message has length ℓ_i. Define $\mathrm{cost}(\mathcal{P})$ and $\mathrm{mcost}(\mathcal{P})$ as

$$\mathrm{cost}(\mathcal{P}) := \sum_{i=1}^{k} \ell_i \,,$$

$$\mathrm{mcost}(\mathcal{P}) := \max_{1 \le i \le k} \ell_i \,.$$

In this way, $\mathrm{cost}(\mathcal{P})$ denotes the *total* communication sent in a protocol whereas $\mathrm{mcost}(\mathcal{P})$ denotes the *maximum* communication sent by any one player.

We now formally define the generalized pointer jumping problems $\textsc{mpj}^n_{m,k}$ and $\widehat{\textsc{mpj}}^n_{m,k}$ in a recursive fashion. We define $\textsc{mpj}^n_{m,2} : [m] \times \{0,1\}^m \to \{0,1\}$ as $\textsc{mpj}^n_{m,2}(i,x) := x_i$, where x_i denotes the ith bit of the string x. In a similar fashion, we define $\widehat{\textsc{mpj}}^n_{m,2} : [m] \times [n]^{[m]} \to [n]$ as $\widehat{\textsc{mpj}}^n_{m,2}(i, f_2) := f_2(i)$. For $k \ge 3$ we then define $\textsc{mpj}^n_{m,k} : [m] \times [n]^{[m]} \times \left([n]^{[n]}\right)^{k-3} \times \{0,1\}^n \to \{0,1\}$ and $\widehat{\textsc{mpj}}^n_{m,k} : [m] \times [n]^{[m]} \times \left([n]^{[n]}\right)^{k-2} \to [n]$ as follows.

$$\textsc{mpj}^n_{m,k}(i, f_2, f_3, \ldots, f_{k-1}, x) := \textsc{mpj}^n_{n,k-1}(f_2(i), f_3, \ldots, f_{k-1}, x)\,, \text{ for } k \ge 3\,,$$

$$\widehat{\textsc{mpj}}^n_{m,k}(i, f_2, f_3, \ldots, f_k) := \widehat{\textsc{mpj}}^n_{n,k-1}(f_2(i), f_3, \ldots, f_k)\,, \text{ for } k \ge 3\,.$$

It will be helpful, at times, to view strings in $\{0,1\}^n$ as functions from $[n]$ to $\{0,1\}$ and use functional notation accordingly. Unrolling the recursion in the above definitions, we see that, for $k \geq 2$,

$$\text{MPJ}^n_{m,k}(i, f_2, \ldots, f_{k-1}, x) \;=\; x \circ f_{k-1} \circ \cdots \circ f_2(i)\,; \tag{2.1}$$

$$\widehat{\text{MPJ}}^n_{m,k}(i, f_2, \ldots, f_k) \;=\; f_k \circ \cdots \circ f_2(i)\,. \tag{2.2}$$

It is often useful to discuss the composition of certain subsets of the inputs. Let $\hat{i}_2 = i$, and for $3 \leq j \leq k$, let $\hat{i}_j = f_{j-1} \circ \cdots \circ f_2(i)$. Similarly, let $\hat{x}_{k-1} = x$, and for $1 \leq j \leq k-2$, let $\hat{x}_j = x \circ f_{k-1} \circ \cdots \circ f_{j+1}$.

Henceforth, we shall drop the superscript n. The most natural formulation of this problem has $m = n$; in this case, we drop the subscript m as well.

Previous work on multiplayer pointer jumping considered only MPJ_k and $\widehat{\text{MPJ}}_k$. In Section 2.4 we prove Theorems 5, 6, and 8 by performing round elimination on $\text{MPJ}_{m,k}$ or $\widehat{\text{MPJ}}_{m,k}$ and shrinking m at each step.

We also consider the subproblems $\text{MPJ}^{\text{perm}}_k$ and $\widehat{\text{MPJ}}^{\text{perm}}_k$ where each f_j above is a bijection from $[n]$ to $[n]$ (equivalently, a permutation of $[n]$). We let $\mathcal{S}_n$ denote the set of all permutations of $[n]$.

2.3 Sublinear Upper Bounds

Here is a rough plan of the proof of our sublinear upper bound. We leverage the fact that a protocol P for $\text{MPJ}^{\text{perm}}_3$ with sublinear communication is known. To be precise:

Fact 13 ([PRS97, Corollary 4.8]). $C(\text{MPJ}^{\text{perm}}_3) = O(n \log\log n / \log n)$.

The exact structure of P will not matter; we shall only use P as a black box. To get a sense for why P might be useful for, say, MPJ_3, note that the players could replace f_2 with a permutation π and just simulate P, and this would work if $\pi(i) = f(i)$. Of course, there is

no way for PLR_1 and PLR_3 to agree on a suitable π without communication. However, as we shall see below, it is possible for them to agree on a small enough *set* of permutations such that either some permutation in the set is suitable, or else only a small amount of side information conveys the desired output bit to PLR_3.

This idea eventually gives us a sublinear protocol for MPJ_3. Clearly, whatever upper bound we obtain for MPJ_3 applies to MPJ_k for all $k \geq 3$. However, we can decrease the upper bound as k increases, by embedding several instances of MPJ_3 into MPJ_k. Finally, we achieve an upper bound for $\widehat{\text{MPJ}}_k$ by leveraging several instances of MPJ_{k-1}. For clarity, we first give a complete proof of Theorem 3 for the case $k = 3$. Then we give a proof of Theorem 3 for general k, before proving Theorem 4.

2.3.1 The PRS Protocol

In this section, we present the $\text{MPJ}_3^{\text{perm}}$ protocol of Pudlák, Rödl and Sgall [PRS97]. In the next section, we'll leverage this upper bound to get a sublinear protocol for MPJ_3.

Theorem 14 ([PRS97, Corollary 4.8]). $C(\text{MPJ}_3^{\text{perm}}) = O(n \log\log n / \log n)$.

Remark. *The proof of Theorem 14 is technical. Furthermore, we use it as a black box, so details of this proof are not necessary to understand the rest of the chapter. The reader is encouraged to skip details of this proof if necessary.*

Proof. For this proof, it will be helpful to think of $\text{MPJ}_3^{\text{perm}}$ as a problem on a bipartite graph $G = (A \cup B, E)$ where π describes the edges in G (i.e., if $b = \pi(a)$ then $(a, b) \in E$), x describes a two-coloring of B, i describes a distinguished start vertex $a \in A$, and the goal is to output the color of $\pi(a)$. For a set of vertices $B' \subseteq B$, we define the *parity* $P_{B'}$ in the natural way: arbitrarily associate one color with "0", the second color with "1", and define

$$P_{B'} := \left(\sum_{b \in B'} x[b] \right) \mod 2 .$$

At a high level, the PRS protocol works in the following way. First, PLR_1, using π, partitions B into clusters $C_1, \ldots, C_r$. Crucially, the construction of this partition will depend only on π; we defer a more complete description of this construction to later in the proof. PLR_1 then sends the parity of each cluster. PLR_2 sees a and sends the color of several $b \in B$. Importantly, PLR_2 sends $x[b]$ for *every* b in the cluster containing $\pi(a)$, except possibly for $\pi(a)$ itself. PLR_3 recovers $x[\pi(a)]$ by selecting the cluster containing $\pi(a)$ and "XOR-ing" out the other bits in the cluster; he outputs $x[\pi(a)] = \left(P_C + \sum_{b \in C \setminus \{\pi(a)\}} x[b]\right) \mod 2$.

The magic in this protocol comes from how PLR_1 chooses the clusters and how PLR_2 chooses which b to send information about, such that for all $a \in A$ and all $\pi \in S_n$, PLR_3 has enough information to recover the answer.

Pudlák, Rödl and Sgall show this is possible using the probabilistic method. Specifically, let H be a random bipartite graph on $A \cup B$, where each edge (a, b) is selected with some probability p to be determined later. Next, we'll construct a protocol $\mathcal{P}_H$ for $\text{MPJ}_3^{\text{perm}}$ based on this random graph. Finally, we'll show that with nonzero probablity, $\mathcal{P}_H$ is an efficient protocol, i.e., that for each a and each π, only $O(n \log\log n / \log n)$ bits need to be sent.

Given H, PLR_2's message is simple: he sends the color of each neighbor of a in H. In other words, PLR_2 sends $x[b]$ for each b such that $(a, b) \in E(H)$. PLR_1's message is less simple to construct. First, he constructs a graph G_π on B, and he puts $(b, b') \in E(G_\pi)$ if and only if $(\pi^{-1}(b), b')$ and $(\pi^{-1}(b'), b)$ are edges in H. Next, we let $\mathcal{C}$ be a clique cover of G_π. The cliques in this cover are the clusters of vertices in B that PLR_1 uses in his message.

Now, it becomes easy to see why $\mathcal{P}_H$ correctly computes $\text{MPJ}_3^{\text{perm}}$. Let $C \in \mathcal{C}$ be the clique containing $\pi(a)$, and let b be some other element of C. Since C is a clique, $(b, \pi(a))$ is an edge of G_π, and therefore by construction $(a, b) \in H$. But since (a, b) is an edge in H, PLR_2 sends $x[b]$. Thus, we see that PLR_2 sends $x[b]$ for each $b \in C$ not including $\pi(a)$, giving PLR_3 enough information to recover $x[\pi(a)]$.

To recap, the protocol of Pudlák, Rödl, and Sgall works as follows:

- PLR_1 sees π and x. Using H, he creates a graph G_π. PLR_1 sends the parity P_C of each clique C in the clique cover $\mathcal{C}$ of G_π.

- PLR_2 sees a and x. Using H, he sends $x[b]$ for all $b \in B$ such that (a, b) is an edge in H.

- PLR_3 sees a and π and must output $x[\pi(a)]$. Using H, he creates the graph G_π. PLR_3 then computes the clique C containing $\pi(a)$. Finally, PLR_3 takes the parity P_C from PLR_1's message and the colors $x[b]$ of all $b \in C \setminus \{\pi(a)\}$ and outputs $x[\pi(a)]$ by computing

$$\begin{aligned}\left(P_C + \sum_{b \in C \setminus \{\pi(a)\}} x[b]\right) \mod 2 &= \left(\sum_{b \in C} x[b] + \sum_{b \in C \setminus \{\pi(a)\}} x[b]\right) \mod 2 \\ &= x[\pi(a)]\,.\end{aligned}$$

We've seen that $\mathcal{P}_H$ correctly outputs $\text{MPJ}_3^{\text{perm}}(a, \pi, x)$. It remains to show that there exists H such that $\mathcal{P}_H$ *efficiently* computes $\text{MPJ}_3^{\text{perm}}$ for all inputs. Recall the communication in $\mathcal{P}_H$: PLR_1 sends one bit for each cluster in G_π, and PLR_2 sends one bit for each neighbor of a in H. H is a random graph. Each possible edge (a, b) is independently selected with probability p. Therefore, for each $a \in A$, we expect a to have pn neighbors. By a Chernoff bound, there exists a constant $\varepsilon > 0$ such that

$$\Pr[a \text{ has more than } (1+\varepsilon)pn \text{ neighbors}] \leq \exp(-\Omega(np^2))\,. \tag{2.3}$$

Setting $p := \log\log n / \log n$, we see that

$$\begin{aligned}n \cdot \Pr[a \text{ has more than } (1+\varepsilon)pn \text{ neighbors}] &\leq n \cdot \exp\left(-\Omega\left(n(\log\log n/\log n)^2\right)\right) \\ &= o(1)\,.\end{aligned}$$

Next, we consider the number of cliques necessary to cover G_π. Recall that $(b, b') \in E(G_\pi)$

if and only if $(\pi^{-1}(b), b')$ and $(\pi^{-1}(b'), b)$ are edges in H, and that each edge in H is independently selected with probability p. It follows that each (b, b') is randomly selected for $E(G_\pi)$ with probability p^2. Simple inspection shows that for any two candidate edges e_1, e_2 in G_π, the edges from H used to determine whether e_1 and e_2 are edges in G_π are disjoint. Thus, G_π is a random graph, with each edge independently selected with probability p^2.

Pudlák, Rödl, and Sgall determine the number of cliques required to cover G_π by considering the number of independent sets needed to cover its complement $\bar{G}_\pi$. Note that $\bar{G}_\pi$ is a random graph, where each edge is selected with probability $q := 1-p^2$. Consider the chromatic number $\chi(\bar{G}_\pi)$ of the complement graph, and suppose that $r = \chi(\bar{G}_\pi)$. Fix any r-coloring of $\bar{G}_\pi$, and let C_i be the set of vertices of color i. C_i is an independent set in $\bar{G}_\pi$, hence it forms a clique in G_π, and therefore $\mathcal{C} := \{C_1, \ldots, C_r\}$ is a clique cover of G_π. Pudlák, Rödl, and Sgall prove the following technical lemma on the chromatic number of random graphs.

Lemma 15 ([PRS97, Theorem 4.4**]).** *For every $\varepsilon > 0$ there exists $\delta > 0$ and n_0 such that for all $7/8 < q < 1 - 1/n^{1/2-\varepsilon}$ and for all $n \geq n_0$,*

$$\Pr\left[\chi(G(n,q)) \leq (1/2+\varepsilon)\frac{-n\log(1-q)}{\log n}\right] \;>\; 1 - \exp(-n^{1+\delta})\,. \tag{2.4}$$

This lemma is an extension of a result by Bollobás, who considered the problem for constant q; we do not develop it further here.

PLR$_1$ sends one bit for each clique in the clique cover of G_π. Recalling that $q = 1 - p^2$ and

that $p := \log\log n/\log n$, we see that

$$\begin{aligned}(1/2+\varepsilon)(-n\log(1-q)/\log n) &= (1/2+\varepsilon)(-2n\log p/\log n)\\ &= (1/2+\varepsilon)\left(\frac{2n}{\log n}\log\left(\frac{\log n}{\log\log n}\right)\right)\\ &= O\left(\frac{n\log\log n}{\log n}\right).\end{aligned}$$

Thus, we see that if H behaves nicely with π, PLR_1 sends only $O(n\log\log n/\log n)$ bits. We require a graph H such that no vertex a has *too* many neighbors, and no G_π requires a clique cover that is *too* large. Call H bad if either there exists $a \in A$ such that a has more than $(1+\varepsilon)pn$ neighbors or there exists π such that the smallest clique cover of G_π has more than $(1/2+\varepsilon)(-n\log(1-q)/\log n)$ cliques; otherwise, call H good. Using Equations 2.3 and 2.4 and a union bound, we see that

$$\begin{aligned}\Pr[H \text{ is bad }] &\leq n!\cdot\Pr\left[\chi(G(n,1-p^2)) > \left(\frac{1}{2}+\varepsilon\right)\frac{-n\log(1-q)}{\log n}\right]\\ &\quad +n\cdot\Pr[a \text{ has more than } (1+\varepsilon)pn \text{ neighbors }]\\ &< n!\cdot\exp(-n^{1+\delta}) + n\cdot\exp(-\Omega(np^2))\\ &= \exp(n\log n)\cdot\exp(-n^{1+\delta}) + n\cdot\exp(-\Omega(np^2))\\ &= o(1).\end{aligned}$$

Hence there exists a good H and a protocol $\mathcal{P}_H$ where players send $O(n\log\log n/\log n)$ bits each. □

2.3.2 A 3-Player Protocol

Following the plan outlined above, we prove Theorem 3 for the case $k = 3$ by plugging Fact 13 into the following lemma, whose proof is the topic of this section.

Lemma 16. *Suppose $\phi : \mathbb{Z} \to (0, 1]$ is a function such that $C(\text{MPJ}_3^{\text{perm}}) = O(n\phi(n))$. Then $C(\text{MPJ}_3) = O(n\sqrt{\phi(n)})$.*

Definition 1. *A set $\mathcal{A} \subseteq \mathcal{S}_n$ of permutations is said to d-cover a function $f : [n] \to [n]$ if, for each $r \in [n]$, at least one of the following conditions holds:*

(i) $\exists\, \pi \in \mathcal{A}$ *such that* $\pi(r) = f(r)$, *or*

(ii) $|f^{-1}(f(r))| > d$.

Lemma 17. *Let $f : [n] \to [n]$ be a function and d be a positive integer. There exists a set $\mathcal{A}_d(f) \subseteq \mathcal{S}_n$, with $|\mathcal{A}_d(f)| \leq d$, that d-covers f.*

Proof. We give an explicit algortihm to construct $\mathcal{A}_d(f)$. Our strategy for each permutation we construct is to fix a set of pointers that cover the range of f and define the rest of the permutation by arbitrarily matching the remaining elements. To be precise, suppose that $\text{Range}(f) = \{s_1, \ldots, s_t\}$. Let $A_i = f^{-1}(s_i)$ be the corresponding fibers of f. Clearly, $\{A_i\}_{i=1}^t$ partition $[n]$. Let $B := [n] - \text{Range}(f)$ denote the elements *not* in the range of f, and write B as $\{b_1, b_2, \ldots, b_{n-t}\}$.

For each $1 \leq i \leq t$, let $A_i = \{a_{i,1}, \ldots, a_{i,|A_i|}\}$ denote the elements of A_i, and for each i and each $1 \leq j \leq d$, define $c_{i,j} := a_{i,\min\{j,|A_i|\}}$. The jth permutation we construct will map $c_{i,j}$ to s_i. Finally, let $C_j := \{c_{1,j}, c_{2,j}, \ldots, c_{t,j}\}$, and let $D_j = \{d_{1,j}, \ldots, d_{n-t,j}\}$ denote the elements not in C_j. We define permutation π_j in the following manner.

$$\pi_j(r) = \begin{cases} s_i & \text{if } r = c_{i,j}, \\ b_i & \text{if } r = d_{i,j}. \end{cases}$$

By construction, π_j defines a bijection between C_j and $\text{Range}(f)$, and between D_j and B; therefore, π_j permutes $[n]$. It remains to verify that this choice of $\mathcal{A}_d(f)$ d-covers f, i.e., to verify that every $r \in [n]$ satisfies at least one of the two conditions in Definition 1. Pick any

$r \in [n]$. Suppose $r \in A_i$, so that $f(r) = s_i$. If $|A_i| \leq d$ then there exists $1 \leq j \leq d$ such that $a_{i,j} = r$. Therefore, $\pi_j(r) = \pi_j(a_{i,j}) = \pi_j(c_{i,j}) = s_i$ and condition (i) holds. On the other hand, $A_i = f^{-1}(s_i) = f^{-1}(f(r))$, so if $|A_i| > d$ then condition (ii) holds. Either way, the proof is complete. □

Proof of Lemma 16.

Let $(i, \pi, x) \in [n] \times \mathcal{S}_n \times \{0,1\}^n$ denote an input for the problem $\text{MPJ}_3^{\text{perm}}$. Then the desired output is $x_{\pi(i)}$. The existence of a protocol P for $\text{MPJ}_3^{\text{perm}}$ with $\text{cost}(P) = O(n\phi(n))$ means that there exist functions

$$\alpha : \mathcal{S}_n \times \{0,1\}^n \to \{0,1\}^m\,, \quad \beta : [n] \times \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m\,, \quad \text{and}$$
$$\gamma : [n] \times \mathcal{S}_n \times \{0,1\}^m \times \{0,1\}^m \to \{0,1\}\,,$$

where $m = O(n\phi(n))$, such that $\gamma(i, \pi, \alpha(\pi, x), \beta(i, x, \alpha(\pi, x))) = x_{\pi(i)}$. The functions α, β and γ yield the messages in P of PLR_1, PLR_2 and PLR_3 respectively.

To design a protocol for MPJ_3, we first let PLR_1 and PLR_3 agree on a parameter d, to be fixed below, and a choice of $\mathcal{A}_d(f)$ for each $f : [n] \to [n]$, as guaranteed by Lemma 17. Now, let $(i, f, x) \in [n] \times [n]^{[n]} \times \{0,1\}^n$ be an input for MPJ_3. Our protocol works as follows.

- PLR_1 sends a two-part message. The first part consists of the strings $\{\alpha(\pi, x)\}_\pi$ for all $\pi \in \mathcal{A}_d(f)$. The second part consists of the bits x_s for $s \in [n]$ such that $|f^{-1}(s)| > d$.
- PLR_2 sends the strings $\{\beta(i, x, \alpha)\}_\alpha$ for all strings α in the first part of PLR_1's message.
- PLR_3 can now output $x_{f(i)}$ as follows. If $|f^{-1}(f(i))| > d$, then she reads $x_{f(i)}$ off from the second part of PLR_1's message. Otherwise, since $\mathcal{A}_d(f)$ d-covers f, there exists a $\pi_0 \in \mathcal{A}_d(f)$ such that $f(i) = \pi_0(i)$. She uses the string $\alpha_0 := \alpha(\pi_0, x)$ from the first part of PLR_1's message and the string $\beta_0 := \beta(i, x, \alpha_0)$ from PLR_2's message to output $\gamma(i, \pi_0, \alpha_0, \beta_0)$.

To verify correctness, we only need to check that PLR_3's output in the "otherwise" case indeed equals $x_{f(i)}$. By the correctness of P, the output equals $x_{\pi_0(i)}$ and we are done, since $f(i) = \pi_0(i)$.

We now turn to the communication cost of the protocol. By the guarantees in Lemma 17, $|\mathcal{A}_d(f)| \leq d$, so the first part of PLR_1's message is at most dm bits long, as is PLR_2's message. Since there can be at most n/d values $s \in [n]$ such that $|f^{-1}(s)| > d$, the second part of PLR_2's message is at most n/d bits long. Therefore the communication cost is at most $2dm + n/d = O(dn\phi(n) + n/d)$. Setting $d = \lceil 1/\sqrt{\phi(n)} \rceil$ gives us a bound of $O(n\sqrt{\phi(n)})$, as desired. □

2.3.3 A k-Player Protocol

We now show how to prove Theorem 3 by generalizing the protocol from Lemma 16 into a protocol for k players.

Lemma 18. *Let $(i, f_2, \ldots, f_{k-1}, x)$ be input for MPJ_k. Then, for all $1 < j < k$,*

$$\text{MPJ}_k(i, f_2, \ldots, x) = \text{MPJ}_3(f_{j-1} \circ \cdots \circ f_2(i), f_j, x \circ f_{k-1} \circ \cdots \circ f_{j+1}).$$

Proof. From the definition of MPJ_k and MPJ_3, we have

$$\begin{aligned} \text{MPJ}_k(i, f_2, \ldots, f_{k-1}, x) &= x \circ f_{k-1} \circ \cdots \circ f_2(i) \\ &= x \circ f_{k-1} \circ \cdots \circ f_{j+1}(f_j(f_{j-1} \circ \cdots \circ f_2(i)) \\ &= \text{MPJ}_3(f_{j-1} \circ \cdots \circ f_2(i), f_j, x \circ f_{k-1} \circ \cdots \circ f_{j+1}) \,. \end{aligned}$$

□

In our protocol for MPJ_k, $\text{PLR}_1, \text{PLR}_j$, and PLR_k will use a modified version of the protocol from Lemma 16 for MPJ_3 on input $(f_{j-1} \circ \cdots \circ f_2(i), f_j, x \circ \cdots \circ f_{j+1})$. Before we get

to the protocol, we need to generalize the technical definition and lemma from the previous subsection.

Definition 2. *A set $\mathcal{A} \subseteq \mathcal{S}_n$ of permutations is said to (S, d)-cover a function $f : [n] \to [n]$ if, for each $r \in S$, at least one of the following conditions holds:*

(i) $\exists\, \pi \in \mathcal{A}$ *such that* $\pi(r) = f(r)$*, or*

(ii) $|S \cap f^{-1}(f(r))| > d$*.*

Lemma 19. *Let $f : [n] \to [n]$ be a function, $S \subseteq [n]$, and d be a positive integer. There exists a set $\mathcal{A}_{S,d}(f) \subseteq \mathcal{S}_n$, with $|\mathcal{A}_{S,d}(f)| \leq d$, that (S, d)-covers f.*

Proof. This proof closely follows the proof of Lemma 17. As before, we give an explicit algorithm to construct $\mathcal{A}_{S,d}(f)$. Suppose $\mathrm{Range}(f) = \{s_1, s_2, \ldots, s_t\}$, and let $\{A_i\}$ and B be defined as in Lemma 17. Let $a_{i,1}, \cdots, a_{i,z_i}$ denote the elements of $A_i \cap S$, and let $a_{i,z_i+1}, \ldots, a_{i,|A_i|}$ denote the remaining elements of A_i. For $1 \leq i \leq t$ and $1 \leq j \leq d$, define $c_{i,j} := a_{i,\min\{j,|A_i|\}}$. As before, the jth permutation we construct will map $c_{i,j}$ to s_i. Finally, let $C_j := \{c_{1,j}, \ldots, c_{t,j}\}$, and let $D_j := \{d_{i,j}, \ldots, d_{n-t,j}\}$ be the elements not in C_j, We define permutation π_j in the following manner.

$$\pi_j(r) = \begin{cases} s_i & \text{if } r = c_{i,j}, \\ b_i & \text{if } r = d_{i,j}. \end{cases}$$

By construction, π_j defines a bijection between C_j and $\mathrm{Range}(f)$, and between D_j and B. It suffices to verify that this choice of $\mathcal{A}_{S,d}(f)$ (S, d)-covers f, i.e., to verify that every $r \in S$ satisfies at least one of the two conditions in Definition 2. Pick any $r \in S$, and suppose $r \in A_i$. If $|A_i \cap S| \leq d$, then there exists $1 \leq j \leq d$ such that $r = c_{i,j}$. Otherwise, condition (ii) holds, since $A_i = f^{-1}(f(r))$. □

Proof of Theorem 3. To design a protocol for MPJ_k, we first let PLR_1 and PLR_k agree on a parameter d, to be fixed below. They also agree on a choice of $\mathcal{A}_{S,d}(f)$ for all $S \subseteq [n]$ and $f : [n] \to [n]$.

Let $(i, f_2, \ldots, f_{k-1}, x)$ denote an input for MPJ_k. Also, let $S_1 = [n]$, and for all $2 \leq j \leq k-1$, let $S_j = \{s \in [n] : |S_{j-1} \cap f_j^{-1}(s)| > d\}$. Our protocol works as follows:

- PLR_1 sends a $(k-1)$-part message. For $1 \leq j < k-2$, the j^{th} part of PLR_1's message consists of the strings $\{\alpha(\pi, \hat{x}_{j+1})\}_\pi$ for each $\pi \in \mathcal{A}_{S_j,d}(f_{j+1})$. Part $k-2$ consists of the strings $\{\alpha(\pi, x)\}_\pi$ for each $\pi \in \mathcal{A}_{S_{k-2},d}(f_{k-1})$. The remaining part consists of the bits x_s for $s \in S_{k-1}$.

- For $2 \leq j \leq k-1$, PLR_j sends the strings $\{\beta(\hat{i}_j, \hat{x}_j, \alpha)\}_\alpha$ for all strings α in the $(j-1)$st part of PLR_1's message.

- PLR_k can now output $x_{\hat{i}_k}$ as follows. If $|S_1 \cap f_2^{-1}(f_2(i))| \leq d$, then $\mathcal{A}_{S_1,d}(f_2)$ (S_1, d)-covers f_2, and there exists $\pi_0 \in \mathcal{A}_{S_1,d}(f_2)$ such that $f_2(i) = \pi_0(i)$. She uses the string $\alpha_0 = \alpha(\pi_0, \hat{x}_2)$ from the first part of PLR_1's message and the string $\beta_0 = \beta(i, \hat{x}_2, \alpha_0)$ from PLR_2's message to output $\gamma_0 = \gamma(i, \pi_0, \alpha_0, \beta_0)$. Similarly, if for some $2 \leq j \leq k-2$ such that $|S_j \cap f_{j+1}^{-1}(f_{j+1}(\hat{i}_{j+1}))| \leq d$, then $\mathcal{A}_{S_j,d}(f_{j+1})$ (S_j, d)-covers f_{j+1}, and there exists a $\pi_0 \in \mathcal{A}_{S_j,d}(f_{j+1})$ such that $f_{j+1}(\hat{i}_{j+1}) = \pi_0(\hat{i}_{j+1})$. She uses the string $\alpha_0 = \alpha(\pi_0, \hat{x}_{j+1})$ from the j^{th} part of PLR_1's message and the string $\beta_0 = \beta(\hat{i}_{j+1}, \hat{x}_{j+1}, \alpha_0)$ from PLR_{j+1}'s message to output $\gamma_0 = \gamma(\hat{i}_{j+1}, \pi_0, \alpha_0, \beta_0)$. Otherwise, $|S_{k-2} \cap f_{k-1}^{-1}(f_{k-1}(\hat{i}_{k-1}))| > d$, hence $\hat{i}_k \in S_{k-1}$, and she reads $x_{\hat{i}_k}$ off from the last part of PLR_1's message.

To verify correctness, we need to ensure that PLR_k always outputs $x \circ f_{k-1} \circ \cdots \circ f_2(i)$. We proceed inductively. If $|S_1 \cap f_2^{-1}(f_2(i))| \leq d$ then there exists $\pi_0 \in \mathcal{A}_{S_1,d}(f_2)$ such that $f_2(i) = \pi_0(i)$, $\alpha_0 = \alpha(\pi_0, \hat{x}_2)$, and $\beta_0 = \beta(i, \hat{x}_2, \alpha_0)$, and PLR_k outputs $\gamma_0 = \gamma(i, \pi_0, \alpha_0, \beta_0) = \hat{x}_2(\pi_0(i)) = x \circ f_{k-1} \circ \cdots \circ f_2(i)$. Otherwise, $|S_1 \cap f_2^{-1}(f_2(i))| > d$, hence $f_2(i) \in S_2$.

Inductively, if $\hat{i}_j \in S_{j-1}$, then either $|S_{j-1} \cap f_j^{-1}(f_j(\hat{i}_j))| \leq d$, or $|S_{j-1} \cap f_j^{-1}(f_j(\hat{i}_j))| > d$. In the former case, there is $\pi_0 \in \mathcal{A}_{S_{j-1},d}(f_j)$ such that $f_j(\hat{i}_j) = \pi_0(\hat{i}_j)$, $\alpha_0(\pi_0, \hat{x}_j)$, and $\beta_0 = \beta(\hat{i}_j, \hat{x}_j, \alpha_0)$, and PLR_k outputs $\gamma_0 = \gamma(\hat{i}_j, \pi_0, \alpha_0, \beta_0) = \hat{x}_j(f_j(\hat{i}_j)) = x \circ f_{k-1} \circ \cdots \circ f_2(i)$. In the latter case, $f_j(\hat{i}_j) \in S_j$. By induction, we have that either PLR_k outputs $x \circ f_{k-1} \circ \cdots \circ f_2(i)$, or $\hat{i}_k \in S_{k-1}$. But in this case, PLR_k outputs $x(\hat{i}_k) = x \circ f_{k-1} \circ \cdots \circ f_2(i)$ directly from the last part of PLR_1's message. Therefore, PLR_k always outputs $x \circ f_{k-1} \circ \cdots \circ f_2(i)$ correctly.

We now turn to the communication cost of the protocol. By Lemma 19, $|\mathcal{A}_{S_j,d}(f_j)| \leq d$ for each $2 \leq j \leq k-1$, hence the first $k-2$ parts of PLR_1's message are each at most dm bits long, as is PLR_j's message for all $2 \leq j \leq k-1$. Also, since for all $2 \leq j \leq k-1$, there are at most $|S_{j-1}|/d$ $s \in S_j$ such that $|S_{j-1} \cap f_j^{-1}(s)| > d$, we must have that $|S_2| \leq |S_1|/d = n/d, |S_3| \leq |S_2|/d \leq n/d^2, |S_j| \leq n/d^{j-1}$, and $|S_{k-1}| \leq n/d^{k-2}$. Therefore, the final part of PLR_1's message is at most n/d^{k-2} bits long, and the total communication cost is at most $2(k-2)dm + n/d^{k-2} = O((k-2)dn\phi(n) + n/d^{k-2})$. Setting $d = \lceil((k-2)\phi(n))^{-1/(k-1)}\rceil$ gives us a bound of $O(n(k\phi(n))^{k-2/k-1})$ as desired. □

Note that except for the first and last players, the input can be quite restrictive. Specifically, for all $2 \leq j \leq k-1$, PLR_j needs to see only $\hat{i}_j$ and $\hat{x}_j$, i.e. PLR_j is both *conservative* and *collapsing*. Despite this severe restriction, we have a sublinear protocol for MPJ_k. As we'll see in Section 2.7, further restricting the input such that PLR_1 is also collapsing yields very high lower bounds.

2.3.4 A k-Player Protocol for $\widehat{\text{MPJ}}_k$

We conclude this section by extending the protocol for MPJ_k to get a protocol for $\widehat{\text{MPJ}}_k$.

Proof of Theorem 4. Let $(i, f_2, \ldots, f_k)$ be input for $\widehat{\text{MPJ}}_k$. For $1 \leq j \leq \log n$, define $x^{(j)}$ to be the n-bit string such that $x^{(j)}(r)$ equals the jth most significant bit of $f_{k-1}(r)$. Note that $(i, f_2, \ldots, f_{k-2}, x^{(j)})$ is a valid input for MPJ_{k-1}. Our protocol for $\widehat{\text{MPJ}}_k$ works as fol-

lows. First, players $\text{PLR}_1, \ldots, \text{PLR}_{k-1}$ use a protocol for MPJ_{k-1} t times in parallel, on inputs $(i, f_2, \ldots, f_{k-2}, x^{(j)})$ for each $1 \leq j \leq t$. As a result of this, PLR_{k-1} learns the t most significant bits of $\hat{i}_k = f_{k-1}(\hat{i}_{k-1})$. There are at most $n/2^t$ values $i^* \in [n]$ whose t most significant bits match those of $\hat{i}_k$. PLR_{k-1} sends $f_k(i^*)$ for each of these possibilities, and PLR_k, seeing $\hat{i}_k$, outputs $f_k(\hat{i}_k)$ from PLR_{k-1}'s message.

The first $k-1$ players run t copies of an MPJ_{k-1} protocol at a cost of $tC(\text{MPJ}_{k-1})$. Additionally, PLR_{k-1} sends $\log n$ bits for each of $n/2^t$ different possible values for $\hat{i}_k$. Therefore, the cost of this protocol is $tC(\text{MPJ}_{k-1}) + n \log n/2^t$. Using the MPJ_{k-1} protocol from Section 2.3.3 and setting $t := \left(2 - \frac{1}{k-1}\right) \log\left(\log n / \log\log n\right)$ gives the desired bound. □

2.4 Lower Bounds for Myopic Protocols

For many of our results in this section, we shall make use of the following sequences of numbers, all of which are parameterized by some $\delta \in \mathbb{R}^+$ (possibly dependent on n and k) to be specified later. Let $a_0 := 0$, and for $\ell > 0$, let $a_\ell := \delta 2^{a_\ell - 1}$. For all $\ell \geq 0$, let $m_\ell := n2^{-a_\ell}$. Note that $m_0 = n$. Also, let $\phi(k)$ be the least δ such that $a_{k-1} \geq 1$.

We now prove the lower bound on myopic MPJ_k protocols. We repeat the main theorem here for convenience:

Theorem 20. *(Precise restatement of Theorem 6). Let $\mathcal{P}$ be a myopic protocol for MPJ_k. Then, $\text{mcost}(\mathcal{P}) > n\phi(k)$.*

We prove this theorem by viewing MPJ_k as a special instance of $\text{MPJ}_{m,k}$ and by using a round elimination lemma. First, we note that $\text{MPJ}_{m,2}$ is just the INDEX problem on m bits. The one-way communication complexity of INDEX is well known; we state it here in terms of $\text{MPJ}_{m,2}$.

Fact 21. *If $\mathcal{P}$ is a protocol for $\text{MPJ}_{m,2}$, then $\text{mcost}(\mathcal{P}) \geq m$.*

The structure of our proof is as follows. We assume the existence of a protocol for MPJ_k in which each player sends at most δn bits. In the round elimination step, we show how to turn a protocol for $\text{MPJ}_{m,k}$ into a protocol for $\text{MPJ}_{m',k-1}$ with the same cost, and with $m' < m$. Repeating this step $k-2$ times, transforms the δn-bit protocol for MPJ_k into a δn-bit protocol for $\text{MPJ}_{m,2}$ with $m > \delta n$, contradicting Fact 21.

The following simple definition and lemma provide the combinatorial hook that permits the round elimination step.

Definition 3. *Let $i \in [\ell]$ and $\mathcal{F} \subseteq [n]^\ell$ be given. The* range *of i in $\mathcal{F}$, denoted $\text{Range}(i, \mathcal{F})$, is defined as:*

$$\text{Range}(i, \mathcal{F}) := \{f(i) : f \in \mathcal{F}\}$$

Lemma 22. *Let $\mathcal{F} \subseteq [n]^\ell$ be given. If $|\mathcal{F}| \geq m^\ell$, then $|\text{Range}(i, \mathcal{F})| \geq m$ for some $i \in [\ell]$.*

Proof. We prove the contrapositive of this statement. Suppose that $|\text{Range}(i, \mathcal{F})| < m$ for all $i \in [\ell]$. Without loss of generality, assume that $\text{Range}(i, \mathcal{F}) \subseteq [m-1]$ for each i, and let $\mathcal{G} := \{f : f(i) \leq m-1 \text{ for all } i \in [\ell]\}$. Its clear that $\mathcal{F} \subseteq \mathcal{G}$. Furthermore, $|\mathcal{G}| = (m-1)^\ell$. Hence, $|\mathcal{F}| \leq |\mathcal{G}| < m^\ell$. □

Lemma 23 (Round Elimination Lemma). *Let $k \geq 3$. If there is a δn-bit myopic protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit myopic protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.*

Proof. In $\text{MPJ}_{m,k}$, PLR_1's input is a function $f_2 : [m] \to [n]$. There are n^m such functions. Since PLR_1 sends at most δn bits, he must send the same message M on $n^m/2^{\delta n}$ distinct f_2. Let $\mathcal{F}$ be the set of inputs for which PLR_1 sends M. It follows that $|\mathcal{F}| \geq n^m/2^{\delta n} = 2^{m \log n - \delta n} = 2^{m(\log n - \delta n/m)} = 2^{m \log m'} = (m')^m$. By Lemma 22, we must have $i \in [m]$ with $|\text{Range}(i, \mathcal{F})| \geq m'$. Fix such an i, and let $S := \text{Range}(i, \mathcal{F})$. Without loss of generality, assume $S = [m']$. [3]

[3]Specifically, if $S \neq [m']$, then fix a permutation $\pi \in S_n$ that maps (a subset of) S to $[m']$. In $\mathcal{Q}$, players

We are now ready to construct a protocol for $\text{MPJ}_{m',k-1}$. Label the players $\text{PLR}_2, \ldots, \text{PLR}_k$. For each $j \in [m']$, the players agree on a $g_j \in \mathcal{F}$ such that $g_j(i) = j$. Then, on input $(j, f_3, \ldots, f_{k-1}, x)$, players simulate $\mathcal{P}$ on input $(i, g_j, f_3, \ldots, f_{k-1}, x)$. Clearly, $\text{cost}(\mathcal{Q}) = \text{cost}(\mathcal{P})$, and since $g_j(i) = j$, we must have

$$\text{MPJ}_{m,k}(i, g_j, f_3, \ldots, f_{k-1}, x) = \text{MPJ}_{m',k-1}(j, f_3, \ldots, f_{k-1}, x)\,.$$

□

Note that the reduction step in the round elimination lemma uses only the first two layers of input, so the lemma can be applied to a much wider range of problems than just $\text{MPJ}_{m,k}$ and to a much wider range of protocols than just myopic protocols. For example, the reduction step only requires that PLR_1 is myopic. More importantly, the lemma applies to $\widehat{\text{MPJ}}_{m,k}$ exactly as stated.

Lemma 24. *Let $k \geq 3$. If there is a δn-bit myopic protocol $\mathcal{P}$ for $\widehat{\text{MPJ}}_{m,k}$, then there is a δn-bit myopic protocol $\mathcal{Q}$ for $\widehat{\text{MPJ}}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.*

Proof of Theorem 20. The main theorem follows by careful application of the Round Elimination Lemma. Suppose $\mathcal{P}$ is a δn-bit myopic protocol for $\text{MPJ}_k = \text{MPJ}_{m_0,k}$. By the Round Elimination Lemma, a δn-bit protocol for $\text{MPJ}_{m_\ell,z}$ yields a δn-bit protocol for $\text{MPJ}_{m',z-1}$, where $m' = n \cdot 2^{-\delta n/m_\ell} = n \cdot 2^{-\delta n/(n2^{-a_\ell})} = n \cdot 2^{-\delta 2^{a_\ell}} = n \cdot 2^{-a_{\ell+1}} = m_{\ell+1}$. Applying the lemma $k-2$ times, we transform $\mathcal{P}$ into a δn-bit protocol for $\text{MPJ}_{m_{k-2},2}$. By Fact 21, we must have $\delta n \geq m_{k-2} = n2^{-a_{k-2}}$, hence $1 \leq \delta 2^{a_{k-2}} = a_{k-1}$. Therefore, $\text{cost}(\mathcal{P}) \geq \phi(k)n$. (Recall that $\phi(k)$ is precisely the least δ such that $a_{k-1} \geq 1$.)

We complete the proof by showing that $\phi(k) > 1/2$. Specifically, we claim that if $\delta \leq 1/2$, then $a_\ell < 1$ for all $\ell > 0$. We prove this claim by induction. In the base case, $a_1 = \delta 2^{a_0} \leq$

agree on g_j such that $\pi(g_j(i)) = j$ and simulate $\mathcal{P}$ on input $(i, g_j, f_3 \circ \pi, \ldots, f_{k-1}, x)$. $f_3(j) = f_3(\pi(g_j(i))) = f_3 \circ \pi(g_j(i))$, and the rest of the proof follows.

$1/2 < 1$, and if $a_\ell < 1$, then $a_{\ell+1} = \delta 2^{a_\ell} < (1/2) \cdot 2^1 = 1$. □

Next, we show how to extend this to an exact lower bound for the total communication of myopic protocols.

Corollary 25. *For all $m \leq n$, any myopic protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$ must have $\text{cost}(\mathcal{P}) \geq m$.*

Proof. We prove this by induction on k. The base case $\text{MPJ}_{m,2}$ is trivial. For the general case, assume that for all $m \leq n$, any protocol for $\text{MPJ}_{m,k-1}$ requires m bits, and suppose there is a protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$ where PLR_1 sends m_1 bits. The reduction in Lemma 23 gives a protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ where $m' = n \cdot 2^{-\delta n/m} = n \cdot 2^{-m_1/m}$. By the induction hypothesis, $\text{cost}(\mathcal{Q}) \geq m'$. Therefore, $\text{cost}(\mathcal{P}) \geq m_1 + m'$. Next, note that

$$m_1 + m' < m \quad \Leftrightarrow \quad m_1 + n \cdot 2^{-m_1/m} < m \tag{2.5}$$

$$\Leftrightarrow \quad n < 2^{m_1/m}(m - m_1) \tag{2.6}$$

$$\Leftrightarrow \quad n < 2^{\alpha} m(1 - \alpha). \tag{2.7}$$

where $\alpha = m'/m \in [0, 1]$. The function $f(x) = 2^x(1 - x)$ is decreasing on all $x \in [0, 1]$, so it achieves its maximal value at $f(0) = 1$. Note that $2^{\alpha} m(1 - \alpha) < m$. Hence if the right hand side of inequality (2.7) holds, then $n < m$. However, by assumption, $m \leq n$, so this cannot be true. Therefore, $m_1 + m' \geq m$, completing the proof. □

Our main theorem shows that no matter how many players are involved, some player must send at least $\phi(k)n > n/2$ bits. For specific k, the constant factor can be improved. For example, a δn-bit protocol for MPJ_3 gives a δn-bit protocol for $\text{MPJ}_{m,2}$ with $m = n \cdot 2^{-\delta}$. By Lemma 21, we must have $n \cdot 2^{-\delta} \leq \delta n$, or $\delta 2^{\delta} \geq 1$. Solving for δ gives a lower bound of $\approx 0.6412n$.

Next we give a similar theorem for $\widehat{\text{MPJ}}_k$.

Theorem 26. *(Restatement of Theorem 8). Fix $2 \leq k < \log^* n$, and let $\mathcal{P}$ be a myopic protocol for $\widehat{\text{MPJ}}_k$. Then, $\text{cost}(\mathcal{P}) \geq n(\log^{(k-1)} n - \log^{(k)} n)$ bits.*

As in the lower bound proof for MPJ_k, we begin with an easy lower bound for $\widehat{\text{MPJ}}_{m,2}$.

Fact 27. *In any deterministic protocol for $\widehat{\text{MPJ}}_{m,2}$, PLR_1 communicates at least $m \log n$ bits.*

Theorem 26 is a direct consequence of the following lemma:

Lemma 28. *If $\delta = \log^{(k-1)} n - \log^{(k)} n$, then $a_j \leq \log^{(k-j)} n - \log^{(k+1-j)} n$ for all $1 \leq j < k$. In particular, $a_{k-1} \leq \log n - \log\log n$.*

Proof. (by induction) For $j = 1$, $a_j = a_1 = \delta = \log^{(k-1)} n - \log^{(k)} n = \log^{(k-j)} n - \log^{(k+1-j)} n$. For the induction step, we have

$$\begin{aligned} a_{j-1} &\leq \log^{(k+1-j)} n - \log^{(k+2-j)} n \\ &= \log\left(\frac{\log^{(k-j)} n}{\log^{(k+1-j)} n}\right) \end{aligned}$$

Therefore, $2^{a_{j-1}} \leq \frac{\log^{(k-j)} n}{\log^{(k+1-j)} n}$, and

$$\begin{aligned} a_j &= \delta 2^{a_{j-1}} \\ &\leq \left(\log^{(k-1)} n - \log^{(k)} n\right)\left(\frac{\log^{(k-j)} n}{\log^{(k+1-j)} n}\right) \\ &= \frac{\log^{(k-1)} n \log^{(k-j)} n}{\log^{(k+1-j)} n} - \frac{\log^{(k)} n \log^{(k-j)} n}{\log^{(k+1-j)} n} \\ &\leq \log^{(k-j)} n - \log^{(k+1-j)} n \end{aligned}$$

where the last inequality is because the positive term is less than $\log^{(k-j)} n$, and the negative term is greater than $\log^{(k+1-j)} n$, for all $2 \leq j < k$. □

Proof of Theorem 26. Let $\delta = \log^{(k-1)} n - \log^{(k)} n$. Suppose we have a protocol for $\widehat{\text{MPJ}}_k$ in

which each player sends δn bits. By Lemma 24, we have a δn-bit protocol for $\widehat{\text{MPJ}}_{m_{k-2},2}$. By Fact 27, such a protocol costs at least $m_{k-2} \log n$ bits. Hence, we must have

$$\begin{aligned} \delta n \geq m_{k-2} \log n &\Leftrightarrow \delta n \geq n 2^{-a_{k-2}} \log n \\ &\Leftrightarrow \delta 2^{a_{k-2}} \geq \log n \\ &\Leftrightarrow a_{k-1} \geq \log n \end{aligned}$$

However, we know by Lemma 28 that $a_{k-1} \leq \log n - \log\log n < \log n$, so we have a contradiction. □

2.5 An Upper Bound for Myopic Protocols

The analysis for the lower bound in the previous section also gives insight as to what myopic protocols *can* do. Specifically, in a protocol for $\text{MPJ}_{m,k}$, we'd like PLR_1's message to give PLR_2 enough information so that $\text{PLR}_2, \ldots, \text{PLR}_k$ can run a protocol for $\text{MPJ}_{m',k-1}$ for some $m' < m$. To do this, we need PLR_1's messages to partition his input space so that for each of his messages M_j and for each $1 \leq i \leq m$, the range size $|\operatorname{Range}(i, M_1)|$ is small.

It turns out that just such a protocol is possible, and that the communication cost matches our lower bound up to $1 + o(1)$ factors. To aid in the analysis of this protocol, we need the following *covering lemma*.

Definition 4. *We say a subset $T \subseteq [m]^d$ is isomorphic to $[m']^d$ and write $T \cong [m']^d$ if $T = T_1 \times \cdots \times T_d$ for sets $T_1, \ldots, T_d \subseteq [m]$, each of size m'.*

Lemma 29. ***(Covering Lemma).*** *For integers $d, m, m' < m \in \mathbb{Z}_{>0}$, let $\mathcal{U}_{m,d} := [m]^d$, and $\mathcal{S}_{m',d} := \{T \subseteq \mathcal{U}_{m,d} : T \cong [m']^d\}$. Then there exists a set $\mathcal{C} \subseteq \mathcal{S}_{m',d}$ of size $|\mathcal{C}| \leq (m/m')^d \cdot$*

$d \ln m + 1$ *such that* $\cup_{T \in \mathcal{C}} T = \mathcal{U}_{m,d}$. *We say that* $\mathcal{C}$ covers $\mathcal{U}_{m,d}$ *and call* $\mathcal{C}$ *an* m'*-covering of* $\mathcal{U}_{m,d}$.

Proof. We use the probabilistic method. Fix $r > (m/m')^d \, d \ln m$, and pick $T_1, \ldots, T_r$ independently and uniformly at random from $\mathcal{S}_{m',d}$. Note that picking T in this way amounts to picking d $[m']$-subsets of $[m]$ independently and uniformly at random. Therefore, for any $p \in \mathcal{U}_{m,d}$, we have $\Pr[p \in T] = (m'/m)^d$. For each $p \in \mathcal{U}_{m,d}$, let $BAD_p := \bigwedge_{1 \le j \le r} (p \notin T_j)$ be the event that p is not covered by any set T_j. Also, let $BAD := \bigvee_{p \in \mathcal{U}_{m,d}} BAD_p$ be the event that *some* p is not covered. From the probability calculation above, and using the fact that $1 + x \le e^x$, we have $\Pr[BAD_p] = \left(1 - (m'/m)^d\right)^r \le e^{-r(m'/m)^d}$. By the union bound, we have $\Pr[BAD] \le m^d \Pr[BAD_p] \le e^{d \ln m - r(m'/m)^d}$. Recall that $r > (m/m')^d \cdot d \ln m$, so $d \ln m - r(m'/m)^d < d \ln m - d \ln m = 0$. Hence, $\Pr[BAD] < e^0 = 1$. Therefore, there must exist a set $\{T_1, \ldots, T_r\}$ of sets isomorphic to $[m']^d$ that cover $\mathcal{U}_{m,d}$. □

Theorem 30. *For all* $k \ge 3$*, there exists a myopic protocol for* MPJ_k *in which each player sends* $\phi(k)n(1 + o(1))$ *bits.*

Proof. We prove this by construction. As a warmup, we give a $(0.65n)$-bit max-communication protocol for MPJ_3. Later, we show how to generalize this to more than 3 players. Recall that we have a $\phi(3)n$-bit lower bound for MPJ_3, where $\phi(3) \approx 0.6412$ is the unique real number δ such that $a_2 = \delta 2^\delta = 1$. In advance, the players fix a $[0.65n]$-covering $\mathcal{C}$ of $[n]^{[n]}$. On input (i, f_2, x), PLR_1 sends $T \in \mathcal{C}$ such that $f_2 \in T$. PLR_2 sees i, x and T, and sends x_j for all $j \in \text{Range}(i, T)$. PLR_3 sees i, f_2 and recovers $x_{f_2(i)}$ from PLR_2's message.

In terms of communication cost, PLR_1 sends $\log |\mathcal{C}|$ bits. By Lemma 29, $|\mathcal{C}| \le (n/0.65n)^n \cdot n \ln n + 1$, hence PLR_1 sends $\log |\mathcal{C}| = n \log(1/0.65)(1 + o(1)) < 0.65n$ bits. PLR_2 sends one bit for each $j \in \text{Range}(i, T)$. Since $T \cong [0.65n]^n$, we must have $|\text{Range}(i, T)| \le 0.65n$. Hence, PLR_2 sends at most $0.65n$ bits, and the maximum communication cost is also 0.65 bits.

For the general case, we construct a protocol $\mathcal{P}$ for MPJ_k as follows. Fix $\delta := \phi(k)$, and for

each $0 \leq j \leq k-2$, players agree in advance on a $[m_{j+1}]$-covering set $\mathcal{C}_{j+1}$ for $\mathcal{U}_{n,m_j}$. Note that by the covering lemma, $\log|\mathcal{C}_{j+1}| = m_j \log(n/m_{j+1})(1+o(1))$. Also note that

$$\begin{aligned} m_j \log(n/m_{j+1}) &= n2^{-a_j} \log\left(n/n2^{-a_{j+1}}\right) \\ &= -n2^{-a_j} \log(2^{-a_{j+1}}) \\ &= n2^{-a_j} a_{j+1} \\ &= n2^{-a_j}(\delta 2^{a_j}) \\ &= \delta n\,. \end{aligned}$$

On input $(i, f_2, \ldots, f_{k-1}, x)$, the players proceed as follows. PLR_1 sees $f_2 \in [n]^{[n]}$ and picks $T_1 \in \mathcal{C}_1$ that contains f_2. PLR_1 communicates T_1 to the rest of the players.

PLR_2 sees $i \in [m]$, $f_3 \in [n]^{[n]}$, and T_1. From i and T_1, PLR_2 computes $R_2 := \text{Range}(i, T_1)$. Note that since T_1 is an $[m_1]$ covering, $|\text{Range}(i, T_1)| = m_1$ for all i. Without loss of generality, assume $R_2 = [m_1]$. Let f_3^* be f_3 restricted to the domain R_2. Note that f_3^* is a function $[m_1] \to [n]$, so $f_3^* \in \mathcal{U}_{n,m_1}$. PLR_2 picks $T_2 \in \mathcal{C}_2$ that contains f_3^* and communicates T_2 to the rest of the players.

Generalizing, PLR_j computes $R_j := \text{Range}(f_{j-1} \circ \cdots \circ f_2(i), T_{j-1})$, which has size m_{j-1} because $T_{j-1} \in \mathcal{C}_{j-1}$. Noting that f_j restricted to R_j is an element in $\mathcal{U}_{n,m_{j-2}}$, PLR_j picks $T_j \in \mathcal{C}_j$ that contains f_j and communicates this to the rest of the players.

PLR_{k-1} computes $R_{k-1} := \text{Range}(f_{k-2} \circ \cdots \circ f_2(i), T_{k-2})$ and sends x_r for each $r \in R_{k-1}$. PLR_k computes $r^* := f_{k-1} \circ f_{k-2} \circ \cdots \circ f_2(i)$ and recovers x_{r^*} from PLR_{k-1}'s message.

For each $1 \leq j \leq k-2$, PLR_j sends $\log|\mathcal{C}_{j+1}| = \delta n(1+o(1))$ bits. PLR_{k-1} sends one bit for each $j \in R_{k-1}$. By construction, $|R_{k-1}| \leq m_{k-2}$. Choosing δ to be the smallest real such that $\delta 2^{a_{k-2}} = a_{k-1} \geq 1$ ensures that $m_{k-2} \leq \delta n$.

In conclusion, we have a protocol $\mathcal{P}$ where each player sends $\delta n(1+o(1))$ bits, where δ

is the smallest real such that $a_{k-1} \geq 1$. Note that this choice of δ exactly matches our lower bound. $\square$

2.6 Randomizing the Lower Bound

Theorems 20 and 26 give strong lower bounds for deterministic protocols for MPJ_k and $\widehat{\text{MPJ}}_k$ respectively. In this section, we show that our technique can also be used to show lower bounds on the randomized complexity of MPJ_k.

Previously, Chakrabarti [Cha07] showed randomized lower bounds of $\Omega(n/k)$ and $\Omega(n \log^{(k-1)} n)$ for MPJ_k and $\widehat{\text{MPJ}}_k$ respectively. The bound for $\widehat{\text{MPJ}}_k$ is for the maximum communication and is tight. The bound for MPJ_k is for the total communication; this bound implies an $\Omega(n/k^2)$ lower bound on the maximum communication. In contrast, we achieve:

Theorem 31. *In any randomized myopic protocol for* MPJ_k*, some player must communicate at least* $\Omega(n/k \log n)$ *bits.*

Our lower bound improves on the bound from [Cha07] for $k = \Omega(\log n)$. To prove this lower bound, we give a round elimination lemma for ε-error distributional protocols for $\text{MPJ}_{m,k}$ under the uniform distribution. By Yao's minimax principle [Yao77], lower bounds on distributional protocols imply lower bounds on randomized protocols. Our "base case" is the lower bound on the ε-error distributional complexity of $\text{MPJ}_{m,k}$, due to Ablayev [Abl96]:

Fact 32. *Any protocol for* $\text{MPJ}_{m,2}$ *that errs on at most an* ε*-fraction of the inputs distributed uniformly must communicate at least* $m\,(1 - H(\varepsilon))$ *bits.*[4]

Lemma 33 (Round Elimination Lemma). *Let* $k \geq 3$*. If there is a* δn*-bit,* ε*-error distributional myopic protocol* $\mathcal{P}$ *for* $\text{MPJ}_{m,k}$*, then there is a* δn*-bit,* $\hat{\varepsilon}$*-error distributional myopic protocol* $\mathcal{Q}$ *for* $\text{MPJ}_{m',k-1}$ *with* $m' = n \cdot 2^{-2\delta \frac{n}{m}}$ *and* $\hat{\varepsilon} = 2n\varepsilon$*.*

[4]The binary entropy function H is defined as: $H(\varepsilon) := -\varepsilon \log \varepsilon - (1-\varepsilon)\log(1-\varepsilon)$.

Proof. For the sake of notation, we let $z := (f_3, \ldots, f_{k-1}, x)$, so the input to $\text{MPJ}_{m,k}$ is (i, f_2, z). Let $\mathcal{P}(i, f_2, z)$ denote the output of $\mathcal{P}$ on input (i, f_2, z). Let

$$\alpha(i, f_2, z) := \begin{cases} 1 & \text{if } \mathcal{P}(i, f_2, z) \neq \text{MPJ}_{m,k}(i, f_2, z) \\ 0 & \text{otherwise} \end{cases}$$

Since $\mathcal{P}$ is an ε-error protocol, we have $\mathbb{E}_{i,f_2,z}[\alpha(i, f_2, z)] = \varepsilon$. Now, let $\hat{\alpha}(i, f_2) := \mathbb{E}_z[\alpha(i, f_2, z)]$, and call (i, f_2) *bad* if $\hat{\alpha}(i, f_2) > 2n\varepsilon$; otherwise, call (i, f_2) *good*. Clearly, $\mathbb{E}_{i,f_2}[\hat{\alpha}(i, f_2)] = \mathbb{E}_{i,f_2,z}[\alpha(i, f_2, z)] = \varepsilon$, so by Markov's inequality, we get $\Pr[(i, f_2) \text{ is bad}] < 1/2n$. Now, let

$$\beta(i, f_2) := \begin{cases} 1 & \text{if } (i, f_2) \text{ is bad} \\ 0 & \text{otherwise} \end{cases}$$

Also, let $\hat{\beta}(f_2) = \mathbb{E}_i[\beta(i, f_2)]$. Call f_2 *bad* if $\hat{\beta}(f_2) \geq 1/n$, and call f_2 *good* otherwise. Note that $\mathbb{E}_{f_2}[\hat{\beta}(f_2)] = \mathbb{E}_{i,f_2}[\beta(i, f_2)] < 1/(2n)$, so by another application of Markov's inequality, we get $\Pr[f_2 \text{ is bad}] < 1/2$. Therefore, f_2 is good with probability at least $1/2$.

Note that if f_2 is good, then $\Pr_i[(i, f_2) \text{ is bad}] < 1/n$. Furthermore, if (i, f_2) were bad for even a single i, then we would have $\Pr_i[(i, f_2) \text{is bad}] \geq 1/n$. Therefore, (i, f_2) is good for *every* i whenever f_2 is good.

The rest of this lemma closely follows the deterministic version. There are n^m functions $f_2 : [m] \to [n]$. Since at least half the functions f_2 are good, there must be at least $n^m/2$ good f_2. Since PLR_1 sends at most δn bits, he must send the same message M_1 on $n^m/(2 \cdot 2^{\delta n})$ distinct good f_2. Let $\mathcal{F}$ be the set of good inputs for which PLR_1 sends M_1. It follows that $|\mathcal{F}| \geq \frac{n^m}{2 \cdot 2^{\delta n}} = 2^{m \log n - 1 - \delta n} > 2^{m \log n - 2\delta n} = (m')^m$. By Lemma 22, we must have $i \in [m]$ with $|\operatorname{Range}(i, \mathcal{F})| \geq m'$. Furthermore, every $f \in \mathcal{F}$ is good, so (i, f) is good for all $f \in \mathcal{F}$. Construct a protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ as we did in Lemma 23. As in Lemma 23, the cost of $\mathcal{Q}$ remains equal to the cost of $\mathcal{P}$, $\text{MPJ}_{m,k}(i, g_j, z) = \text{MPJ}_{m',k-1}(j, z)$, and that $\mathcal{Q}(j, z) =$

$\mathcal{P}(i, g_j, z)$. Finally, we get

$$\begin{aligned}\Pr_{j,z}[\mathcal{Q}(i,z) \neq \text{MPJ}_{m',k-1}(j,z)] &= \Pr_{j,z}[\mathcal{P}(i,g_j,z) \neq \text{MPJ}_{m,k}(i,g_j,z)] \\ &= \Pr_{j,z}[\alpha(i,g_j,z) = 1] \\ &\leq 2n\varepsilon\,.\end{aligned}$$

where the inequality holds because (i, g_j) is good for every j. □

Proof of Theorem 31. Let $\varepsilon = 1/3$ and $\delta = 1/32$, and suppose an ε-error randomized protocol for MPJ_k exists where each player sends at most $t = \frac{n}{48\delta \ln 2(\log 3 + (k-2)\log(2n))} = \Omega(\frac{n}{k \log n})$ bits. By Chernoff bounds, there exists an $\hat{\varepsilon} := \varepsilon\,(2n)^{-(k-2)}$-error randomized protocol $\mathcal{P}$ for MPJ_k, where each player sends δn bits. By Yao's minimax lemma, there is a deterministic protocol where each player sends δn bits that errs on an $\hat{\varepsilon}$ fraction of inputs, distributed uniformly.

Set $a_0 = 0, a_\ell = 2\delta 2^{a_{\ell-1}}$, and $m_\ell = n2^{-a_\ell}$. Note that $a_0 < 1/8$, and if $a_{\ell-1} < 1/8$, then $a_\ell = 2\delta 2^{a_{\ell-1}} < 1/8$, so by induction, $a_\ell < 1/8$ for all ℓ. Using Lemma 33 $k-2$ times, we get a δn-bit, ε-error protocol for $\text{MPJ}_{m_{k-2},2}$. Combining this with Fact 32, we get

$$\begin{aligned}\delta n \geq m_{k-2}\,(1 - H(1/3)) &\Leftrightarrow \delta n \geq n2^{-a_{k-2}}\,(1 - H(1/3)) \\ &\Leftrightarrow \delta 2^{a_{k-2}} \geq 1 - H(1/3) \\ &\Leftrightarrow a_{k-1}/2 \geq 1 - H(1/3)\,.\end{aligned}$$

However, we have already seen that $a_{k-1}/2 < 1/16 < 1 - H(1/3)$, so this is a contradiction. □

2.7 Collapsing Protocols: A Lower Bound

Let $F : \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_k \to \mathscr{B}$ be a k-player NOF communication problem and P be a protocol for F. We say that PLR_j is *collapsing* in P if her message depends only on $x_1, \ldots, x_{j-1}$ and the function $g_{x,j} : \mathscr{A}_1 \times \mathscr{A}_2 \times \cdots \times \mathscr{A}_j \to \mathscr{B}$ given by $g_{x,j}(z_1, \ldots, z_j) = F(z_1, \ldots, z_j, x_{j+1}, \ldots, x_k)$. For pointer jumping, this amounts to saying that PLR_j sees all layers $1, \ldots, j-1$ of edges (i.e., the layers preceding the one on her forehead), but not layers $j+1, \ldots, k$; however, she does see the result of following the pointers from each vertex in layer j. Still more precisely, if the input to MPJ_k (or $\widehat{\text{MPJ}}_k$) is $(i, f_2, \ldots, f_k)$, then the only information PLR_j gets is $i, f_2, \ldots, f_{j-1}$ and the composition $f_k \circ f_{k-1} \circ \cdots \circ f_{j+1}$.

We say that a protocol is collapsing if every player involved is collapsing. We shall prove Theorem 10 by contradiction. Assume that there is a collapsing protocol P for MPJ_k in which every player sends less than $n - \frac{1}{2}\log n - 2$ bits. We shall construct a pair of inputs that differ only in the last layer (i.e., the Boolean string on PLR_k's forehead) and that cause players 1 through $k-1$ to send the exact same sequence of messages. This will cause PLR_k to give the same output for both these inputs. But our construction will ensure that the desired outputs are unequal, a contradiction. To aid our construction, we need some definitions and preliminary lemmas.

Definition 5. *A string $x \in \{0,1\}^n$ is said to be* consistent *with $(f_1, \ldots, f_j, \alpha_1, \ldots, \alpha_j)$ if, in protocol P, for all $h \leq j$, PLR_h sends the message α_h on seeing input $(f_1, \ldots, f_{h-1}, x \circ f_j \circ f_{j-1} \circ \cdots \circ f_{h+1})$ and previous messages $\alpha_1, \ldots, \alpha_{h-1}$.[5] A subset $T \subseteq \{0,1\}^n$ is said to be consistent with $(f_1, \ldots, f_j, \alpha_1, \ldots, \alpha_j)$ if x is consistent with $(f_1, \ldots, f_j, \alpha_1, \ldots, \alpha_j)$ for all $x \in T$.*

[5] It is worth noting that, in Definition 5, x is not to be thought of as an input on PLR_k's forehead. Instead, in general, it is the composition of the rightmost $k-j$ layers of the input graph.

Definition 6. *For strings $x, x' \in \{0,1\}^n$ and $a, b \in \{0,1\}$, define the sets*

$$I_{ab}(x, x') \;:=\; \{i \in [n] :\; (x_i, x'_i) = (a, b)\}\,.$$

A pair of strings (x, x') is said to be a crossing pair *if for all $a, b \in \{0,1\}$, $I_{ab}(x, x') \neq \emptyset$. A set $T \subseteq \{0,1\}^n$ is said to be* crossed *if it contains a crossing pair and* uncrossed *otherwise. The* weight *of a string $x \in \{0,1\}^n$ is defined to be the number of 1s in x, and denoted $|x|$.*

For the rest of this section, we assume (without loss of generality) that n is large enough and even.

Lemma 34. *If $T \subseteq \{0,1\}^n$ is uncrossed, then $|\{x \in T :\; |x| = n/2\}| \leq 2$.*

Proof. Let x and x' be distinct elements of T with $|x| = |x'| = n/2$. For $a, b \in \{0,1\}$, define $t_{ab} = |I_{ab}(x, x')|$. Since $x \neq x'$, we must have $t_{01} + t_{10} > 0$. An easy counting argument shows that $t_{01} = t_{10}$ and $t_{00} = t_{11}$. Since T is uncrossed, (x, x') is not a crossing pair, so at least one of the numbers t_{ab} must be zero. It follows that $t_{00} = t_{11} = 0$, so x and x' are bitwise complements of each other.

Since this holds for any two strings in $\{x \in T :\; |x| = n/2\}$, that set can have size at most 2. □

Lemma 35. *Suppose $t \leq n - \frac{1}{2}\log n - 2$. If $\{0,1\}^n$ is partitioned into 2^t disjoint sets, then one of those sets must be crossed.*

Proof. Let $\{0,1\}^n = T_1 \sqcup T_2 \sqcup \cdots \sqcup T_m$ be a partition of $\{0,1\}^n$ into m uncrossed sets. Define $X := \{x \in \{0,1\}^n :\; |x| = n/2\}$. Then $X = \bigcup_{i=1}^m (T_i \cap X)$. By Lemma 34,

$$|X| \;\leq\; \sum_{i=1}^{m} |T_i \cap X| \;\leq\; 2m\,.$$

Using Stirling's approximation, we can bound $|X| > 2^n/(2\sqrt{n})$. Therefore, $m > 2^{n-\frac{1}{2}\log n-2}$. □

Proof of Theorem 10. Set $t = n - \frac{1}{2}\log n - 2$. Recall that we have assumed that there is a collapsing protocol P for MPJ_k in which every player sends at most t bits. We shall prove the following statement by induction on j, for $j \in [k-1]$.

(*) There exists a partial input $(i = f_1, f_2, \ldots, f_j) \in [n] \times \left([n]^{[n]}\right)^{j-1}$, a sequence of messages $(\alpha_1, \ldots, \alpha_j)$ and a crossing pair of strings $(x, x') \in (\{0,1\}^n)^2$ such that both x and x' are consistent with $(f_1, \ldots, f_j, \alpha_1, \ldots, \alpha_j)$, whereas $x \circ f_j \circ \cdots \circ f_2(i) = 0$ and $x' \circ f_j \circ \cdots \circ f_2(i) = 1$.

Considering (*) for $j = k-1$, we see that PLR_k must behave identically on inputs $(i, f_2, \ldots, f_{k-1}, x)$ and $(i, f_2, \ldots, f_{k-1}, x')$. Therefore, she must err on one of these two inputs. This will give us the desired contradiction.

To prove (*) for $j = 1$, note that PLR_1's message, being at most t bits long, partitions $\{0,1\}^n$ into at most 2^t disjoint sets. By Lemma 35, one of these sets, say T, must be crossed. Let (x, x') be a crossing pair in T and let α_1 be the message that PLR_1 sends on seeing a string in T. Fix $i = f_1$ such that $i \in I_{01}(x, x')$. These choices are easily seen to satisfy the conditions in (*).

Now, suppose (*) holds for a particular $j \geq 1$. Fix the partial input $(f_1, \ldots, f_j)$ and the message sequence $(\alpha_1, \ldots, \alpha_j)$ as given by (*). We shall come up with appropriate choices for f_{j+1}, α_{j+1} and a new crossing pair (y, y') to replace (x, x'), so that (*) is satisfied for $j+1$. Since PLR_{j+1} sends at most t bits, she partitions $\{0,1\}^n$ into at most 2^t subsets (the partition might depend on the choice of $(f_1, \ldots, f_j, \alpha_1, \ldots, \alpha_j)$).

As above, by Lemma 35, she sends a message α_{j+1} on some crossing pair (y, y'). Choose f_{j+1} so that it maps $I_{ab}(x, x')$ to $I_{ab}(y, y')$ for all $a, b \in \{0,1\}$; this is possible because $I_{ab}(y, y') \neq \emptyset$. Then, for all $i \in [n]$, $x_i = y_{f_{j+1}(i)}$ and $x'_i = y'_{f_{j+1}(i)}$. Hence, $x = y \circ f_{j+1}$

and $x' = y' \circ f_{j+1}$. Applying the inductive hypothesis and the definition of consistency, it is straightforward to verify the conditions of (*) with these choices for f_{j+1}, α_{j+1}, y and y'. This completes the proof. □

2.8 Collapsing Protocols: An Upper Bound

We now turn to proving Theorem 11 by constructing appropriate collapsing protocols for $\widehat{\text{MPJ}}_k^{\text{perm}}$. Our protocols use what we call *bucketing schemes*, which have the flavor of the conservative protocol of Damm et al. [DJS98]. For any function $f \in [n]^{[n]}$ and any $S \subseteq [n]$. let $\mathbf{1}_S$ denote the indicator function for S; that is, $\mathbf{1}_S(i) = 1 \leftrightarrow i \in S$. Also, let $f|_S$ denote the function f restricted to S. f_S can be seen as a list of numbers $\{i_s\}$, one for each $s \in S$. Players will often need to send $\mathbf{1}_S$ and $f|_S$ together in a single message. This is because later players might not know S, and will therefore be unable to interpret $f|_S$ without $\mathbf{1}_S$. Let $\langle m_1, \ldots, m_t \rangle$ denote the concatenation of messages $m_1, \ldots, m_t$. As before, it will be instructive to first consider the special case $k = 3$ in detail.

Definition 7. *A* bucketing scheme *on a set X is an ordered partition $\mathcal{B} = (B_1, \ldots, B_t)$ of X into* buckets. *For $x \in X$, we write $\mathcal{B}[x]$ to denote the integer j such that $B_j \ni x$.*

We actually prove our upper bound for problems slightly more general than $\widehat{\text{MPJ}}_k^{\text{perm}}$. To be precise, for an instance $(i, f_2, \ldots, f_k)$ of $\widehat{\text{MPJ}}_k$, we allow any one of $f_2, \ldots, f_k$ to be an arbitrary function in $[n]^{[n]}$. The rest of the f_js are required to be permutations, i.e., in $\mathcal{S}_n$. For $k = 3$, this leads to two cases.

Theorem 36. *There is an $O(n \log\log n)$-communication collapsing protocol for instances (i, f_2, f_3) of $\widehat{\text{MPJ}}_3$ in which $f_3 \in \mathcal{S}_n$.*

Proof. Assume, without loss of generality, that n is a power of 2. The players agree on the bucketing scheme $\mathcal{B} = (B_1, \ldots, B_{\log n})$ on $[n]$ defined by $B_j := \{r \in [n] : \lceil (r \log n)/n \rceil = j\}$.

Note that each $|B_j| \leq \lceil n/\log n \rceil$ and that a bucket can be described using $\lceil \log\log n \rceil$ bits. Upon input (i, f_2, f_3):

- PLR_1 sees $f_3 \circ f_2$ and sends $\alpha := \langle \mathcal{B}[f_3(f_2(1))], \mathcal{B}[f_3(f_2(2))], \ldots, \mathcal{B}[f_3(f_2(n))] \rangle$.
- PLR_2 sees i, f_3, and α. From α, she recovers $b := \mathcal{B}[f_3(f_2(i))]$ and hence B_b. She determines the set $S := \{h \in [n] : f_3(h) \in B_b\}$. Note that the definitions guarantee that $f_2(i) \in S$. She sends $\beta := \langle \mathbf{1}_S, f_3|_S \rangle$.
- PLR_3 sees i, f_2, α, and β. She computes $j = f_2(i)$. Since $j \in S$, she determines $f_3(j)$ from β and outputs that.

The protocol is clearly correct. As for the communication cost, PLR_1's message uses $n\lceil \log\log n \rceil$ bits. In PLR_2's message, $\mathbf{1}_S$ takes up n bits and $f_3|_S$ takes up $|S| \log n$ bits. To finish the proof, note that $S = f_3^{-1}(B_b)$. Since f_3 is a permutation, this gives $|S| = |B_b| \leq \lceil n/\log n \rceil$ and we are done. □

When bucketing, it is important to ensure that each player sees only a limited number of elements that could go in the bucket sent by the previous player because this in turn limits the size of the message he needs to sends to the next player. For example, in the above protocol, PLR_2 learns from PLR_1's message the identity of the bucket containing the desired output $f_3(f_2(i))$. Since f_3 permutes $[n]$, she is guaranteed to see only $\lceil n/\log n \rceil$ possibilities for this output. In the next two protocols, ensuring a similar bound is less trivial.

Theorem 37. *There is an $O(n \log\log n)$-communication collapsing protocol for instances (i, f_2, f_3) of $\widehat{\text{MPJ}}_3$ in which $f_2 \in \mathcal{S}_n$.*

Proof. For a function $f : [n] \to [n]$ and an integer $t \in [n]$, we define a bucketing scheme $\mathcal{B}^*(f, t)$ on $\text{Range}(f)$ via the following algorithm.

Algorithm $\mathcal{B}^*(f, t)$

1 $i \leftarrow 0\,;\ S \leftarrow \mathrm{Range}(f)$

2 **while** $S \neq \emptyset$

3 **do** $i \leftarrow i + 1$

4 viewing S in ascending order, move elements from S into B_i until $|f^{-1}(B_i)| \geq \frac{n}{t}$

5 **return** $(B_1, \ldots, B_i)$

We collect the salient properties of this scheme into the following lemma, omitting the easy proof.

Lemma 38. *The bucketing scheme $\mathcal{B}^*(f, t)$ is completely determined by the sorted list of values in $(f(1), f(2), \ldots, f(n))$. The scheme has at most t buckets. If B is one of those buckets and $m = \max\{s : s \in B\}$, then $|f^{-1}(B \setminus \{m\})| < n/t$.* □

Returning to the proof of Theorem 37, let us assume without loss of generality that n is a power of 2. We now describe our protocol. On input (i, f_2, f_3):

- PLR$_1$ sees $f_3 \circ f_2$ and computes $\mathcal{B} := \mathcal{B}^*(f_3 \circ f_2, \log n)$. She then sends the message $\alpha := \langle \mathcal{B}[f_3(f_2(1))], \mathcal{B}[f_3(f_2(2))], \ldots, \mathcal{B}[f_3(f_2(n))] \rangle$.

- PLR$_2$ sees i, f_3, and α, and computes $\mathcal{B} := \mathcal{B}^*(x, \log n)$; this computation is correct by Lemma 38, because f_2 is a permutation. From α, she recovers $b := \mathcal{B}[f_3(f_2(i))]$ and hence B_b. She determines $m := \max\{s : s \in B_b\}$ and $S := \{h \in [n] : f_3(h) \in B_b \setminus \{m\}\}$. Note that the definitions guarantee that either $f_2(i) \in S$ or else $f_3(f_2(i)) = m$. She sends $\beta := \langle \mathbf{1}_S, f_3|_S, m \rangle$.

- PLR$_3$ sees i, f_2, α, and β. If $f_2(i) \in S$, then she recovers $f_3(f_2(i))$ from the first two parts of β and outputs that. Otherwise, she outputs m from the third part of β.

Again, this protocol is clearly correct. As for the communication cost, note that it requires only $\lceil \log \log n \rceil$ bits to describe a bucket in $\mathcal{B}$. Therefore, PLR$_1$'s message uses $n\lceil \log \log n \rceil$

bits. As before, PLR_2's message requires $n + |S| \log n + \log n$ bits. To finish the proof, note that $S = f_3^{-1}(B_b \setminus \{m\})$ and use Lemma 38 to conclude that $|S| < n/\log n$. $\square$

The final collapsing protocol is for k players when all but one of the layers is a permutation. For $j = 1, \ldots, k-1$ let $b_j = \log^{(k-j)}(n)$ and $\hat{f}_j = f_k \circ \cdots \circ f_{j+1}$.

Theorem 39. *There is an $O(n \log^{(k-1)} n)$ protocol for $\widehat{\text{MPJ}}_k$ when all but one of $f_2, \ldots, f_k$ are permutations.*

Proof. This protocol is a hybrid of the two previous approaches. Without loss of generality, let f_{j+1} be the layer that is *not* a permutation. Note that for $1 \leq j^* \leq j$, $\{\hat{f}_{j^*}\}$ all represent (up to permutation) the same list of numbers. Similarly, for $j < j^* \leq k$, $\{\hat{f}_{j^*}\}$ all represent $[n]$ (up to permutation). Let $\mathcal{B}^*(f, t)$ denote the bucketing scheme used in Theorem 37, and for $t \in [n]$, define the bucketing scheme $\mathcal{B}_t = (B_1, \ldots, B_t)$ on $[n]$ by $B_j := \{r \in [n] : \lceil (rt)/n \rceil = j\}$. Note that $\mathcal{B}_{\log n}$ is equivalent to the bucketing scheme used in Theorem 36. On input $(i, f_2, \ldots, f_k)$,

- PLR_1 sees $\hat{f}_1$, computes $\mathcal{B} \leftarrow \mathcal{B}^*(\hat{f}_1, 2^{b_1})$, and sends $\langle \mathcal{B}[\hat{f}_1(1)], \ldots, \mathcal{B}[\hat{f}_1(n)] \rangle$.

- PLR_2 sees $\hat{i}_2, \hat{f}_2$, and the message from PLR_1. PLR_2 computes $\mathcal{B} \leftarrow \mathcal{B}^*(\hat{f}_2, 2^{b_1})$ and $\mathcal{B}' \leftarrow \mathcal{B}^*(\hat{f}_2, 2^{b_2})$. From PLR_1's message, he recovers $B \leftarrow \mathcal{B}[\hat{f}_1(i)]$. Let k_2 be the largest element of B, and let $S_2 = \{i \in [n] : \hat{f}_2(i) \in B \wedge \hat{f}_2(i) \neq k_2\}$. PLR_2 sends $\langle \mathbf{1}_{S_2}, \{\mathcal{B}'[\hat{f}_2(s)] : s \in S_2\}, k_2 \rangle$.

- PLR_3 sees $\hat{i}_3, \hat{f}_3$ and PLR_2's message. PLR_3 then computes $\mathcal{B} \leftarrow \mathcal{B}^*(\hat{f}_3, 2^{b_2})$ and $\mathcal{B}' \leftarrow \mathcal{B}^*(\hat{f}_3, 2^{b_3})$. First, PLR_3 checks to see if $\hat{i}_3 = f_2(i) \in S_2$. If $f_2(i) \notin S_2$, then he announces $\hat{f}_2(f_2(i)) = k_2$ and the protocol ends. Otherwise, he recovers $B \leftarrow \mathcal{B}[\hat{f}_2(f_2(i))]$ from the second part of PLR_2's message. Let k_3 be the greatest element of B, and let $S_3 = \{i \in [n] : \hat{f}_3(i) \in B \wedge \hat{f}_3(i) \neq k_3\}$. PLR_2 sends $\langle \mathbf{1}_{S_3}, \{\mathcal{B}'[\hat{f}_3(s)] : s \in S_3\}, k_3 \rangle$.

$\vdots$

- PLR_j sees $\hat{i}_j, \hat{f}_j$, and PLR_{j-1}'s message. PLR_j then computes $\mathcal{B} \leftarrow \mathcal{B}^*(\hat{f}_j, 2^{b_{j-1}})$ and $\mathcal{B}' \leftarrow \mathcal{B}_{b_j}$. PLR_j checks to see if $\hat{i}_j = f_{j-1}(\hat{i}_{j-1}) \in S_{j-1}$. if $\hat{i}_j \notin S_{j-1}$ then he announces $\hat{f}_{j-1}(f_{j-1}(\hat{i}_{j-1})) = \hat{f}_{j-1}(\hat{i}_j) = k_{j-1}$ and the protocol ends. Otherwise, he recovers $B \leftarrow \mathcal{B}[\hat{f}_{j-1}(f_{j-1}(\hat{i}_{j-1}))]$ from the second part of player $(j-1)$'s message. Let k_j be the greatest element of B, and let $S_j = \{i \in [n] : \hat{f}_j(i) \in B \wedge \hat{f}_j \neq k_j\}$. PLR_j sends $\langle \mathbf{1}_{S_j}, \{\mathcal{B}'[\hat{f}_j(s)] : s \in S_j\}, k_j \rangle$.

- PLR_{j+1} sees $\hat{i}_{j+1}, \hat{f}_{j+1}$, and PLR_j's message. PLR_{j+1} then computes $\mathcal{B} = \mathcal{B}_{b_{j+1}}$ and $\mathcal{B}' = \mathcal{B}_{b_{j+1}}$. PLR_{j+1} checks to see if $\hat{i}_{j+1} = f_j(\hat{i}_j) \in S_j$. If $\hat{i}_{j+1} \notin S_j$, then PLR_{j+1} announces $k_j = \hat{f}_j(f_j(\hat{i}_j))$ and the protocol ends. Otherwise,he recovers $B \leftarrow \mathcal{B}[\hat{f}_j(f_j(\hat{i}_j))]$. Let $S_{j+1} = \{i \in [n] : \hat{f}_{j+1}(i) \in B\}$. PLR_{j+1} sends $\langle \mathbf{1}_{S_{j+1}}, \{\mathcal{B}'[\hat{f}_{j+1}(s)] : s \in S_{j+1}\} \rangle$.

$\vdots$

- PLR_k sees $\hat{i}_k$ and PLR_{k-1}'s message and outputs $f_k(\hat{i}_k)$.

We claim that this protocol costs $O(n \log^{(k-1)} n)$ and correctly outputs $\widehat{\text{MPJ}}_k(i, f_2, \ldots, f_k)$. For all $j^* < j$, PLR_{j^*} buckets elements using the $\mathcal{B}^*$ scheme. Then, by definition of S_{j^*}, PLR_{j^*+1} knows that either $\hat{f}_{j^*}[f_{j^*}(\hat{i}_{j^*})] = k_{j^*}$, or that $f_{j^*}(\hat{i}_{j^*}) \in S_{j^*}$. In the former case, $k_{j^*} = \widehat{\text{MPJ}}_k(i, f_2, \ldots, f_k)$, and player j^*+1 outputs it directly, ending the protocol. In the latter case, he recovers the bucket B for $\hat{f}_{j^*}(f_{j^*}(\hat{i}_{j^*}))$ from the second part of player j^*s message. PLR_{j^*+1} identifies the elements of $\hat{f}_{j^*+1}$ that are elements of B and buckets these. The latter players bucket elements based on the $\mathcal{B}_t$ bucketing scheme. The correctness of this part of the protocol follows directly from Theorem 36. Finally, PLR_1 sends $b_1 = \log^{(k-1)} n$ bits to identify the bucket for each $i \in [n]$. Each successive PLR_j using the $\mathcal{B}^*$ scheme uses $n + b_j(n/b_j) + \log n =$ bits, and each PLR_j using the $\mathcal{B}_t$ scheme uses $n + b_j(n/b_j)$ bits. Thus, the maximum communication cost of this protocol is $O(n \log^{(k-1)} n)$ bits. When k is constant, this also gives a bound on the total communication cost. However, when $k = \omega(1)$, the total communication cost is $O(n \log^{(k-1)} n + kn)$ since later players each send $\Theta(n)$ bits. We address this issue in

two steps. First, we show an $O(n \log^{(k-1)} n)$ protocol for all $k \leq \log^* n$. Then, we handle the case where $k > \log^* n$.

Suppose then that $k \leq \log^* n$. Note that in the previous protocol, no matter which bucketing scheme PLR_j uses, he divides the input space into $2^{b_j} = b_{j+1}$ buckets, and therefore it costs b_j bits to describe each bucket. Note also that the resulting set S_{j+1} has size $n/2^{b_j} = n/b_{j+1}$. To get the communication cost to telescope, each PLR_j divides his input space into $2^{2b_j} = (b_{j+1})^2$ buckets. It costs twice as many bits to describe each bucket, but the resulting set S_{j+1} will have size $|S_{j+1}| \leq n/(b_{j+1})^2$. Thus, the second part of PLR_j's message will cost $2b_j n/(b_j^2) = 2n/b_j$ bits. Instead of sending n bits to send the characteristic vector of S_j, PLR_j can use the following fact and describe S_j using only $2nb_{j+1}/b_j^2 < 2n/b_j$ bits.

Fact 40. *For all* $1 \leq k \leq n$,

$$\log \binom{n}{n/k} \leq \frac{n \log k}{k}$$

In this new protocol, PLR_1 sends $n \log^{(k-1)} n$ bits, and for $1 < j < k$, PLR_j sends at most $3n/b_j + \log n$ bits. The total cost of the protocol is at most

$$\begin{aligned} n \log^{(k-1)} n + 3n \sum_{j-2}^{k-1} \frac{1}{\log^{(k-j)} n} + k \log n &< n \log^{(k-1)} n + O(n) + 3n \sum_{r=1}^{k-2} \frac{1}{\log^{(r)} n} \\ &< n \log^{(k-1)} n + \frac{3n}{\log^{(k-2)} n} \left(1 + \frac{1}{2} + \frac{1}{4} + \ldots\right) \\ &= n \log^{(k-1)} n \,, \end{aligned}$$

where the last inequality holds because $2 \log x \leq x$ for all $x \geq 2$ and $\log^{(r)} n > 2$ for all $r < \log^* n - 1$.

Finally, if $k > \log^* n$, then players can follow a $O(n)$ protocol by having all but the last $\log^* n$ players not communicating, and have the rest of the players use a $O(n \log^{(k-1)} n) = O(n)$ bit protocol. □

2.9 Concluding Remarks

We have presented the first nontrivial upper bound on the NOF communication complexity of the Boolean problem MPJ_k, showing that $C(\text{MPJ}_k) = o(n)$. A lower bound of $\Omega(n)$ had seemed *a priori* reasonable, but we show that it fails. One plausible line of attack on lower bounds for MPJ_k is to treat it as a *direct sum* problem: at each player's turn, it seems that n different paths need to be followed in the input graph, so it seems that an information theoretic approach (as in Bar-Yossef et al. [BJKS02] or Chakrabarti [Cha07]) could lower bound $C(\text{MPJ}_k)$ by n times the complexity of some simpler problem. However, it appears that such an approach would naturally yield a lower bound of the form $\Omega(n/\xi(k))$, as in Conjecture 1, which we have falsified.

The most outstanding open problem regarding MPJ_k is to resolve Conjecture 2. A less ambitious, but seemingly difficult, goal is to get tight bounds on $C(\text{MPJ}_3)$, closing the gap between our $O(n\sqrt{\log\log n/\log n})$ upper bound and Wigderson's $\Omega(\sqrt{n})$ lower bound. A still less ambitious question is prove that MPJ_3 is harder than its very special subproblem TPJ_3 (defined in Section 2.1.1). Our $n - O(\log n)$ lower bound for collapsing protocols is a step in the direction of improving the known lower bounds. We hope our technique provides some insight about the more general problem.

Chapter 3

Distributed Functional Monitoring

The notion of *distributed functional monitoring* was recently introduced by Cormode, Muthukrishnan and Yi [CMY08] to initiate a formal study of the communication cost of certain fundamental problems arising in distributed systems, especially sensor networks. In this model, each of k sites reads a stream of tokens and is in communication with a central coordinator, who wishes to continuously monitor some function f of σ, the union of the k streams. The goal is to minimize the number of bits communicated by a protocol that correctly monitors $f(\sigma)$, to within some small error. As in previous work, we focus on a threshold version of the problem, where the coordinator's task is simply to maintain a single output bit, which is 0 whenever $f(\sigma) \leq \tau(1-\varepsilon)$ and 1 whenever $f(\sigma) \geq \tau$. Following Cormode et al., we term this the $(k, f, \tau, \varepsilon)$ functional monitoring problem.

In previous work, some upper and lower bounds were obtained for this problem, with f being a frequency moment function, e.g., F_0, F_1, F_2. Importantly, these functions are *monotone*. Here, we further advance the study of such problems, proving two new classes of results. First, we study the effect of non-monotonicity of f on our ability to give nontrivial monitoring protocols, by considering $f = F_p$ with deletions allowed, as well as $f = H$. Second, we prove new lower bounds on this problem when $f = F_p$, for several values of p. In [ABC09], we

provide nontrivial monitoring protocols when f is either H, the empirical Shannon entropy of a stream, or any of a related class of entropy functions (Tsallis entropies). These are the first nontrivial algorithms for distributed monitoring of non-monotone functions.

3.1 Introduction

Energy efficiency is a key issue in sensor network systems. Communication, which typically uses power-hungry radio, is a vital resource whose usage needs to be minimized [EGHK99]. Several other distributed systems have a similar need for minimizing communication. This is the primary motivation for our present work, which is a natural successor to the recent work of Cormode, Muthukrishnan and Yi [CMY08], who introduced a clean formal model to study this issue. The formalization, known as *distributed functional monitoring*, involves a multi-party communication model consisting of k *sites* (the sensors, in a sensor network) and a single central *coordinator*. Each site asynchronously receives "readings" from its environment, formalized as a *data stream* consisting of *tokens* from a discrete universe. The union of these streams defines an overall input stream σ that the coordinator wishes to monitor continuously, using an appropriate protocol involving private two-way communication channels between the coordinator and each site. Specifically, the coordinator wants to continuously maintain approximate knowledge of some nonnegative real-valued function f of σ. (We assume that f is invariant under permutations of σ, which justifies our use of "union" above, rather than "concatenation.")

As is often the case in computer science, the essence of this problem is captured by a threshold version with Boolean outputs. Specifically, we have a threshold $\tau \in \mathbb{R}_+$ and an approximation parameter $\varepsilon \in \mathbb{R}_+$, and we require the coordinator to continuously maintain an output bit, which should be 0 whenever $f(\sigma) \leq \tau(1-\varepsilon)$ and 1 whenever $f(\sigma) \geq \tau$.[1]

[1]Clearly, a solution to the value monitoring problem solves this threshold version, and the value monitoring problem can be solved by running, in parallel, several copies of a solution to this threshold version with geomet-

Following [CMY08], we call this the $(k, f, \tau, \varepsilon)$ functional monitoring problem. This formulation of the problem combines aspects of streaming algorithms, sketching and communication complexity.

Motivation. Plenty of recent research has studied such continuous monitoring problems, for several special classes of functions f (see, e.g., [BO03, DGGR04, CMZ06, SSK07]). Applications have arisen not only in sensor networks, but also in more general network and database settings. However, most of this past work had not provided formal bounds on communication cost, an issue that was first addressed in detail in [CMY08], and that we continue to address here. Philosophically, the study of such monitoring problems is a vast generalization of Slepian-Wolf style distributed source coding [SW73] in much the same way that communication complexity is a vast generalization of basic source coding in information theory. Furthermore, while the problems and the model we consider here are strongly reminiscent of streaming algorithms, there are notable additional challenges: for instance, maintaining an approximate count of the total number of tokens received is a nontrivial problem in our setting, but is trivial in the streaming model. For a more detailed discussion of prior research, we refer the reader to [CMY08] and the references therein.

Our Results and Comparison with Prior Work. Our work studies $(k, f, \tau, \varepsilon)$ functional monitoring for two natural classes of functions f: the empirical Shannon entropy H and the frequency moments F_p. For an input stream σ of tokens from the universe $[n] := \{1, 2, \ldots, n\}$, let f_i denote the number of appearances of i in σ, where $i \in [n]$. For $p \geq 0$, the pth frequency moment $F_p(\sigma)$ is defined to be $\sum_{i=1}^{n} f_i^p$. Note that p can be non-integral or zero: indeed, using the convention $0^0 = 0$ makes $F_0(\sigma)$ equal to the number of distinct tokens in σ. These functions F_p capture important statistical properties of the stream and have been studied heavily in the streaming algorithms literature [AMS99, Mut03]. The stream σ also implicitly defines

rically spaced thresholds.

a probability distribution over $[n]$, given by $\Pr[i] = f_i/m$, where m is the length of σ. For various applications, especially ones related to anomaly detection in networks, the entropy of this distribution — also called the empirical entropy of the stream — is a measure of interest. Abusing notation somewhat, we denote this as $H(\sigma)$, when the underlying entropy measure is Shannon entropy: thus, $H(\sigma) = \sum_{i=1}^{n} (f_i/m) \log(m/f_i)$.[2]

We study the effect of *non-monotonicity* of f on the $(k, f, \tau, \varepsilon)$ problem: the bounds of Cormode et al. [CMY08] crucially exploited the fact that the functions being monitored were monotone nondecreasing. We obtain two new classes of results. First, we prove lower bounds for monitoring $f = F_p$ with *deletions* allowed: i.e., the stream can contain "negative tokens" that effectively delete earlier tokens. In contrast with the good upper bounds in [CMY08] for monitoring F_p *without* deletions (a monotone problem), we show that essentially no nontrivial upper bounds are possible. Using similar techniques, we also give a lower bound for monitoring H that is necessarily much milder, and in the same ballpark as our upper bound.

Secondly, we prove new lower bounds for the monotone problems $f = F_p$, without deletions, for various values of p. These improve or are incomparable with previous bounds [CMY08]; see Table 3.1 for a side-by-side comparison.

In [ABC09], we provide nontrivial monitoring protocols for H, and the related functions T_α. For this, we suitably extend recent sketching algorithms such as those due to Bhuvanagiri and Ganguly [BG06] and Harvey et al. [HNO08]. These are the first nontrivial algorithms for monitoring non-monotone functions. Our algorithms, which are simple and easily usable, can monitor continuously until the end of the stream, even as the $f(\sigma)$ crosses the threshold multiple times. This is the desired behavior when monitoring non-monotone functions.

Notation, etc. We now define some notation that we use at various points in this chapter. We use $|\sigma|$ to denote the length of the stream σ and $\sigma_1 \circ \sigma_2$ to denote the concatenation: σ_1 followed

[2]Throughout this chapter we use "$\log$" to denote logarithm to the base 2 and "$\ln$" to denote natural logarithm.

Problem	Previous Results	Our Results
H, deterministic	$O(m)$, trivially	$\Omega(k\varepsilon^{-1/2}\log m)$
H, randomized		$\widetilde{O}(k\varepsilon^{-3}\log^4 m)$, $\Omega(\varepsilon^{-1/2}\log m)$
F_p, dels., determ.		$\Omega(m)$
F_p, dels., rand.		$\Omega(m/k)$
F_1, deterministic	$O(k\log(1/\varepsilon))$, $\Omega(k\log(1/(\varepsilon k)))$	$\Omega(k\log(1/\varepsilon))$
F_0, randomized	$\widetilde{O}(k/\varepsilon^2)$, $\Omega(k)$	$\Omega(1/\varepsilon)$, $\Omega(1/\varepsilon^2)$ if round-based
$F_p, p>1$, rand.	$\widetilde{O}(k^2/\varepsilon + (\sqrt{k}/\varepsilon)^3)$, $\Omega(k)$, for $p=2$	$\Omega(1/\varepsilon)$, $\Omega(1/\varepsilon^2)$ if round-based

Table 3.1: Summary of our results (somewhat simplified) and comparison with previous work [CMY08]. Dependence on τ is not shown here, but is stated in the relevant theorems.

by σ_2. We typically use $S_1, \ldots, S_k$ to denote the k sites, and C to denote the coordinator, in a $(k, f, \tau, \varepsilon)$ functional monitoring protocol. We tacitly assume that randomized protocols use a public coin and err with probability at most $1/3$ at each point of time. These assumptions do not lose generality, as shown by appropriate parallel repetition and the private-versus-public-coin theorem of Newman [New91]. We use m to denote the overall input length (i.e., number of tokens) seen by the protocol under consideration. We state our communication bounds in terms of m, k and ε, and sometimes τ.

3.2 Formal Definition

In this section, we formally define *distributed functional monitoring*. Let $\sigma^1, \ldots, \sigma^k$ denote aribtrary streams of tokens from a finite universe $\mathcal{U} = [m]$, and let σ denote the union of $\sigma^1, \ldots, \sigma^k$. The following definition formalizes our model.

Definition 1. *For all $k \in \mathbb{N}_+$, all functions $f : \mathcal{U}^* \to \mathbb{R}_+$, and all $\varepsilon, \tau \in \mathbb{R}_+$, the $(k, f, \tau, \varepsilon)$-distributed functional monitoring problem is defined as follows. Each of k sites receives a stream of tokens σ^i from $\mathcal{U}$ and has a bidirectional communication channel with a central*

coordinator, who must continuously output 0 *whenever* $f(\sigma) \leq \tau(1-\varepsilon)$ *and* 1 *whenever* $f(\sigma) \geq \tau$.

In this chapter, we are interested in protocols which solve the $(k, f, \tau, \varepsilon)$-distributed functional monitoring problem for several different functions f. Our goal is to minimize the amount of communication required.

3.3 Lower Bounds for Non-Monotone Functions

We now give lower bounds for estimating entropy, and later, F_p. We give deterministic bounds first, and then randomized bounds. We abuse notation and let H denote both the empirical entropy of a stream and the binary entropy function $H : [0,1] \to [0,1]$ given by $H(x) = -x \log x - (1-x)\log(1-x)$.

Theorem 41. *For any* $\varepsilon < 1/2$ *and* $m \geq k/\sqrt{\varepsilon}$*, a deterministic algorithm solving* $(k, H, \tau, \varepsilon)$ *functional monitoring must communicate* $\Omega(k\varepsilon^{-1/2}\log(\varepsilon m/k))$ *bits.*

Proof. We use an adversarial argument that proceeds in rounds. Each round, the adversary will force the protocol to send at least one bit. The result will follow by showing a lower bound on the number of rounds r that the adversary can create, using no more than m tokens. Let $\tau = 1$, and let z be the unique positive real such that $H(\frac{z}{2z+1}) = 1-\varepsilon$. Note that this implies $H(\frac{z}{2z+1}) > 1/2 > H(1/10)$, whence $\frac{z}{2z+1} > 1/10$, hence $z > 1/8$. An estimation of H using calculus shows that $z = \Theta(1/\sqrt{\varepsilon})$. Fix a monitoring protocol $\mathcal{P}$. The adversary only uses tokens from $\{0,1\}$, i.e., the stream will induce a two-point probability distribution.

The adversary starts with a "round 0" in which he sends nine 1s followed by a 0 to site S_1. Note that at the end of round 0, the entropy of the stream is $H(1/10) < 1/2$. For $i \in \{0, 1, \ldots, r\}$, let a_i denote the number of 0s and b_i the number of 1s in the stream at the end of round i. Then $a_0 = 1$ and $b_0 = 9$. For all $i > 0$, the adversary maintains the invariant that

$b_i = \lceil a_i(z+1)/z \rceil$. This ensures that at the end of round i, the empirical entropy of the stream is

$$H\left(\frac{a_i}{a_i+b_i}\right) \;\leq\; H\left(\frac{a_i}{a_i(1+(z+1)/z)}\right) \;=\; H\left(\frac{z}{2z+1}\right) \;=\; 1-\varepsilon\,,$$

which requires the coordinator to output 0.

Consider the situation at the start of round i, where $i \geq 1$. If each player were to receive $\lceil (b_{i-1}-a_{i-1})/k \rceil$ 0-tokens in this round, then at some point the number of 0s in the stream would equal the number of 1s, which would make the empirical entropy equal to 1 and require the coordinator to change his output to 1. Therefore, there must exist a site S_{j_i}, $j_i \in [k]$, who would communicate upon receiving these many 0-tokens in round i. In actuality, the adversary does the following in round i: he sends these many 0s to S_{j_i}, followed by as many 1s as required to restore the invariant, i.e., to cause $b_i = \lceil a_i(z+1)/z \rceil$. Clearly, this strategy forces at least one bit of communication per round. It remains to bound r from below. Note that the adversary's invariant implies $b_i - a_i \leq a_i/z + 1$ and $a_i + b_i \leq a_i(2z+1)/z + 1 = a_i(2+1/z)+1$. Therefore, we have

$$a_i \;=\; a_{i-1} + \left\lceil \frac{b_{i-1}-a_{i-1}}{k} \right\rceil \;\leq\; a_{i-1} + \left\lceil \frac{1+a_{i-1}/z}{k} \right\rceil \;\leq\; a_{i-1}\left(1+\frac{1}{zk}\right)+2\,.$$

Setting $\alpha = (1+1/zk)$ and iterating gives $a_r \leq a_0\alpha^r + 2(\alpha^r-1)/(\alpha-1) = a_0\alpha^r + 2zk(\alpha^r-1) = \alpha^r(a_0+2zk) - 2zk$. Using our upper bound on a_i+b_i, the above inequality, and the facts that $a_0 = 1$ and that $z > 1/8$, we obtain

$$\begin{aligned}
a_r + b_r &\leq \alpha^r\,(1+2zk)\,(2+1/z) - 2zk(2+1/z) + 1 \\
&\leq (2+1/z)\,(1+2zk)\,\alpha^r \\
&\leq (2+1/z)\,(1+2zk)\,e^{r/zk} \\
&\leq 60zke^{r/zk}\,.
\end{aligned}$$

Therefore, we can have $a_r + b_r \leq m$, provided $r \leq zk\ln(m/(60zk))$. Recalling that $z = \Theta(1/\sqrt{\varepsilon})$, we get the claimed lower bound of $\Omega(k\varepsilon^{-1/2}\log(\varepsilon m/k))$. $\square$

Our next lower bounds are for functional monitoring of frequency moments when we allow for deletions. Specifically, we now consider *update streams* that consist of tokens of the form (i, v), where $i \in [n]$ and $v \in \{-1, 1\}$, to be thought of as updates to a vector $(f_1, \ldots, f_n)$ of frequencies. The vector is initially zero and is updated using $f_i \leftarrow f_i + v$ upon receipt of the token (i, v); in this way, each update either adds or deletes one copy of item i.

As usual, we let m denote the length of an update stream whose tokens are distributed amongst several sites. Our next results essentially show that **no** nontrivial savings in communication is possible for the problem of monitoring frequency moments in this setting. These bounds highlight the precise problem caused by the non-monotonicity of the function being monitored. They should be contrasted with the much smaller upper bounds achievable in the monotone case, when there are no deletions (see Table 3.1).

Our proofs are again adversarial and proceed in rounds. They use appropriate instantiations of the following generic lemma.

Definition 2. *An update stream is said to be* positive *if it consists entirely of tokens from $[n] \times \{1\}$, i.e., insertions only. The* inverse *of an update stream $\sigma = \langle (i_1, v_1), \ldots, (i_m, v_m)\rangle$ is defined to be $\sigma^{-1} := \langle (i_m, -v_m), \ldots, (i_1, -v_1)\rangle$. A function $G : \mathbb{Z}_+^n \to \mathbb{R}_+$ on frequency vectors is said to be* monotone *if G is nondecreasing in each parameter, separately. We extend such a G to a function on streams (or update streams) in the natural way, and write $G(\sigma)$ to denote $G(\vec{f})$, where $\vec{f}$ is the frequency vector determined by σ.*

Lemma 42. *Let $G : \mathbb{Z}_+^n \to \mathbb{R}_+$ be monotone and let $\mathcal{P}$ be a protocol for the $(k, G, \tau, \varepsilon)$ functional monitoring problem with deletions allowed. Let $\sigma_0, \sigma_1, \ldots, \sigma_k$ be a collection of positive update streams such that (1) $G(\sigma_0) \leq \tau(1-\varepsilon)$, and (2) $G(\sigma_0 \circ \sigma_1 \circ \ldots \circ \sigma_k) \geq \tau$. If $\mathcal{P}$ is a deterministic protocol, then at least $\left\lfloor |(m - |\sigma_0|) \,/\, (2 \cdot \max_{j\in[k]}\{|\sigma_j|\})\right\rfloor$ are communicated.*

If $\mathcal{P}$ is a δ-error randomized protocol, then the expected number of bits communicated is at least $((1-\delta)/k) \cdot \left\lfloor (m - |\sigma_0|) \,/\, (2 \cdot \max_{j \in [k]}\{|\sigma_j|\}) \right\rfloor$.

Proof. Let $S_1, \ldots, S_k$ be the k sites involved in $\mathcal{P}$. The adversary will send certain tokens to certain sites, maintaining the invariant that the coordinator is always required to output 0. In round 0, the adversary sends σ_0 to S_1; by condition (1), this maintains the invariant.

Let $s = \max_{j \in [k]}\{|\sigma_j|\}$ and $r = \lfloor (m - |\sigma_0|)/2s \rfloor$. The adversary uses r additional rounds maintaining the additional invariant that at the start of each such round the value of G is $G(\sigma_0)$. Consider round i, where $i \in [r]$. By condition (2), if the adversary were to send σ_j to S_j in this round, for each $j \in [k]$, the coordinator's output would have to change to 1.

Suppose $\mathcal{P}$ is a deterministic protocol. Then, since the coordinator's output would have to change to 1, there must exist a site S_{j_i}, with $j_i \in [k]$, that would have to communicate upon receiving σ_{j_i} in this round. In actuality, the adversary sends $\sigma_{j_i} \circ \sigma_{j_i}^{-1}$ to S_{j_i} and nothing to any other site in round i. Clearly, this maintains both invariants and causes at least one bit of communication. Also, this adds at most $2s$ tokens to the overall input stream. Thus, the adversary can cause r bits of communication using $|\sigma_0| + 2sr \leq m$ tokens in all, which proves the claim for deterministic protocols.

The proof when $\mathcal{P}$ is a δ-error randomized protocol proceeds in a similar manner. The difference is that each round i has an associated collection of probabilities $(p_{i1}, \ldots, p_{ik})$, where $p_{ij} = \Pr[S_j \text{ communicates in round } i \text{ upon receiving } \sigma_j]$. As before, condition (2) implies that were each S_j to receive σ_j in this round, correctness would require C's output to change to 1. Thus,

$$1 - \delta \;\leq\; \Pr[\mathcal{P} \text{ is correct}] \;\leq\; \Pr[C \text{ receives a bit in round } i] \;\leq\; \textstyle\sum_{j=1}^{k} p_{ij}\,,$$

where the final inequality uses a union bound. Therefore, there exists a site S_{j_i}, with $j_i \in [k]$, having $p_{ij_i} \geq (1-\delta)/k$. Again, as in the deterministic case, the adversary actually sends

$\sigma_{j_i} \circ \sigma_{j_i}^{-1}$ to S_{j_i} and nothing to any other site in round i. By linearity of expectation, the expected total communication with r rounds is at least $r(1-\delta)/k$, which proves the lemma. □

The theorems that follow are for randomized protocols with error $\delta = 1/3$.

Theorem 43. *The expected communication cost of a randomized $(k, F_0, \tau, \varepsilon)$ functional monitoring protocol that allows for deletions is $\Omega(\min\{m/k, m/\varepsilon\tau\})$.*

Proof. Let $a := \max\{1, \lceil \frac{\tau\varepsilon}{k} \rceil\}$, and instantiate σ_0 as a stream of $\tau - ka$ distinct elements and $\sigma_1, \ldots, \sigma_k$ each as a stream of a distinct elements. Note that $ka \geq \tau\varepsilon$, so $F_0(\sigma_0) = \tau - ka \leq \tau(1-\varepsilon)$. Furthermore, note that $F_0(\sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_k) = \tau$, hence the streams satisfy the conditions of Lemma 42 with $G = F_0$. Applying that lemma, and noting that $|\sigma_j| = a$ gives us a lower bound of $((1-\delta)/k) \cdot \lfloor (m - |\sigma_0|)/(2a) \rfloor = \Omega(\min\{m/k, m/\varepsilon\tau\})$ for large enough m. □

Note that Lemma 42 implies a slightly stronger result for deterministic protocols that monitor frequency moments; however, a linear lower bound is already known, even without deletions, by the same techniques used in [AMS99] to prove lower bounds in the streaming model.

The proof of the next theorem is similar to that of Theorem 43.

Theorem 44. *The expected communication cost of a randomized $(k, F_p, \tau, \varepsilon)$ monitoring protocol (with $p > 0$) that allows deletions is $\Omega(\min\{m/k, mp/\tau^{1/p}\varepsilon\})$.*

Proof. Let $s_0 := (1-\varepsilon)^{1/p}\tau^{1/p}$, $s_1 := \tau^{1/p}$, and $a := 1 + \lceil (s_1 - s_0)/k \rceil$. Instantiate σ_0 as a stream of $\lfloor s_0 \rfloor$ insertions of the token "1", and instantiate $\sigma_1, \ldots, \sigma_k$ each as a stream of a insertions of the token "1". Note that

$$F_p(\sigma_0) \leq s_0^p = \tau(1-\varepsilon)\,,$$

and that

$$\begin{aligned} F_p(\sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_k) &\geq (\lfloor s_0 \rfloor + k\,(1 + \lceil (s_1 - s_0)/k \rceil))^p \\ &\geq (\lfloor s_0 \rfloor + 1 + s_1 - s_0)^p \\ &\geq s_1^p = \tau\,. \end{aligned}$$

hence the streams satisfy the conditions of Lemma 42 with $G = F_p$. Applying that lemma, and noting that $|\sigma_j| = a$ gives us a lower bound of $((1-\delta)/k) \cdot \lfloor (m - |\sigma_0|)/(2a) \rfloor$, which is $\Omega(\min\{m/k, mp/\tau^{1/p}\varepsilon\})$ for large enough m. □

Theorem 45. *The expected communication cost of a randomized $(k, H, \tau, \varepsilon)$ functional monitoring protocol is $\Omega(\varepsilon^{-1/2} \log(\varepsilon m/k))$ bits.*

We note that Yi and Zhang [YZ09] study problems similar to ours but in terms of competitive ratio. The bounds in this section rely on the construction of hard instances which might not be possible in their case.

3.4 Frequency Moments Without Deletions: New Bounds

We finish with another set of lower bounds, this time for monitoring F_p (for various p) without deletions. Our bounds either improve or are incomparable with previous lower bounds: see Table 3.1.

Theorem 46. *A deterministic protocol that solves $(k, F_1, \tau, \varepsilon)$ functional monitoring must communicate at least $\Omega\big(k \log \frac{k+\tau}{k+\varepsilon\tau}\big)$ bits. In particular, when $\tau \geq k/\varepsilon^{\Omega(1)}$, it must communicate $\Omega\left(k \log(1/\varepsilon)\right)$ bits.*

Proof. Again we use an adversary, who proceeds in rounds: each round, he gives just enough tokens to a single site to force that site to communicate.

Let $a_0 = 0$ and, for $i \geq 1$, let a_i be the total number of tokens received by all sites (i.e., the value of F_1 for the input stream) at the end of round i. The adversary maintains the invariant that $a_i \leq \tau(1-\varepsilon)$, so that the coordinator must always output 0. For $j \in [k]$, let b_{ij} be the maximum number of tokens that site j can receive in round i without being required to communicate. The correctness of the protocol requires $a_{i-1} + \sum_{j=1}^{k} b_{ij} < \tau$, for otherwise the desired output can change from 0 to 1 without the coordinator having received any communication. Let $j^* = \operatorname{argmin}_{j\in[k]}\{b_{ij}\}$. In round i, the adversary sends $b_{ij^*}+1$ tokens to site j^*, forcing it to communicate. We have

$$a_i = a_{i-1} + b_{ij^*} + 1 \leq a_{i-1} + \frac{\tau - a_{i-1}}{k} + 1 = 1 + \frac{\tau}{k} + \left(1 - \frac{1}{k}\right) a_{i-1}\,.$$

Letting $\alpha = 1 - 1/k$ and iterating the above recurrence gives

$$a_i \leq (1+\tau/k)(1-\alpha^i)/(1-\alpha) = (k+\tau)(1-\alpha^i).$$

Now note that $\alpha \geq e^{-2/k}$, so when $i \leq r := \frac{k}{2}\ln\frac{k+\tau}{k+\varepsilon\tau}$, we have $\alpha^i \geq \frac{k+\varepsilon\tau}{k+\tau}$, so that

$$a_i \leq (\tau+k)\left(\frac{k+\tau-k-\varepsilon\tau}{k+\tau}\right) = \tau(1-\varepsilon)\,.$$

This shows that the adversary can maintain the invariant for up to r rounds, forcing $\Omega(r)$ bits of communication, as claimed. □

Our next lower bounds use reductions from a fundamental problem in communication complexity: the Gap Hamming distance problem. We discuss Gap Hamming Distance more in Chapter 4. In this chapter, we use a parameterized version of Gap Hamming Distance, denoted GHD_c, where $c \in \mathbb{R}_+$ is a parameter. In this problem, Alice and Bob are given $x, y \in \{0,1\}^n$ respectively and want to output 1 if $\Delta(x,y) \geq \frac{n}{2} + c\sqrt{n}$ and 0 if $\Delta(x,y) \leq \frac{n}{2} - c\sqrt{n}$; they don't care what happens if the input satisfies neither of these conditions. We shall need the following

lower bounds on the randomized communication complexity $\mathrm{R}(\mathrm{GHD}_c)$, as well as the one-way randomized communication complexity (where the only communication is from Alice to Bob) $\mathrm{R}^{\rightarrow}(\mathrm{GHD}_c)$. Proofs of these bounds can be found in [Woo07]. Further background background on the problem can be found in Woodruff [Woo07] and in Chapter 4.

Theorem 47. *Suppose $c > 0$ is a constant. Then $\mathrm{R}(\mathrm{GHD}_c) = \Omega(\sqrt{n})$ and $\mathrm{R}^{\rightarrow}(\mathrm{GHD}_c) = \Omega(n)$. Here, the Ω notation hides factors dependent upon c.*[3]

It is conjectured that the general randomized bound is in fact as strong as the one-way version. This is not just a tantalizing conjecture about a basic communication problem. Settling it would have important consequences because, for instance, the Gap Hamming distance problem is central to a number of results in streaming algorithms. As we shall soon see, it would also have consequences for our work here.

Conjecture 48. *For sufficiently small constants c, we have $\mathrm{R}(\mathrm{GHD}_c) = \Omega(n)$.*

Remark. *For a long time, it was conjectured that the general randomized bound is as strong as the one-way version. In Chapter 4, we make the first progress on this conjecture, showing first an $\Omega(n/2^{O(k^2)})$ and then an improved $\tilde{\Omega}(n/k^2)$ bound on the communication complexity of any k-round randomized protocol for GHD_c. However, these results have little bearing on our work here, because the reduction below uses a potentially large value of k (up to $k \leq n$). However, a very recent result of Chakrabarti and Regev [CR10] finally showed a general lower bound of $R(\mathrm{GHD}_c) = \Omega(n)$. Thus, the bounds in Theorems 49 and 50 are now $\Omega(1/\varepsilon^2)$, and Theorem 51 is subsumed. We believe this work is of independent interest and there for include these results for pedagogical value.*

Theorem 49. *For any $\varepsilon \leq 1/2$, a randomized protocol for $(k, F_0, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(1/\varepsilon)$ bits.*

[3]The bounds in [Woo07] restrict the range of c, but this turns out not to be necessary.

Proof. We give a reduction from GHD_1. Let $\mathcal{P}$ be a randomized protocol for $(k, F_0, \tau, \varepsilon)$ functional monitoring. Set $N := \lfloor 1/\varepsilon^2 \rfloor$ and $\tau = 3N/2 + \sqrt{N}$. We design a two-party public coin randomized communication protocol $\mathcal{Q}$ for GHD_1 on N-bit inputs that simulates a run of $\mathcal{P}$ involving the coordinator, C, and two sites, S_1 and S_2. Let $x \in \{0,1\}^N$ be Alice's input in $\mathcal{Q}$ and let $y \in \{0,1\}^N$ be Bob's input. Alice creates a stream $\sigma_a := \langle a_1, \ldots, a_N \rangle$ of tokens from $[N] \times \{0,1\}$ by letting $a_i := (i, x_i)$ and Bob similarly creates a stream $\sigma_b := \langle b_1, \ldots, b_N \rangle$, where $b_i := (i, y_i)$. They then simulate a run of $\mathcal{P}$ where S_1 first receives all of σ_a after which S_2 receives all of σ_b. They output whatever the coordinator would have output at the end of this run.

The simulation itself occurs as follows: Alice maintains the state of S_1, Bob maintains the state of S_2, and they *both* maintain the state of C. Clearly, this can be done by having Alice send to Bob all of S_1's messages to C plus C's messages to S_2 (and having Bob act similarly). The total communication in $\mathcal{Q}$ is at most that in $\mathcal{P}$.

We now show that $\mathcal{Q}$ is correct. By construction, the combined input stream $\sigma = \sigma_a \circ \sigma_b$ seen by $\mathcal{P}$ has $2\Delta(x,y)$ tokens with frequency 1 each and $N - \Delta(x,y)$ tokens with frequency 2 each. Therefore $F_0(\sigma) = N + \Delta(x,y)$. When $\Delta(x,y) \geq N/2 + \sqrt{N}$, we have $F_0(\sigma) \geq \tau$ and $\mathcal{Q}$, following $\mathcal{P}$, correctly outputs 1. On the other hand, when $\Delta(x,y) \leq N/2 - \sqrt{N}$, we have

$$F_0(\sigma) \;\leq\; \frac{3N}{2} - \sqrt{N} \;=\; \tau\left(1 - \frac{2\sqrt{N}}{3N/2 + \sqrt{N}}\right) \;\leq\; \tau\left(1 - \frac{1}{\sqrt{N}}\right) \;\leq\; \tau(1-\varepsilon)\,.$$

Thus $\mathcal{Q}$ correctly outputs 0. Since $\mathcal{Q}$ is correct, by Theorem 47, it must communicate at least $\Omega(\sqrt{N}) = \Omega(1/\varepsilon)$ bits. Therefore, so must $\mathcal{P}$. □

Theorem 50. *For any $\varepsilon < 1/2$ and any constant $p > 1$, a randomized protocol for $(k, F_p, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(1/\varepsilon)$ bits.*

Proof. For simplicity, we assume here that $p \geq 2$. As before, we reduce from GHD_1 on

$N := \lfloor 1/\varepsilon^2 \rfloor$-bit inputs. For this reduction, we set $\tau := (N/2 + \sqrt{N})2^p + (N - 2\sqrt{N})$. Let $\mathcal{P}$ be a protocol for $(k, F_p, \tau, \varepsilon)$ functional monitoring. We design a protocol $\mathcal{Q}$ for GHD_1 on input (x, y) that simulates a run of $\mathcal{P}$ involving two sites, creating two streams $\langle (i, x_i) \rangle_{i \in [N]}$ and $\langle (i, y_i) \rangle_{i \in [N]}$, exactly as before; however, in this reduction, the output of $\mathcal{Q}$ is the *opposite* of the coordinator's output at the end of the run of $\mathcal{P}$.

We now show that $\mathcal{Q}$ is correct. The input stream σ seen by $\mathcal{P}$ has the same frequency distribution as before, which means that

$$F_p(\sigma) = 2\Delta(x, y) + (N - \Delta(x, y)) \cdot 2^p = N \cdot 2^p - \Delta(x, y)(2^p - 2)\,.$$

When $\Delta(x, y) \leq N/2 - \sqrt{N}$, we have

$$\begin{aligned} F_p(\sigma) &\geq N \cdot 2^p - (N/2 - \sqrt{N})(2^p - 2) \\ &= (N/2 + \sqrt{N})2^p + (N - 2\sqrt{N}) \\ &= \tau\,. \end{aligned}$$

Therefore $\mathcal{P}$ outputs 1, which means $\mathcal{Q}$ correctly outputs 0. On the other hand, when $\Delta(x, y) \geq N/2 + \sqrt{N}$, we have

$$\begin{aligned} F_p(\sigma) &\leq N \cdot 2^p - (N/2 + \sqrt{N})(2^p - 2) \\ &= \tau \left(1 - \frac{2\sqrt{N}2^p - 4\sqrt{N}}{(N/2 + \sqrt{N}) \cdot 2^p + (N - 2\sqrt{N})} \right) \\ &\leq \tau(1 - 1/\sqrt{N}) \\ &\leq \tau(1 - \varepsilon)\,, \end{aligned}$$

where the penultimate inequality uses $p \geq 2$. Therefore $\mathcal{P}$ outputs 0, whence $\mathcal{Q}$ correctly outputs 1. Theorem 47 now implies that $\mathcal{Q}$, and hence $\mathcal{P}$, must communicate $\Omega(\sqrt{N}) =$

$\Omega(1/\varepsilon)$ bits. □

We remark that if Conjecture 48 holds (for a favorable c), then the lower bounds in Theorems 49 and 50 would improve to $\Omega(1/\varepsilon^2)$. This further strengthens the motivation for settling the conjecture.

We also consider a restricted, yet natural, class of protocols that we call *round-based* protocols; the precise definition follows. Note that all nontrivial protocols in [CMY08] are round-based, which illustrates the naturalness of this notion.

Definition 3. *A* round-based *protocol for* $(k, f, \tau, \varepsilon)$ *functional monitoring is one that proceeds in a series of rounds numbered* $1, \ldots, r$. *Each round has the following four stages. (1) Coordinator* C *sends messages to the sites* S_i*, based on the past communication history. (2) Each* S_i *read its tokens and sends messages to* C *from time to time, based on these tokens and the Stage* 1 *message from* C *to* S_i*. (3) At some point, based on the messages it receives,* C *decides to end the current round by sending a special, fixed, end-of-round message to each* S_i*. (4) Each* S_i *sends* C *a final message for the round, based on all its knowledge, and then resets itself, forgetting all previously read tokens and messages.*

It is possible to improve the lower bounds above by restricting to round-based protocols, as in Definition 3. The key is that if the functional monitoring protocol $\mathcal{P}$ in the proofs of Theorems 49 and 50 is round-based, then the corresponding communication protocol $\mathcal{Q}$ only requires messages from Alice to Bob. This is because Alice can now simulate the coordinator C and *both* sites S_1 and S_2, during $\mathcal{P}$'s processing of σ_a: she knows that S_2 receives no tokens at this time, so she has the information needed to compute any messages that S_2 might need to send. Consider the situation when Alice is done processing her tokens. At this time the Stage 4 message (see Definition 3) from S_1 to C in the current round has been determined, so Alice can send this message to Bob. From here on, Bob has all the information needed to continue simulating S_1, because he knows that S_1 receives no further tokens. Thus, Bob can simulate $\mathcal{P}$

to the end of the run.

Theorem 51. *Suppose p is either 0 or a constant greater than 1. For any $\varepsilon \leq 1/2$, a round-based randomized protocol for $(k, F_p, \tau, \varepsilon)$ functional monitoring must communicate $\Omega(1/\varepsilon^2)$ bits.*

Proof. We use the observations in the preceding paragraph, proceed as in the proofs of Theorems 49 and 50 above, and plug in the one-way communication lower bound from Theorem 47. □

Chapter 4

Gap Hamming Distance: The First Multi-Round Lower Bound

The Gap-Hamming-Distance problem arose in the context of proving space lower bounds for a number of key problems in the data stream model. In this problem, Alice and Bob have to decide whether the Hamming distance between their n-bit input strings is large (i.e., at least $n/2+\sqrt{n}$) or small (i.e., at most $n/2-\sqrt{n}$); they do not care if it is neither large nor small. This $\Theta(\sqrt{n})$ gap in the problem specification is crucial for capturing the approximation allowed to a data stream algorithm.

Thus far, for randomized communication, an $\Omega(n)$ lower bound on this problem was known only in the one-way setting [Woo04]. We prove an $\Omega(n)$ lower bound for randomized protocols that use any constant number of rounds.

As a consequence we conclude, for instance, that ε-approximately counting the number of distinct elements in a data stream requires $\Omega(1/\varepsilon^2)$ space, even with multiple (a constant number of) passes over the input stream. This extends earlier one-pass lower bounds, answering a long-standing open question. We obtain similar results for approximating the frequency moments and for approximating the empirical entropy of a data stream.

In the process, we also obtain tight $n - \Theta(\sqrt{n}\log n)$ lower and upper bounds on the one-way deterministic communication complexity of the problem. Finally, we give a simple combinatorial proof of an $\Omega(n)$ lower bound on the one-way randomized communication complexity.

4.1 Introduction

Our focus here is on the Gap-Hamming-Distance problem. To the best of our knowledge, this problem was first formally studied by Indyk and Woodruff [IW03] in FOCS 2003. They studied the problem in the context of proving space lower bounds for the Distinct Elements problem in the data stream model. We shall discuss their application shortly, but let us first define our communication problem precisely.

The Problem In the Gap-Hamming-Distance problem, Alice receives a Boolean string $x \in \{0,1\}^n$ and Bob receives $y \in \{0,1\}^n$. They wish to decide whether x and y are "close" or "far" in the Hamming sense. That is, they wish to output 0 if $\Delta(x,y) \leq n/2 - \sqrt{n}$ and 1 if $\Delta(x,y) \geq n/2 + \sqrt{n}$. They do not care about the output if neither of these conditions holds. Here, Δ denotes Hamming distance. In the sequel, we shall be interested in a parametrized version of the problem, where the thresholds are set at $n/2 \pm c\sqrt{n}$, for some parameter $c \in \mathbb{R}^+$.

Our Results While we prove a number of results about the Gap-Hamming-Distance problem here, there is a clear "main theorem" that we wish to highlight. Technical terms appearing below are defined precisely in Section 4.2.

Theorem 52 (Main Theorem, Informal). *Suppose a randomized $\frac{1}{3}$-error protocol solves the Gap-Hamming-Distance problem using k rounds of communication. Then, at least one message must be $n/2^{O(k^2)}$ bits long. In particular, any protocol using a constant number of rounds must communicate $\Omega(n)$ bits in some round. In fact, these bounds apply to deterministic protocols with low distributional error under the uniform distribution.*

At the heart of our proof is a round elimination lemma that lets us "eliminate" the first round of communication, in a protocol for the Gap-Hamming-Distance problem, and thus derive a shorter protocol for an "easier" instance of the same problem. By repeatedly applying this lemma, we eventually eliminate all of the communication. We also make the problem instances progressively easier, but, if the original protocol was short enough, at the end we are still left with a nontrivial problem. The resulting contradiction lower bounds the length of the original protocol. We note that this underlying "round elimination philosophy" is behind a number of key results in communication complexity [MNSW98, Sen03, CR04, ADHP06, Cha07, VW07, CJP08].

Besides the above theorem, we also prove tight lower *and upper* bounds of $n - \Theta(\sqrt{n} \log n)$ on the one-way deterministic communication complexity of Gap-Hamming-Distance. Only $\Omega(n)$ lower bounds were known before. We also prove an $\Omega(n)$ one-way randomized communication lower bound. This matches earlier results, but our proof has the advantage of being purely combinatorial. (We recently learned that Woodruff [Woo09] had independently discovered a similar combinatorial proof. We present our proof nevertheless, for pedagogical value, as it can be seen as a generalization of our deterministic lower bound proof.)

Motivation and Relation to Prior Work We now describe the original motivation for studying the Gap-Hamming-Distance problem. Later, we discuss the consequences of our Theorem 52. In the data stream model, one wishes to compute a real-valued function of a massively long input sequence (the data stream) using very limited space, hopefully sublinear in the input length. To get interesting results, one almost always needs to allow randomized approximate algorithms. A key problem in this model, that has seen much research [FM85, AMS99, BJK$^+$04, IW03, Woo09], is the Distinct Elements problem: the goal is to estimate the number of distinct elements in a stream of m elements (for simplicity, assume that the elements are drawn from the universe $[m] := \{1, 2, \ldots, m\}$).

An interesting solution to this problem would give a nontrivial tradeoff between the quality of approximation desired and the space required to achieve it. The best such result [BJK$^+$04] achieved a multiplicative $(1+\varepsilon)$-approximation using space $\widetilde{O}(1/\varepsilon^2)$, where the $\widetilde{O}$-notation suppresses $\log m$ and $\log(1/\varepsilon)$ factors. It also processed the input stream in a single pass, a very desirable property. Soon afterwards, Indyk and Woodruff [IW03] gave a matching $\Omega(1/\varepsilon^2)$ space lower bound for one-pass algorithms for this problem, by a reduction from the Gap-Hamming-Distance communication problem. In SODA 2004, Woodruff [Woo04] improved the bound, extending it to the full possible range of subconstant ε, and also applied it to the more general problem of estimating frequency moments $F_p := \sum_{i=1}^{n} f_i^p$, where f_i is the frequency of element i in the input stream. A number of other natural data stream problems have similar space lower bounds via reductions from Gap-Hamming, a more recent example being the computation of the empirical entropy of a stream [CCM07].

The idea behind the reduction is quite simple; as a concrete example, suppose there exists a streaming algorithm $\mathcal{A}$ for F_0. Alice and Bob can convert their Gap-Hamming inputs into suitable streams of integers. A protocol for Gap-Hamming-Distance is obtained in the following manner: Alice processes $\mathcal{A}$ on her stream. Then, she sends the *contents* of the memory to Bob in a single message, after which Bob processes $\mathcal{A}$ on his stream. Thus, Alice and Bob can estimate the number of distinct elements in the concatenation of their streams. By setting the approximation factor correctly, Alice and Bob convert an ε-approximate streaming algorithm for F_0 into a protocol for Gap-Hamming. In this way, an $\Omega(n)$ one-way communication lower bound for Gap-Hamming-Distance translates into an $\Omega(1/\varepsilon^2)$ one-pass space lower bound for F_0. Much less simple was the proof of the communication lower bound itself. Woodruff's proof [Woo04] required intricate combinatorial arguments and a fair amount of complex calculations. Jayram et al. [JKS08] later provided a rather different proof, based on a simple geometric argument, coupled with a clever reduction from the INDEX problem.[1] A version

[1] In the INDEX problem, Alice has an n-bit string x and sends a single message to Bob, who sees $i \in [n]$

of this proof is given in Woodruff's Ph.D. thesis [Woo07]. In Section 4.5, we provide a still simpler direct combinatorial proof, essentially from first principles.

All of this left open the tantalizing possibility that a second pass over the input stream could drastically reduce the space required to approximate the number of distinct elements — or, more generally, the frequency moments F_p. Perhaps $\widetilde{O}(1/\varepsilon)$ space was possible? This was a long-standing open problem [Kum06] in data streams. Yet, some thinking about the underlying Gap-Hamming communication problem suggested that the linear lower bound ought to hold for general communication protocols, not just for one-way communication. This prompted the following natural conjecture.

Conjecture 53. *A $\frac{1}{3}$-error randomized communication protocol for the Gap-Hamming-Distance problem must communicate $\Omega(n)$ bits in total, irrespective of the number of rounds of communication.*

An immediate consequence of the above conjecture is that a second pass does *not* help beat the $\Omega(1/\varepsilon^2)$ space lower bound for the aforementioned streaming problems; in fact, no constant number of passes helps. Our Theorem 52 does *not* resolve Conjecture 53. However, it *does* imply the $\Omega(1/\varepsilon^2)$ space lower bound with a constant number of passes. This is because we *do* obtain a linear communication lower bound with a constant number of rounds.

Remark. *Subsequent to the work in this chapter and the improved lower bound of Chapter 5, Chakrabarti and Regev [CR10] answered Conjecture 53 in the affirmative. The techniques they employ are completely different than ours, and rather involved. We include these results for completeness.*

---

and must output x_i. Ablayev [Abl96] gave an $\Omega(n)$ lower bound for randomized protocols that compute INDEX. Note that this problem is inherently one-way—if the communication is from Bob to Alice, then a trivial $O(\log n)$ protocol exists.

Finer Points To better understand our contribution here, it is worth considering some finer points of previously known lower bounds on Gap-Hamming-Distance, including some "folklore" results. The earlier one-way $\Omega(n)$ bounds were *inherently* one-way, because the INDEX problem has a trivial two-round protocol. Also, the nature of the reduction implied a distributional error lower bound for Gap-Hamming only under a somewhat artificial input distribution. Our bounds here, including our one-way randomized bound, overcome this problem, as does the recent one-way bound of Woodruff [Woo09]: they apply to the uniform distribution. As noted by Woodruff [Woo09], this has the desirable consequence of implying space lower bounds for the Distinct Elements problem under weaker assumptions about the input stream: it could be random, rather than adversarial.

Intuitively, the uniform distribution is the hard case for the Gap-Hamming problem. The Hamming distance between two uniformly distributed n-bit strings is likely to be just around the $n/2 \pm \Theta(\sqrt{n})$ thresholds, which means that a protocol will have to work hard to determine which threshold the input is at. Indeed, this line of thinking suggests an $\Omega(n)$ lower bound for distributional complexity — under the uniform distribution — on the *gapless* version of the problem. Our proofs here confirm this intuition, at least for a constant number of rounds.

It is relatively easy to obtain an $\Omega(n)$ lower bound on the *deterministic* multi-round communication complexity of the problem. One can directly demonstrate that the communication matrix contains no large monochromatic rectangles (see, e.g. [Woo07]). Indeed, the argument goes through even with gaps of the form $n/2 \pm \Theta(n)$, rather than $n/2 \pm \Theta(\sqrt{n})$. It is also easy to obtain an $\Omega(n)$ bound on the randomized complexity of the gapless problem, via a reduction from DISJOINTNESS. Unfortunately, the known hard distributions for DISJOINTNESS are far from uniform, and DISJOINTNESS is actually very easy under a uniform input distribution. So, this reduction does not give us the results we want. Incidentally, an even easier reduction from DISJOINTNESS yields an arbitrary-round $\Omega(\sqrt{n})$ lower bound for Gap-Hamming-Distance; this result is folklore.

Furthermore, straightforward rectangle-based methods (discrepancy/corruption) fail to effectively lower bound the randomized communication complexity of our problem. This is because there *do* exist very large near-monochromatic rectangles in its communication matrix. This can be seen, e.g., by considering all inputs (x, y) with $x_i = y_i = 0$ for $i \in [O(\sqrt{n})]$.

Connection to Decision Trees and Quantum Communication We would like to bring up two other illuminating observations. Consider the following query complexity problem: the input is a string $x \in \{0,1\}^n$ and the desired output is 1 if $|x| \geq n/2 + \sqrt{n}$ and 0 if $|x| \leq n/2 - \sqrt{n}$. Here, $|x|$ denotes the Hamming weight of x. The model is a randomized decision tree whose nodes query individual bits of x, and whose leaves give outputs in $\{0,1\}$. It is not hard to show that $\Omega(n)$ queries are needed to solve this problem with $\frac{1}{3}$ error. Essentially, one can do no better than sampling bits of x at random, and then $\Omega(1/\varepsilon^2)$ samples are necessary to distinguish a biased coin that shows heads with probability $\frac{1}{2} + \varepsilon$ from one that shows heads with probability $\frac{1}{2} - \varepsilon$.

The Gap-Hamming-Distance problem can be seen as a generalization of this problem to the communication setting. Certainly, any efficient decision tree for the query problem implies a correspondingly efficient communication protocol, with Alice acting as the querier and Bob acting as the responder (say). Conjecture 53 says that no better communication protocols are possible for this problem.

This query complexity connection brings up another crucial point. The *quantum* query complexity of the above problem can be shown to be $O(\sqrt{n})$, by the results of Nayak and Wu [NW99]. This in turn implies an $O(\sqrt{n}\log n)$ quantum communication protocol for Gap-Hamming, essentially by carefully "implementing" the quantum query algorithm, as in Razborov [Raz02]. Therefore, any technique that seeks to prove an $\Omega(n)$ lower bound for Gap-Hamming (under classical communication) must necessarily fail for quantum protocols. This rules out several recently-developed methods, such as the factorization norms method of Linial

and Shraibman [LS07] and the pattern matrix method of Sherstov [She08].

4.2 Basic Definitions, Notation and Preliminaries

We begin with definitions of our central problem of interest, and quickly recall some standard definitions from communication complexity. Along the way, we also introduce some notation that we use in the rest of the paper.

Definition 4. *For strings $x, y \in \{0,1\}^n$, the Hamming distance between x and y, denoted $\Delta(x,y)$, is defined as the number of coordinates $i \in [n]$ such that $x_i \neq y_i$.*

Definition 5 (Gap-Hamming-Distance problem). *Suppose $n \in \mathbb{N}$ and $c \in \mathbb{R}^+$. The c-Gap-Hamming-Distance partial function, on n-bit inputs, is denoted $\mathrm{GHD}_{c,n}$ and is defined as follows.*

$$\mathrm{GHD}_{c,n}(x,y) \;=\; \begin{cases} 1\,, & \text{if } \Delta(x,y) \geq n/2 + c\sqrt{n}\,, \\ 0\,, & \text{if } \Delta(x,y) \leq n/2 - c\sqrt{n}\,, \\ \star\,, & \text{otherwise.} \end{cases}$$

We also use $\mathrm{GHD}_{c,n}$ to denote the corresponding communication problem where Alice holds $x \in \{0,1\}^n$, Bob holds $y \in \{0,1\}^n$, and the goal is for them to communicate and agree on an output bit that matches $\mathrm{GHD}_{c,n}(x,y)$. By convention, $\star$ matches both 0 and 1.

Protocols Consider a communication problem $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1,\star\}^n$ and a protocol $\mathcal{P}$ that attempts to solve f. We write $\mathcal{P}(x,y)$ to denote the output of $\mathcal{P}$ on input (x,y): note that this may be a random variable, dependent on the internal coin tosses of $\mathcal{P}$, if $\mathcal{P}$ is a randomized protocol. A deterministic protocol $\mathcal{P}$ is said to be correct for f if $\forall\,(x,y) : \mathcal{P}(x,y) = f(x,y)$ (the "$=$" is to be read as "matches"). It is said to have *distributional error*

ε under an input distribution ρ if $\Pr_{(x,y)\sim\rho}[\mathcal{P}(x,y) \neq f(x,y)] \leq \varepsilon$. A *randomized protocol* $\mathcal{P}$, using a public random string r, is said to be have error ε if $\forall\,(x,y) : \Pr_r[\mathcal{P}(x,y) \neq f(x,y)] \leq \varepsilon$. A protocol $\mathcal{P}$ is said to be a *k-round protocol* if it involves exactly k messages, with Alice and Bob taking turns to send the messages; by convention, we usually assume that Alice sends the first message and the recipient of the last message announces the output. A 1-round protocol is also called a *one-way protocol*, since the entire communication happens in the Alice $\rightarrow$ Bob direction.

Communication Complexity The deterministic communication complexity $D(f)$ of a communication problem f is defined to be the minimum, over deterministic protocols $\mathcal{P}$ for f, of the number of bits exchanged by $\mathcal{P}$ for a worst-case input (x,y). By suitably varying the class of protocols over which the minimum is taken, we obtain, e.g., the ε-error randomized, one-way deterministic, ε-error one-way randomized, and ε-error ρ-distributional deterministic communication complexities of f, denoted $R_\varepsilon(f)$, $D^{\rightarrow}(f)$, $R^{\rightarrow}_\varepsilon(f)$, and $D_{\rho,\varepsilon}(f)$, respectively. When the error parameter ε is dropped, it is tacitly assumed to be $\frac{1}{3}$; as is well-known, the precise value of this constant is immaterial for asymptotic bounds.

Definition 6 (Near-Orthogonality). *We say that strings $x, y \in \{0,1\}^n$ are c-near-orthogonal, and write $x \perp_c y$, if $|\Delta(x,y) - n/2| < c\sqrt{n}$. Here, c is a positive real quantity, possibly dependent on n. Notice that $\mathrm{GHD}_{c,n}(x,y) = \star \Leftrightarrow x \perp_c y$.*

The distribution of the Hamming distance between two uniform random n-bit strings — equivalently, the distribution of the Hamming weight of a uniform random n-bit string — is just an unbiased binomial distribution $\mathrm{Binom}(n, \frac{1}{2})$. We shall use the following (fairly loose) bounds on the tail of this distribution (see, e.g., Feller [Fel68]).

Fact 54. *Let $T_n(c) = \Pr_x[x \not\perp_c 0^n]$, where x is distributed uniformly at random in $\{0,1\}^n$. Let*

$T(c) = \lim_{n\to\infty} T_n(c)$. *Then*

$$2^{-3c^2-2} \;\leq\; T(c) \;\approx\; \frac{e^{-2c^2}}{c\sqrt{2\pi}} \;\leq\; 2^{-c^2}\,.$$

There are two very natural input distributions for $\text{GHD}_{c,n}$: the uniform distribution on $\{0,1\}^n \times \{0,1\}^n$, and the (non-product) distribution that is uniform over all inputs for which the output is precisely defined. We call this latter distribution $\mu_{c,n}$.

Definition 7 (Distributions). *For $n \in \mathbb{N}$, $c \in \mathbb{R}^+$, let $\mu_{c,n}$ denote the uniform distribution on the set $\{(x,y) \in \{0,1\}^n \times \{0,1\}^n : x \not\perp_c y\}$. Also, let $\mathcal{U}_n$ denote the uniform distribution on $\{0,1\}^n$.*

Using Fact 54, we can show that for a constant c and suitably small ε, the distributional complexities $D_{\mathcal{U}_n \times \mathcal{U}_n, \varepsilon}(\text{GHD}_{c,n})$ and $D_{\mu_{c,n},\varepsilon}(\text{GHD}_{c,n})$ are within constant factors of each other. This lets us work with the latter and draw conclusions about the former. The latter has the advantage that it is meaningful for any $\varepsilon < \frac{1}{2}$, whereas the former is only meaningful if $\varepsilon < \frac{1}{2}T(c)$.

Let $\mathcal{B}(x,r)$ denote the Hamming ball of radius r centered at x. We need to use the following bounds on the volume (i.e., size) of a Hamming ball. Here, $H : [0,1] \to [0,1]$ is the binary entropy function.

Fact 55. *If $r = c\sqrt{n}$, then $(\sqrt{n}/c)^r < |\mathcal{B}(x,r)| < n^r$.*

Fact 56. *If $r = \alpha n$ for some constant $0 < \alpha < 1$, then $|\mathcal{B}(x,r)| \leq 2^{nH(\alpha)}$.*

4.3 Main Theorem: Multi-Round Lower Bound

4.3.1 Some Basics

In order to prove our multi-round lower bound, we need a simple — yet, powerful — combinatorial lemma, known as Sauer's Lemma [Sau72]. For this, we recall the concept of Vapnik-Chervonenkis dimension. Let $S \subseteq \{0,1\}^n$ and $I \subseteq [n]$. We say that S shatters I if the set obtained by restricting the vectors in S to the coordinates in I has the maximum possible size, $2^{|I|}$. We define $\text{VC-dim}(S)$ to be the maximum $|I|$ such that S shatters I.

Lemma 57 (Sauer's Lemma). *Suppose $S \subseteq \{0,1\}^n$ has $\text{VC-dim}(S) < d$. Then*

$$|S| \;\leq\; \sum_{i=0}^{d} \binom{n}{i}.$$

When $d = \alpha n$ for some constant α, then the above sum can be upper bounded by $2^{nH(\alpha)}$. This yields the following corollary.

Corollary 58. *If $|S| \geq 2^{nH(\alpha)}$, for a constant α, then $\text{VC-dim}(S) \geq \alpha n$.*

We now turn to the proof proper. It is based on a round elimination lemma that serves to eliminate the first round of communication of a GHD protocol, yielding a shorter protocol, but for GHD instances with weakened parameters. To keep track of all relevant parameters, we introduce the following notation.

Definition 8. *A $[k, n, s, c, \varepsilon]$-protocol is a deterministic k-round protocol for $\text{GHD}_{c,n}$ that errs on at most an ε fraction of inputs, under the input distribution $\mu_{c,n}$, and in which each message is s bits long.*

The next lemma gives us the "end point" of our round elimination argument.

Lemma 59. *There exists no $[0, n, s, c, \varepsilon]$-protocol with $n > 1$, $c = o(\sqrt{n})$, and $\varepsilon < \frac{1}{2}$.*

Proof. With these parameters, $\mu_{c,n}$ has nonempty support. This implies $\Pr_{\mu_{c,n}}[\text{GHD}_{c,n}(x,y) = 0] = \Pr_{\mu_{c,n}}[\text{GHD}_{c,n}(x,y) = 1] = \frac{1}{2}$. Thus, a 0-round deterministic protocol, which must have constant output, cannot achieve error less than $\frac{1}{2}$. □

4.3.2 The Round Elimination Lemma

The next lemma is the heart of our proof. To set up its parameters, we set $t_0 = (48 \ln 2) \cdot 2^{11k}$, $t = 2^{15k}$, and $b = T^{-1}(1/8)$, and we define a sequence $\langle (n_i, s_i, c_i, \varepsilon_i) \rangle_{i=0}^{k}$ as follows:

$$\left.\begin{aligned} n_0 &= n\,, & n_{i+1} &= n_i/3\,, \\ s_0 &= t_0 s\,, & s_{i+1} &= t s_i\,, \\ c_0 &= 10\,, & c_{i+1} &= 2c_i\,, \\ \varepsilon_0 &= 2^{-2^{11k}}\,, & \varepsilon_{i+1} &= \varepsilon_i / T(c_{i+1})\,. \end{aligned}\right\} \text{ for } i > 0\,. \tag{4.1}$$

Lemma 60 (Round Elimination for GHD). *Suppose $0 \le i < k$ and $s_i \le n_i/20$. Suppose there exists a $[k-i, n_i, s_i, c_i, \varepsilon_i]$-protocol. Then there exists a $[k-i-1, n_{i+1}, s_{i+1}, c_{i+1}, \varepsilon_{i+1}]$-protocol.*

Proof. Let $(n, s, c, \varepsilon) = (n_i, s_i, c_i, \varepsilon_i)$ and $(n', s', c', \varepsilon') = (n_{i+1}, s_{i+1}, c_{i+1}, \varepsilon_{i+1})$. Also, let $\mu = \mu_{c,n}$, $\mu' = \mu_{c',n'}$, $\text{GHD} = \text{GHD}_{c,n}$ and $\text{GHD}' = \text{GHD}_{c',n'}$. Let $\mathcal{P}$ be a $[k-i, n, s, c, \varepsilon]$-protocol. Assume, WLOG, that Alice sends the first message in $\mathcal{P}$.

Call a string $x_0 \in \{0,1\}^n$ "good" if

$$\Pr_{(x,y)\sim\mu}[\mathcal{P}(x,y) \ne \text{GHD}(x,y) \mid x = x_0] \le 2\varepsilon\,. \tag{4.2}$$

By the error guarantee of $\mathcal{P}$ and Markov's inequality, the number of good strings is at least 2^{n-1}. There are $2^s \le 2^{n/20}$ different choices for Alice's first message. Therefore, there is a set $M \subseteq \{0,1\}^n$ of good strings such that Alice sends the same first message $\mathfrak{m}$ on every input

$x \in M$, with $|M| \geq 2^{n-1-n/20} \geq 2^{nH(1/3)}$. By Corollary 58, $\text{VC-dim}(M) \geq n/3$. Therefore, there exists a set $I \subseteq [n]$, with $|I| = n/3 = n'$, that is shattered by M. For strings $x' \in \{0,1\}^{n'}$ and $x'' \in \{0,1\}^{n-n'}$, we write $x' \circ x''$ to denote the string in $\{0,1\}^n$ formed by plugging in the bits of x' and x'' (in order) into the coordinates in I and $[n] \setminus I$, respectively.

We now give a suitable $(k-i-1)$-round protocol $\mathcal{Q}$ for GHD′, in which Bob sends the first message. Consider an input $(x', y') \in \{0,1\}^{n'} \times \{0,1\}^{n'}$, with Alice holding x' and Bob holding y'. By definition of shattering, there exists an $x'' \in \{0,1\}^{n-n'}$ such that $x := x' \circ x'' \in M$. Alice and Bob agree beforehand on a suitable x for each possible x'. Suppose Bob were to pick a uniform random $y'' \in \{0,1\}^{n-n'}$ and form the string $y := y' \circ y''$. Then, Alice and Bob could simulate $\mathcal{P}$ on input (x, y) using only $k - i - 1$ rounds of communication, with Bob starting, because Alice's first message in $\mathcal{P}$ would always be $\mathfrak{m}$. Call this randomized protocol $\mathcal{Q}_1$. We define $\mathcal{Q}$ to be the protocol obtained by running t instances of $\mathcal{Q}_1$ in parallel, using independent random choices of y'', and outputting the majority answer. Note that the length of each message in Q is $ts = s'$. We shall now analyze the error.

Suppose $x'' \perp_b y''$. Let $d_1 = \Delta(x, y) - n/2$, $d_2 = \Delta(x', y') - n'/2$ and $d_3 = \Delta(x'', y'') - (n - n')/2$. Note that $d_1 = d_2 + d_3$. Also,

$$\begin{aligned} |d_1| &\geq |d_2| - |d_3| \\ &\geq c'\sqrt{n'} - b\sqrt{n - n'} \\ &\geq \frac{(c' - b\sqrt{2})\sqrt{n}}{\sqrt{3}} \\ &\geq c\sqrt{n}\,, \end{aligned}$$

where we used (4.1) and our choice of b. Thus, $x \not\perp_c y$. The same calculation also shows that d_1 and d_2 have the same sign, as $|d_2| > |d_3|$. Therefore $\text{GHD}(x, y) = \text{GHD}'(x', y')$.

For the rest of the calculations in this proof, fix an input x' for Alice, and hence, x'' and x as well. For a fixed y', let $\mathcal{E}(y')$ denote the event that $\mathcal{P}(x, y) \neq \text{GHD}(x, y)$: note that y'' remains

random. Using the above observation (at step (4.4) below), we can bound the probability that $\mathcal{Q}_1$ errs on input (x', y') as follows.

$$\begin{aligned}
\Pr_y\left[Q_1(x', y') \neq \textsc{ghd}'(x', y') \mid y'\right] \;\leq\; & \qquad (4.3)\\
& \Pr_y[\mathcal{P}(x, y) \neq \textsc{ghd}(x, y) \vee \textsc{ghd}(x, y) \neq \textsc{ghd}'(x', y') \mid y']\\
& \leq \Pr_{y''}\left[\mathcal{E}(y')\right] + \Pr_y\left[\textsc{ghd}(x, y) \neq \textsc{ghd}'(x', y') \mid y'\right]\\
& \leq \Pr_{y''}\left[\mathcal{E}(y')\right] + \Pr_{y''}\left[x'' \not\perp_b y''\right] \qquad (4.4)\\
& \leq \Pr_{y''}\left[\mathcal{E}(y')\right] + T(b)\\
& = \Pr_{y''}\left[\mathcal{E}(y')\right] + 1/8\,, \qquad (4.5)
\end{aligned}$$

where step (4.5) follows from our choice of b. To analyze $\mathcal{Q}$, notice that during the t-fold parallel repetition of $\mathcal{Q}_1$, y' remains fixed while y'' varies. Thus, it suffices to understand how the repetition drives down the sum on the right side of (4.5). Unfortunately, for some values of y', the sum may exceed $\frac{1}{2}$, in which case it will be driven *up*, not down, by the repetition. To account for this, we shall bound the *expectation* of the first term of that sum, for a random y'.

To do so, let $z \sim \mu \mid x$ be a random string independent of y. Notice that z is uniformly distributed on a subset of $\{0,1\}^n$ of size $2^n T(c)$, whereas y is uniformly distributed on a subset of $\{0,1\}^n$ of size $2^n T(c')$. (We are now thinking of x as being fixed and both y' and y'' as being random.) Therefore,

$$\begin{aligned}
\mathbb{E}_{y'}\left[\Pr_{y''}\left[\mathcal{E}(y')\right]\right] \;=\; & \Pr_y\left[\mathcal{E}(y')\right] \qquad (4.6)\\
& = \Pr_y\left[\mathcal{P}(x, y) \neq \textsc{ghd}(x, y)\right]\\
& \leq \Pr_z\left[\mathcal{P}(x, z) \neq \textsc{ghd}(x, z)\right] \cdot \frac{T(c)}{T(c')}\\
& \leq 2\varepsilon T(c)/T(c')\,, \qquad (4.7)
\end{aligned}$$

where (4.7) holds because x, being good, satisfies (4.2). Thus, by Markov's inequality,

$$\Pr_{y'}\left[\Pr_{y''}\left[\mathcal{E}(y')\right] \geq \frac{1}{8}\right] \;\leq\; 16\varepsilon T(c)/T(c')\,. \tag{4.8}$$

If, for a particular y', the *bad event* $\Pr_{y''}[\mathcal{E}(y')] \geq \frac{1}{8}$ does *not* occur, then the right side of (4.5) is at most $1/8 + 1/8 = 1/4$. In other words, $\mathcal{Q}_1$ errs with probability at most $1/4$ for this y'. By standard Chernoff bounds, the t-fold repetition in $\mathcal{Q}$ drives this error down to $(e/4)^{t/4} \leq 2^{-t/10} \leq \varepsilon_0 \leq \varepsilon$. Combining this with (4.8), which bounds the probability of the bad event, we get

$$\begin{aligned}\Pr_{y',r}\left[\mathcal{Q}(x',y') \neq \textsc{ghd}'(x',y')\right] \;&\leq\; 16\varepsilon T(c)/T(c') + \varepsilon \\ &\leq\; \varepsilon/T(c') \\ &=\; \varepsilon'\,,\end{aligned}$$

where r denotes the internal random string of $\mathcal{Q}$ (i.e., the collection of y''s used).

Note that this error bound holds for *every* fixed x', and thus, when $(x',y') \sim \mu'$. Therefore, we can fix Bob's random coin tosses in $\mathcal{Q}$ to get the desired $[k-i-1, n', s', c', \varepsilon']$-protocol. □

4.3.3 The Lower Bound

Having established our round elimination lemma, we obtain our lower bound in a straightforward fashion.

Theorem 61 (Multi-round Lower Bound). *Let $\mathcal{P}$ be a k-round $\frac{1}{3}$-error randomized communication protocol for $\textsc{ghd}_{c,n}$, with $c = O(1)$, in which each message is s bits long. Then*

$$s \;\geq\; \frac{n}{2^{O(k^2)}}\,.$$

Remark. *This is a formal restatement of Theorem 52.*

Proof. For simplicity, assume $c \leq c_0 = 10$. Our proof easily applies to a general $c = O(1)$ by a suitable modification of the parameters in (4.1). Also, assume $n \geq 2^{4k^2}$, for otherwise there is nothing to prove.

By repeating $\mathcal{P}$ $(48 \ln 2) \cdot 2^{11k} = t_0$ times, in parallel, and outputting the majority of the answers, we can reduce the error to $2^{-2^{11k}} = \varepsilon_0$. The size of each message is now $t_0 s = s_0$. Fixing the random coins of the resulting protocol gives us a $[k, n_0, s_0, c_0, \varepsilon_0]$-protocol $\mathcal{P}_0$.

Suppose $s_i \leq n_i/20$ for all i, with $0 \leq i < k$. We then repeatedly apply Lemma 60 k times, starting with $\mathcal{P}_0$. Eventually, we end up with a $[0, n_k, s_k, c_k, \varepsilon_k]$-protocol. Examining (4.1), we see that $n_k = n/3^k$, $s_k = 2^{15k^2} s_0 = (48 \ln 2) 2^{15k^2+11k} s$, and $c_k = 10 \cdot 2^k$. Notice that $n_k \geq 2^{4k^2}/3^k > 1$ and $c_k = o(\sqrt{n_k})$. We also see that $\langle c_i \rangle_{i=1}^k$ is an increasing sequence, whence $\varepsilon_{i+1}/\varepsilon_i = 1/T(c_{i+1}) \leq 1/T(c_k) \leq 2^{3{c_k}^2+2}$, where the final step uses Fact 54. Thus,

$$\begin{aligned}
\varepsilon_k &\leq \varepsilon_0 \left(2^{3c_k^2+2}\right)^k \\
&= 2^{-2^{11k}} \cdot 2^{(3(10 \cdot 2^k)^2+2) \cdot k} \\
&= 2^{-2^{11k}+300k \cdot 2^{2k}+2k} \\
&< \frac{1}{2}.
\end{aligned}$$

In other words, we have a $[0, n_k, s_k, c_k, \varepsilon_k]$-protocol with $n_k > 1$, $c_k = o(\sqrt{n_k})$ and $\varepsilon_k < \frac{1}{2}$. This contradicts Lemma 59.

Therefore, there must exist an i such that $s_i \geq n_i/20$. Since $\langle s_i \rangle_{i=1}^k$ is increasing and $\langle n_i \rangle_{i=1}^k$ is decreasing, $s_k \geq n_k/20$. By the above calculations, $(48 \ln 2) 2^{15k^2+11k} s \geq n/(20 \cdot 3^k)$, which implies $s \geq n/2^{O(k^2)}$, as claimed. □

Notice that, for constant k, the argument in the above proof in fact implies a lower bound for deterministic protocols with small enough constant distributional error under $\mu_{c,n}$. This, in

turn, extends to distributional error under the uniform distribution, as remarked earlier.

4.4 Tight Deterministic One-Way Bounds

The main result of this section is the following.

Theorem 62. $D^{\rightarrow}(\text{GHD}_{c,n}) = n - \Theta(\sqrt{n}\log n)$ *for all constant* c.

Definition 9. *Let* $x_1, x_2, y \in \{0,1\}^n$. *We say that* y witnesses x_1 *and* x_2 *or that* y *is a witness for* (x_1, x_2) *if* $x_1 \not\perp_c y$, $x_2 \not\perp_c y$, *and* $\text{GHD}_{c,n}(x_1, y) \neq \text{GHD}_{c,n}(x_2, y)$.

Intuitively, if (x_1, x_2) have a witness, then they cannot be in the same message set. For if Alice sent the same message on x_1 and x_2 and Bob's input y was a witness for (x_1, x_2) then whatever Bob were to output, the protocol would err on either (x_1, y) or (x_2, y). The next lemma characterizes which (x_1, x_2) pairs have witnesses.

Lemma 63. *For all* $x_1, x_2 \in \{0,1\}^n$, *there exists* y *that witnesses* (x_1, x_2) *if and only if* $\Delta(x_1, x_2) \geq 2c\sqrt{n}$.

Proof. On the one hand, suppose y witnesses (x_1, x_2). Then assume WLOG that $\Delta(x_1, y) \leq n/2 - c\sqrt{n}$ and $\Delta(x_2, y) \geq n/2 + c\sqrt{n}$. By the triangle inequality, $\Delta(x_1, x_2) \geq \Delta(x_2, y) - \Delta(x_1, y) = 2c\sqrt{n}$. Conversely, suppose $\Delta(x_1, x_2) \geq 2c\sqrt{n}$. Let $L = \{i : x_1[i] = x_2[i]\}$, and let $R = \{i : x_1[i] \neq x_2[i]\}$. Suppose y agrees with x_1 on all coordinates from R and half the coordinates from L. Then, $\Delta(x_1, y) = |L|/2 = (n - \Delta(x_1, x_2))/2 \leq n/2 - c\sqrt{n}$. Furthermore, y agrees with x_2 on *no* coordinates from R and half the coordinates from L, so $\Delta(x_1, y) = |L|/2 + |R| \geq n/2 + c\sqrt{n}$. □

We show that it is both necessary and sufficient for Alice to send different messages on x_1 and x_2 whenever $\Delta(x_1, x_2)$ is "large". To prove this, we need the following theorem, due to Bezrukov [Bez87] and a simple claim that is proved using the probabilistic method.

Theorem 64. *Call a subset $A \subseteq \{0,1\}^n$ d-maximal if it is largest, subject to the constraint that $\Delta(x,y) \leq d$ for all $x, y \in A$.*

1. *If $d = 2t$ then $\mathcal{B}(x,t)$ is d-maximal for any $x \in \{0,1\}^n$.*
2. *If $d = 2t+1$ then $\mathcal{B}(x,t) \cup \mathcal{B}(y,t)$ is d-maximal for any $x, y \in \{0,1\}^n$ such that $\Delta(x,y) = 1$.* □

Claim 65. *It is possible to cover $\{0,1\}^n$ with at most $2^{n-O(\sqrt{n}\log n)}$ Hamming balls, each of radius $c\sqrt{n}$.* □

Proof. We use the probabilistic method. Let $r := c\sqrt{n}$. For $x \in \{0,1\}^n$, let $\mathcal{B}_x := \mathcal{B}(x,r)$ be the Hamming ball of radius r centered at x. For a t to be determined later, pick $x_1, \ldots, x_t$ independently and uniformly at random from $\{0,1\}^n$. We want to show that with nonzero probability, the universe $\{0,1\}^n$ is covered by these t Hamming balls $\mathcal{B}_{x_1}, \ldots, \mathcal{B}_{x_t}$.

Now, fix any $x \in \{0,1\}^n$ and any $1 \leq i \leq t$. Since x_i was picked uniformly at random, each x is equally likely to be in $\mathcal{B}_{x_i}$. Therefore,

$$\Pr[x \in \mathcal{B}_{x_i}] = \frac{|\mathcal{B}_{x_i}|}{2^n} \geq 2^{\theta(\sqrt{n}\log n)-n}$$

where inequality stems from Fact 55.

Let $BAD_x = \bigwedge_{1 \leq i \leq t} x \notin \mathcal{B}_{x_i}$ be the event that x is not covered by any of the Hamming balls we picked at random, and let $BAD = \bigvee BAD_x$ be the event that *some* x is not covered by the Hamming balls. We want to limit $\Pr[BAD]$. BAD_x occurs when $x \notin \mathcal{B}_{x_i}$ for all x_i. Therefore, using $1 - x \leq e^{-x}$ for all real x,

$$\Pr[BAD_x] = \left(1 - 2^{\theta(\sqrt{n}\log n)-n}\right)^t \leq \mathrm{e}^{-t \cdot 2^{\theta(\sqrt{n}\log n)-n}}.$$

By the union bound,

$$\Pr[BAD] \leq 2^n \Pr[BAD_x] = 2^{n - \frac{t}{\ln 2} 2^{\theta(n\sqrt{n})-n}}.$$

Picking $t = \ln 2(n+1)2^{n-\theta(\sqrt{n}\log n)} = 2^{n-\theta(\sqrt{n}\log n)}$ ensures that $\Pr[BAD] < 1$. Therefore, there exists a set of $t = 2^{n-\theta(\sqrt{n}\log n)}$ Hamming balls of radius $c\sqrt{n}$ that cover $\{0,1\}^n$. □

Proof of Theorem 62. For the lower bound, suppose for the sake of contradiction that there is a protocol where Alice sends only $n - c\sqrt{n}\log n$ bits. By the pigeonhole principle, there exists a set $M \subseteq \{0,1\}^n$ of inputs of size $|M| \geq 2^n / 2^{n-c\sqrt{n}\log n} = 2^{c\sqrt{n}\log n} = n^{c\sqrt{n}}$ upon which Alice sends the same message. By Theorem 64, the Hamming ball $\mathcal{B}(x, c\sqrt{n})$ is $2c\sqrt{n}$-maximal, and by Fact 55, $|\mathcal{B}(x, c\sqrt{n})| < |M|$. Therefore, there must be $x_1, x_2 \in M$ with $\Delta(x_1, x_2) > 2c\sqrt{n}$. By Lemma 63, there exists a y that witnesses (x_1, x_2). No matter what Bob outputs, the protocol errs on either (x_1, y) or on (x_2, y).

For a matching upper bound, Alice and Bob fix a covering $\mathcal{C} = \{\mathcal{B}(x_0, r)\}$ of $\{0,1\}^n$ by Hamming balls of radius $r = c\sqrt{n}$. On input x, Alice sends Bob the Hamming ball $\mathcal{B}(x_0, r)$ containing x. Bob selects some $x' \in \mathcal{B}(x_0, r)$ such that $x' \not\perp_c y$ and outputs $\text{GHD}(x', y)$. The correctness of this protocol follows from Lemma 63, as $\Delta(x, x') \leq 2c\sqrt{n}$ since they are both in $\mathcal{B}(x_0, c\sqrt{n})$. The cost of the protocol is given by Claim 65, which shows that it suffices for Alice to send $\log\left(2^{n-O(\sqrt{n}\log n)}\right) = n - O(\sqrt{n}\log n)$ bits to describe each Hamming ball. □

4.5 One Round Randomized Lower Bound

Next, we develop a one-way lower bound for randomized protocols. Note that our lower bound applies to the uniform distribution, which, as mentioned in Section 4.1, implies space lower bounds for the Distinct Elements problem under weaker assumptions about the input stream. Woodruff [Woo09] recently proved similar results, also for the uniform distribution. We in-

clude our lower bound as a natural extension of the deterministic bound.

Theorem 66. $R^{\rightarrow}_{\varepsilon}(\text{GHD}_{c,n}) = \Omega(n)$.

Proof. For the sake of clarity, fix $c = 2$ and $\varepsilon = 1/10$, and suppose $\mathcal{P}$ is a one-round, ε-error, $o(n)$-bit protocol for $\text{GHD}_{c,n}$.

Definition 10. *For $x \in \{0,1\}^n$, let $Y_x := \{y : x \not\perp_2 y\}$. Say that x is* good *if $\Pr_{y \in Y_x}[\mathcal{P}(x,y) = \text{GHD}(x,y)] \leq 2\varepsilon$. Otherwise, call x* bad.

By Markov's inequality, at most a $1/2$-fraction of x are *bad*. Next, fix Alice's message $\mathfrak{m}$ to maximize the number of *good* x, and let $M = \{\text{good } x \in \{0,1\}^n : \text{ Alice sends } \mathfrak{m} \text{ on input } x\}$. It follows that

$$|M| \geq 2^{n-1}/2^{o(n)} > 2^{n(1-o(1))}.$$

Our goal is to show that since $|M|$ is large, we must err on a $> 2\varepsilon$-fraction of $y \in Y_x$ for some $x \in M$, contradicting the goodness of x. Note that it suffices to show that a 4ε fraction of $y \in Y_{x_1}$ witness x_1 and x_2.

$|M| \geq 2^{n(1-o(1))}$, so by Fact 56 and Theorem 64, There exist x_1, x_2 with $\Delta(x_1, x_2) \geq 1 - o(1)$. Next, we'd like to determine the probability that a random $y \in Y_{x_1}$ witnesses (x_1, x_2). Without loss of generality, let $x_1 = 0^n$. Let $w(x) := \Pr_{y \in Y_{x_1}}[\text{GHD}(x,y) \neq \text{GHD}(x_1,y)]$. The following lemma shows that $w(x)$ is an increasing function of $|x|$.

Lemma 67. *For all $x, x' \in \{0,1\}^n$, $w(x) \geq w(x') \Leftrightarrow |x| \geq |x'|$, with equality if and only if $|x| = |x'|$.*

Proof. If $|x| = |x'|$, then $w(x) = w(x')$ by symmetry. Further, note that $\text{GHD}(x,y) = 0$ if and only if $\text{GHD}(-x,y) = 1$. Therefore, it suffices to handle the case where $|y| \leq n/2 - c\sqrt{n}$ and $\text{GHD}(\vec{0},y) = 0$.

For the rest of the proof, we assume that $x_i = x'_i$, except for the nth coordinate, where $x_n = 0$ and $x'_n = 1$. Thus, $|x| = |x'| - 1$. We show that $w(x) < w(x')$; the rest of the lemma follows by induction.

Let Y be the set of strings with Hamming weight $|y| \leq n/2 - c\sqrt{n}$. Partition Y into the following three sets:

- $A := \{y : |y| = n/2 + c\sqrt{n} \wedge y_n = 0\}$.
- $B := \{y : |y| < n/2 + c\sqrt{n} \wedge y_n = 0\}$.
- $C := \{y : y_n = 1\}$.

Note the one-to-one correspondence between strings in B and strings in C obtained by flipping the nth bit. Now, consider any $y \in B$ such that y witnesses $(\vec{0}, x')$ but not $(\vec{0}, x)$. Flipping the nth bit of y yields a string $y' \in C$ such that Y witnesses $(\vec{0}, x)$ but not $(\vec{0}, x')$. Hence among $y \in B \cup C$ there is an equal number of witnesses for x and x'. For any $y \in A$, $y_n = 0$, whence $|y - x'| = |y - x| + 1$. Therefore, any y that witnesses $(\vec{0}, x)$ must also witness $(\vec{0}, x')$, whence $w(x) \leq w(x')$. $\square$

We compute $w(x)$ by conditioning on $|y|$:

$$w(x) = \sum_{n_1=1}^{n/2-c\sqrt{n}} \left(\Pr\left[\Delta(x,y) \geq n/2 + c\sqrt{n} \mid |y| = n_1\right] \cdot \Pr[|y| = n_1]\right) .$$

Fix $|x| =: m$, pick a random y with $|y| = n_1$, and suppose there are k coordinates i such that $x_i = y_i$. Then, $\Delta(x,y) = (m-k) + (n_1 - k) = m + n_1 - 2k$. Hence,

$$\Delta(x,y) \geq n/2 + c\sqrt{n} \iff k \leq \frac{m+n_1}{2} - \frac{n}{4} - \frac{c}{2}\sqrt{n}\,.$$

Note that given a random y with weight $|y| = n_1$, the probability that exactly k of m coordinates have $x_i = y_i = 1$ follows the hypergeometric distribution $\mathrm{Hyp}(k; n, m, n_1)$. Therefore, we can

express the probability $\Pr_{|y|=n_1}[\Delta(x,y) \geq n/2 + c\sqrt{n}]$ as

$$\Pr_{|y|=n_1}\left[\Delta(x,y) \geq n/2 + c\sqrt{n}\right] \;=\; \sum_{k \leq \frac{m+n_1}{2} - \frac{n}{4} - \frac{c}{2}\sqrt{n}} \mathrm{Hyp}(k; n, m, n_1)\,.$$

Finally, we show that $w(x) > 4\varepsilon$ for a suitably large constant $|x|$ with the following claims, whose proofs require tight tail bounds for binomial and hypergeometric distributions and are left to Section 4.7.

Claim 68. *Conditioned on $|y| \leq n/2 - 2\sqrt{n}$, we have $\Pr[|y| \geq n/2 - 2.1\sqrt{n}] \leq \frac{1}{3}$.*

Claim 69. *For all $d < n/2 - 2.1\sqrt{n}$, we have $\Pr[\Delta(x_2, y) \geq n/2 + d\sqrt{n}] \geq 0.95$.*

Its easy to see from the previous two claims that $w(x) > 0.95 \cdot (2/3) > 4\varepsilon$. □

4.6 Concluding Remarks

Our most important contribution in this chapter was to prove a multi-round lower bound on a fundamental problem in communication complexity, the Gap-Hamming Distance problem. As a consequence, we extended several known $\Omega(1/\varepsilon^2)$-type space bounds for various data stream problems, such as the Distinct Elements problem, to multi-pass algorithms. These resolve long-standing open questions.

4.7 Proofs of Technical Lemmas

Many claims in this chapter require tight upper and lower tail bounds for binomial and hypergeometric distributions. We use Chernoff bounds where they apply. For other bounds, we approximate using normal distributions. We use Feller [Fel68] as a reference.

Definition 11. *For $x \in \mathbb{R}$, let $\phi(x) := e^{-x^2/2}/\sqrt{2\pi}$ and*

$$N(x) := \int_x^\infty \phi(y)dy.$$

$N(x)$ is the cumulative distribution function of the normal distribution. We use it in Fact 54 to approximate $T(x)$. Here, we'll also use it to approximate tails of the binomial and hypergeometric distributions.

Lemma 70 (Feller, Chapter VII, Lemma 2.). *For all $x > 0$,*

$$\phi(x)\left(\frac{1}{x} - \frac{1}{x^3}\right) < N(x) < \phi(x)\frac{1}{x}.$$

In the next two theorems, we let $a \sim b$ denote $a = b(1 \pm o(1))$.

Theorem 71 (Feller, Chapter VII, Theorem 2.). *For fixed z_1, z_2,*

$$Pr[n/2 + (z_1/2)\sqrt{n} \leq |y| \leq n/2 + (z_2/2)\sqrt{n}] \sim N(z_1) - N(z_2).$$

Theorem 72. *For any γ such that $\gamma = \omega(1)$ and $\gamma = o(n^{1/6})$, we have*

$$\sum_{k > n/2 + \gamma\sqrt{n}/2} \binom{n}{k} \sim N(\gamma).$$

Claim 73 (Restatement of Claim 68). *Conditioned on $|y| \leq n/2 - 2\sqrt{n}$,*

$$Pr_{|y| \leq n/2 - 2\sqrt{n}}[|y| \geq n/2 - 2.1\sqrt{n}] \leq 0.0828.$$

Proof. By Theorem 71 and Lemma 70, we have

$$\begin{aligned}\Pr[n/2 - 2.1\sqrt{n} \leq |y| \leq n/2 - 2\sqrt{n}] &\sim N(4) - N(4.2) \\ &\leq \phi(4)/4 - \phi(4.2)(4.2^{-1} - 4.2^{-3}) \\ &\leq 2.0219 * 10^{-5}\end{aligned}$$

By Fact 54, $\Pr[|y| \leq n/2 - 2\sqrt{n}] \geq 2^{-3 \cdot 2^2 - 2} = 2^{-14} = 6.1035 \cdot 10^{-5}$. Putting the two terms together, we get

$$\Pr[|y| \geq n/2 - 2.1\sqrt{n} | |y| \leq n/2 - 2\sqrt{n}] \leq \frac{2.0219 \cdot 10^{-5}}{6.1035 \cdot 10^{-5}} \leq 1/3.$$

□

Claim 74 (Restatement of Claim 69). *For all $d < n/2 - 2.1\sqrt{n}$,*

$$\Pr[\Delta(x_2, y) \geq n/2 + 2\sqrt{n}] \geq 0.95.$$

Proof. The proof follows from the following claim, instantiated with $c = 2$ and $\alpha = 2.1$. □

Claim 75. *For all $\alpha > c$, $|x| = \gamma n$, and all $\gamma \geq 1 - (1 - c/\alpha)/4$,*

$$\Pr_{|y| = n/2 - \alpha\sqrt{n}}[\Delta(x, y) \geq n/2 + c\sqrt{n}] \geq 1 - \exp\left(-\frac{2(\alpha - c)\alpha^2(1 + o(1))}{3\alpha + c}\right).$$

Proof. Let $m := |x| = \gamma n$ and let $n_1 = n/2 - \alpha\sqrt{n}$. Then, the probability that a random y with $|y| = n_2$ can be expressed using the hypergeometric distribution $\mathrm{Hyp}(k; n, m, n_1)$. Let the m set bits of x be the defects. The probability of k of the n_1 bits of y are defective is

$\mathrm{Hyp}(k; n, m, n_1)$. Note that $\Delta(x, y) = (m - k) + (n_1 - k) = m + n_1 - 2k$. Therefore,

$$\Delta(x, y) \geq n/2 + c\sqrt{n} \Leftrightarrow k \leq \frac{m + n_1}{2} - \frac{n}{4} - \frac{c}{2}\sqrt{n}. = \frac{\gamma n}{2} - \frac{\alpha + c}{2}\sqrt{n}$$

We express the probability $\Pr_{|y|=n_1}[\Delta(x, y) \geq n/2 + c\sqrt{n}]$ as

$$\Pr_{|y|=n_1}[\Delta(x, y) \geq n/2 + c\sqrt{n}] = \Pr_{K \sim \mathrm{Hyp}(k;n,m,n_1)}[K \leq \frac{\gamma n}{2} - \frac{\alpha + c}{2}\sqrt{n}].$$

Next, we use a concentration of measure result due to Hush and Scovel [HS05]. Here, we present a simplified version.

Theorem 76 (Hush, Scovel). *Let $m = \gamma n > n_1 = n/2 - \alpha\sqrt{n}$, and let $\beta = n/m(n - m)$.*

$$\Pr[K - E[K] > \eta] < \exp(-2\beta\eta^2(1 + o(1))).$$

The expected value of a random variable K distributed according to $\mathrm{Hyp}(K; n, m, n_1)$ is

$$E[K] = \frac{mn_1}{n} = \frac{\gamma n}{n}\left(\frac{n}{2} - \alpha\sqrt{n}\right) = \frac{\gamma n}{2} - \gamma\alpha\sqrt{n}.$$

Set $\eta := (\alpha - c)\sqrt{n}/4$. Note that

$$\begin{aligned} E[K] + \eta &= \frac{\gamma n}{2} - \gamma\alpha\sqrt{n} + \frac{\alpha - c}{4}\sqrt{n} \\ &\leq \frac{\gamma n}{2} - \frac{\alpha + c}{2}\sqrt{n} \\ &= \frac{m + n_1}{2} - \frac{n}{4} - \frac{c}{2}\sqrt{n}. \end{aligned}$$

where the inequality holds because $\gamma \geq 1 - (1 - c/\alpha)/4$. Note also that $(1 - c/\alpha)/4 =$

$(\alpha - c)/4\alpha$, so $1 - (1 - c/\alpha)/4 = (3\alpha + c)/4\alpha$. By Theorem 76,

$$\begin{aligned}
\Pr[K > \frac{\gamma n}{2} - \frac{\alpha + c}{2}\sqrt{n}] &= \Pr[K - E[K] > \eta] \\
&< \exp\left(-\frac{2n\eta^2(1 + o(1))}{m(n - m)}\right) \\
&= \exp\left(-\frac{2(\alpha - c)^2(1 + o(1))}{16\gamma(1 - \gamma)}\right) \\
&\leq \exp\left(-\frac{2(\alpha - c)^2(4\alpha)^2(1 + o(1))}{16(\alpha - c)(3\alpha + c)}\right) \\
&= \exp\left(-\frac{2(\alpha - c)\alpha^2(1 + o(1))}{3\alpha + c}\right).
\end{aligned}$$

It follows that $\Pr[K \leq \frac{\gamma n}{2} - \frac{\alpha+c}{2}\sqrt{n}] \geq 1 - \exp\left(-\frac{2(\alpha-c)\alpha^2(1+o(1))}{3\alpha+c}\right)$. □

Claim 77. *For any $x_L \in \{0,1\}^{n_L}$, $\text{GHD}(x_L, y_L)$ is defined for at least a $\left(e^{-2(c')^2}/5c'\right)$-fraction of $y_L \in \{0,1\}^{n_L}$.*

Proof. Without loss of generality, assume $x_L = \vec{0}$. Then, $\text{GHD}(x_L, y_L)$ is defined for all y such that $|y| \leq n_L/2 - c'\sqrt{n_L}$ or $|y| \geq n_L/2 + c'\sqrt{n_L}$. Note that for any constant $x > c'$,

$$\begin{aligned}
&\Pr_y\left[|y| \leq \frac{n_L}{2} - c'\sqrt{n_L}\right] \\
&\quad \geq \Pr\left[\frac{n_L}{2} - x\sqrt{n_L} \leq |y| \leq \frac{n_L}{2} - c'\sqrt{n_L}\right] \\
&\quad \geq N(2c') - N(2x) \\
&\quad \geq \phi(2c')\left(\frac{1}{2c'} - \frac{1}{(2c')^3}\right) - \frac{\phi(2x)}{2x} \\
&\quad = \frac{e^{-(2c')^2/2}}{\sqrt{2\pi}}(\left(\frac{1}{2c'} - \frac{1}{(2c')^3}\right) - \frac{e^{-2x^2}}{2x\sqrt{2\pi}} \\
&\quad \geq \frac{e^{-2(c')^2}}{10c'}.
\end{aligned}$$

$\Pr[|y| \geq n_L/2 + c'\sqrt{n_L}]$ is bounded in the same fashion. □

Chapter 5

Improving the Gap Hamming Distance Lower Bounds Through Better Round Elimination

Gap Hamming Distance is a well-studied problem in communication complexity, in which Alice and Bob have to decide whether the Hamming distance between their respective n-bit inputs is less than $n/2 - \sqrt{n}$ or greater than $n/2 + \sqrt{n}$. We show that every k-round bounded-error communication protocol for this problem sends a message of at least $\Omega(n/(k^2 \log k))$ bits. This lower bound has an exponentially better dependence on the number of rounds than the previous best bound, due to Brody and Chakrabarti. Our communication lower bound implies strong space lower bounds on algorithms for a number of data stream computations, such as approximating the number of distinct elements in a stream.

5.1 Introduction

5.1.1 The Communication Complexity of the Gap Hamming Distance Problem

Communication complexity studies the communication requirements of distributed computing. In its simplest and best-studied setting, two players, Alice and Bob, receive inputs x and y, respectively, and are required to compute some function $f(x, y)$. Clearly, for most functions f, the two players need to communicate to solve this problem. The basic question of communication complexity is the *minimal amount* of communication needed. By abstracting away from the resources of local computation time and space, communication complexity gives us a bare-bones but elegant model of distributed computing. It is interesting for its own sake but is also useful as one of our main sources of lower bounds in many other models of computation, including data structures, circuits, Turing machines, VLSI, and streaming algorithms. The basic results are excellently covered in the book of Kushilevitz and Nisan [KN97], but many additional fundamental results have appeared since its publication in 1997.

One of the few basic problems whose randomized communication complexity is not yet well-understood is the *Gap Hamming Distance* (GHD) problem, defined as follows.

> GHD: Alice receives input $x \in \{0,1\}^n$ and Bob receives input $y \in \{0,1\}^n$, with the promise that $|\Delta(x, y) - n/2| \geq \sqrt{n}$, where Δ denotes the Hamming distance. Decide whether $\Delta(x, y) < n/2$ or $\Delta(x, y) > n/2$.

Mind the gap between $n/2 - \sqrt{n}$ and $n/2 + \sqrt{n}$, which is what makes this problem interesting and useful. Indeed, the communication complexity of the gapless version, where there is no promise on the inputs, can easily be seen to be linear (for instance by a reduction from disjointness). The gap makes the problem easier, and the question is how it affects the communication complexity: does it remain linear? A gap size of $\Theta(\sqrt{n})$ is the natural choice—a $\Theta(1)$

fraction of the inputs lie inside the promise area for this gap size, and as we'll see below, it is precisely this choice of gap size that has strong implications for streaming algorithms lower bounds. Moreover, understanding the complexity of the $\sqrt{n}$-gap version can be shown to imply a complete understanding of the GHD problem for all gaps.

Randomized protocols for GHD and more general problems can be obtained by sampling. Suppose for instance that it is promised that either $\Delta(x,y) \leq (1/2-\gamma)n$ or $\Delta(x,y) \geq (1/2+\gamma)n$. Choosing an index $i \in [n]$ at random, the predicate $[x_i \neq y_i]$ is a coin flip with heads probability $\leq 1/2-\gamma$ in the first case and $\geq 1/2+\gamma$ in the second. It is known that flipping such a coin $\Theta(1/\gamma^2)$ times suffices to distinguish these two cases with probability at least $2/3$. Hence if we use shared randomness to choose $\Theta(1/\gamma^2)$ indices, we obtain a one-round bounded-error protocol with communication $\Theta(1/\gamma^2)$ bits. In particular, for GHD (where $\gamma = 1/\sqrt{n}$), the communication is $\Theta(n)$ bits, which is no better than the trivial upper bound of n when Alice just sends x to Bob.

What about lower bounds? Indyk and Woodruff [IW03] managed to prove a linear lower bound for the case of one-round protocols for GHD, where there is only one message from Alice to Bob (see also [Woo04, JKS08]). However, going beyond one-round bounds turned out to be quite a difficult problem. Recently, Brody and Chakrabarti [BC09] obtained linear lower bounds for all *constant*-round protocols:

Theorem 78. *[BC09] Every k-round bounded-error protocol for GHD sends a message of length $\frac{n}{2^{O(k^2)}}$.*

In fact our bound is significant as long as the number of rounds is $k \leq c_0\sqrt{\log n}$, for a universal constant c_0. Regarding lower bounds that hold irrespective of the number of rounds, an easy reduction gives an $\Omega(\sqrt{n})$ lower bound (which is folklore): take an instance of the gapless version of the problem on $x, y \in \{0,1\}^{\sqrt{n}}$ and "repeat" x and y $\sqrt{n}$ times each. This blows up the gap from 1 to $\sqrt{n}$, giving an instance of GHD on n bits. Solving this n-bit instance

of GHD solves the $\sqrt{n}$-bit instance of the gapless problem. Since we have a linear lower bound for the latter, we obtain a general $\Omega(\sqrt{n})$ bound for GHD.[1]

5.1.2 Our results

Our main result is an improvement of the bound of Brody and Chakrabarti, with an exponentially better dependence on the number of rounds:

Theorem 79. *Every k-round bounded-error protocol for GHD sends a message of length* $\Omega\left(\frac{n}{k^2 \log k}\right)$.

In fact we get a bound for the more general problem of distinguishing distance $\Delta(x, y) \leq (1/2 - \gamma)n$ from $\Delta(x, y) \geq (1/2 + \gamma)n$, as long as $\gamma = \Omega(1/\sqrt{n})$: for this problem every k-round protocol sends a message of $\Omega\left(\frac{1}{k^2 \log k} \frac{1}{\gamma^2}\right)$ bits.

Like the result of [BC09], our lower bound deteriorates with the number of rounds. Also like their result, our proof is based on *round elimination*, an important framework for proving communication lower bounds. Our proof contains an important insight into this framework that we now explain.

A communication problem usually involves a number of parameters, such as the input size, an error bound, and in our case the gap size. The round elimination framework consists of showing that a k-round protocol solving a communication problem for a class $\mathcal{C}$ of parameters can be turned into a $(k-1)$-round protocol for an easier class $\mathcal{C}'$, provided the message communicated in the first round is short. This fact is then applied repeatedly to obtain a 0-round protocol (say), for some nontrivial class of instances. The resulting contradiction can then be

[1] In fact the same proof lower-bounds the *quantum* communication complexity; a linear quantum lower bound for the gapless version follows easily from Razborov's work [Raz02] and the observation that $\Delta(x, y) = |x| + |y| - 2|x \wedge y|$. However, as Brody and Chakrabarti observed, in the quantum case this $\sqrt{n}$ lower bound is essentially tight: there is a bounded-error quantum protocol, based on a well-known quantum algorithm for approximate counting, that communicates $O(\sqrt{n} \log n)$ qubits. This also implies that lower bound techniques which apply to quantum protocols, such as discrepancy, factorization norms [LS07, LS08], and the pattern matrix method [She08], cannot prove better bounds for classical protocols.

recast as a communication lower bound. Historically, the easier class $\mathcal{C}'$ has contained *smaller input lengths*[2] than those in $\mathcal{C}$.

In contrast to previous applications of round elimination, we manage to *avoid shrinking the input length*: the simplification will instead come from a slight deterioration in the error parameter. Here is how this works. If Alice's first message is short, then there is a specific message and a large set A of inputs on which Alice would have sent that message. Roughly speaking, we can use the largeness of A to show that *almost any* input $\tilde{x}$ for Alice is close to A in Hamming distance. Therefore, Alice can "move" $\tilde{x}$ to its nearest neighbor, x, in A: this makes her first message redundant, as it is constant for all inputs $x \in A$. Since x and $\tilde{x}$ have small Hamming distance, it is likely that both pairs $(\tilde{x}, y)$ and (x, y) are on the same side of the gap, i.e. have the same GHD value. Hence the correctness of the new protocol, which is one round shorter, is only mildly affected by the move. Eliminating all k rounds in this manner, while carefully keeping track of the accumulating errors, yields a lower bound of $\Omega(n/(k^4 \log^2 k))$ on the maximum message length of any k-round bounded-error protocol for GHD.

Notice that this lower bound is slightly weaker than the $\Omega(n/(k^2 \log k))$ bound stated above. To obtain the stronger bound, we leave the purely combinatorial setting and analyze a version of GHD *on the unit sphere*:[3] Alice's input is now a unit vector $x \in \mathbb{R}^n$ and Bob's input is a unit vector $y \in \mathbb{R}^n$, with the promise that either $x \cdot y \geq 1/\sqrt{n}$ or $x \cdot y \leq -1/\sqrt{n}$ (as we show below in Section 5.2, this version and the Boolean one are essentially equivalent in terms of communication complexity). Alice's input is now close to the large, constant-message set A in *Euclidean distance*. The rest of the proof is as outlined above, but the final bound is stronger than in the combinatorial proof for reasons that are discussed in Section 5.2.2. Although this

[2] In fact, $\mathcal{C}$ and $\mathcal{C}'$ are often designed such that an instance in $\mathcal{C}$ is a "direct sum" of several independent instances in $\mathcal{C}'$.

[3] The idea of going to the unit sphere was also used by Jayram et al. [JKS08] for a simplified one-round lower bound. As we will see in Section 5.2, doing so is perhaps even more natural than working with the combinatorial version; in particular it is then easy to make GHD into a *dimension-independent* problem.

proof uses arguments from high-dimensional geometry, such as measure concentration, it arguably remains conceptually simpler than the one in [BC09].

Related work. The round elimination technique was formalized in Miltersen et al. [MNSW98] and dates back even further, at least to Ajtai's lower bound for predecessor data structures [Ajt88]. For us, the most relevant previous use of this technique our result from Chapter 4, where a weaker lower bound is proved on GHD.

As in the previous chapter, this proof identifies a large subset A of inputs on which Alice sends the same message. The "largeness" of A is used to identify a suitable subset of $(n/3)$ coordinates such that Alice can "lift" any $(n/3)$-bit input $\tilde{x}$, defined on these coordinates, to some n-bit input $x \in A$. In the resulting protocol for $(n/3)$-bit inputs, the first message is now constant, hence redundant, and can be eliminated.

The input size thus shrinks from n to $n/3$ in one round elimination step. As a result of this constant-factor shrinkage, the Brody-Chakrabarti final lower bound necessarily decays exponentially with the number of rounds. Our proof crucially avoids this shrinkage of input size by instead considering the *geometry* of the set A, and exploiting the natural invariance of the GHD predicate to small perturbations of the inputs.

Remark. After we obtained our results, a subset of the authors independently proved an optimal $\Omega(n)$ lower bound, independent of the number of rounds [CR10]. However, the techniques they introduce are completely different, and rather involved. Our result, through its relatively simple and elegant proof, should be of independent interest to the community.

5.1.3 Applications to Streaming

The introduction of gapped versions of the Hamming distance problem [IW03] was motivated by the streaming model of computation, in particular the problem of approximating the number

of distinct elements in a data stream. For many data stream problems, including the distinct elements problem, the goal is to output a multiplicative approximation of some real-valued quantity. Usually, both *randomization* and *approximation* are required. When both are allowed, there are often remarkably space-efficient solutions.

As Indyk and Woodruff showed, *communication lower bounds* for the Gap Hamming Distance problem imply *space lower bounds* on algorithms that output the number of distinct elements in a data stream up to a multiplicative approximation factor $1 \pm \gamma$. The reduction from GHD works as follows. Alice converts her n-bit string $x = x_1 x_2 \cdots x_n$ into a stream of tuples $\sigma = \langle (1, x_1), (2, x_2), \ldots, (n, x_n) \rangle$. Bob converts y into $\tau = \langle (1, y_1), (2, y_2), \ldots, (n, y_n) \rangle$ in a similar fashion. Using a streaming algorithm for the distinct elements problem, Alice processes σ and sends the memory contents to Bob, who then processes τ starting from where Alice left off. In this way, they estimate the number of distinct elements in $\sigma \circ \tau$. Note that each element in σ is unique, and that elements in τ are distinct from elements in σ precisely when $x_i \neq y_i$. Hence, an accurate approximation ($\gamma = \Omega(1/\sqrt{n})$ is required) for the number of distinct elements in $\sigma \circ \tau$ gives an answer to the original GHD instance. This reduction can be extended to multi-pass streaming algorithms in a natural way: when Bob is finished processing τ, he sends the memory contents back to Alice, who begins processing σ a second time. Generalizing, it is easy to see that a p-pass streaming algorithm gives a $(2p-1)$-round communication protocol, where each message is the memory contents of the streaming algorithm. Accordingly, a lower bound on the length of the largest message of $(2p-1)$-round protocols gives a space lower bound for the p-pass streaming algorithm.

Thus, the one-round linear lower bound by Indyk and Woodruff [IW03] yields the desired $\Omega(1/\gamma^2)$ (one-pass) space lower bound for the streaming problem. Similarly, our new communication lower bounds imply $\Omega(1/(\gamma^2 p^2 \log p))$ space lower bounds for p-pass algorithms for the streaming problem. This improves on previous bounds for all $p = o(n^{1/4}/\sqrt{\log n})$.

Organization of the chapter. We start with some preliminaries in Section 5.2, including a discussion of the key measure concentration results that we will use, both for the sphere and for the Hamming cube, in Section 5.2.2. In Section 5.3 we prove our main result, while in Section 5.4 we give the simple combinatorial proof of the slightly weaker result mentioned above.

5.2 Preliminaries

Notation. For $x, y \in \mathbb{R}^n$, let $d(x,y) := \|x - y\|$ be the Euclidean distance between x and y, and $x \cdot y$ their inner product. For $z \in \mathbb{R}$, define $\mathrm{sgn}(z) := 0$ if $z \geq 0$, and $\mathrm{sgn}(z) = 1$ otherwise. For a set $S \subseteq \mathbb{R}^n$, let $d(x,S)$ be the infimum over all $y \in S$ of $d(x,y)$. The unique rotationally-invariant probability distribution on the n-dimensional sphere $\mathbb{S}^{n-1}$ is the Haar measure, which we denote by ν. When we say that a vector is taken from the uniform distribution over a measurable subset of the sphere, we will always mean that it is distributed according to the Haar measure, conditioned on being in that subset.

Define the max-cost of a communication protocol to be the length of the longest *single* message sent during an execution of the protocol, for a worst-case input. We use $R^k_\varepsilon(f)$ to denote the minimal max-cost amongst all two-party, k-round, public-coin protocols that compute f with error probability at most ε on every input (here a "round" is one message).

5.2.1 Problem Definition

We will prove our lower bounds for the problem $\mathcal{GHD}_{d,\gamma}$, where d is an integer and $\gamma > 0$. In this problem Alice receives a d-dimensional unit vector x, and Bob receives a d-dimensional unit vector y, with the promise that $|x \cdot y| \geq \gamma$. Alice and Bob should output $\mathrm{sgn}(x \cdot y)$.

We show that $\mathcal{GHD}_{n,1/\sqrt{n}}$ has essentially the same randomized communication complexity as the problem GHD that we defined in the introduction. Generalizing that definition, for any

$g > 0$ define the problem $\mathrm{GHD}_{n,g}$, in which the input is formed of two n-bit strings x and y, with the promise that $|\Delta(x,y) - n/2| \geq g$, where Δ is the Hamming distance. Alice and Bob should output 0 if $\Delta(x,y) < n/2$ and 1 otherwise.

The following proposition shows that for any $\sqrt{n} \leq g \leq n$, the problems $\mathrm{GHD}_{n,g}$ and $\mathcal{GHD}_{d,\gamma}$ are essentially equivalent from the point of view of randomized communication complexity (with shared randomness) as long as $d \geq n$ and $\gamma = \Theta(g/n)$. It also shows that the randomized communication complexity of $\mathcal{GHD}_{d,\gamma}$ is independent of the dimension d of the input, as long as d is large enough with respect to γ.

Proposition 1. *For every $\varepsilon > 0$, there is a constant $C_0 = C_0(\varepsilon)$ such that for all integers $k, d \geq 0$ and $\sqrt{n} \leq g \leq n$, we have $R^k_{2\varepsilon}(\mathcal{GHD}_{d,C_0 g/n}) \leq R^k_\varepsilon(\mathrm{GHD}_{n,g}) \leq R^k_\varepsilon(\mathcal{GHD}_{n,2g/n})$.*

Proof. We begin with the right-hand inequality. The idea is that a $\mathrm{GHD}_{n,g}$ protocol can be obtained by applying a given $\mathcal{GHD}$ protocol to a suitably transformed input. Let $x, y \in \{0,1\}^n$ be two inputs to $\mathrm{GHD}_{n,g}$. Define $\tilde{x} = \frac{1}{\sqrt{n}}\left((-1)^{x_i}\right)_{i\in[n]}$ and $\tilde{y} = \frac{1}{\sqrt{n}}\left((-1)^{y_i}\right)_{i\in[n]}$. Then $\tilde{x}, \tilde{y} \in \mathbb{S}^{n-1}$. Moreover, $\tilde{x}\cdot\tilde{y} = 1 - 2\Delta(x,y)/n$. Therefore, if $\Delta(x,y) \geq n/2 + g$ then $\tilde{x}\cdot\tilde{y} \leq -2g/n$, and if $\Delta(x,y) \leq n/2 - g$ then $\tilde{x}\cdot\tilde{y} \geq 2g/n$. This proves $R^k_\varepsilon(\mathrm{GHD}_{n,g}) \leq R^k_\varepsilon(\mathcal{GHD}_{n,2g/n})$.

For the left inequality, let x and y be two unit vectors (in any dimension) such that $|x\cdot y| \geq \gamma$, where $\gamma = C_0 g/n$. Note that since $g \geq \sqrt{n}$, we have $n = \Omega(\gamma^{-2})$. Using shared randomness, Alice and Bob pick a sequence of vectors $w_1, \ldots, w_n$, each independently and uniformly drawn from the unit sphere. Define two n-bit strings $\tilde{x} = (\mathrm{sgn}(x\cdot w_i))_{i\in[n]}$ and $\tilde{y} = (\mathrm{sgn}(y\cdot w_i))_{i\in[n]}$. Let $\alpha = \cos^{-1}(x\cdot y)$ be the angle between x and y. Then a simple argument (used, e.g., by Goemans and Williamson [GW95]) shows that the probability that a random unit vector w is such that $\mathrm{sgn}(x\cdot w) \neq \mathrm{sgn}(y\cdot w)$ is exactly α/π. This means that for each i, the bits $\tilde{x}_i$ and $\tilde{y}_i$ differ with probability $\frac{1}{\pi}\cos^{-1}(x\cdot y)$, independently of the other bits of $\tilde{x}$ and $\tilde{y}$. The first few terms in the Taylor series expansion of $\cos^{-1}$ are $\cos^{-1}(z) = \frac{\pi}{2} - z - \frac{z^3}{6} + O(z^5)$. Hence, for each i, $\Pr_{w_i}(\tilde{x}_i \neq \tilde{y}_i) = 1/2 - \Theta(x\cdot y)$, and these events are independent for different i.

Choosing C_0 sufficiently large, with probability at least $1 - \varepsilon$, the Hamming distance between $\tilde{x}$ and $\tilde{y}$ is at most $n/2 - g$ if $x \cdot y \geq \gamma$, and it is at least $n/2 + g$ if $x \cdot y \leq -\gamma$. □

5.2.2 Concentration of Measure

It is well known that the Haar measure ν on a high-dimensional sphere is tightly concentrated around the equator—around *any* equator, which makes it a fairly counterintuitive phenomenon. The original phrasing of this phenomenon, usually attributed to P. Lévy [Lév51], goes by showing that among all subsets of the sphere, the one with the smallest "boundary" is the spherical cap $S_\gamma^x = \{y \in \mathbb{S}^{n-1} : x \cdot y \geq \gamma\}$. The following standard volume estimate will prove useful (see, e.g., [Bal97], Lemma 2.2).

Fact 80. *Let $x \in \mathbb{S}^{n-1}$ and $\gamma > 0$. Then $\nu(S_\gamma^x) \leq e^{-\gamma^2 n/2}$.*

Given a measurable set A, define its *t-boundary* $A_t := \{x \in \mathbb{S}^{n-1} : d(x, A) \leq t\}$, for any $t > 0$. At the core of our results will be the standard fact that, for any not-too-small set A, the set A_t contains almost all the sphere, even for moderately small values of t.

Fact 81 (Concentration of measure on the sphere). *For any measurable $A \subseteq \mathbb{S}^{n-1}$ and any $t > 0$,*

$$\Pr(x \in A) \Pr(x \notin A_t) \leq 4\, e^{-t^2 n/4}, \tag{5.1}$$

where the probabilities are taken according to the Haar measure on the sphere.

Proof. The usual measure concentration inequality for the sphere (Theorem 14.1.1 in [Mat02]) says that for any set $B \subseteq \mathbb{S}^{n-1}$ of measure at least $1/2$ and any $t' > 0$,

$$\Pr(x \notin B_{t'}) \leq 2\, e^{-(t')^2 n/2}.$$

This suffices to prove the fact if $\Pr(x \in A) \geq 1/2$, so assume that $\Pr(x \in A) < 1/2$. Let t_0 be such that A_{t_0} has measure $1/2$; such a t_0 exists by continuity. Applying measure concentration to $B = A_{t_0}$ gives

$$\Pr(x \notin A_{t'+t_0}) \leq 2\, e^{-(t')^2 n/2}, \tag{5.2}$$

for all $t' > 0$, while applying it to $B = \overline{A_{t_0}}$ yields

$$\Pr(x \in A_{t_0 - t''}) \leq \Pr(x \notin B_{t''}) \leq 2\, e^{-(t'')^2 n/2} \tag{5.3}$$

for all $t'' \leq t_0$, since $A_{t_0-t''}$ is included in the complement of $(\overline{A_{t_0}})_{t''}$. Taking $t'' = t_0$ gives us $\Pr(x \in A) \leq 2\, e^{-t_0^2 n/2}$. If $t \leq t_0$ then this suffices to prove the inequality. Otherwise, set $t' := t - t_0$ in (5.2) and $t'' := t_0$ in (5.3) and multiply the two inequalities to obtain the required bound, by using that $t_0^2 + (t - t_0)^2 \geq t^2/2$ (which holds since $2t_0^2 + t^2/2 - 2t\, t_0 = (\sqrt{2}t_0 - t/\sqrt{2})^2 \geq 0$). $\square$

Why the sphere? In Section 5.4 we give a proof of a slightly weaker lower bound than the one in our main result by using measure concentration facts on the Hamming cube only. We present those useful facts now, together with a brief discussion of the differences, in terms of concentration of measure phenomenon, between the Haar measure on the sphere and the uniform distribution over the hypercube. These differences point to the reasons why the proof of Section 5.4 gives an inferior bound.

On the Hamming cube, the analogous notion of spherical cap is the Hamming ball: let $T_c^x = \{y \in \{0,1\}^n : \Delta(x,y) \leq n/2 - c\sqrt{n}\}$ be the Hamming ball of radius $n/2 - c\sqrt{n}$ centered at x. The analogue of Fact 80 is given by the Chernoff bound:

Fact 82. *For all $c > 0$, we have $2^{-n}|T_c^x| \leq e^{-2c^2}$.*

A result similar to Lévy's, attributed to Harper [Har66], states that among all subsets (of

the Hamming cube) of a given size, the ball is the one with the smallest boundary. Following a similar proof as for Fact 81, one can get the following statement for the Hamming cube (see e.g. Corollary 4.4 in [Bar05]):

Fact 83 (Concentration of measure on the Hamming cube). *Let $A \subseteq \{0,1\}^n$ be any set, and define $A_c = \{x \in \{0,1\}^n : \exists y \in A,\ \Delta(x,y) \leq c\sqrt{n}\}$. Then*

$$\Pr(x \in A) \Pr(x \notin A_c) \leq e^{-c^2}, \tag{5.4}$$

where the probabilities are taken according to the uniform distribution on the Hamming cube.

To compare these two statements, embed the Hamming cube in the sphere by mapping $x \in \{0,1\}^n$ to the vector $v_x = \frac{1}{\sqrt{n}}((-1)^{x_i})_{i\in[n]}$, so that two strings of Hamming distance $c\sqrt{n}$ are mapped to vectors with Euclidean distance $2\sqrt{c}/n^{1/4}$. While on the sphere inequality (5.1) indicates that most points are at distance roughly $1/\sqrt{n}$ from any set of measure half, if we are restricted to the Hamming cube then very few points are at a corresponding Hamming distance of 1 from, say, the set of all strings with fewer than $n/2$ 1s, which has measure roughly $1/2$ in the cube. This difference is crucial: it indicates that the n-dimensional cube is too rough an approximation of the n-dimensional sphere for our purposes, perhaps explaining why our combinatorial bound in Section 5.4 yields a somewhat weaker dependence on the number of rounds.

5.3 Main Result

Our main result is the following.

Theorem 84. *Let $0 \leq \varepsilon \leq 1/50$. There exist constants C, C' depending only on ε such that the following holds for any $\gamma > 0$ and any integers $n \geq \varepsilon^2/(4\gamma^2)$ and $k \leq C'/(\gamma \ln(1/\gamma))$: if*

$\mathcal{P}$ is a randomized ε-error k-round communication protocol for $\mathcal{GHD}_{n,\gamma}$ then some message has length at least $\frac{C}{k^2 \ln k} \cdot \frac{1}{\gamma^2}$ bits.

Using Proposition 1 we immediately get a lower bound for the Hamming cube version $\mathrm{GHD} = \mathrm{GHD}_{n,\sqrt{n}}$:

Corollary 85. *Any ε-error k-round randomized protocol for GHD communicates $\Omega(n/(k^2 \ln k))$ bits.*

This follows from Theorem 84 when $k = o(\sqrt{n}/\log n)$. If k is larger, then the bound stated in the Corollary is in fact weaker than the general $\Omega(\sqrt{n})$ lower bound which we sketched in the introduction.

5.3.1 Proof Outline

We now turn to the proof of Theorem 84. Let ε, γ and n be as in the statement of the theorem. Since lowering n only makes the $\mathcal{GHD}_{n,\gamma}$ problem easier, for the rest of this section we assume that $n := \varepsilon^2/(4\gamma^2)$ is fixed, and for simplicity of notation we write $\mathcal{GHD}_\gamma$ for $\mathcal{GHD}_{n,\gamma}$.

Measurability. Before proceeding with the proof, we first need to handle a small technicality arising from the continuous nature of the input space: namely, that the distributional protocol might make decisions based on subsets of the input space that are not measurable. To make sure that this does not happen, set $\delta = \gamma/6$ and consider players Alice and Bob who first round their inputs to the closest vector in a fixed δ-net, and then proceed with an ε-error protocol for $\mathcal{GHD}_{\gamma/2}$. Since by definition rounding to the δ-net moves any vector a distance at most δ, the rounding will affect the inner product $x \cdot y$ by at most $2\delta + \delta^2 \leq \gamma/2$. As a result, Alice and Bob will succeed with probability $1-\varepsilon$ provided they are given valid inputs to $\mathcal{GHD}_\gamma$. Hence any randomized ε-error protocol for $\mathcal{GHD}_{\gamma/2}$ can be transformed into a randomized ε-error protocol for $\mathcal{GHD}_\gamma$ with the same communication, but which initially rounds its inputs to a

discrete set. We prove a lower bound on the latter type of protocol. This will ensure that all sets encountered in the proof are measurable.

Distributional complexity. By Yao's principle it suffices to lower-bound the *distributional complexity*, i.e., to analyze *deterministic* protocols that are correct with probability $1-\varepsilon$ under some input distribution. As our input distribution for $\mathcal{GHD}_\gamma$ we take the distribution that is uniform over the inputs satisfying the promise $|x \cdot y| \geq \gamma$. Given our choice of n, Claim 87 below guarantees that the $\nu\times\nu$-measure of non-promise inputs is at most ε. Hence it will suffice to lower-bound the distributional complexity of protocols making error at most $2\,\varepsilon$ under the distribution $\nu \times \nu$. We define an *ε-protocol* to be a deterministic communication protocol for $\mathcal{GHD}_{n,\gamma}$ whose error under the distribution $\nu \times \nu$ is at most ε, where we say that a protocol $\mathcal{P}$ makes an error if $\mathcal{P}(x, y) \neq \mathrm{sgn}(x, y)$.

We prove a lower bound on the maximum length of a message sent by any ε-protocol, via round elimination. The main reduction step is given by the following technical lemma:

Lemma 86 (Round Elimination on the sphere). *Let $\varepsilon, \gamma > 0$, $n = \varepsilon^2/(4\gamma^2)$, and $1 \leq \kappa \leq k$. Assume there is a κ-round ε-protocol $\mathcal{P}$ such that the first message has length bounded as $c_1 \leq C_1 \frac{n}{k^2 \ln k} - 7\ln(2k)$ where C_1 is a universal constant. Then there is a $(\kappa-1)$-round ε'-protocol $\mathcal{Q}$ (obtained by eliminating the first message of $\mathcal{P}$), where $\varepsilon' \leq \left(1 + \frac{1}{k}\right)\varepsilon + \frac{1}{16k}$.*

Before proving this lemma in Section 5.3.2, we show how it implies Theorem 84.

Proof. [Proof of Theorem 84] We will show that in any k-round $(2\,\varepsilon)$-protocol, there is a message sent of length at least $C_1 n/(k^2 \ln k) - 7\ln(2k)$. The discussion in the "Distributional complexity" paragraph above shows this suffices to prove the theorem, by setting $C = C_1\varepsilon^2/8$, and choosing C' small enough so that the bound on k in the statement of the theorem implies that $7\ln(2k) < C_1 n/(2k^2 \ln k)$.

Let $\mathcal{P}$ be a k-round $(2\,\varepsilon)$-protocol, and assume for contradiction that each round of communication uses at most $C_1 n/(k^2 \ln k) - 7\ln(2k)$ bits. Solving the recurrence

$\varepsilon_\kappa = (1+1/k)\varepsilon_{\kappa-1} + 1/(16k)$, $\varepsilon_0 = 2\,\varepsilon$ gives $\varepsilon_\kappa = (1+1/k)^\kappa(2\,\varepsilon + 1/16) - 1/16$, so that applying Lemma 86 k times leads to a *0-round* protocol for $\mathcal{GHD}_\gamma$ that errs with probability at most $\varepsilon' \leq e\,(2\,\varepsilon + 1/16) - 1/16 \leq 1/4$ over the input distribution $\nu \times \nu$. We have reached a contradiction: such a protocol needs communication and hence cannot be 0-round. Hence $\mathcal{P}$ must send a message of length at least $C_1 n/(k^2 \ln k) - 7\ln(2k)$.

□

5.3.2 The Main Reduction Step

Proof. [Proof of Lemma 86] Let $\mathcal{P}(x, y)$ denote the output of the protocol on input x, y. Define $x \in \mathbb{S}^{n-1}$ to be *good* if $\Pr_{\nu\times\nu}(P(x, y) \text{ errs } |x) \leq (1 + 1/k)\varepsilon$. By Markov's inequality, at least a $1/(k+1)$-fraction of x (distributed according to ν) are good. For a given message m, let A_m be the set of all good x on which Alice sends m as her first message. The sets A_m, over all messages $m \in \{0,1\}^{c_1}$, form a partition of the set of good x. Define $m_1 := \text{argmax}_m \nu(A_m)$ and let $A := A_{m_1}$. We then have $\nu(A) \geq \frac{1}{k+1} 2^{-c_1} \geq e^{-c_1 - \ln(k+1)}$.

We now define protocol $\mathcal{Q}$. Alice receives an input $\tilde{x}$, Bob receives $\tilde{y}$, both distributed according to ν. Alice computes the point $x \in A$ that is closest to $\tilde{x}$, and Bob sets $y := \tilde{y}$. They run protocol $\mathcal{P}(x, y)$ without Alice sending the first message, so Bob starts and proceeds as if he received m_1 from Alice.

To prove the lemma, it suffices to bound the error probability ε' of $\mathcal{Q}$ with input $\tilde{x}, \tilde{y}$ distributed according to $\nu \times \nu$. Define $d_1 = 2\sqrt{\frac{c_1 + 6\ln(2k) + 2}{n}}$. We consider the following bad events:

- $\text{BAD}_1 : d(\tilde{x}, A) > d_1$,
- $\text{BAD}_2 : P(x, y) \neq \text{sgn}(x \cdot y)$,
- $\text{BAD}_3 : d(\tilde{x}, A) \leq d_1$ but $\text{sgn}(x \cdot y) \neq \text{sgn}(\tilde{x} \cdot \tilde{y})$.

If none of those events occurs, then protocol $\mathcal{P}$ outputs the correct answer. We bound each of them separately, and will conclude by upper bounding ε' with a union bound.

The first bad event can be easily bounded using the measure concentration inequality from Fact 81. Since $\tilde{x}$ is uniformly distributed in $\mathbb{S}^{n-1}$ and $\Pr(A) \geq e^{-c_1 - \ln(k+1)}$, we get

$$\Pr(\text{BAD}_1) \leq 4\, e^{-d_1^2 n/4 + c_1 + \ln(k+1)} \leq 4\, e^{-5\ln(2k) - 2} \leq \frac{1}{32k}.$$

The second bad event has probability bounded by $(1 + 1/k)\,\varepsilon$ by the goodness of x. Now consider event BAD_3. Without loss of generality, we may assume that $\tilde{x} \cdot \tilde{y} = \tilde{x} \cdot y > 0$ but $x \cdot y < 0$ (the other case is treated symmetrically). In order to bound BAD_3, we will use two claims. The first shows that the probability that $\tilde{x} \cdot y$ is close to 0 for a random $\tilde{x}$ and y is small. The second uses measure concentration to show that, if $\tilde{x} \cdot y$ is not too close to 0, then moving $\tilde{x}$ to the nearby x is unlikely to change the sign of the inner product.

Claim 87. *Let x, y be distributed according to ν. For any real $\alpha \geq 0$, we have $\Pr(0 \leq x \cdot y \leq \alpha) \leq \alpha\sqrt{n}$.*

Proof. With ω_n the volume of the n-dimensional Euclidean unit ball, we write (see for example [BGK+98], Lemma 5.1)

$$\Pr(0 \leq x \cdot y \leq \alpha) \;=\; \frac{(n-1)\,\omega_{n-1}}{n\,\omega_n} \int_0^\alpha (1-t^2)^{\frac{n-3}{2}} \mathrm{dt} \;\leq\; \alpha\sqrt{n},$$

where we used $\frac{\omega_{n-1}}{\omega_n} < \sqrt{\frac{n+1}{2\pi}} < \sqrt{n}$. □

Claim 88. *Let $x, \tilde{x}$ be two fixed unit vectors at distance $\|x - \tilde{x}\| = d \in [0, d_1]$, and $0 < \alpha \leq 1/(4\sqrt{n})$. Let y be taken according to ν. Then $\Pr(\tilde{x} \cdot y \geq \alpha \wedge x \cdot y < 0) \leq\; e^{-\alpha^2 n/(8d_1^2)}$.*

Proof. Note that $x \cdot \tilde{x} = 1 - \|x - \tilde{x}\|^2/2 = 1 - d^2/2$. Since the statement of the lemma is

rotationally-invariant, we may assume without loss of generality that

$$\begin{aligned}\tilde{x} &= (1, 0, 0 \ldots, 0),\\ x &= (1 - d^2/2, -\sqrt{d^2 - d^4/4}, 0, \ldots, 0),\\ y &= (y_1, y_2, y_3, \ldots, y_n).\end{aligned}$$

Therefore, $y_1 \geq \alpha$ when $\tilde{x} \cdot y \geq \alpha$. Note that

$$x \cdot y = x_1 y_1 + x_2 y_2 \geq (1 - d^2/2)\alpha - \sqrt{d^2 - d^4/4}\, y_2.$$

Hence the event $\tilde{x} \cdot y \geq \alpha \wedge x \cdot y < 0$ implies

$$y_2 > \frac{(1 - d^2/2)\,\alpha}{\sqrt{d^2 - d^4/4}} \geq \frac{\alpha}{2d},$$

where we used the fact that $d \leq d_1 \leq 1$, given our assumption on c_1. By Fact 80, the probability that, when y is sampled from ν, y_2 is larger than $\alpha/(2d)$ is at most $e^{-\alpha^2 n/(8d^2)}$. Hence the probability that both $\tilde{x} \cdot y \geq \alpha$ and $x \cdot y < 0$ happen is at most as much. □

Setting $\alpha = 1/(128k\sqrt{n})$, by Claim 87 we find that the probability that $0 \leq \tilde{x} \cdot y \leq \alpha$ is at most $1/(128k)$. Furthermore, the probability that $\tilde{x} \cdot y \geq \alpha$ and $x \cdot y < 0$ is at most $\exp\left(-\frac{n}{2^{19}k^2(c_1 + 6\ln(2k)+2)}\right)$ by Claim 88. This bound is less than $1/(128k)$ given our assumption on c_1, provided C_1 is a small enough constant. Putting both bounds together, we see that

$$\Pr(\tilde{x} \cdot y \geq 0 \wedge x \cdot y < 0) < 1/(64k).$$

The event that $\tilde{x} \cdot y < 0$ but $x \cdot y \geq 0$ is bounded by $1/(64k)$ in a similar manner. Hence, $\Pr(\text{BAD}_3) < 1/(32k)$. Taking the union bound over all three bad events concludes the proof of the lemma. □

5.4 A Simple Combinatorial Proof

In this section we present a combinatorial proof of the following:

Theorem 89. *Let $0 \leq \varepsilon \leq 1/50$. There exists a constant C'' depending on ε only, such that the following holds for any $g \leq C''\sqrt{n}$ and $k \leq n^{1/4}/(1024 \log n)$: if $\mathcal{P}$ is a randomized ε-error k-round communication protocol for $\mathrm{GHD}_{n,g}$ then some message has length at least $\frac{n}{(512k)^4 \log^2 k}$ bits.*

Even though this is a weaker result than Theorem 84, its proof is simpler and is based on concentration of measure in the Hamming cube rather than on the sphere (we refer to Section 5.2.2 for a high-level comparison of the two proofs). Interestingly, the dependence on the number of rounds that we obtain is quadratically worse than that of the proof using concentration on the sphere.

We proceed as in Section 5.3.1, observing that it suffices to lower-bound the distributional complexity of $\mathrm{GHD}_{n,g}$ under a distribution uniform over the inputs satisfying the promise $|\Delta(x,y) - n/2| \geq g$. In fact, as we did before, by taking C'' small enough we can guarantee that the number of non-promise inputs is at most $\varepsilon\, 2^n$. Hence it will suffice to lower-bound the distributional complexity of protocols making error at most $2\,\varepsilon$ under the uniform input distribution. We define an *ε-protocol* to be a deterministic communication protocol for GHD whose distributional error under the uniform distribution is at most ε. The following is the analogue of Lemma 86, from which the proof of Theorem 89 follows as in Section 5.3.1.

Lemma 90 (Round Elimination on the Hamming cube). *Let $\varepsilon > 0$ and κ, k be two integers such that $k \geq 128$ and $1 \leq \kappa \leq k \leq n^{1/4}/(1024 \log n)$. Assume that there is a κ-round ε-protocol $\mathcal{P}$ such that the first message has length bounded by $c_1 \leq n/((512k)^4 \log^2 k)$. Then there exists a $(\kappa - 1)$-round ε'-protocol $\mathcal{Q}$ (obtained by eliminating the first message of $\mathcal{P}$) where $\varepsilon' \leq \left(1 + \frac{1}{k}\right)\varepsilon + \frac{1}{16k}$.*

Proof. Define $x \in \{0,1\}^n$ to be *good* if $\Pr(P(x,y) \text{ errs } |x) \leq (1+1/k)\varepsilon$. By Markov's inequality, at least a $1/(k+1)$-fraction of $x \in \{0,1\}^n$ are good. For a given message m, let $A_m := \{\text{good } x : \text{ Alice sends } m \text{ given } x\}$. The sets A_m, over all messages $m \in \{0,1\}^{c_1}$, together form a partition of the set of good x. Define $m_1 := \text{argmax}_m |A_m|$, and let $A := A_{m_1}$. By the pigeonhole principle, we have $|A| \geq \frac{1}{k+1} 2^{n-c_1}$.

We now define protocol $\mathcal{Q}$. Alice receives an input $\tilde{x}$, Bob receives $\tilde{y}$, uniformly distributed. Alice computes the string $x \in A$ that is closest to $\tilde{x}$ in Hamming distance, and Bob sets $y := \tilde{y}$. They run protocol $\mathcal{P}(x,y)$ without Alice sending the first message, so Bob starts and proceeds as if he received the fixed message m_1 from Alice.

To prove the lemma, it suffices to bound the error probability ε' of $\mathcal{Q}$ under the uniform distribution. Define $d_1 := 9\sqrt{n}/((1024k)^2 \log k)$. As in the proof of Lemma 86, we consider the following bad events:

- $\text{BAD}_1 : \Delta(x,\tilde{x}) > d_1\sqrt{n}$,
- $\text{BAD}_2 : P(x,y) \neq \text{GHD}(x,y)$,
- $\text{BAD}_3 : \Delta(x,\tilde{x}) \leq d_1\sqrt{n}$ but $\text{GHD}(\tilde{x},y) \neq \text{GHD}(x,y)$.

If none of those events occurs, then protocol $\mathcal{P}$ outputs the correct answer. We bound each of them separately, and will conclude by a union bound. BAD_1 is easily bounded using Fact 83, which implies

$$\Pr(\tilde{x} \notin A_{d_1}) \leq e^{-81n/((1024k)^4 \log^2 k)} 2^{c_1 + \log(k+1)} \leq \frac{2}{k^2} \leq \frac{1}{32k},$$

given our assumptions on c_1 and k. The second bad event is bounded by $(1+1/k)\,\varepsilon$, by definition of A.

We now turn to BAD_3. The event that $\text{GHD}(\tilde{x},y) \neq \text{GHD}(x,y)$ only depends on the relative distances between x, $\tilde{x}$, and y, so we may apply a shift to assume that $x = (0,\ldots,0)$.

Without loss of generality, we assume that $\Delta(\tilde{x}, y) > n/2$ and $|y| < n/2$ (the error bound when $\Delta(\tilde{x}, y) < n/2$ and $|y| > n/2$ is proved in a symmetric manner). Note that, since y is uniformly random (subject to $|y| < n/2$), by a standard head estimate for the binomial distribution with probability at least $1 - 1/(128k)$ we have $|y| \leq n/2 - \sqrt{n}/(128k)$ (this is analogous to the estimate from Claim 87 that we used in the continuous setting). Hence we may assume that this holds with an additive loss of at most $1/(128k)$ in the error. Now

$$\Delta(\tilde{x}, y) > n/2 \iff |\tilde{x}| + |y| - 2|\tilde{x} \cap y| > n/2 \iff |\tilde{x} \cap y| < \frac{|\tilde{x}| + |y| - n/2}{2}.$$

It is clear that the worst case in this statement is for $|y| = n/2 - \sqrt{n}/(128k)$ and $|\tilde{x}| = \Delta(x, \tilde{x}) = d_1\sqrt{n}$. By symmetry, the probability that this event happens is the same as if we fix any y of the correct weight, and $\tilde{x}$ is a random string of weight $d_1\sqrt{n}$. The expected intersection size is $|y||\tilde{x}|/n = |\tilde{x}|/2 - d_1/(128k)$, and so by Hoeffding's inequality (see e.g. the bound on the tail of the hypergeometric distribution given in [Chv79]), for $a = \sqrt{n}/(256k) - d_1/(128k)$, we have

$$\Pr\left(|\tilde{x} \cap y| \leq \frac{|\tilde{x}| + |y| - n/2}{2}\right) = \Pr\left(|\tilde{x} \cap y| \leq \mathbb{E}[|\tilde{x} \cap y|] - a\right) \leq e^{-2a^2/(d_1\sqrt{n})}.$$

Given our choice of d_1 we have $a \geq 3\sqrt{n}/(4 \cdot 256k)$, and hence the upper bound is at most $1/k^2 \leq 1/(128k)$, given our assumption on k. Applying the union bound over all bad events then yields the lemma. $\square$

Chapter 6

Conclusions

Communication Complexity plays a central role in proving lower bounds and hardness results in theoretical computer science. This thesis described some of our work in this area and gives new lower bounds for three communications problems, with applications to circuit complexity, wireless sensor networks, and streaming algorithms.

We give several new lower bounds for multiparty pointer jumping for both myopic and collapsing protocols. While these lower bounds do not extend to general MPJ_k protocols, we hope that the intuition from these results will prove fruitful to proving a general lower bound that would place MPJ_k outside ACC^0. We also leverage the work of Pudlák, Rödl, and Sgall [PRS97] to achieve the first nontrivial $o(n)$ upper bound for MPJ_k, thus showing that the complexity of MPJ_k is deeper than previously suspected.

For the distributed functional monitoring problem, we consider several non-monotone functions and provide a suite of new lower bounds, including a generic adversarial lower bound technique. Together with our upper bound for monitoring empricial entropy [ABC09], we tell a contrasting story for functional monitoring of non-monotone functions. When deletions are disallowed, efficient non-monotone functional monitoring is possble; however, when deletions are permitted, essentially nothing nontrivial is possible, even when the monitoring protocol is

randomized.

Finally, we give the first multiround lower bound for the Gap-Hamming-Distance problem, answering a long-standing open problem [Kum06]. As a result, we extend $\Omega(1/\varepsilon^2)$ lower bounds for streaming algorithms that estimate frequency moments F_k to a constant number of passes. Our result also implies a lower bound for streaming algorithms that use a constant number of passes to compute the empirical entropy of a stream, given the work of Chakrabarti et al. [CCM07].

Several of our results use the Round Elimination Lemma. Together, these results show the power of this classic lower bound technique. In particular, the $\tilde{\Omega}(n/k^2)$ lower bound for Gap-Hamming-Distance from Chapter 5 is the first round elimination result that remains nontrivial even for a $\omega(\log n)$ number of rounds. Its use of isoperimetry appears to be a promising line of attack for other problems.

Bibliography

[AB07] Sanjeev Arora and Boaz Barak. *Complexity Theory: A Modern Approach*. Available online at ⟨http://www.cs.princeton.edu/theory/complexity/⟩, 2007.

[ABC09] Chrisil Arackaparambil, Joshua Brody, and Amit Chakrabarti. Functional monitoring without monotonicity. In *Proc. 36th International Colloquium on Automata, Languages and Programming*, pages 95–106, 2009.

[Abl96] Farid Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theoretical Computer Science*, 175(2):139–159, 1996.

[ADHP06] Micah Adler, Erik D. Demaine, Nicholas J. A. Harvey, and Mihai Pătraşcu. Lower bounds for asymmetric communication channels and distributed source coding. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 251–260, 2006.

[Ajt88] Miklós Ajtai. A lower bound for finding predecessors in Yao's cell probe model. *Combinatorica*, 8:235–247, 1988.

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999. Preliminary version in *Proc. 28th Annu. ACM Symp. Theory Comput.*, pages 20–29, 1996.

[Bal97] K. Ball. An elementary introduction to modern convex geometry. *Flavors of Geometry*, 31, 1997.

[Bar05] A. Barvinok. Lecture notes on measure concentration, 2005.

[BC09] Joshua Brody and Amit Chakrabarti. A multi-round communication lower bound for gap hamming and some consequences. In *Proc. 24th Annual IEEE Conference on Computational Complexity*, pages 358–368, 2009.

[Bez87] Sergei Bezrukov. Specification of the maximal sized subsets of the unit cube with respect to given diameter. *Problems of Information Transmission*, 1:106–109, 1987.

[BF99] Paul Beame and Faith E. Fich. Optimal bounds for the predecessor problem. In *Proc. 31st Annual ACM Symposium on the Theory of Computing*, pages 295–304, 1999.

[BG06] Lakshminath Bhuvanagiri and Sumit Ganguly. Estimating entropy over data streams. In *Proc. 14th Annual European Symposium on Algorithms*, pages 148–159, 2006.

[BGK+98] A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovász, and M. Simonovits. Approximation of diameters: Randomization doesn't help. In *Proc. 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 244–251, 1998.

[BHK01] László Babai, Thomas P. Hayes, and Peter G. Kimmel. The cost of the missing bit: Communication complexity with help. *Combinatorica*, 21(4):455–488, 2001.

[BJK+04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *Proc. 6th International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 128–137, 2004.

[BJKS02] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 209–218, 2002.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. Syst. Sci.*, 45(2):204–232, 1992.

[BO03] Brian Babcock and Chris Olston. Distributed top-k monitoring. In *Proc. Annual ACM SIGMOD Conference*, pages 28–39, 2003.

[BPS05] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. In *Proc. 32nd International Colloquium on Automata, Languages and Programming*, pages 1176–1188, 2005.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Comput. Complexity*, 4:350–366, 1994.

[CCGL03] Amit Chakrabarti, Bernard Chazelle, Benjamin Gum, and Alexey Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. *Disc. Comput. Geom.: The Goodman-Pollack Festschrift*, pages 313–328, 2003. Preliminary version in *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 305–311, 1999.

[CCM07] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for computing the entropy of a stream. In *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 328–335, 2007.

[CFL83] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proc. 15th Annual ACM Symposium on the Theory of Computing*, pages 94–99, 1983.

[Cha07] Amit Chakrabarti. Lower bounds for multi-player pointer jumping. In *Proc. 22nd Annual IEEE Conference on Computational Complexity*, pages 33–45, 2007.

[Chv79] Vaclav Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

[CJP08] Amit Chakrabarti, T. S. Jayram, and Mihai Pătraşcu. Tight lower bounds for selection in randomly ordered streams. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 720–729, 2008.

[CMY08] Graham Cormode, S. Muthukrishnan, and Ke Yi. Algorithms for distributed functional monitoring. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1076–1085, 2008.

[CMZ06] Graham Cormode, S. Muthukrishnan, and Wei Zhuang. What's different: Distributed, continuous monitoring of duplicate-resilient aggregates on data streams. In *Proc. 22nd International Conference on Data Engineering*, page 57, 2006.

[CR04] Amit Chakrabarti and Oded Regev. An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 473–482, 2004.

[CR10] Amit Chakrabarti and Oded Regev. A tight lower bound for gap hamming distance. Private communication, 2010.

[DGGR04] Abhinandan Das, Sumit Ganguly, Minos N. Garofalakis, and Rajeev Rastogi. Distributed set expression cardinality estimation. In *Proc. 30th International Conference on Very Large Data Bases*, pages 312–323, 2004.

[DJS98] Carsten Damm, Stasys Jukna, and Jiří Sgall. Some bounds on multiparty communication complexity of pointer jumping. *Comput. Complexity*, 7(2):109–127, 1998. Preliminary version in *Proc. 13th International Symposium on Theoretical Aspects of Computer Science*, pages 643–654, 1996.

[EGHK99] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *MOBICOM*, pages 263–270, 1999.

[Fel68] William Feller. *An Introduction to Probability Theory and its Applications*. John Wiley, New York, NY, 1968.

[FM85] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.*, 31(2):182–209, 1985.

[GM07] Sudipto Guha and Andrew McGregor. Lower bounds for quantile estimation in random-order and multi-pass streaming. In *Proc. 34th International Colloquium on Automata, Languages and Programming*, pages 704–715, 2007.

[GM08] Sudipto Guha and Andrew Mcgregor. Tight lower bounds for multi-pass stream computation via pass elimination. In *Proc. 35th International Colloquium on Automata, Languages and Programming*, pages 760–772, 2008.

[Gol09] Alexander Golynski. Cell probe lower bounds for succinct data structures. In *Proc. 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 625–634, 2009.

[Gro06] Andre Gronemeier. NOF-multiparty information complexity bounds for pointer jumping. In *Proc. 31st International Symposium on Mathematical Foundations of Computer Science*, 2006.

[GW95] Michel Goemans and David Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.

[Har66] L. H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1:385–394, 1966.

[HG91] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Comput. Complexity*, 1:113–129, 1991.

[HNO08] Nicholas J. A. Harvey, Jelani Nelson, and Krzysztof Onak. Sketching and streaming entropy via approximation theory. In *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 489–498, 2008.

[HS05] Don Hush and Clint Scovel. Concentration of the hypergeometric distribution. *Statistics and Probability Letters*, 75(2):127–132, 2005.

[IW03] Piotr Indyk and David Woodruff. Tight lower bounds for the distinct elements problem. In *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 283–289, 2003.

[JKS08] T. S. Jayram, R. Kumar, and D. Sivakumar. The one way communication complexity of hamming distance. *Theory of Computing*, 4(1):129–135, 2008.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, Cambridge, 1997.

[Kum06] Ravi Kumar. Story of distinct elements, 2006. talk at IITK Workshop on Algorithms for Data Structures.

[KW90] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM J. Disc. Math.*, 3(2):255–265, 1990. Preliminary version in *Proc. 20th Annual ACM Symposium on the Theory of Computing*, pages 539–550, 1988.

[Lév51] P. Lévy. *Problèmes concrets d'analyse fonctionnelle*. Gauthier-Villars, 1951.

[LS07] Nati Linial and Adi Shraibman. Lower bounds in communication complexity based on factorization norms. In *Proc. 39th Annual ACM Symposium on the Theory of Computing*, pages 699–708, 2007.

[LS08] T. Lee and A. Shraibman. Disjointness is hard in the multi-party number-on-the-forehead model. In *Proc. 23rd Annual IEEE Conference on Computational Complexity*, pages 81–91, 2008.

[Mat02] Jiří Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, 2002.

[Mil94] Peter Bro Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proc. 26th Annual ACM Symposium on the Theory of Computing*, pages 625–634, 1994.

[MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998. Preliminary version in *Proc. 27th Annual ACM Symposium on the Theory of Computing*, pages 103–111, 1995.

[Mut03] S. Muthukrishnan. Data streams: Algorithms and applications. In *Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, page 413, 2003.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991.

[NW93] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SICOMP*, 22(1):211–219, 1993. Preliminary version in *Proc. 23rd Annu. ACM Symp. Theory Comput.*, pages 419–429, 1991.

[NW99] Ashwin Nayak and Felix Wu. The quantum query complexity of approximating the median and related statistics. In *Proc. 31st Annual ACM Symposium on the Theory of Computing*, pages 384–393, 1999.

[Păt10] Mihai Pătraşcu. Towards polynomial lower bounds for dynamic problems. In *Proc. 42nd Annual ACM Symposium on the Theory of Computing*, 2010.

[PRS97] Pavel Pudlák, Vojtěch Rödl, and Jiří Sgall. Boolean circuits, tensor ranks and communication complexity. *SIAM J. Comput.*, 26(3):605–633, 1997.

[PT06] Mihai Pătraşcu and Mikkel Thorup. Time-space trade-offs for predecessor search. In *Proc. 38th Annual ACM Symposium on the Theory of Computing*, pages 232–240, 2006.

[PV10] Mihai Pătraşcu and Emanuele Viola. Cell-probe lower bounds for succinct partial sums. In *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[Raz02] Alexander A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya of the Russian Academy of Science, Mathematics*, 67:0204025, 2002.

[Sau72] N. Sauer. On the density of families of sets. *J. Combin. Theory Ser. A*, 13:145–147, 1972.

[Sen03] Pranab Sen. Lower bounds for predecessor searching in the cell probe model. In *Proc. 18th Annual IEEE Conference on Computational Complexity*, pages 73–83, 2003.

[She08] Alexander A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 85–94, New York, NY, USA, 2008. ACM.

[SSK07] Izchak Sharfman, Assaf Schuster, and Daniel Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Trans. Database Syst.*, 32(4), 2007.

[SV08] Pranab Sen and S. Venkatesh. Lower bounds for predecessor searching in the cell probe model. *J. Comput. Syst. Sci.*, 74(3):364–385, 2008.

[SW73] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory*, 19(4):471–480, 1973.

[VW07] Emanuele Viola and Avi Wigderson. One-way multi-party communication lower bound for pointer jumping with applications. In *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 427–437, 2007.

[Woo04] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 167–175, 2004.

[Woo07] David P. Woodruff. *Efficient and Private Distance Approximation in the Communication and Streaming Models*. PhD thesis, MIT, 2007.

[Woo09] David Woodruff. The average case complexity of counting distinct elements. In *Proc. 12th International Conference on Database Theory*, 2009.

[Xia92] B. Xiao. *New Bounds in Cell Probe Model*. PhD thesis, UC San Diego, 1992.

[Yao77] Andrew C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proc. 18th Annual IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[Yao90] Andrew C. Yao. On ACC and threshold circuits. In *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 619–627, 1990.

[YZ09] Ke Yi and Qin Zhang. Multi-dimensional online tracking. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1098–1107, 2009.