58th Annual IEEE Symposium on Foundations of Computer Science

A Rounds vs. Communication Tradeoff for Multi-Party Set Disjointness

Mark Braverman
Institute for Advanced Studies and
Computer Science Department, Princeton University
Princeton, NJ, JSA
Email: mbraverm@cs.princeton.edu

Rotem Oshman
Computer Science Department
Tel Aviv University
Tel Aviv, Israel
Email: roshman@tau.ac.il

***Abstract*—In the set disjointess problem, we have k players, each with a private input $X^i \subseteq [n]$, and the goal is for the players to determine whether or not their sets have a global intersection. The players communicate over a shared blackboard, and we charge them for each bit that they write on the board.**

We study the trade-off between the number of interaction rounds we allow the players, and the total number of bits they must send to solve set disjointness. We show that if R rounds of interaction are allowed, the communication cost is $\tilde{\Omega}(nk^{1/R}/R^4)$, which is nearly tight. We also leverage our proof to show that *wellfare maximization with unit demand bidders* cannot be solved efficiently in a small number of rounds: here, we have k players bidding on n items, and the goal is to find a matching between items and player that bid on them which approximately maximizes the total number of items assigned. It was previously shown by Alon et. al. that $\Omega(\log\log k)$ rounds of interaction are required to find an assignment which achieves a constant approximation to the maximum-wellfare assignment, even if each player is allowed to write $n^{\epsilon(R)}$ bits on the board in each round, where $\epsilon(R) = \exp(-R)$. We improve this lower bound to $\Omega(\log k/\log\log k)$, which is known to be tight up to a $\log\log k$ factor.

I. Introduction

Set disjointness is a classical problem in communication complexity: we have two players, Alice and Bob, and they receive sets $X, Y \subseteq \{1, \ldots, n\}$ (respectively). Their goal is to determine whether or not their sets intersect, that is, whether $X \cap Y = \emptyset$. Kalyanasundaram and Schnitger [1] and Razborov [2] showed that $\Omega(n)$ bits of communication are required, even for randomized protocols, and this lower bound is one of the most widely-used lower bounds in communication complexity.

The extension of set disjointness to the multi-party, number-in-hand setting is as follows: we have k players, each with a private input $X^i \subseteq \{1, \ldots, n\}$, and the goal is to determine whether there is a global intersection, that is, whether $\bigcap_{i=1}^k X^i = \emptyset$ or not. Communication between the players can be over a shared blackboard, where players write messages that are seen by all other players, or over point-to-point channels, in which case players can only send each other private messages. We focus here on the shared blackboard model. The multi-party version of set disjointness reduces to many natural problems; for example, most of the lower bounds in [3], [4], which include pointwise problems like computing the pointwise-and of the input vectors, and graphs problems such as connectivity, triangle-freeness and others, can be shown by reduction from disjointness. Promise versions of set disjointness also have applications in streaming [5], [6], [7], [8].

Much of the interest in multi-party (number-in-hand) communication complexity comes from distributed computing (e.g., [4], [9], [10], [11] and others). In this area we usually care about *round complexity* rather than *total communication complexity*: we typically assume that every processor in the system has some limited number of bits it can send in each round, and we ask how many rounds we need to solve a given problem [12]. If the total communication complexity of the problem is high compared to the total bandwidth per round, then of course we get a lower bound on the number of rounds required to solve it. However, some problems have low overall communication complexity, but cannot be solved efficiently in a small number of rounds. We show that set disjointness in the shared blackboard model falls into this class: it is known that the problem can be solved with a total of $\Theta(n \log k)$ bits [9], but we show that if we are restricted to R rounds of interaction, then the communication complexity of set disjointness is $\tilde{\Omega}(nk^{1/R}/R^4)$. Our lower bound is nearly tight, as there is an easy $\tilde{O}(nk^{1/R})$ upper bound. (When $R = \Theta(\log k/\log\log k)$ we reach nearly-optimal communication complexity, $\tilde{\Theta}(n)$, so it does not make sense to consider larger R. The factor $1/R^4$ is therefore at most polylogarithmic in k.)

We also show a similar result for wellfare maximization with unit-demand bidders, studied in [13], [14]. In this problem we have k players bidding on n items, and our goal is to find a matching between items and players that bid on them, such that the total number of items assigned is at least a constant fraction of the maximum possible. The input for wellfare maximization is represented as an unweighted bipartite graph, with players on one side and items on the other, with edges between the players and the items they are interested in. Our goal is to find and output a matching whose size is at least some fixed constant fraction of the size of the maximum matching.

It was shown in [13] that wellfare maximization requires high communication for simultaneous protocols, and in [14] that if each player is restricted to sending only $n^{\epsilon(R)}$ bits per

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.22

round (where $\epsilon(R) = \exp(-R)$), then $\Omega(\log\log k)$ rounds are required to find a matching that is within a constant fraction of the maximum matching. (The value of n used in [14] is exponential in k.) It was previously known that the problem can be solved in $O(\log k)$ rounds [13], but the true round complexity remained open.

Here we show that for k bidders and $n = \Theta(k)$ items, finding a constant approximation to the wellfare maximization in R rounds requires $\Omega(k^{1+1/R}/R^2)$ bits of communication, implying that $\Omega(\log k/\log\log k)$ rounds are required if each player can only send k^{ϵ} bits per round. This is tight up to the $\log\log k$ factor [13].

A. A Round-Efficient Protocol for Disjointness

In [9] we showed that the communication complexity of disjointness in the shared blackboard model is $\Theta(n \log k)$, where n is the size of the universe and k is the number of players. The lower bound of $\Omega(n \log k)$ allows one to get logarithmic lower bounds for distributed systems where each player has small bandwidth per round of communication, so that the total bandwidth per round is linear. However, the simple upper bound we gave in [9] requires as many as n *rounds* of communication, raising the question: what is the trade-off between rounds and communication for disjointness in the shared blackboard model?

The following simple protocol achieves communication complexity $\tilde{O}(nk^{1/R})$ in R rounds. Let us represent the inputs to the players by their characteristic vectors $X^1, \ldots, X^k \in \{0,1\}^n$, where $X^i_j = 1$ iff element j appears in player i's input. To solve set disjointness, we try to find, for each coordinate $j = 1, \ldots, n$, a "witness" to the fact that element j is not in the intersection: a player i with $X^i_j = 0$. If for each coordinate we can find such a witness then the intersection is empty.

The most naïve approach would be to have all the players write on the board the indices of all coordinates in which their input is zero, but this can incur communication as high as $O(nk)$. Instead, for each coordinate, we "guess" how many players have input zero in this coordinate, and have players write their zero on the board with probability corresponding to this guess, so that with high probability we will find at least one player with a zero, but not too many.

More precisely, for each $i \in [n]$, let Z_i be the number of players that have $X^i_j = 0$ (that is, the number of players missing element i from their inputs). In each round $r = 0, \ldots, R-1$, we target coordinates i that have $k^{(R-(r-1))/R} \leq Z_i \leq k^{(R-r)/R}$, going from coordinates where many players have zero to coordinates where only a small number do. Each player announces each zero coordinate in its input with probability $\tilde{\Theta}(k^{(r-1-R)/R})$, unless this coordinate already appears on the board from some previous round. (We cannot afford to re-write coordinates that we already caught in previous rounds.) At the end of the protocol we conclude that there is an intersection iff some coordinate does not appear on the board.

In round r, if indeed $k^{(R-(r-1))/R} \leq Z_i \leq k^{(R-r)/R}$, then in expectation coordinate i will be written on the board between $\tilde{\Theta}(1)$ and $\tilde{\Theta}(k^{1/R})$ times, requiring $\log n$ bits each time it is written. On the other hand, if $Z_i > k^{(R-r)/R}$ then we probably already saw coordinate i in some previous round, so it will not be announced in the current round at all (this is important, otherwise too many players would announce it in round r); and if $0 < Z_i < k^{(R-(r-1))/R}$, it is also not likely that coordinate i will be announced, but we will catch it in some future round.

Notice the structure of the protocol: first we dispense with coordinates where many players have zero, by having each player that has zero send a weak signal about it — i.e., announce it with probability only $1/k^{(R-1)/R}$. Then we gradually move to coordinates where fewer and fewer players have zero, by having players that have zero send stronger and stronger signals, increasing the probability that they announce the coordinate. Our lower bound shows that this structure is in some sense inherent: one *has* to first target coordinates where many players have zero, and gradually focus on "less popular" coordinates.

B. Related Work

a) Brief overview of work on multi-party communication and information complexity: There are two main models studied in multi-party communication complexity, the *shared blackboard model* studied here, and the *point-to-point model* (also called "message-passing"). Information complexity techniques have found application in both.[1]

Information complexity was first applied to multi-party computation in [6], which studies set disjointness with a promise: the players are promised that either the sets intersect at exactly one element, or the sets are pairwise disjoint. The communication complexity of promise set disjointness is $\Theta(n/k)$ [7]. In the current paper we use the notion of *conditional information cost* developed in [6], and also follow the technique by which they decompose the problem into many smaller problems (*direct sum*). In [6] the authors were able to prove a lower bound proved of $\Omega(n/k^2)$ on promise set disjointness, and this was gradually improved until [7] gave the tight lower bound of $\Omega(n/k)$. The problem is notable for its connections to streaming algorithms [5].

In the point-to-point model, information-theoretic ideas were first applied in [15] to provide lower bounds on problems of frequency moment estimation, and later in [4], [16], [11] to other graph, estimation and linear algebra problems. In [17] a tight lower bound was shown for the set disjointness problem in the point-to-point model, by defining an appropriate notion of information cost for this model, and proving a lower bound on it. We later proved a tight lower bound for disjointness

[1] An entirely different line of work on multi-party communication complexity is *number-on-forehead* communication, where each player can see the inputs to all the other players, but not its own input. This model is very different in nature from the ones we consider, which feature private ("number-in-hand") inputs.

in the shared blackboard model [9], again using information complexity.

The lower bounds mentioned above, as well as information-complexity lower bounds for two-party communication, tend follow the same structure: first, one proves that the overall problem can be decomposed into many copies of some smaller problem, whose costs add up to the cost of the large problem; this is called *direct sum*. Then one gives a lower bound for the smaller problem, using properties of communication protocols in the model under consideration. This approach, pioneered in [18], [6], is the one we follow in this paper.

b) Round vs. communication tradeoffs: There are by now many bounded-round communication lower bounds (although in some cases they are obtained "along the way" to an unrestricted-round lower bound). For example, in [19] it is shown that set disjointness with a specific sparsity promise requires $\Theta(n \log^{(} R)n)$ bits of communication for two players, where $\log^{(} i)n$ is the log iterated i times. The upper bound of [19] was extended to the multi-player problem of finding an intersection in [10].

A useful technique for proving round-restricted lower bounds is *round elimination*: given an r-round protocol that uses "too little communication" (i.e., would contradict the lower bound we wish to prove), we construct an $(r-1)$-round protocol on instances that are smaller or simpler by some other measure, by cleverly embedding the smaller instance in a larger one, sampling the first message of the r-round protocol *without looking at the input*, and then running the remaining rounds as usual. If this is done carefully, one can show that the resulting $(r-1)$-round protocol simulates the r-round protocol well, and therefore has small error. Eventually we eliminate all rounds, but still have a non-trivial problem which cannot be solved without communication, leading to a contradiction. This is the approach used in [13] to obtain a lower bound of $\Omega(\log\log k)$ on the number of rounds required to find a constant approximation to the maximum matching in a bipartite graph.

To our knowledge, we are the first to use the direct sum approach in order to show a round/communication tradeoff.

C. Organization

The remainder of the paper is organized as follows. In Section II we introduce basic notations and definitions. In Section III we describe the rounds/communication trade-off for set disjointness, omitting some proofs for lack of space. Finally, in Section IV we outline how the proof for set disjointness is modified to obtain a lower bound on approximate maximum matching.

II. Preliminaries

a) The shared blackboard model: We have k players, with private inputs $X_1, \ldots, X_k$. The goal of the players is to compute some function $f(X_1, \ldots, X_k)$ of their inputs. The players communicate over a *shared blackboard*: in each round of communication, each player may write a message on the board, and then all players observe all the messages written. After R rounds, the protocol halts, and the output is a function of the contents of the board. (In the problems we study here, the size of the output is small compared to the overall communication complexity, so it does not matter if we require the answer to be written on the board or not; we choose to require it here for simplicity.)

A slightly delicate point is whether players are *required* to write at least one bit in each round, or whether they can stay silent and not be charged. The latter allows player to convey one bit of information "for free" by their choice of whether to stay silent or not; we therefore adopt here the convention that each player must write at least one bit on the board in each round. This means that an R-round protocol has communication at least $R \cdot k$, but for our purposes, since we work with a $R = O(\log k)$ and show a lower bound of $\Omega(nk^{1/R}/R^3)$, this is not significant when n is large. We impose no upper bound on the number of bits a player can write in one round.

The protocols we consider are randomized, but we restrict the players to use only private random coins. Since our lower bound is information-theoretic, we can generate "public randomness" for free, by having some player write its private randomness on the board. This reveals no information about its input, but it may cost one additional round.

b) Notation: We use boldface letters to denote random variables. Subscripts indicate the coordinates of disjointness, and superscripts indicate the players; for example, X_j^i is the j-th coordinate of player i.

The messages sent in round r are denoted by $\mathbf{\Pi}_r$, a random variable which is a tuple of k messages, and player i's message in round r is denoted $\mathbf{\Pi}_r^i$. The transcript up to round r, inclusive, is denoted $\mathbf{\Pi}_{\leq r}$.

Because the notation can become cumbersome quickly, we adopt some short-hand notation. If $\mathbf{A}$ is a random variable, we will denote a concrete value to $\mathbf{A}$ by a, and we will also use a as short-hand notation for the event $\mathbf{A} = a$ where this is not confusing. In particular, an r-round partial transcript of the protocol will be denoted by $\pi_{\leq r}$, and we also use $\pi_{\leq r}$ as short-hand notation for the event that the transcript generated in the first r rounds was $\pi_{\leq r}$.

If μ is a distribution on inputs, then $\mu(\pi_{\leq r})$ denotes the probability that the partial transcript $\pi_{\leq r}$ will be generated when inputs are drawn from μ. (The protocol is fixed throughout.) We will also let $\mu|\pi_{\leq r}$ denote the distribution on the remainder of the transcript conditioned on the transcript so far being $\pi_{\leq r}$, and $\mu(\pi_{r+1}|\pi_{\leq r}), \mu(\pi_{r+1}^i|\pi_{\leq r})$ denote the probability that the next-round messages after round r will be π_{r+1} or that player i's message in round $r+1$ will be π_{r+1}^i, respectively.

c) Problem statements: In this paper we study the following problems.

- Disjointness: for $X^1, \ldots, X^k \in \{0,1\}^n$,

$$\textsc{Disj}_{n,k}(X^1, \ldots, X^k) = \neg \bigvee_{j=1}^{n} \left(\bigwedge_{i=1}^{k} X_j^i \right).$$

- Boolean AND: for $x^1, \ldots, x^k \in \{0,1\}$,
$$\textsc{And}_k(x^1, \ldots, x^k) = \bigwedge_{i=1}^{k} x^i.$$
A lower bound on AND is the main component in our lower bound for disjointness. We omit the subscript k when clear from the context.
- Approximate maximum matching / wellfare maximization with unit demands: the input is a bipartite graph on $k \times n$ nodes, with each player $i \in [k]$ receiving the edges adjacent to node i on the left side of the graph. The goal is to output a matching shose size is within $(1-\epsilon)$ of the size of the maximum matching in the graph. Note that the output is *the matching*, not the size of the matching.

d) Background on information theory: Our lower bound is based on *information complexity* [18] and follows the framework introduced in [6]. We require the following basic notions from information theory.

Definition 1 (Entropy and conditional entropy)**.** *The* entropy *of a random variable* $\mathbf{X} \sim \mu$ *with support* $\mathcal{X}$ *is given by*
$$H(\mathbf{X}) = \sum_{x \in \mathcal{X}} \Pr_{\mu}[\mathbf{X} = x] \log \frac{1}{\Pr_{\mu}[\mathbf{X} = x]}.$$
For two random variables $\mathbf{X}, \mathbf{Y}$ *with joint distribution* μ*, the* conditional entropy of $\mathbf{X}$ given $\mathbf{Y}$ *is*
$$H(\mathbf{X} \mid \mathbf{Y}) = \mathop{\mathrm{E}}_{\mathbf{y} \sim \mu(\mathbf{Y})} \sum_{x \in \mathcal{X}} \Pr_{\mu(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})}[\mathbf{X} = x] \log \frac{1}{\Pr_{\mu(\mathbf{X} \mid \mathbf{Y}=\mathbf{y})}[\mathbf{X} = x]}.$$

Definition 2 (KL divergence)**.** *Given two distributions* μ_1, μ_2 *with support* $\mathcal{X}$*, the* KL divergence *of* μ_1 *from* μ_2 *is*
$$\mathsf{D}\left(\frac{\mu_1}{\mu_2}\right) = \sum_{x \in \mathcal{X}} \mu_1(x) \log \frac{\mu_1(x)}{\mu_2(x)}.$$

Definition 3 (Mutual information and conditional mutual information)**.** *The* mutual information *between two random variables* $\mathbf{X}, \mathbf{Y}$ *is*
$$\mathrm{I}(\mathbf{X}; \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X} \mid \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \mathbf{X}).$$
The conditional mutual information between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ *is*
$$\begin{aligned}\mathrm{I}(\mathbf{X}; \mathbf{Y} \mid \mathbf{Z}) &= H(\mathbf{X} \mid \mathbf{Z}) - H(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) \\ &= H(\mathbf{Y} \mid \mathbf{Z}) - H(\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}).\end{aligned}$$

Mutual information and KL divergence are related as follows:
$$\begin{aligned}\mathrm{I}(\mathbf{X}; \mathbf{Y}) &= \mathsf{D}\left(\frac{\mu(\mathbf{X}, \mathbf{Y})}{\mu(\mathbf{X})\mu(\mathbf{Y})}\right) \\ &= \mathop{\mathrm{E}}_{\mathbf{y} \sim \mu(\mathbf{Y})} \mathsf{D}\left(\frac{\mu(\mathbf{X} \mid \mathbf{Y} = \mathbf{y})}{\mu(\mathbf{X})}\right) \\ &= \mathop{\mathrm{E}}_{\mathbf{x} \sim \mu(\mathbf{X})} \mathsf{D}\left(\frac{\mu(\mathbf{Y} \mid \mathbf{X} = \mathbf{x})}{\mu(\mathbf{Y})}\right),\end{aligned}$$
and similarly for conditional mutual information.

For convenience, if $\mu \sim \mathrm{Bernoulli}(p)$ and $\eta \sim \mathrm{Bernoulli}(q)$, we denote $\mathsf{D}\left(\frac{\mu}{\eta}\right)$ by $\mathsf{D}\left(\frac{p}{q}\right)$. We require the following three technical lemmas:

Lemma 1. *Let* $q \in (0, 1/3)$ *and* $\gamma \in (-1, 1/2)$*. Then*
$$\mathsf{D}\left(\frac{(1+\gamma)q}{q}\right) \geq \frac{1}{4 \ln 2} \cdot q\gamma^2.$$

Lemma 2. *Let* $q \in (0, 1/2)$ *and* $\gamma \geq 1.5$ *such that* $\gamma q \leq 1$*. Then*
$$\mathsf{D}\left(\frac{\gamma q}{q}\right) \geq q\gamma/10.$$

And finally,

Lemma 3. *Let* $p \geq 2q$ *and let* $\alpha > 1/2$*, such that* $\alpha q + (1 - \alpha p) \in (0,1)$*. Then*
$$\mathsf{D}\left(\frac{p}{\alpha q + (1-\alpha)p}\right) \geq p/(32 \ln 2).$$

III. Lower Bound for Set Disjointness

We describe first the disjointness lower bound, and then outline how it is adapted to obtain a lower bound on wellfare maximization.

For set disjointness we show the following lower bound:

Theorem 4. *Let* $R \cdot k \leq n \leq 2^{k^{1/R}}/8$*. Then any protocol that solves* $\textsc{Disj}_{n,k}$ *in* R *rounds with worst-case error at most* $1/100$ *requires* $\Omega(nk^{1/R}/(R^4 \log n))$ *bits of communication in the worst case.*

The reason we require $n \geq R \cdot k$ is that in our model, at least $R \cdot k$ bits are sent per round. We also require $n \leq 2^{k^{1/R}}/8$, but if n is larger than this, then $n \cdot k^{1/R} = O(n \log n)$, so the two-player lower bound of $\Omega(n)$ [1] is already within $\log n$ of optimal.

A. Information Complexity and Direct Sum

In order to bound the number of bits that players must communicate, we bound the information they must reveal about their inputs. We introduce a distribution on inputs — in our case, the distribution is of the form μ^n, where μ is a distribution on a single coordinate (that is, μ is on $\{0,1\}^k$), and μ^n is the product distribution where the coordinates are drawn iid from μ.

To quantify the information the players reveal about their inputs when they execute a protocol Π, we use *conditional information cost*, introduced in [6]:
$$\mathop{\mathrm{CIC}}_{\mu}(\Pi) = \mathop{\mathrm{I}}_{\mu}(\mathbf{\Pi}; \mathbf{X} \mid \mathbf{D}),$$
where $\mathbf{\Pi}$ is a random variable denoting the transcript of the protocol, $\mathbf{X}$ is the input to the players, and $\mathbf{D}$ is an auxiliary

variable also governed by the distribution μ, and used to break dependencies between the players.

Because $\mathrm{I}(\mathbf{A};\mathbf{B}|\mathbf{C}) \leq \mathrm{H}(\mathbf{A})$ for any three random variables $\mathbf{A},\mathbf{B},\mathbf{C}$, if we can bound the conditional information cost of the protocol from below, we obtain a bound on the entropy, and therefore the length, of its transcripts. Unlike communication, information cost is *additive*, which makes it useful in proving lower bounds.

For a problem P, we define the information complexity of P under μ with worst-case error δ:

$$\operatorname*{CIC}_{\mu,\delta}(P) = \inf_{\Pi}\left\{\operatorname*{CIC}_{\mu}(\Pi)\right\},$$

where the infimum is taken over all protocols that solve P with *worst-case error* at most δ on any input.

a) Direct sum: Following [6], the first step in our lower bound is to use a *direct sum* reduction to reduce the set disjointness lower bound to a lower bound on Boolean AND. The direct sum theorem of [6] asserts the following:

Theorem 5 (Direct sum [6])**.** *Let μ be an input distribution with an auxiliary variable $\mathbf{D}$, such that*

(1) The inputs $\mathbf{X}^1,\ldots,\mathbf{X}^k$ are independent conditioned on any value $\mathbf{D}=d$, and,

(2) $\Pr_\mu\left[\bigwedge_{i=1}^k \mathbf{X}^i = 1\right] = 0$.

Then:

$$\operatorname*{CIC}_{\mu^n,\delta}(\textsc{Disj}_{n,k}) \geq n \cdot \operatorname*{CIC}_{\mu,\delta}(\textsc{And}_k).$$

Condition (2) may appear odd, because it seems to make the problem easy (the answer is always 0). However, recall that we required small *worst-case* error, even on the input 1^k, which is not in the support. Thus, the protocol cannot "use" the fact that the answer is 0 under our distribution.

Importantly, the proof of the direct sum theorem is a reduction which *preserves the number of rounds*: given an R-round protocol for $\textsc{Disj}_{n,k}$, [6] constructs from it an R-round protocol for $\textsc{And}_n$ with $1/n$ the information cost. This allows us to use the reduction when showing a round/communication trade-off.

In our case we use a slight modification: our input distribution μ has $\Pr_\mu\left[\bigwedge_{i=1}^k \mathbf{X}^i = 1\right] \leq 1/n^2$, but the probability of 1^k is not zero. Nevertheless, when inputs to disjointness are drawn from μ^n, the probability that *any* coordinate will have an $\textsc{And}$ value of 1 is bounded by $1/n$, and therefore we can get:

$$\operatorname*{CIC}_{\mu^n,\delta}(\textsc{Disj}_{n,k}) \geq n \cdot \operatorname*{CIC}_{\mu,\delta+1/n}(\textsc{And}_k).$$

Since we assume that the error is a sufficiently small constant, the increase of $1/n$ is not significant.

We prove a lower bound of $\tilde{\Omega}(k^{1/R}/R^4)$ on $\mathrm{CIC}_{\mu,\delta+1/n}(\textsc{And}_k)$ with R rounds, and thereby obtain a lower bound of $\tilde{\Omega}(nk^{1/R}/R^4)$ on the communication complexity of $\textsc{Disj}_{n,k}$ with R rounds.

B. The Hard Distribution

We now describe the distribution μ we use to show the R-round lower bound on $\textsc{And}_k$.

The distribution μ is composed of $R{+}1$ "slices" $\mu_0,\ldots,\mu_R$. In each slice μ_s, the inputs to the players are iid Bernoulli random variables, with probability p_s of being zero. We choose p_s such that in slice s, typically $\Theta(k^{1-s/R})$ players get zero inputs: in the first slice, $s=0$, we set $p_s = 1/2$, and for the other slices, $s=1,\ldots,R$, we set

$$p_s = 2k^{-s/R}\log n.$$

specifically, in slices $s>1$, the inputs to all the players are iid Bernoulli variables with probability $2k^{s/R}\log n$ of being zero, and in slice 0 the inputs are iid Bernoulli $1/2$.

The overall input distribution μ is generated by choosing a slice $\mathbf{S}$ uniformly at random, and drawing the input $\mathbf{X}$ from $\mu_{\mathbf{S}}$.

Notice that the inputs are independent given the slice $\mathbf{S}$; we use $\mathbf{S}$ as the dependence-breaking variable $\mathbf{D}$ we require for the direct sum theorem.

a) Intuition for the choice of the distribution μ: Intuitively, the goal of the protocol is to expose some player whose input is zero, in order to become convinced that the answer to $\textsc{And}$ is zero. To see why our input distribution is hard, consider a simple protocol where in each round r we have some fixed probability q_r, and each player that got zero announces this fact with probability q_r (and otherwise stays silent). We would like to set q_r as high as possible, in order to find a zero quickly. How high can we afford to set q_r while respecting an overall communication budget of $O(k^{1/R}/\operatorname{poly}(R))$?

In the first round, we do not know which slice the input was drawn from; with probability $1/R$ it was drawn from μ_0, where $\Theta(k)$ players get zero. Therefore, if we are aiming for total communication (or information) $O(k^{1/R}/\operatorname{poly}(R))$, we cannot set q_1 greater than $\Theta(k^{1/R-1}/\operatorname{poly}(R))$. However, this means that if we are *not* in slice μ_0, but rather in some higher slice, then with good probability no zeroes will be announced: the total number of zeroes in this case is probably $O(k^{1-1/R}/\operatorname{poly}(R))$ (ignoring the $\log n$ factor for the time being).

After the first round, if the input was not drawn from slice μ_0, we have now learned this fact, because no zeroes were announced. But we made very little progress on the remaining slices, and in particular, it is unlikely that a zero was found, and we still do not know which slice the input is drawn from.

In the second round and onwards the picture is similar. In round r, slices $\mu_{r-1},\ldots,\mu_R$ all remain possible, and we cannot afford to set q_r greater than $\Theta(k^{r/R-1}/\operatorname{poly}(R))$, otherwise we would use more than $O(k^{1/R}/\operatorname{poly}(R))$ communication / information if the input is drawn from μ_{r-1}. We therefore expect that on slices higher than μ_{r-1} no zeroes will be announced, and we will have learned very little about the input.

Finally, after R rounds, we are still "in the game" only if the input is drawn from μ_R; if the input was drawn from some lower slice, we probably have found a zero already. However, if the input is drawn from μ_R, we know almost nothing about it. In particular the protocol cannot distinguish μ_R from the input 1^k, and this is a problem, because the answer on those two cases is almost certainly different: on 1^k the correct output is 1, but on slice μ_R the correct output is almost certainly 0.

b) The "all-one" distribution, ν: In order to compare the behavior of the protocol on μ, where the correct answer is almost certainly zero, to its behavior on the all-one input, let us denote by ν the "input distribution" where all players get one. We also think of μ, the slices μ_s, and ν as distributions on *transcripts* of the protocol, obtained by drawing an input from the respective distribution and then generating a transcript of the protocol on that input.

In our proof we measure the probability of "good events" under ν, the all-one input. When we say, e.g., "we are likely to reach a transcript satisfying…", we are referring to the transcript distribution under ν.

We show that with high probability (under ν), after $s \leq R$ rounds, the transcript $\pi_{\leq s}$ generated so far has $\mu_r(\pi_{\leq s}) \approx \nu(\pi_{\leq s})$ for any slice $r \geq s$. In other words, the transcript is roughly as likely to be generated when the input is drawn from any slice μ_r for $r \geq s$ as it is when all players get 1. This formalizes our intuition that for "surviving slices" $r \geq s$, we have learned very little about the input. In particular, under ν, after R rounds we are very likely to reach a transcript on which the output is 1; therefore, on μ_R, which "survives" all R rounds, we are also likely to reach a transcript on which the output is 1, so the protocol errs with high probability.

To show this, we charge the difference between $\mu_r(\pi_{\leq s})$ and $\nu(\pi_{\leq s})$ to the information cost of the protocol. Because we are working with product distributions, the information revealed by the protocol is the sum of the information each player reveals about its own input; we show that when a player's message distribution "deviates significantly" under μ_s and ν, the player's contribution to the information cost is large, and therefore most players' messages have similar probabilities under μ_s and ν.

Let us now make this intuition more precise.

C. Measuring the Protocol's Progress

Suppose that after s rounds s the protocol's transcript is $\pi_{\leq s}$. We measure the "information" that player i has revealed about its input in $\pi_{\leq s}$ by comparing the likelihood of the transcript $\pi_{\leq s}$ is when player i has zero, to its likelihood when player i has one. As long as the two are close to each other, we "know only a little" about the input of player i.

More formally, for a player i, a partial transcript $\pi_{\leq s} = \pi_1, \ldots, \pi_s$, and a round $r \leq s$, let

$$\lambda_r^i(\pi_{\leq s}) := \frac{\Pr\left[\pi_r^i \mid \pi_{<r}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_r^i \mid \pi_{<r}, \mathbf{X}^i = 1\right]},$$

and let

$$\Lambda_s^i(\pi_{\leq s}) := \sum_{r=1}^{s} \log \lambda_r^i(\pi_{\leq s})$$

$$= \sum_{r=1}^{s} \log \frac{\Pr\left[\pi_r^i \mid \pi_{<r}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_r^i \mid \pi_{<r}, \mathbf{X}^i = 1\right]}.$$

We often omit the transcript $\pi_{\leq s}$ and simply write λ_t^i, Λ_s^i when the partial transcript is clear from the context.

Here, $\Pr\left[\pi_r \mid \pi_{<r}, \mathbf{X}^i = b\right]$ denotes the probability under the protocol Π that player i sends its round-r message π_r^i in $\pi_{\leq s}$, given that the transcript up to round r is $\pi_{<r}$ and its input is b. (This probability does not depend on the input distribution, only the protocol.) For any slice μ_r (and indeed, for any product distribution on inputs) we have:

Property 1.

$$\frac{\mu_r(\mathbf{X}^i = 0|\pi_{\leq s})}{\mu_r(\mathbf{X}^i = 1|\pi_{\leq s})} = \frac{\mu_r(\pi_{\leq s}|\mathbf{X}^i = 0)}{\mu_r(\pi_{\leq s}|\mathbf{X}^i = 1)} = 2^{\Lambda_s^i}.$$

Thus, if we can show that $\Lambda_s^i(\pi_{\leq s})$ is small in absolute value, we know that on all slices, player i's input is close to its prior, even conditioned on the transcript $\pi_{\leq s}$. Equivalently, we know that the probability of transcript $\pi_{\leq s}$ being generated under slice r (or under ν) is not sensitive to player i's input.

D. Good Players and Transcripts

Even though the average player cannot afford to "say a lot" about its input, a small number of players might do so. We call such players *bad players*, and we discard them from our argument (by essentially fixing their inputs to 1). Players that have not revealed a lot of information about their input are called *good*.

Definition 4 (Good players). *We say that player i is* good in $\pi_{\leq s}$ *if for each prefix $\pi_{\leq t}$, $t \leq s$, we have*

$$\left|\log \lambda_t^i(\pi_{\leq s})\right| \leq \frac{1}{2R}.$$

Let $G(\pi_{\leq s}) \subseteq [k]$ denote the set of players that are good in $\pi_{\leq s}$.

In our induction, we control the number of bad players, and show that with high probability it remains $O(k^{s/R}/\operatorname{poly}(R))$ after s rounds, by "charging" each good player that becomes bad to the information cost of the protocol.

Unfortunately, it is not enough to control the number of bad players: even though each good player only revealed a little information, together it can add up to too much. We need to show that the *total* progress made by the protocol is roughly $k^{s/R}$, that is, all good players *together* have revealed roughly that much information.

We refer to $\left|\sum_{i \in G(\pi_{\leq s})} \Lambda_s^i\right|$ as the *total badness* of the transcript. A transcript that satisfies both conditions — a small number of bad players, and low total badness — is called *good*.

Definition 5 (Good transcripts). *We say that a partial transcript $\pi_{\leq s}$ is* good *if:*
(1) At most $k^{s/R}/(8 \log n)$ players are bad in $\pi_{\leq s}$, that is,

$$|[k] \setminus G(\pi_{\leq s})| \leq \frac{k^{s/R}}{8 \log n},$$

and

(2) The sum *of the Λ_s^i for the good players is bounded in absolute value:*

$$\left|\sum_{i\in G(\pi_{\le s})}\Lambda_s^i\right| \le \frac{k^{s/R}}{2\log n}.$$

We denote by $\mathcal{G}_{\le s}$ the set of good s-round transcripts. Our goal will be to show that $\mathcal{G}_{\le R}$ has large probability under ν. Also, let $\mathcal{G}_{\le s}^{(1)} \supseteq \mathcal{G}_{\le s}$ and $\mathcal{G}_{\le s}^{(2)} \supseteq \mathcal{G}_{\le s}$ denote the set of transcripts that satisfy conditions (1) and (2), respectively, in the definition of a good transcript.

E. Properties of Good Transcripts

Condition (2) in the definition of a good transcript, which controls $\left|\sum_{i\in G(\pi_{\le s})}\Lambda_s^i\right|$, allows us to relate the probability of good transcripts under ν and under μ_s, as follows:

Lemma 6. *If $\pi_{\le s}$ is good, then $\nu(\pi_{\le s}) \le 4\mu_s(\pi_{\le s})$.*

For good players we also get a relationship in the other direction on their next message:

Lemma 7. *If $\pi_{\le s}$ is good, and i is a good player in $\pi_{\le s}$, then $\mu_s(\pi_{s+1}^i|\pi_{\le s}) \ge \nu(\pi_{s+1}^i|\pi_{\le s})/2$.*

We use this connection to relate the expected divergence for good players under $\nu|\pi_{\le s}$ to that under $\mu_s|\pi_{\le s}$, as follows:

Corollary 8. *If $\pi_{\le s}$ is good, then*

$$\begin{aligned}&\mathop{\mathrm{E}}_{\mathbf{\Pi}_{s+1}\sim\nu|\pi_{\le s}}\left[\sum_{i\in G(\pi_{\le s})}\mathsf{D}\left(\frac{\mu_s(\mathbf{X}^i|\pi_{\le s},\mathbf{\Pi}_{s+1}^i}{\mu_s(\mathbf{X}^i|\pi_{\le s}}\right)\right]\\ &\le 2\mathop{\mathrm{E}}_{\mathbf{\Pi}_{s+1}\sim\mu_s|\pi_{\le s}}\left[\sum_{i\in G(\pi_{\le s})}\mathsf{D}\left(\frac{\mu_s(\mathbf{X}^i|\pi_{\le s},\mathbf{\Pi}_{s+1}^i}{\mu_s(\mathbf{X}^i|\pi_{\le s}}\right)\right].\end{aligned}$$

Corollary 8 will be useful to us because as we said, we analyze the behavior of the protocol under ν; in particular we want to bound from below the probability that the transcript remains good after s rounds, $\nu(\mathcal{G}_{\le s})$. However, the information cost of the protocol is computed under the distribution μ, which is a convex combination of the slices μ_s, not under ν. Corollary 8 allows us to translate between the behavior of the protocol under μ_s and its behavior under ν.

Finally, since we will be interested in showing that most good players remain good, let us define:

Definition 6 (Breaking bad)**.** *We say that player i* breaks bad *in round r if $i \in G(\pi_{<r})$ but $i \notin G(\pi_{\le r})$, that is, i was good up to round r but became bad afterwards. Let $\mathbf{B}_r^i$ be an indicator for this event.*

F. Analyzing the Behavior of Good Players

Recall that our high-level intuition is that in round $s+1$, not too many players can afford to announce that their input is zero, otherwise the information cost of the protocol would be too high on slice s. We would like to translate this intuition into a bound on the number of good players that go bad in round $s+1$: "announcing a zero" corresponds to becoming bad by revealing too much information. If player i was good in $\pi_{\le s}$ but becomes bad in $\pi_{\le s+1}$, we would like to "charge" this to the information revealed by the protocol in round $s+1$ under slice s, by showing that the ratio between $\Pr\left[\pi_{s+1}^i \mid \pi_{\le s}, \mathbf{X}^i = 0\right]$ and $\mu_s\left(\pi_{s+1}^i|\pi_{\le s}\right)$ is $\mathrm{poly}(1/R)$. (That is, there is a noticeable difference between the probability that player i sends π_{s+1}^i when its input is 0, compared to the prior probability under μ_s that player i sends π_{s+1}^i in round $s+1$, given that the transcript so far is $\pi_{\le s}$.)

A small technical obstacle stands in our way: the definition of good players says nothing about slice s, and instead measures the ratio between the probability when the player's input is 0 and when it is 1 (because ultimately, we want to compare against the all-one input assignment). We overcome this as follows.

Similar to the definition of λ_r^i and Λ_s^i, we define a "transcript probability quotient" under μ_s, as follows:

$$\gamma_{s,r}^i(\pi_{\le s}) := \frac{\Pr\left[\pi_r^i \mid \pi_{<r}, \mathbf{X}^i = 0\right]}{\mu_s\left(\pi_r^i|\pi_{<r}\right)},$$

and

$$\Gamma_s^i := \sum_{r=1}^{s}\log\gamma_{s,r}^i.$$

Note that s has two meanings here: it is both the number of rounds in the transcript $\pi_{\le s}$ with respect to which Γ_s^i is defined, and the number of the slice μ_s with respect to which we measure the difference in the probability of $\pi_{\le s}$. This is no coincidence, as each slice μ_s is used to argue good behavior in round $s+1$ of the protocol, and to do this we need to control the behavior up to and including round s. After round $s+1$, we no longer use slice s in the argument, because the players may very well have discovered whether or not the input was drawn from this slice.

By analogy to Property 1 for Λ_s^i, the quantity Γ_s^i also measures sensitivity to player i's input, this time in the following sense:

Property 2. We can think of Γ_s^i as measuring the effect that player i's input has on the probability that $\pi_{\le s}$ will be generated:

$$\begin{aligned}\frac{\mu_s\left(\pi_{\le s}|\mathbf{X}^i=0\right)}{\mu_s\left(\pi_{\le s}\right)} &= \frac{\prod_{t=1}^s\prod_{j=1}^k\Pr\left[\pi_t^i \mid \pi_{<t}, \mathbf{X}^i=0\right]}{\prod_{t=1}^s\prod_{j=1}^k\mu_s\left(\pi_t^j|\pi_{<t}\right)}\\ &= \prod_{t=1}^s\frac{\Pr\left[\pi_t^i \mid \pi_{<t}, \mathbf{X}^i=0\right]}{\mu_s\left(\pi_t^i|\pi_{<t}\right)} = 2^{\Gamma_s^i}.\end{aligned}$$

This is because the inputs to the players are independent under μ_s, and hence the other players' messages are independent of player i's input given the transcript so far. Also, by Bayes,

$$\frac{\Pr_{\mu_s|\pi_{\le s}}\left[\mathbf{X}^i=0\right]}{\Pr_{\mu_s}\left[\mathbf{X}^i=0\right]} = \frac{\mu_s\left(\pi_{\le s}|\mathbf{X}^i=0\right)}{\mu_s\left(\pi_{\le s}\right)} = 2^{\Gamma_s^i}. \quad \square$$

By keeping track of $|\log\lambda_1^i|, \ldots, |\log\lambda_s^i|$ and of Λ_s^i in the definition of good players and transcripts, we also implicitly control $|\log\gamma_1^i|, \ldots, |\log\gamma_s^i|$ and Γ_s^i:

Lemma 9. *For any* $s, r \leq R$, *we have* $\operatorname{sgn}\left(\log\gamma_{s,r}^i\right) = \operatorname{sgn}\left(\log\lambda_r^i\right)$ *and* $\left|\log\gamma_{s,r}^i\right| \leq \left|\log\lambda_r^i\right|$.

We also get that for good transcripts, the posterior probability of a good player's input being zero is not much different than the prior:

Corollary 10. *If* $i \in G(\pi_{\leq s})$, *then*

$$\frac{1}{2} \leq \frac{\Pr_{\mu_s|\pi_{\leq s}}\left[\mathbf{X}^i = 0\right]}{\Pr_{\mu_s}\left[\mathbf{X}^i = 0\right]} \leq 2.$$

In particular, for a good player, the posterior probability that the player's input is zero never exceeds 1/2:

Corollary 11. *If* $i \in G(\pi_{\leq s})$, *then* $\Pr_{\mu_s|\pi_{\leq s}}\left[\mathbf{X}^i = 0\right] \leq 1/2$.

In the induction step we will need to argue that for a good player i, if $\left|\log\gamma_{s,s+1}^i\right|$ is small, then so is $\left|\log\lambda_{s+1}^i\right|$. And indeed, when either of the two quantities is not too large, they are close to each other:

Lemma 12. *If player* i *is good in* $\pi_{\leq s}$, *and in addition we have either* $\left|\log\gamma_{s,s+1}^i\right| \leq 1$ *or* $\left|\log\lambda_{s+1}^i\right| \leq 1$, *then* $\left|\log\lambda_{s+1}^i\right| \leq 4\left|\log\gamma_{s,s+1}^i\right|$.

G. The Information Cost of Players that Break Bad

As we explained informally above, in order to establish that condition (1) in the definition of a good transcript holds w.h.p., we show that each player that breaks bad in round $s+1$ contributes noticeably to the information cost of round $s+1$ on slice μ_s.

First, we show that good players that incur large $\gamma_{s,s+1}^i$ contribute significantly to the divergence; since $\gamma_{s,s+1}^i$ and λ_{s+1}^i are related as we saw above, this will allow us to bound the number of players that break bad. (We made no attempt to optimize the constants in the proof.)

Lemma 13. *If* $i \in G(\pi_{\leq s})$ *and* $|\log\gamma_{s,s+1}^i| \geq 1/(8R)$ *then*

$$\mathsf{D}\left(\frac{\mu_s(\mathbf{X}^i|\pi_{\leq s+1})}{\mu_s(\mathbf{X}^i|\pi_{\leq s})}\right) \geq k^{-s/R}/(800R^2).$$

Proof sketch. Let $p = \Pr_{\mu_s|\pi_{\leq s}}\left[\mathbf{X}^i = 0\right]$ and let us abbreviate $\gamma = \gamma_{s,s+1}^i$. By Bayes,

$$\begin{aligned}
&\Pr_{\mu_s|\pi_{\leq s+1}}\left[\mathbf{X}^i = 0\right] = \\
&= \frac{\mu_s\left(\pi_{s+1}|\pi_{\leq s}, \mathbf{X}^i = 0\right)\Pr_{\mu_s|\pi_{\leq s}}\left[\mathbf{X}^i = 0\right]}{\mu_s\left(\pi_{s+1} \mid \pi_{\leq s}\right)} \\
&= p \cdot \gamma = p(1 + (\gamma - 1)).
\end{aligned}$$

If $\gamma \geq 1.5$, then by Lemma 2

$$\mathsf{D}\left(\frac{p(1+(\gamma-1))}{p}\right) \geq p\gamma/10.$$

Recall that under each slice, the prior probability of getting 0 is at least $k^{-s/R}/2$, and thus by Corollary 10 we have $p \geq k^{-s/R}/4$. Since $R \geq 1$,

$$\mathsf{D}\left(\frac{p(1+(\gamma-1))}{p}\right) \geq \frac{k^{-s/R}}{4} \cdot \frac{1.5}{10} > \frac{k^{-s/R}}{40R^2}.$$

On the other hand, if $\gamma < 1.5$, then by Lemma 1,

$$\begin{aligned}
\mathsf{D}\left(\frac{p(1+(\gamma-1))}{p}\right) &\geq \frac{1}{4\ln 2}p \cdot (\gamma - 1)^2 \\
&\geq \frac{1}{6}\frac{k^{-s/R}}{4}(\gamma - 1)^2.
\end{aligned}$$

Using the fact that $|\log\gamma| \geq 1/(8R)$ (i.e., γ is not too close to 1), we can bound this from below by

$$\mathsf{D}\left(\frac{p(1+(\gamma-1))}{p}\right) \geq \frac{k^{-s/R}}{800R^2}.$$

□

For good players that have small $\left|\log\lambda_{s+1}^i\right|$ we can get a bound directly in terms of $\log^2\lambda_{s+1}^i$ and of $(\lambda_{s+1}^i - 1)^2$:

Lemma 14. *If* $i \in G(\pi_{\leq s+1})$, *then*

$$\begin{aligned}
\mathsf{D}\left(\frac{\mu_s(\mathbf{X}^i|\pi_{\leq s+1})}{\mu_s(\mathbf{X}^i|\pi_{\leq s})}\right) &\geq (k^{-s/R}/800)\log^2\lambda_{s+1}^i \\
&\geq (k^{-s/R}\ln^2 2/800)\left(\lambda_{s+1}^i - 1\right)^2.
\end{aligned}$$

Recall that the information cost incurred by round $s+1$ of the protocol is the *expectation* of the divergence from Lemma 13. Thus, as a corollary of Lemma 13, and using Corollary 8, we get:

Lemma 15. *Let* $\pi_{\leq s}$ *be a good transcript. Then*

$$\mathop{\mathrm{E}}_{\mathbf{\Pi}_{s+1}\sim\nu|\pi_{\leq s}}\left[\sum_i \mathbf{B}_{s+1}^i\right] \leq 1600R^2k^{s/R}\mathop{\mathrm{I}}_{\mu_s|\pi_{\leq s}}(\mathbf{\Pi}_{s+1}; \mathbf{X}).$$

Finally, if "most" partial transcript up to round s are good, then conditioning on $\mathcal{G}_{\leq s}$, being good up to round s, does not increase the information cost of round $s+1$ by much. Using this fact, and using Lemma 6 to relate expectations under ν and under μ_s, we get:

Corollary 16. *If* $\nu(\mathcal{G}_{\leq s}) \geq 1/2$ *(where* $\mathcal{G}_{\leq s}$ *is the set of good* s*-round transcripts), then*

$$\mathop{\mathrm{E}}_{\nu|\mathcal{G}_{\leq s}}\left[\sum_i \mathbf{B}_{s+1}^i\right] \leq O(R^2k^{s/R}\mathop{\mathrm{CIC}}_{\mu}(\Pi)) \leq O\left(\frac{k^{(s+1)/R}}{R\log n}\right).$$

Thus, by Markov, if the transcript up to round s is good, then the first condition of being a good transcript (Definition 5) continues to hold in round $s+1$ with high probability. Next we turn our attention to condition (2).

H. Bounding the Total Badness

a) High-level overview: Recall that the total badness of a transcript $\pi_{\leq s}$ is defined as $\left|\sum_{i \in G(\pi_{\leq s})} \Lambda_s^i\right|$, where $\Lambda_s^i(\pi_{\leq s}) = \sum_{r=1}^{s} \log \lambda_r^i(\pi_{\leq s})$.

In order to bound the total badness and maintain the second condition of Definition 5, we need a more fine-grained connection between the information revealed by player i in message Π_{s+1}^i, and its contribution to the increment in the total badness, $\Delta_{s+1}^i - \Delta_s^i = \log \lambda_{s+1}^i$. In Lemma 13, we showed (roughly) that if the increment is at least $\Omega(1/R)$ then the information revealed is at least $\Omega(1/R^2)$, but now we need to charge $\log \lambda_{s+1}^i$ directly to the divergence, with no minimum threshold. We can only do this when $\log \lambda_{s+1}^i$ is small to start with, which is why we throw out bad players.

Even then, for a specific message π_{s+1}^i, the true relationship between $\log \lambda_{s+1}^i$ and the divergence can be, in the worst case, *quadratic*, matching the bound we showed in Lemma 13. If we proceed carelessly here, we could incur a quadratic blow-up throughout, yielding a lower bound of $\Omega(k^{1/(2R)}/\operatorname{poly}(R))$ instead of $\Omega(k^{1/R}/\operatorname{poly}(R))$ (which is significant when R is constant).

Instead of considering an individual, specific message π_{s+1}^i and relating the "damage" it causes (the increment in $\left|\Lambda^i\right|$) to the divergence of $\mu_s(\mathbf{X}^i | \pi_{\leq s+1})$ from $\mu_s(\mathbf{X}^i | \pi_{\leq s})$, we consider the *expectation* over the next-round messages. The idea is that the increment caused by the next-round message $\mathbf{\Pi}_{s+1}^i$ is positive for some messages π_{s+1}^i and negative for others, and we can show that the first-order term cancels out. Indeed, since $\log x \approx x - 1 - (x-1)^2/2$ and $\log^2 x \approx (x-1)^2/2$ for x close to 1 (with a $\ln 2$ factor which we are going to ignore here), for good players we would ideally get:

$$\begin{aligned}
&\left| \mathop{\mathrm{E}}_{\pi_{s+1}^i \sim \nu | \pi_{\leq s}} \left[\log \frac{\Pr\left[\pi_{s+1}^i \mid \pi_{\leq s}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right]} \right] \right| \\
&\approx \Bigg| \mathop{\mathrm{E}}_{\pi_{s+1}^i \sim \nu | \pi_{\leq s}} \Bigg[\frac{\Pr\left[\pi_{s+1}^i \mid \pi_{\leq s}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right]} - \\
&\qquad -1 - \frac{1}{2} \left(\frac{\Pr\left[\pi_{s+1}^i \mid \pi_{\leq s}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right]} - 1 \right)^2 \Bigg] \Bigg| \\
&\approx \Bigg| \sum_{\pi_{s+1}^i} \left(\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right] \cdot \frac{\Pr\left[\pi_{s+1}^i \mid \pi_{\leq s}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right]} \right) \\
&\qquad - \sum_{\pi_{s+1}^i} \Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right] \\
&\qquad - \frac{1}{2} \mathop{\mathrm{E}}_{\pi_{s+1}^i \sim \nu | \pi_{\leq s}} \left[\log^2 \frac{\Pr\left[\pi_{s+1}^i \mid \pi_{\leq s}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right]} \right] \Bigg| \\
&= \left| 1 - 1 - \frac{1}{2} \mathop{\mathrm{E}}_{\pi_{s+1}^i \sim \nu | \pi_{\leq s}} \left[\log^2 \frac{\Pr\left[\pi_{s+1}^i \mid \pi_{\leq s}, \mathbf{X}^i = 0\right]}{\Pr\left[\pi_{s+i}^i \mid \pi_{\leq s}, \mathbf{X}^i = 1\right]} \right] \right| \\
&= \frac{1}{4} \left| \mathop{\mathrm{E}}_{\pi_{s+1}^i \sim \nu | \pi_{\leq s}} \left[\log^2 \lambda_{s+1}^i \right] \right|. \qquad (1)
\end{aligned}$$

Now that we have cancelled out the first-order terms, we can use Lemma 14 to relate $\log^2 \lambda_{s+1}^i$ to the divergence player i contributes, and thereby relate the expected increment in Λ_s^i to player i's contribution to the information cost.

We say that this computation is what we would "ideally" do, because it is not quite true: a player that is good in $\pi_{\leq s}$ can have a non-zero probability of turning bad in round $s+1$, in which case we can no longer apply the approximation of the log we used in (1), as the log is not close to 1. Nevertheless, using the fact that under ν good players have only a small probability of turning bad, we are able to bound the expectation in terms of the divergence, and also show concentration for the sum over all good players. That is, for good players we show that with high probability,

$$\left| \sum_{\text{good players}} \left(\Lambda_{s+1}^i - \Lambda_s^i \right) \right| \leq k^{s+1}/\operatorname{poly}(R),$$

so that condition (2) in the definition of a good transcript is maintained.

b) A more detailed proof sketch: In order not to charge for bad players when we bound the total badness, let us define:

$$L^i(\pi_{s+1}^i) := \begin{cases} \log \lambda_{s+1}^i & \text{if player } i \text{ is good in } \pi_{\leq s+1}, \\ 0 & \text{otherwise.} \end{cases}$$

As always, we omit the message π_{s+1}^i from our notation when clear from the context.

With this notation, if $\pi_{\leq s}$ is good, then in order to show that $\pi_{\leq s+1}$ satisfies condition (2) of Definition 5, it suffices to show that

$$\left| \sum_{i=1}^{k} L^i \right| \leq \frac{k^{(s+1)/R}}{4 \log n}.$$

We then have:

$$\begin{aligned}
\left| \sum_{i \in G(\pi_{\leq s+1})} \Lambda_{s+1}^i \right| &\leq \left| \sum_{i \in G(\pi_{\leq s})} \Lambda_s^i \right| + \left| \sum_{i \in G(\pi_{\leq s+1})} \log \lambda_{s+1}^i \right| \\
&\leq \frac{k^{s/R}}{2 \log n} + \left| \sum_{i=1}^{k} L^i \right| \\
&\leq \frac{k^{s/R}}{2 \log n} + \frac{k^{(s+1)/R}}{4 \log n} \leq \frac{k^{(s+1)/R}}{2 \log n},
\end{aligned}$$

as required.

Let us relate the expected value of L^i to the information revealed by player i in round $s+1$.

As we explained above, our calculations are complicated by the fact that players break bad with non-zero probability, and when they do we cannot control their contribution, λ_{s+1}^i, by charging it against the information cost. To handle this, define

$$P_0^i(\pi_{\leq s}) = \Pr\left[i \notin G(\mathbf{\Pi}_{\leq s+1}) \mid \pi_{\leq s}, \mathbf{X}^i = 0\right],$$

and

$$P_1^i(\pi_{\leq s}) = \Pr\left[i \notin G(\mathbf{\Pi}_{\leq s+1}) \mid \pi_{\leq s}, \mathbf{X}^i = 1\right].$$

As usual we omit the transcript and write simply P_0^i, P_1^i where reasonable.

Now we can bound a player's expected contribution L^i as follows:

Lemma 17. *If $\pi_{\le s}$ is good and $i \in G(\pi_{\le s})$, then*

$$\begin{aligned}&\left|\mathop{\mathrm{E}}_{\mathbf{\Pi}^i_{s+1}\sim\nu|\pi_{\le s}}\left[\mathbf{L}^i\right]\right|\\ &\le 2\left((6400k^{s/R}\log n)\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{\Pi}^i_{s+1};\mathbf{X}^i\right)+P_0^i+P_1^i\right).\end{aligned}$$

The lemma is proven by repeating the "idealized" calculation we did in (1), but this time taking into consideration the event that player i breaks bad.

Summing across all players, we obtain:

Corollary 18. *We have*

$$\begin{aligned}&\left|\mathop{\mathrm{E}}_{\nu|\pi_{\le s}}\left[\sum_{i=1}^{k}\mathbf{L}^i\right]\right|\\ &\le 2\left(6400\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{\Pi}_{s+1};\mathbf{X}\right)+\sum_{i\in G(\pi_{\le s})}\left(P_0^i+P_1^i\right)\right).\end{aligned}$$

We see that in order to control the increment to $\sum_i \Lambda^i$, we must bound the sum of the probabilities that the good players go bad when their inputs are 0, and the same sum when their inputs are 1.

One of the two is easy: by definition,

$$P_1^i = \mathop{\mathrm{Pr}}_{\nu|\pi_{\le s}}\left[\mathbf{B}^i = 1\right],$$

that is, P_1^i is the probability that i breaks bad under ν. Corollary 16 gives us a bound on the sum of the P_1^i for good players. However, we do not have a bound on the P_0^i's, and indeed, since the probability that $\mathbf{X}^i = 0$ is very low under μ_s, we cannot bound the P_0^i's by making a similar argument under the slice μ_s.

To bound $\sum_i P_0^i$, we show that whenever P_0^i is large compared to P_1^i, player i contributes to the information cost in proportion to P_0^i. Intuitively, if P_1^i is small, but P_0^i is large, then somehow player i gives a lot of information about its input: when the input is 0 player i is likely to break bad, but when the input is 1 player i is not. An external observer *knows* if a given player is bad or not, because this is only a function of the transcript. Therefore, in this scenario, player i reveals by the fact that it broke bad that its input was probably 0.

This intuition is made formal as follows.

Lemma 19. *If $\pi_{\le s}$ is good, $i \in G(\pi_{\le s})$, and $P_0^i \ge 2P_1^i$, then*

$$\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X}^i;\mathbf{\Pi}^i_{s+1}\right) \ge \frac{k^{-s/R}}{64\ln 2}\cdot P_0^i.$$

Proof. Recall that $P_0^i = \Pr\left[\mathbf{B}^i = 1 \mid \pi_{\le s}, \mathbf{X}^i = 0\right]$. By the data-processing inequality, since $\mathbf{B}^i$ is a function of $\mathbf{\Pi}^i_{s+1}$ (given $\mathbf{\Pi}_{\le s} = \pi_{\le s}$),

$$\begin{aligned}\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X}^i;\mathbf{\Pi}^i_{s+1}\right) &\ge \mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X}^i;\mathbf{B}^i\right)\\ &\ge \mathop{\mathrm{Pr}}_{\mu_s|\pi_{\le s}}\left[\mathbf{X}^i = 0\right]\mathsf{D}\left(\begin{array}{c}\mathbf{B}^i|\mathbf{X}^i=0\\ \hline\hline \mathbf{B}^i\end{array}\right)\\ &\ge \frac{k^{-s/R}}{2}\mathsf{D}\left(\begin{array}{c}\mathbf{B}^i|\mathbf{X}^i=0\\ \hline\hline \mathbf{B}^i\end{array}\right).\end{aligned}$$

In the last step we used the fact that i is good, so by Corollary 10, the probability under $\mu_s|\pi_{\le s}$ that i gets zero is at least half the prior under μ_s, which for $s \ge 1$ is at least $k^{-s/R}\log n$. For slice 0, $\mu_0|\pi_{\le 0} = \mu_0$, and the probability that player i gets 0 is $1/2$, which satisfies $1/2 \ge k^{-0/R}/2$.

Now let us relate the divergence to P_0^i, using Lemma 3. By definition, $\mathbf{B}^i|\mathbf{X}^i = 0$ is Bernoulli with probability P_0^i of being 1, while $\mathbf{B}^i$ is Bernoulli with probability $qP_1^i+(1-q)P_0^i$, where $q = \Pr_{\mu_s|\pi_{\le s}}\left[\mathbf{X}^i = 1\right] \ge 1/2$ (by Corollary 11). By Lemma 3,

$$\mathsf{D}\left(\begin{array}{c}\mathbf{B}^i|\mathbf{X}^i=0\\ \hline\hline \mathbf{B}^i\end{array}\right) \ge \frac{P_0^i}{32\ln 2}.$$

The claim follows. □

Now we can use this bound, together with Corollary 18, to get the bound we wanted, relating the sum of the L^i to the information cost.

Corollary 20. *If $\pi_{\le s}$ is good,*

$$\left|\mathop{\mathrm{E}}_{\nu|\pi_{\le s}}\left[\sum_{i=1}^{k}\mathbf{L}^i\right]\right| \le 20,000\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{\Pi}_{s+1};\mathbf{X}\right).$$

Proof. Let $A \subseteq G(\pi_{\le s})$ be the set of good players who have $P_0^i < 2P_1^i$, and let $B = G(\pi_{\le s}) \setminus A$ be the good players who have $P_0^i \ge 2P_1^i$. We bound the contribution of the terms $P_0^i + P_1^i$ to the absolute value of the expectation as follows:

$$\begin{aligned}&\sum_{i\in G(\pi_{\le s})}\left(P_0^i+P_1^i\right) = \sum_{i\in A}\left(P_0^i+P_1^i\right)+\sum_{i\in B}\left(P_0^i+P_1^i\right)\\ &\le 3\sum_{i\in A}P_1^i+\sum_{i\in B}P_1^i+45k^{s/R}\sum_{i\in B}\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X}^i;\mathbf{\Pi}^i_{s+1}\right)\\ &\le 3\sum_{i\in G(\pi_{\le s})}P_1^i+45k^{s/R}\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X};\mathbf{\Pi}_{s+1}\right)\\ &= 3\mathop{\mathrm{E}}_{\nu|\pi_{\le s}}\left[\sum_{i=1}^{k}\mathbf{B}^i\right]+45k^{s/R}\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X};\mathbf{\Pi}_{s+1}\right)\\ &\le 5000R^2k^{s/R}\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X};\mathbf{\Pi}_{s+1}\right).\end{aligned}$$

The last step used Lemma 15. □

To establish concentration, we also bound the variance of the sum $\sum_{i=1}^{k}\mathbf{L}^i$:

Lemma 21. *If $\pi_{\le s}$ is good and $i \in G(\pi_{\le s})$,*

$$\mathop{\mathrm{Var}}_{\nu|\pi_{\le s}}\left[\mathbf{L}^i\right] \le O(1)\cdot k^{s/R}R^2\log n\cdot\mathop{\mathrm{I}}_{\mu_s|\pi_{\le s}}\left(\mathbf{X}^i;\mathbf{\Pi}^i_{s+1}\right).$$

Corollary 22. *Whenever $\pi_{\leq s}$ is a good transcript with $\mathrm{I}_{\mu_s|\pi_{\leq s}}(\mathbf{\Pi}_{s+1};\mathbf{X}) \leq k^{1/R}/(CR^2 \log n)$, where C is a large constant, we have*

$$, \Pr_{\mathbf{\Pi}_{s+1}\sim\nu|\pi_{\leq s}} \left[\left| \sum_{i \in G(\mathbf{\Pi}_{\leq s+1})} \lambda_{s+1}^{i} \right| \right] \leq 1/(16R).$$

Ultimately, we obtain:

Corollary 23. *If $\nu(\mathcal{G}_{\leq s}) \geq 1/2$ (where $\mathcal{G}_{\leq s}$ is the set of good s-round transcripts), then*

$$\Pr_{\nu|\mathcal{G}_{\leq s}} \left[\mathbf{\Pi}_{\leq s+1} \notin \mathcal{G}_{\leq s+1}^{(2)} \right] \leq \frac{5}{16R}.$$

I. Putting Everything Together

We have seen that given that we have a good transcript after s rounds, the probability under ν that the transcript will fail to satisfy condition (1) of Definition 5 after the next round is at most $1/(4R)$, and the probability that it fails to satisfy condition (2) is at most $5/(16R)$. Together, the probability that after $s+1$ rounds the transcript will no longer be good is at most $9/(16R)$.

The empty (0-round) transcript is, of course, good. Therefore:

$$\begin{aligned} \nu\left(\mathcal{G}_{\leq R}\right) &= \prod_{s=1}^{R} \nu\left(\mathcal{G}_{\leq s} | \mathcal{G}_{\leq s-1}\right) \\ &\geq \left(1 - \frac{9}{16R}\right)^{R} \geq \frac{1}{4}. \end{aligned}$$

Next we show that this implies that the protocol errs with high probability on the last slice, μ_R.

Let T_1 be the set of (complete, R-round) transcripts on which the output is 1. Under ν, the all-one input, the correct output is always 1, and therefore, if δ is the error of the protocol,

$$\begin{aligned} 1 - \delta &\leq \nu(T_1) \\ &= \nu(T_1 \cap \mathcal{G}_{\leq R}) + \nu(T_1 \setminus \mathcal{G}_{\leq R}) \\ &\leq \nu(T_1 \cap \mathcal{G}_{\leq R}) + \nu(\overline{\mathcal{G}_{\leq R}}) \leq \nu(T_1 \cap \mathcal{G}_{\leq R}) + 3/4. \end{aligned}$$

Now let us go back to the last slice, μ_R. The probability that the protocol outputs 1 under μ_R is at least

$$\begin{aligned} \Pr_{\mu_R}[\text{output = 1}] &\geq \sum_{\pi \in T_1 \cap \mathcal{G}_{\leq R}} \mu_R(\pi) \\ &\geq \frac{1}{4} \sum_{\pi \in T_1 \cap \mathcal{G}_{\leq R}} \nu(\pi) \qquad \text{(By Lemma 6)} \\ &\geq \frac{1/4 - \delta}{4} > \frac{1}{50}. \qquad \text{(Since } \delta \leq 1/100\text{)} \end{aligned}$$

However, under μ_R, each player gets 0 with iid probability $2k^{R/R} \log n = 2 \log n$, so the probability that *no* player got 0 is smaller than $1/n^2$. This means that the correct answer is 1 with probability at most $1/n^2$, and therefore, under μ_R the protocol errs with probability greater than $1/100$, contradicting our assumption.

IV. Lower Bound for Approximate Maximum Matching

For approximate maximum matching (referred to as wellfare maximization with unit demands in [13], [14]), we show:

Theorem 24. *Let ϵ be a sufficiently small constant. Then there is a constant δ, such that any protocol that outputs a $(1-\epsilon)$-approximation to the maximum matching in R rounds in bipartite graphs of size $2k$ with success probability $1-\delta$ requires $\Omega(k^{1+1/R}/R^4))$ bits of communication in the worst case.*

The idea behind this lower bound is to convert our lower bound for disjointness into a lower bound for matching: we construct a bipartite graph, where one side corresponds to the players $1, \ldots, k$, and the other side corresponds to the element of the set disjointness universe. We connect a player i to an element j if element j is *missing* from player i's input, that is, if $\mathbf{X}_j^i = 0$. Finding a large matching then corresponds to finding a player that got zero for many coordinates, and this task is at least as hard as disjointness.

The one difference between the two problems is that when we solve disjointness, it is perfectly acceptable for one player to serves as "witness" that *many* elements are not in the intersection, but when we want to output a matching, we cannot assign one player to more than one zero coordinate in its input. Still, our input distribution from Section III can be modified so that with high probability, for most coordinates, there is exactly one player that has a zero in this coordinate. On such inputs, a protocol that solves disjointness must implicitly find a large matching.

A. The Modified Input Distribution

We modify the distribution μ that we constructed in Section III as follows. Call the resulting distribution η.

- We remove the $\log n$ factor everywhere, as we no longer need to ensure that with very high probability the answer to AND is zero. So, on slice $s \geq 1$, which we denote η_s, the inputs are iid Bernoulli variables with probability $k^{-s/R}/2$ of being 0.
- The last slice, η_R, where very few players get 0, will now have constant weight α; that is, $\Pr[\mathbf{S} = R] = \alpha$. We choose α a sufficiently large constant compared to the approximation factor ϵ. (This requires that ϵ not be too large.)
- We add the all-one input, ν, into the distribution, as "slice -1". It also receives constant weight, β, which is small compared to α.
- The remaining slices, η_s for $s \in \{0, \ldots, R-1\}$, each get weight $\Pr[\mathbf{S} = s] = (1 - \alpha - \beta)/R$ for each $s \in \{-1, \ldots, R-1\}$.

Now consider the problem FINDZERO$_k$, defined as follows: each player i receives a bit $X^i \in \{0, 1\}$. The output of the protocol is either the name of a player $i \in [k]$, or $\perp$. If the protocol outputs some name $i \in [k]$, then it must be that $X^i = 0$, that is, the protocol must have found a zero. However, the protocol is only *required* to output the name of a player when exactly one player got zero, and otherwise it is allowed to output $\perp$.

If the error is sufficiently small, FINDZERO is "harder" than AND on η: with constant probability (α), the input is drawn from η_R, where again with constant probability (roughly $1/\sqrt{e}$) exactly one player gets zero. Finding this player means that the protocol must distinguish η_R from $\eta_{-1} = \nu$ with (small) constant probability, because if the input is drawn from ν (all one), the output must be $\perp$. (Note that in the FINDZERO problem we can amplify the success probability of a protocol that has even an arbitrarily small (but constant) success probability $p < 1/2$: we repeat the protocol in parallel, and each time it outputs the name of a player, we check if this player indeed got zero. The verification step costs at most one bit of information. Repeating sufficiently many times raises the success probability as high as desired.)

Our lower bound for AND is easily modified to work with the distribution η by adapting the constants and thresholds for "goodness" appropriately. After R rounds, we get that the distribution of the output under η_R is similar to that under $\eta_{-1} = \nu$, and this contradicts the correctness of the protocol.

Now let FINDZERO^n be the bitwise composition of n independent instances of FINDZERO$_k$: the input to each player is a vector $X^i \in \{0,1\}^n$, and the output should be a vector where each coordinate j contains the answer to FINDZERO$_k$ on coordinate j of the inputs to the players. We require that the marginal success probability on the average coordinate be at least $1-\delta$.

By a direct sum argument, the information cost of FINDZERO$_k^n$ is n times that of FINDZERO$_k$. Note that this time the direct sum is "pointwise": a protocol for FINDZERO$_k^n$ must solve every coordinate with good (marginal) probability, regardless of the values in the other coordinates. This is why we can drop the requirement that $\Pr\left[\bigwedge_{i=1}^k \mathbf{X}^i = 1\right] = o(1)$. (In contrast, the direct sum argument for disjointness must circumvent the fact that the output is the *disjunction* of the answers on individual coordinates.)

Finally, we get a lower bound for approximate maximum matching by the reduction we described above, for $k = n$. Formally, given inputs $(X^1, \ldots, X^k)$ to FINDZERO$_k^n$, we construct a bipartite graph, with player nodes $p_1, \ldots, p_k$ and element nodes $v_1, \ldots, v_k$; each player node p_i is connected to those element nodes v_j where $X_j^i = 0$. Because in slice η_R each element node chooses to be a neighbor of each player node with probability $1/(2k)$, with very high probability the subgraph induced by slice-R element nodes and all player nodes contains a matching of size $\Theta(\alpha k)$ over element nodes that each have a single player node connected to them (i.e., coordinates where a single player got zero). On the other hand, all the other slices together have only roughly $\Theta((1-\alpha)k)$ nodes, so they do not contribute much to the matching.

If α is large enough compared to ϵ, we get that for some small constant δ, any protocol that outputs a $(1-\epsilon)$-approximation to the maximum matching must match at least a δ-fraction of element nodes that have a single neighbor (a large, uniformly random subset). This means it must solve FINDZERO$_k^n$ with marginal success probability roughly δ on the average coordinate, assuming it succeeds in finding a large matching with high probability.

REFERENCES

[1] B. Kalyanasundaram and G. Schnitger, "The probabilistic communication complexity of set intersection," *SIAM Journal on Discrete Mathematics*, vol. 5, no. 4, pp. 545–557, Nov. 1992.

[2] Razborov, "On the distributed complexity of disjointness," *TCS: Theoretical Computer Science*, vol. 106, 1992.

[3] J. M. Phillips, E. Verbin, and Q. Zhang, "Lower bounds for number-in-hand multiparty communication complexity, made easy," in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '12, 2012, pp. 486–501.

[4] D. P. Woodruff and Q. Zhang, "Distributed computation does not help," *CoRR*, vol. abs/1304.4636, 2013.

[5] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," *Journal of Computer and System Sciences*, vol. 58, no. 1, pp. 137 – 147, 1999.

[6] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, "An information statistics approach to data stream and communication complexity," *J. Comput. Syst. Sci.*, vol. 68, no. 4, pp. 702–732, 2004.

[7] A. Gronemeier, "Asymptotically optimal lower bounds on the nih-multiparty information," *arXiv preprint arXiv:0902.1609*, 2009.

[8] O. Weinstein and D. P. Woodruff, *The Simultaneous Communication of Disjointness with Applications to Data Streams*, 2015, pp. 1082–1093.

[9] M. Braverman and R. Oshman, "On information complexity in the broadcast model," in *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, 2015, pp. 355–364.

[10] J. Brody, A. Chakrabarti, R. Kondapally, D. P. Woodruff, and G. Yaroslavtsev, "Beyond set disjointness: The communication complexity of finding the intersection," in *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*, ser. PODC '14, 2014, pp. 106–113.

[11] Y. Li, X. Sun, C. Wang, and D. P. Woodruff, "On the communication complexity of linear algebraic problems in the message passing model," in *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, 2014, pp. 499–513.

[12] D. Peleg, *Distributed Computing: A Locality-Sensitive Approach*, ser. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2000.

[13] S. Dobzinski, N. Nisan, and S. Oren, "Economic efficiency requires interaction," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, 2014, pp. 233–242.

[14] N. Alon, N. Nisan, R. Raz, and O. Weinstein, "Welfare maximization with limited interaction," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 1499–1512.

[15] D. P. Woodruff and Q. Zhang, "Tight bounds for distributed functional monitoring," in *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, 2012, pp. 941–960.

[16] ——, "An optimal lower bound for distinct elements in the message passing model," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, 2014, pp. 718–733.

[17] M. Braverman, F. Ellen, R. Oshman, T. Pitassi, and V. Vaikuntanathan, "A tight bound for set disjointness in the message-passing model," in *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, 2013, pp. 668–677.

[18] A. Chakrabarti, Y. Shi, A. Wirth, and A. C.-C. Yao, "Informational complexity and the direct sum problem for simultaneous message complexity," in *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001*, pp. 270–278.

[19] M. Saglam and G. Tardos, "On the communication complexity of sparse set disjointness and exists-equal problems," in *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, 2013, pp. 678–687.