

# Tight Bounds on the Round Complexity of the Distributed Maximum Coverage Problem\*

Sepehr Assadi<sup>†</sup>

Sanjeev Khanna<sup>†</sup>

## Abstract

We study the maximum  $k$ -set coverage problem in the following distributed setting. A collection of input sets  $S_1, \dots, S_m$  over a universe  $[n]$  is partitioned across  $p$  machines and the goal is to find  $k$  sets whose union covers the most number of elements. The computation proceeds in rounds where in each round machines communicate information to each other. Specifically, in each round, all machines simultaneously send a message to a central coordinator who then communicates back to all machines a summary to guide the computation for the next round. At the end of the last round, the coordinator outputs the answer. The main measures of efficiency in this setting are the approximation ratio of the returned solution, the communication cost of each machine, and the number of rounds of computation.

Our main result is an asymptotically tight bound on the tradeoff between these three measures for the distributed maximum coverage problem. We first show that any  $r$ -round protocol for this problem either incurs a communication cost of  $k \cdot m^{\Omega(1/r)}$  or only achieves an approximation factor of  $k^{\Omega(1/r)}$ . This in particular implies that any protocol that simultaneously achieves good approximation ratio ( $O(1)$  approximation) and good communication cost ( $\tilde{O}(n)$  communication per machine), essentially requires *logarithmic* (in  $k$ ) number of rounds. We complement our lower bound result by showing that there exist an  $r$ -round protocol that achieves an  $\frac{e}{e-1}$ -approximation (essentially best possible) with a communication cost of  $k \cdot m^{O(1/r)}$  as well as an  $r$ -round protocol that achieves a  $k^{O(1/r)}$ -approximation with only  $\tilde{O}(n)$  communication per each machine (essentially best possible).

We further use our results in this distributed setting to obtain new bounds for maximum coverage in two other main models of computation for massive datasets, namely, the dynamic streaming model and the MapReduce model.

## 1 Introduction

A common paradigm for designing scalable algorithms for problems on massive data sets is to distribute the computation by partitioning the data across multiple machines interconnected via a communication network. The machines can then jointly compute a function on the union of their inputs by exchanging messages. A well-studied and important case of this paradigm is

the *coordinator model* (see, e.g., [34, 54, 60]). In this model, computation proceeds in rounds, and in each round, all machines simultaneously send a message to a central coordinator who then communicates back to all machines a summary to guide the computation for the next round. At the end, the coordinator outputs the answer. Main measures of efficiency in this setting are the *communication cost*, i.e., the total number of bits communicated by each machine, and the *round complexity*, i.e., the number of rounds of computation.

The distributed coordinator model (and the closely related *message-passing* model<sup>1</sup>) has been studied extensively in recent years (see, e.g., [54, 23, 59, 60, 61], and references therein). Traditionally, the focus in this model has been on optimizing the communication cost and round complexity issues have been ignored. However, in recent years, motivated by application to big data analysis such as MapReduce computation, there have been a growing interest in obtaining round efficient protocols for various problems in this model (see, e.g., [3, 4, 43, 41, 37, 52, 30, 13, 38, 10]).

In this paper, we study the *maximum coverage* problem in the coordinator model: A collection of input sets  $\mathcal{S} := \{S_1, \dots, S_m\}$  over a universe  $[n]$  is arbitrarily partitioned across  $p$  machines, and the goal is to select  $k$  sets whose union covers the most number of elements from the universe. Maximum coverage is a fundamental optimization problem with a wide range of applications (see, e.g., [47, 45, 58, 36] for some applications). As an illustrative example of submodular maximization, the maximum coverage problem has been studied in various recent works focusing on scalable algorithms for massive data sets including in the coordinator model (e.g., [41, 52]), MapReduce framework (e.g., [27, 48]), and the streaming model (e.g. [20, 51]); see Section 1.1 for a more comprehensive summary of previous results.

Previous results for maximum coverage in the distributed model can be divided into two main categories: one on hand, we have *communication efficient* protocols that only need  $\tilde{O}(n)$  communication and achieve a con-

\*A full version of this paper is available on arXiv.

<sup>†</sup>Department of Computer and Information Science, University of Pennsylvania. Supported in part by National Science Foundation grants CCF-1552909 and CCF-1617851. Email: {sassadi, sanjeev}@cis.upenn.edu.

<sup>1</sup>In absence of any restriction on round complexity, these two models are equivalent; see, e.g., [54].

stant factor approximation, but require a *large* number of rounds of  $\Omega(p)$  [15, 51]<sup>2</sup>. On the other hand, we have *round efficient* protocols that achieve a constant factor approximation in  $O(1)$  rounds of communication, but incur a *large* communication cost  $k \cdot m^{\Omega(1)}$  [48].

This state-of-the-affairs, namely, communication efficient protocols that require a large number of rounds, or round efficient protocols that require a large communication cost, raises the following natural question: *Does there exist a truly efficient distributed protocol for maximum coverage, that is, a protocol that simultaneously achieves  $\tilde{O}(n)$  communication cost,  $O(1)$  round complexity, and gives a constant factor approximation?* This is the precisely the question addressed in this work.

**1.1 Our Contributions** Our first result is a *negative* resolution of the aforementioned question. In particular, we show that,

**RESULT 1.** *For any integer  $r \geq 1$ , any  $r$ -round protocol for distributed maximum coverage either incurs  $k \cdot m^{\Omega(1/r)}$  communication per machine or has an approximation factor of  $k^{\Omega(1/r)}$ .*

Prior to our work, the only known lower bound for distributed maximum coverage was due to McGregor and Vu [51] who showed an  $\Omega(m)$  communication lower bound for any protocol that achieves a better than  $\left(\frac{e}{e-1}\right)$ -approximation (regardless of number of rounds and even if the input is *randomly* distributed); see also [9]. Indyk et al. [41] also showed that no composable coresets (a restricted family of single round protocols) can achieve a better than  $\tilde{\Omega}(\sqrt{k})$  approximation without communicating essentially the whole input (which is known to be tight [30]). However, no super constant lower bounds on approximation ratio were known for this problem for arbitrary protocols even for one round of communication. Our result on the other hand implies that to achieve a constant factor approximation with an  $O(n^c)$  communication protocol (for a fixed constant  $c > 0$ ),  $\Omega\left(\frac{\log k}{\log \log k}\right)$  rounds of communication are required.

In establishing Result 1, we introduce a general framework for proving communication complexity lower bounds for *bounded round* protocols in the distributed coordinator model. This framework, formally introduced in Section 4, captures many of the existing multi-party communication complexity lower bounds

<sup>2</sup>We remark that the algorithms of [15, 51] are originally designed for the streaming setting and in that setting are quite efficient as they only require one or a constant number of passes over the stream. However, implementing one pass of a streaming algorithm in the coordinator model directly requires  $p$  rounds of communication.

in the literature for bounded-round protocols including [33, 46, 13, 12] (for one round a.k.a simultaneous protocols), and [7, 8] (for multi-round protocols). We believe our framework will prove useful for establishing distributed lower bound results for other problems, and is thus interesting in its own right.

We complement Result 1 by giving protocols that show that its bounds are essentially *tight*.

**RESULT 2.** *For any integer  $r \geq 1$ , there exist  $r$ -round protocols that achieve:*

1. *an approximation factor of (almost)  $\frac{e}{e-1}$  with  $k \cdot m^{O(1/r)}$  communication per machine, or*
2. *an approximation factor of  $O(r \cdot k^{1/r+1})$  with  $\tilde{O}(n)$  communication per machine.*

Results 1 and 2 together provide a near complete understanding of the tradeoff between the approximation ratio, the communication cost, and the round complexity of protocols for the distributed maximum coverage problem for any fixed number of rounds.

The first protocol in Result 2 is quite general in that it works for maximizing any monotone submodular function subject to a cardinality constraint. Previously, it was known how to achieve a 2-approximation distributed algorithm for this problem with  $m^{O(1/r)}$  communication and  $r$  rounds of communication [48]. However, the previous best  $\left(\frac{e}{e-1}\right)$ -approximation distributed algorithm for this problem with *sublinear* in  $m$  communication due to Kumar et al. [48] requires at least  $\Omega(\log n)$  rounds of communication. As noted above, the  $\left(\frac{e}{e-1}\right)$  is information theoretically the best approximation ratio possible for any protocol that uses sublinear in  $m$  communication [51].

The second protocol in Result 2 is however tailored heavily to the maximum coverage problem. Previously, it was known that an  $O(\sqrt{k})$  approximation can be achieved via  $\tilde{O}(n)$  communication [30] per machine, but no better bounds were known for this problem in multiple rounds under  $\text{poly}(n)$  communication cost. It is worth noting that since an adversary may assign all sets to a single machine, a communication cost of  $\tilde{O}(n)$  is essentially best possible bound. We now elaborate on some applications of our results.

**Dynamic Streams.** In the *dynamic (set) streaming* model, at each step, either a new set is inserted or a previously inserted set is deleted from the stream. The goal is to solve the maximum coverage problem on the sets that are present at the end of the stream. A *semi-streaming* algorithm is allowed to make one or a small number of passes over the stream and use only

$O(n \cdot \text{poly}\{\log m, \log n\})$  space to process the stream and compute the answer. The streaming setting for the maximum coverage problem and the closely related set cover problem has been studied extensively in recent years [58, 28, 14, 24, 35, 32, 15, 39, 25, 11, 20, 26, 51, 9, 36]. Previous work considered this problem in *insertion-only* streams and more recently in the *sliding window* model; to the best of our knowledge, no non-trivial results were known for this problem in dynamic streams. Our Results 1 and 2 imply the first upper and lower bounds for maximum coverage in dynamic streams.

Result 1 together with a recent characterization of multi-pass dynamic streaming algorithms [5] proves that any semi-streaming algorithm for maximum coverage in dynamic streams that achieves any *constant approximation* requires  $\Omega\left(\frac{\log n}{\log \log n}\right)$  passes over the stream. This is in sharp contrast with insertion-only streams in which semi-streaming algorithms can achieve (almost) 2-approximation in a single pass [15] or (almost)  $\left(\frac{e}{e-1}\right)$ -approximation in a constant number of passes [51] (constant factor approximations are also known in the sliding window model [26, 36]). To our knowledge, this is the first *multi-pass* dynamic streaming lower bound that is based on the characterization of [5]. Moreover, as maximum coverage is a special case of submodular maximization (subject to cardinality constraint), our lower bound extends to this problem and settles an open question of [36] on the space complexity of submodular maximization in dynamic streams.

We complement this result by showing that one can implement the first algorithm in Result 2 using proper *linear sketches* in dynamic streams, which imply an (almost)  $\left(\frac{e}{e-1}\right)$ -approximation semi-streaming algorithm for maximum coverage (and monotone submodular maximization) in  $O(\log m)$  passes. As a simple application of this result, we can also obtain an  $O(\log n)$ -approximation semi-streaming algorithm for the set cover problem in dynamic stream that requires  $O(\log m \cdot \log n)$  passes over the stream.

**MapReduce Framework.** In the MapReduce model, there are  $p$  machines each with a memory of size  $s$  such that  $p \cdot s = O(N)$ , where  $N$  is the total memory required to represent the input. MapReduce computation proceeds in synchronous rounds where in each round, each machine performs some local computation, and at the end of the round sends messages to other machine to guide the computation for the next round. The total size of messages received by each machine, however, is restricted to be  $O(s)$ . Following [44], we require both  $p$  and  $s$  to at be at most  $N^{1-\Omega(1)}$ . The main complexity measure of interest in this model is typically the number of rounds. Maximum coverage and submodular

maximization have also been extensively studied in the MapReduce model [27, 22, 48, 53, 41, 52, 30, 31, 19].

Proving round complexity lower bounds in the MapReduce framework turns out to be a challenging task (see, e.g., [56] for implication of such lower bounds to long standing open problems in complexity theory). As a result, most previous work on lower bounds concerns either communication cost (in a fixed number of rounds) or specific classes of algorithms (for round lower bounds); see, e.g., [1, 21, 55, 42] (see [56] for more details). Our results contribute to the latter line of work by characterizing the power of a large family of MapReduce algorithms for maximum coverage.

Many existing techniques for MapReduce algorithms utilize the following paradigm which we call the *sketch-and-update* approach: each machine sends a summary of its input, i.e., a sketch, to a *single* designated machine which processes these sketches and computes a single combined sketch; the original machines then receive this combined sketch and update their sketch computation accordingly; this process is then continued on the updated sketches. Popular algorithmic techniques belonging to this framework include composable coresets (e.g., [15, 17, 18, 41]), the filtering method (e.g., [50]), linear-sketching algorithms (e.g., [3, 4, 43, 2]), and the sample-and-prune technique (e.g., [48, 40]), among many others.

We use Result 1 to prove a lower bound on the power of this approach for solving maximum coverage in the MapReduce model. We show that any MapReduce algorithm for maximum coverage in the sketch-and-update framework that uses  $s = m^\delta$  memory per machine requires  $\Omega\left(\frac{1}{\delta}\right)$  rounds of computation. Moreover, both our algorithms in Result 2 belong to the sketch-and-update framework and can be implemented in the MapReduce model. In particular, the round complexity of our first algorithm for monotone submodular maximization (subject to cardinality constraint) in Result 2 matches the best known algorithm of [31] with the benefit of using sublinear communication (the algorithm of [31], in each round, incurs a linear (in input size) communication cost). We remark that the algorithm in [31] is however more general in that it supports a larger family of constraints beside the cardinality constraint we study in this paper.

**1.2 Organization** The rest of the paper is organized as follows. We start with preliminaries and notation in Section 2. We then present a high level technical overview of our paper in Section 3. In Section 4, we present our framework for proving communication complexity lower bounds for bounded-round protocols. Section 5.1 is dedicated to the proof of Result 1. We

sketch the proof of Result 2 in Section 6. Due to space limitations, many of the proofs and details are deferred to the full version of the paper. In particular, we defer the detailed discussion on the applications of our results to dynamic set streams and the MapReduce model to the full version of the paper.

## 2 Preliminaries

**Notation.** For a collection of sets  $\mathcal{C} = \{S_1, \dots, S_t\}$ , we define  $c(\mathcal{C}) := \cup_{i \in [t]} S_i$ , i.e., the set of elements covered by  $\mathcal{C}$ . For a tuple  $X = (X_1, \dots, X_t)$  and index  $i \in [t]$ ,  $X^{<i} := (X_1, \dots, X_{i-1})$  and  $X^{-i} := (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_t)$ . We use sans serif fonts to denote random variables, i.e.,  $\mathbf{X}$ .

For a random variable  $\mathbf{X}$  over a support  $\Omega_{\mathbf{X}}$ ,  $\text{dist}(\mathbf{X})$  denotes the distribution of  $\mathbf{X}$  and  $|\mathbf{X}| := \log |\Omega_{\mathbf{X}}|$ . We use  $\mathbb{H}(\mathbf{X})$  and  $\mathbb{I}(\mathbf{X}; \mathbf{Y})$  to denote the *Shannon entropy* of  $\mathbf{X}$  and *mutual information* of  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively. For any two distributions  $\mu$  and  $\nu$  over the same probability space,  $\mathbb{D}(\mu \parallel \nu)$  and  $\|\mu - \nu\|$  denote the *Kullback-Leibler divergence* and the *total variation distance* between  $\mu$  and  $\nu$ , respectively.

**2.1 Tools From Information Theory** The proof of the following basic properties of entropy and mutual information can be found in [29] (see Chapter 2).

FACT 2.1. *Let  $A, B$ , and  $C$  be three (possibly correlated) random variables.*

1.  $0 \leq \mathbb{H}(A) \leq |A|$ , and  $\mathbb{H}(A) = |A|$  iff  $A$  is uniformly distributed over its support.
2.  $\mathbb{I}(A; B \mid C) \geq 0$ . The equality holds iff  $A$  and  $B$  are independent conditioned on  $C$ .
3.  $\mathbb{H}(A \mid B, C) \leq \mathbb{H}(A \mid B)$ . The equality holds iff  $A \perp C \mid B$ .
4.  $\mathbb{I}(A, B; C) = \mathbb{I}(A; C) + \mathbb{I}(B; C \mid A)$  (chain rule of mutual information).
5. Suppose  $f(A)$  is a deterministic function of  $A$ , then  $\mathbb{I}(f(A); B \mid C) \leq \mathbb{I}(A; B \mid C)$  (data processing inequality).

We also use the following two standard propositions, regarding the effect of conditioning on mutual information. The proofs are standard (see the full version).

PROPOSITION 2.1. *For variables  $A, B, C, D$ , if  $A \perp D \mid C$ , then,  $\mathbb{I}(A; B \mid C) \leq \mathbb{I}(A; B \mid C, D)$ .*

PROPOSITION 2.2. *For variables  $A, B, C, D$ , if  $A \perp D \mid B, C$ , then,  $\mathbb{I}(A; B \mid C) \geq \mathbb{I}(A; B \mid C, D)$ .*

For two distributions  $\mu$  and  $\nu$  over the same probability space, the *Kullback-Leibler divergence* between  $\mu$  and  $\nu$  is defined as  $\mathbb{D}(\mu \parallel \nu) := \mathbb{E}_{a \sim \mu} \left[ \log \frac{\Pr_{\mu}(a)}{\Pr_{\nu}(a)} \right]$ .

FACT 2.2. *For random variables  $A, B, C$ ,*

$$\mathbb{I}(A; B \mid C) = \mathbb{E}_{(b,c)} \left[ \mathbb{D}(\text{dist}(A \mid C = c) \parallel \text{dist}(A \mid B = b, C = c)) \right].$$

We denote the *total variation distance* between two distributions  $\mu$  and  $\nu$  over the same probability space  $\Omega$  by  $\|\mu - \nu\| = \frac{1}{2} \cdot \sum_{x \in \Omega} |\Pr_{\mu}(x) - \Pr_{\nu}(x)|$ .

The following Pinskers' inequality bounds the total variation distance between two distributions based on their KL-divergence,

FACT 2.3. (PINSKER'S INEQUALITY) *For any two distributions  $\mu$  and  $\nu$ ,  $\|\mu - \nu\| \leq \sqrt{\frac{1}{2} \cdot \mathbb{D}(\mu \parallel \nu)}$ .*

FACT 2.4. *Suppose  $\mu$  and  $\nu$  are two distributions for an event  $\mathcal{E}$ , then,  $\Pr_{\mu}(\mathcal{E}) \leq \Pr_{\nu}(\mathcal{E}) + \|\mu - \nu\|$ .*

**2.2 Communication Complexity Model** We prove our lower bound for distributed protocols using the framework of communication complexity, and in particular in the (number-in-hand) multiparty communication model with shared blackboard: there are  $p$  players (corresponding to machines) receiving inputs  $(x_1, \dots, x_p)$  from a prior distribution  $\mathcal{D}$  on  $\mathcal{X}_1 \times \dots \times \mathcal{X}_p$ . The communication happens in rounds and in each round, the players *simultaneously* write a message to a *shared blackboard* visible to all parties. The message sent by any player  $i$  in each round can only depend on the input of the player, i.e.,  $x_i$ , the current content of the blackboard, i.e., the messages communicated in previous rounds, and public and private randomness. In addition to  $p$  players, there exists a central party called the referee (corresponding to the coordinator) who only sees the content of the blackboard and public randomness and is responsible for outputting the answer in the final round.

For a protocol  $\pi$ , we use  $\Pi = (\Pi_1, \dots, \Pi_p)$  to denote the transcript of the messages communicated by all players, i.e., the content of the blackboard. The *communication cost* of a protocol  $\pi$ , denoted by  $\|\pi\|$ , is the sum of worst-case length of the messages communicated by all players, i.e.,  $\|\pi\| := \sum_{i=1}^p |\Pi_i|$ . We further refer to  $\max_{i \in [p]} |\Pi_i|$  as the *per-player communication cost* of  $\pi$ . We remark that this model is identical to the distributed setting introduced earlier if we allow the coordinator to communicate with machines free of charge. As a result, communication lower bounds in this model imply identical communication lower bounds for distributed

protocols. We refer the reader to the excellent text by Kushilevitz and Nisan [49] for more details on communication complexity.

**2.3 Submodular Maximization with Cardinality Constraint** Let  $V = \{a_1, \dots, a_m\}$  be a ground set of  $m$  items. For any set function  $f : 2^V \rightarrow \mathbb{R}$  and any  $A \subseteq V$ , we define the *marginal contribution* to  $f$  as a set function  $f_A : 2^V \rightarrow \mathbb{R}$  such that for all  $B \subseteq V$ ,  $f_A(B) = f(A \cup B) - f(A)$ . When clear from the context, we abuse the notation and for  $a \in V$ , use  $f(a)$  and  $f_A(a)$  instead of  $f(\{a\})$  and  $f_A(\{a\})$ , respectively. A function  $f$  is *submodular* iff for all  $A \subseteq B \subseteq V$  and for all  $a \in V$ ,  $f_B(a) \leq f_A(a)$ . A submodular function  $f$  is additionally *monotone* iff  $\forall A \subseteq B \subseteq V$ ,  $f(A) \leq f(B)$ .

The maximum coverage problem is a special case of maximizing a monotone submodular function subject to a cardinality constraint of  $k$ , i.e., finding  $A^* \in \arg \max_{A: |A|=k} f(A)$ : for any set  $S$  in maximum coverage we can have an item  $a_S \in V$  and for each  $A \subseteq V$ , define  $f(A) = |\bigcup_{a_S \in A} S|$ . It is easy to verify that  $f(\cdot)$  is monotone submodular.

### 3 Technical Overview

**Lower Bounds (Result 1).** Let us start by sketching our proof for simultaneous protocols. We provide each machine with a collection of sets from a family of sets with small pairwise intersection such that *locally*, i.e., from the perspective of each machine, all these sets look alike. At the same time, we ensure that *globally*, one set in each machine is *special*; think of a special set as covering a *unique* set of elements across the machines while all other sets are mostly covering a set of *shared* elements. The proof now consists of two parts: (i) use the simultaneity of the communication to argue that as each machine is oblivious to identity of its special set, it cannot convey enough information about this set using limited communication, and (ii) use the bound on the size of the intersection between the sets to show that this prevents the coordinator to find a good solution.

The strategy outlined above is in fact at the core of many existing lower bounds for simultaneous protocols in the coordinator model including [33, 46, 13, 12] (a notable exception is the lower bound of [12] on estimating matching size in *sparse* graphs). For example, to obtain the hard input distributions in [46, 13] for the maximum matching problem, we just need to switch the sets in the small intersecting family above with edge-disjoint induced matchings in a Ruzsa-Szemerédi graph [57] (see also [6] for more details on these graphs). The first part of the proof that lower bounds the communication cost required for finding the special induced matchings (corresponding to special sets above), remains quite similar;

however, we now need an entirely different argument for proving the second part, i.e., the bound obtained on the approximation ratio. This observation raises the following question: can we “automate” the task of proving a communication lower bound in these arguments so that one can focus solely on the second part of the argument, i.e., proving the approximation lower bound subject to each machine not being able to find its special entity, e.g., special sets in the coverage problem and special induced matchings in the maximum matching problem?

We answer this question in the affirmative by designing a framework for proving communication lower bounds of the aforementioned type. We design an abstract hard input distribution using the ideas above and prove a general communication lower bound in this abstraction. This reduces the task of proving a communication lower bound for any specific problem to designing suitable combinatorial objects that roughly speaking enforce the importance of “special entities” discussed above. We emphasize that this second part may still be a non-trivial challenge; for instance, lower bounds for matchings in [46, 13] rely on Ruzsa-Szemerédi graphs to prove this part. Nevertheless, automating the task of proving a communication lower bound in our framework allows one to focus solely on a combinatorial problem and entirely bypass the information-theoretic arguments needed to prove the communication lower bound.

We further extend our framework to multi round protocols by building on the recent multi-party round elimination technique of [7] and its extension in [8]. At a high level, in the hard instances of  $r$ -round protocols, each machine is provided with a collection of instances of the same problem but on a “lower dimension”, i.e., defined on a smaller number of machines and input size. One of these instances is a special one in that it needs to be solved by the machines in order to solve the original instance. Again, using the simultaneity of the communication in one round, we show that the first round of communication cannot reveal enough information about this special instance and hence the machines need to solve the special instance in only  $r - 1$  rounds of communication, which is proven to be hard inductively. Using the abstraction in our framework allows us to solely focus on the communication aspects of this argument, independent of the specifics of the problem at hand. This allows us to provide a more direct and simpler proof than [7, 8], which is also applicable to a wider range of problems (the results in [7, 8] are for the setting of combinatorial auctions). However, although simpler than [7, 8], this proof is still far from being simple - indeed, it requires a delicate information-theoretic argument (see Section 4 for further details). This complexity of proving a multi-

round lower bound in this model is in fact another motivation for our framework. To our knowledge, the only previous lower bounds specific to bounded round protocols in the coordinator model are those of [7, 8]; we hope that our framework facilitates proving such lower bounds in this model (understanding the power of bounded round protocols is regarded as an interesting open question in the literature; see, e.g., [60]).

Finally, we prove the lower bound for maximum coverage using this framework by designing a family of sets which we call *randomly nearly disjoint*; roughly speaking the sets in this family have the property that any suitably small *random* subset of one set is essentially disjoint from any other set in the family. A reader familiar with [25] may realize that this definition is similar to the *edifice* set-system introduced in [25]; the main difference here is that we need every *random* subsets of each set in the family to be disjoint from other sets, as opposed to a *pre-specified* collection of sets as in edifices [25]. As a result, the algebraic techniques of [25] do not seem suitable for our purpose and we prove our results using different techniques. The lower bound then follows by instantiating the hard distribution in our framework with this family for maximum coverage and proving the approximation lower bound.

**Upper Bounds (Result 2).** We achieve the first algorithm in Result 2, namely an  $\left(\frac{e}{e-1}\right)$ -approximation algorithm for maximum coverage (and submodular maximization), via an implementation of a thresholding greedy algorithm (see, e.g., [16, 25]) in the distributed setting using the sample-and-prune technique of [48] (a similar thresholding greedy algorithm was used recently in [51] for streaming maximum coverage). The main idea in the sample-and-prune technique is to sample a collection of sets from the machines in each round and send them to the coordinator who can build a partial greedy solution on those sets; the coordinator then communicates this partial solution to each machine and in the next round the machines only sample from the sets that can have a substantial marginal contribution to the partial greedy solution maintained by the coordinator, hence essentially pruning the input sets. Using a different greedy algorithm and a more careful choice of the threshold on the necessary marginal contribution from each set, we show that an  $\left(\frac{e}{e-1}\right)$ -approximation can be obtained in constant number of rounds and sublinear communication (as opposed to the approach of [48] which requires  $\Omega(\log n)$  rounds).

The second algorithm in Result 2, namely a  $k^{O(1/r)}$ -approximation algorithm for any number of rounds  $r$ , however is more involved and is based on a new iterative sketching method specific to the maximum coverage

problem. Recall that in our previous algorithm the machines are mainly “observers” and simply provide the coordinator with a sample of their input; our second algorithm is in some sense on the other extreme. In this algorithm, each machine is responsible for computing a suitable sketch of its input, which roughly speaking, is a collection of sets that tries to “represent” each optimal set in the input of this machine. The coordinator is also maintaining a greedy solution that is updated based on the sketches received from each machine. The elements covered by this collection are shared by the machines to guide them towards the sets that are “misrepresented” by the sketches computed so far, and the machines update their sketches for the next round accordingly. We show that either the greedy solution maintained by the coordinator is already a good approximation or the final sketches computed by the machines are now a good representative of the optimal sets and hence contain a good solution.

#### 4 A Framework for Distributed Lower Bounds

We introduce a general framework for proving communication complexity lower bounds for *bounded round* protocols in the distributed coordinator model. Consider a *decision* problem<sup>3</sup>  $\mathcal{P}$  defined by the family of functions  $\mathcal{P}_s : \{0, 1\}^s \rightarrow \{0, 1\}$  for any integer  $s \geq 1$ ; we refer to  $s$  as *size* of the problem and to  $\{0, 1\}^s$  as its *domain*. Note that  $\mathcal{P}_s$  can be a partial function, i.e., not necessarily defined on its whole domain. An instance  $I$  of problem  $\mathcal{P}_s$  is simply a binary string of length  $s$ . We say that  $I$  is a **Yes** instance if  $\mathcal{P}_s(I) = 1$  and is a **No** instance if  $\mathcal{P}_s(I) = 0$ . For example,  $\mathcal{P}_s$  can denote the decision version of the maximum coverage problem over  $m$  sets and  $n$  elements with parameter  $k$  (in which case  $s$  would be a fixed function of  $m$ ,  $n$ , and  $k$  depending on the representation of the input) such that there is a relatively large gap (as a function of, say,  $k$ ) between the value of optimal solution in **Yes** and **No** instances. We can also consider the problem  $\mathcal{P}_s$  in the distributed model, whereby we distribute each instance between the players. The distributed coverage problem for instance, can be modeled here by partitioning the sets in the instances of  $\mathcal{P}_s$  across the players.

To prove a communication lower bound for some problem  $\mathcal{P}$ , one typically needs to design a hard input distribution  $\mathcal{D}$  on instances of the problem  $\mathcal{P}$ , and then show that distinguishing between the **Yes** and **No** cases in instances sampled from  $\mathcal{D}$ , with some sufficiently large probability, requires large communication. Such a

<sup>3</sup>While we present our framework for decision problems, with some modifications, it also extends to *search* problems. We elaborate more on this in the full version of the paper.

distribution inevitably depends on the specific problem  $\mathcal{P}$  at hand. We would like to *abstract out* this dependence to the underlying problem and design a *template* hard distribution for any problem  $\mathcal{P}$  using this abstraction. Then, to achieve a lower bound for a particular problem  $\mathcal{P}$ , one only needs to focus on the problem specific parts of this template and *design* them according to the problem  $\mathcal{P}$  at hand. We emphasize that obviously we are not going to prove a communication lower bound for every possible distributed problem; rather, our framework reduces the problem of proving a communication lower bound for a problem  $\mathcal{P}$  to designing appropriate problem-specific *gadgets* for  $\mathcal{P}$ , which determine the strength of the lower bound one can ultimately prove using this framework. With this plan in mind, we now describe a high level overview of our framework.

#### 4.1 A High Level Overview of the Framework

Consider any decision problem  $\mathcal{P}$ ; we construct a recursive family of distributions  $\mathcal{D}_0, \mathcal{D}_1, \dots$  where  $\mathcal{D}_r$  is a hard input distribution for  $r$ -round protocols of  $\mathcal{P}_{s_r}$ , i.e., for instances of size  $s_r$  of the problem  $\mathcal{P}$ , when the input is partitioned between  $p_r$  players. Each instance in  $\mathcal{D}_r$  is a careful “combination” of many sub-instances of problem  $\mathcal{P}_{s_{r-1}}$  over different subsets of  $p_{r-1}$  players, which are sampled (essentially) from  $\mathcal{D}_{r-1}$ . We ensure that a small number of these sub-instances are “special” in that to solve the original instance of  $\mathcal{P}_{s_r}$ , at least one of these instances of  $\mathcal{P}_{s_{r-1}}$  (over  $p_{r-1}$  players) needs to be solved necessarily. We “hide” the special sub-instances in the input of players in a way that locally, no player is able to identify them and show that the first round of communication in any protocol with a small communication is spent only in identifying these special sub-instances. We then inductively show that as solving the special instance is hard for  $(r-1)$ -round protocols, the original instance must be hard for  $r$ -round protocols as well.

We now describe this distribution in more detail. The  $p_r$  players in the instances of distribution  $\mathcal{D}_r$  are partitioned into  $g_r$  groups  $P_1, \dots, P_{g_r}$ , each of size  $p_{r-1}$  (hence  $g_r = p_r/p_{r-1}$ ). For every group  $i \in [g_r]$  and every player  $q \in P_i$ , we create  $w_r$  instances  $I_1^i, \dots, I_{w_r}^i$  of the problem  $\mathcal{P}_{s_{r-1}}$  sampled from the distribution  $\mathcal{D}_{r-1}$ . The domain of each instance  $I_j^i$  is the same across all players in  $P_i$  and is different (i.e., disjoint) between any two  $j \neq j' \in [w_r]$ ; we refer to  $w_r$  as the *width parameter*. The next step is to *pack* all these instances into a single instance  $I^i(q)$  for the player  $q$ ; this is one of the places that we need a problem specific gadget, namely a *packing function*<sup>4</sup> that can pack  $w_r$

instances of problem  $\mathcal{P}_{s_{r-1}}$  into a single instance of problem  $\mathcal{P}_{s'_r}$  for some  $s'_r \geq s_r$ . We postpone the formal description of the packing functions to the next section, but roughly speaking, we require each player to be able to construct the instance  $I^i(q)$  from the instances  $I_1^i, \dots, I_{w_r}^i$  and vice versa. As such, even though each player is given as input a single instance  $I^i$ , we can think of each player as conceptually “playing” in  $w_r$  different instances  $I_1^i, \dots, I_{w_r}^i$  of  $\mathcal{P}_{s_{r-1}}$  instead.

In each group  $i \in [g_r]$ , one of the instances, namely  $I_{j^*}^i$  for  $j^* \in [w_r]$ , is the *special instance* of the group: if we combine the inputs of players in  $P_i$  on their special instance  $I_{j^*}^i$ , we obtain an instance which is sampled from the distribution  $\mathcal{D}_{r-1}$ . On the other hand, all other instances are *fooling instances*: if we combine the inputs of players in  $P_i$  on their instance  $I_j^i$  for  $j \neq j^*$ , the resulting instance is *not* sampled from  $\mathcal{D}_{r-1}$ ; rather, it is an instance created by picking the input of each player *independently* from the corresponding marginal of  $\mathcal{D}_{r-1}$  ( $\mathcal{D}_{r-1}$  is *not* a product distribution, thus these two distributions are not identical). Nevertheless, by construction, each player is oblivious to this difference and hence is unaware of which instance in the input is the special instance (since the marginal distribution of a player’s input is identical under the two distributions).

Finally, we need to combine the instances  $I^1, \dots, I^{g_r}$  to create the final instance  $I$ . To do this, we need another problem specific gadget, namely a *re-labeling function*. Roughly speaking, this function takes as input the index  $j^*$ , i.e., the index of the special instances, and instances  $I^1, \dots, I^{g_r}$  and create the final instance  $I$ , while “prioritizing” the role of special instances in  $I$ . By prioritizing we mean that in this step, we need to ensure that the value of  $\mathcal{P}_{s_r}$  on  $I$  is the same as the value of  $\mathcal{P}_{s_{r-1}}$  on the special instances. At the same time, we also need to ensure that this additional relabeling does not reveal the index of the special instance to each individual player, which requires a careful design depending on the problem at hand.

The above family of distributions is parameterized by the sequences  $\{s_r\}$  (size of instances),  $\{p_r\}$  (number of players), and  $\{w_r\}$  (the width parameters), plus the packing and relabeling functions. Our main result in this section is that if these sequences and functions satisfy some natural conditions (similar to what discussed

<sup>4</sup>note that a similar notion to a packing function is captured via a collection of disjoint blocks of vertices in [7] (for finding large matchings), Ruzsa-Szemerédi graphs in [12] (for estimating maximum matching size), and a family of small-intersecting sets in [8] (for finding good allocations in combinatorial auctions). In this work, we use the notion of randomly nearly disjoint set-systems defined in Section 5.1. See the full version of the paper for more details on the connection between this framework and previous work.

<sup>4</sup>For a reader familiar with previous work in [12, 7, 8], we

above), then any  $r$ -round protocol for the problem  $\mathcal{P}_{s_r}$  on the distribution  $\mathcal{D}_r$  requires  $\Omega_r(w_r)$  communication.

We remark that while we state our communication lower bound only in terms of  $w_r$ , to obtain any interesting lower bound using this technique, one needs to ensure that the width parameter  $w_r$  is relatively large in the size of the instance  $s_r$ ; this is also achieved by designing suitable packing and labeling functions (as well as a suitable representation of the problem). However, as “relatively large” depends heavily on the problem at hand, we do not add this requirement to the framework explicitly. A discussion on possible extensions of this framework as well as its connection to previous work appears in the full version of the paper.

## 4.2 The Formal Description of the Framework

We now describe our framework formally. As stated earlier, to use this framework for proving a lower bound for any specific problem  $\mathcal{P}$ , one needs to define appropriate problem-specific gadgets. These gadgets are functions that map multiple instances of  $\mathcal{P}_s$  to a single instance  $\mathcal{P}_{s'}$  for some  $s' \geq s$ . The exact application of these gadgets would become clear shortly in the description of our hard distribution.

**DEFINITION 4.1. (PACKING FUNCTION)** For  $s' \geq s \geq 1$  and  $w \geq 1$ , we refer to a function  $\sigma$  which maps any tuple of instances  $(I_1, \dots, I_w)$  of  $\mathcal{P}_s$  to a single instance  $I$  of  $\mathcal{P}_{s'}$  as a packing function of width  $w$ .

**DEFINITION 4.2. (LABELING FAMILY)** For  $s'' \geq s' \geq 1$  and  $g \geq 1$ , we refer to a family of functions  $\Phi = \{\phi_i\}$ , where each  $\phi_i$  is a function that maps any tuple of instances  $(I^1, \dots, I^g)$  of  $\mathcal{P}_{s'}$  to a single instance  $I$  of  $\mathcal{P}_{s''}$  as a  $g$ -labeling family, and to each function in this family, as a labeling function.

We start by designing the following recursive family of hard distributions  $\{\mathcal{D}_r\}_{r \geq 0}$ , parametrized by sequences  $\{p_r\}_{r \geq 0}$ ,  $\{s_r\}_{r \geq 0}$ , and  $\{w_r\}_{r \geq 0}$ . We require  $\{p_r\}_{r \geq 0}$  and  $\{s_r\}_{r \geq 0}$  to be *increasing* sequences and  $\{w_r\}_{r \geq 0}$  to be *non-increasing*. In two places marked in the distribution, we require one to design the aforementioned problem-specific gadgets for the distribution.

**Distribution  $\mathcal{D}_r$ :** A template hard distribution for  $r$ -round protocols of  $\mathcal{P}$  for any  $r \geq 1$ .

**Parameters:**  $p_r$ : number of players,  $s_r$ : size of the instance,  $w_r$ : width parameter,  $\sigma_r$ : packing function, and  $\Phi_r$ : labeling family.

1. Let  $P$  be the set of  $p_r$  players and define  $g_r := \frac{p_r}{p_{r-1}}$ ; partition the players in  $P$  into  $g_r$  groups

$P_1, \dots, P_{g_r}$  each containing  $p_{r-1}$  players.

2. **Design** a packing function  $\sigma_r$  of width  $w_r$  which maps  $w_r$  instances of  $\mathcal{P}_{s_{r-1}}$  to an instance of  $\mathcal{P}_{s'_r}$  for some  $s_{r-1} \leq s'_r \leq s_r$ .
3. Pick an instance  $I_r^* \sim \mathcal{D}_{r-1}$  over the set of players  $[p_{r-1}]$  and domain of size  $s_{r-1}$ .
4. For each group  $P_i$  for  $i \in [g_r]$ :
  - (a) Pick an index  $j^* \in [w_r]$  uniformly at random and create  $w_r$  instances  $I_1^i, \dots, I_{w_r}^i$  of problem  $\mathcal{P}_{s_{r-1}}$  as follows:
    - (i) Each instance  $I_j^i$  for  $j \in [w_r]$  is over the players  $P_i$  and domain  $D_j^i = \{0, 1\}^{s_{r-1}}$ .
    - (ii) For index  $j^* \in [w_r]$ ,  $I_{j^*}^i = I_r^*$  by mapping (arbitrarily)  $[p_{r-1}]$  to  $P_i$  and domain of  $I_r^*$  to  $D_{j^*}^i$ .
    - (iii) For any other index  $j \neq j^*$ ,  $I_j^i \sim \mathcal{D}_{r-1}^\otimes := \otimes_{q \in P_i} \mathcal{D}_{r-1}(q)$ , i.e., the *product of marginal distribution* of the input to each player  $q \in P_i$  in  $\mathcal{D}_{r-1}$ .
  - (b) Map all the instances  $I_1^i, \dots, I_{w_r}^i$  to a single instance  $I^i$  using the function  $\sigma_r$ .
5. **Design** a  $g_r$ -labeling family  $\Phi_r$  which maps  $g_r$  instances of  $\mathcal{P}_{s'_r}$  to a single instance  $\mathcal{P}_{s_r}$ .
6. Pick a labeling function  $\phi$  from  $\Phi$  uniformly at random and map the  $g_r$  instances  $I^1, \dots, I^{g_r}$  of  $\mathcal{P}_{s'_r}$  to the output instance  $I$  of  $\mathcal{P}_{s_r}$  using  $\phi$ .
7. The input to each player  $q \in P_i$  in the instance  $I$ , for any  $i \in [g_r]$ , is the input of player  $q$  in the instance  $I^i$ , after applying the mapping  $\phi$  to map  $I^i$  to  $I$ .

We remark that in the above distribution, the “variables” in each instance sampled from  $\mathcal{D}_r$  are the instances  $I_1^i, \dots, I_{w_r}^i$  for all groups  $i \in [g_r]$ , the index  $j^* \in [w]$ , and both the choice of labeling family  $\Phi_r$  and the labeling function  $\phi$ . On the other hand, the “constants” across all instances of  $\mathcal{D}_r$  are parameters  $p_r, s_r$ , and  $w_r$ , the choice of grouping  $P_1, \dots, P_{g_r}$ , and the packing function  $\sigma_r$ .

To complete the description of this recursive family of distributions, we need to explicitly define the distribution  $\mathcal{D}_0$  between  $p_0$  players over  $\{0, 1\}^{s_0}$ . We let  $\mathcal{D}_0 := \frac{1}{2} \cdot \mathcal{D}_0^{\text{Yes}} + \frac{1}{2} \cdot \mathcal{D}_0^{\text{No}}$ , where  $\mathcal{D}_0^{\text{Yes}}$  is a distribution over Yes instances of  $\mathcal{P}_{s_0}$  and  $\mathcal{D}_0^{\text{No}}$  is a distribution over No instances. The choice of distributions  $\mathcal{D}_0^{\text{Yes}}$  and  $\mathcal{D}_0^{\text{No}}$



are again *problem-specific*.

We start by describing the main properties of the packing and labeling functions that are required for our lower bound. For any player  $q \in P_i$ , define  $I^i(q) := (I_1^i(q), \dots, I_{w_r}^i(q))$ , where for any  $j \in [w_r]$ ,  $I_j^i(q)$  denotes the input of player  $q$  in the instance  $I_j^i$ . We require the packing and labeling functions to be *locally computable* defined as follows.

**DEFINITION 4.3. (LOCALLY COMPUTABLE)** *We say that the packing function  $\sigma_r$  and the labeling family  $\Phi_r$  are locally computable iff any player  $q \in P_i$  for  $i \in [g_r]$ , can compute the mapping of  $I^i(q)$  to the final instance  $I$ , locally, i.e., only using  $\sigma_r$ , the sampled labeling function  $\phi \in \Phi_r$ , and input  $I^i(q)$ .*

We use  $\phi_q$  to denote the *local mapping* of player  $q \in P_i$  for mapping  $I^i(q)$  to  $I$ ; since  $\sigma_r$  is fixed in the distribution  $\mathcal{D}_r$ , across different instances sampled from  $\mathcal{D}_r$ ,  $\phi_q$  is only a function of  $\phi$ . The input to each player  $q \in P_i$  is *uniquely* determined by  $I^i(q)$  and  $\phi_q$ .

Inside each instance  $I$  sampled from  $\mathcal{D}_r$ , there exists a unique *embedded* instance  $I_r^*$  which is sampled from  $\mathcal{D}_{r-1}$ . Moreover, this instance is essentially “copied”  $g_r$  times, once in each instance  $I_{j^*}^i$  for each group  $P_i$ . We refer to the instance  $I_r^*$  as well as its copies  $I_{j^*}^1, \dots, I_{j^*}^{g_r}$  as *special instances* and to all other instances as *fooling instances*. We require the packing and labeling functions to be *preserving*, defined as follows.

**DEFINITION 4.4. ( $\gamma$ -PRESERVING)** *We say that the packing function and the labeling family are  $\gamma$ -preserving for a parameter  $\gamma \in (0, 1)$ , iff*

$$\Pr_{I \sim \mathcal{D}_r} (\mathcal{P}_{s_r}(I) = \mathcal{P}_{s_{r-1}}(I_r^*)) \geq 1 - \gamma.$$

*In other words, the value of  $\mathcal{P}_{s_r}$  on an instance  $I$  should be equal to the value of  $\mathcal{P}_{s_{r-1}}$  on the embedded special instance  $I_r^*$  of  $I$  w.p.  $1 - \gamma$ .*

Recall that the packing function  $\sigma_r$  is a deterministic function that depends only on the distribution  $\mathcal{D}_r$  itself and not any specific instance (and hence the underlying special instances); on the other hand, the preserving property requires the packing and labeling functions to somehow “prioritize” the special instances over the fooling instances (in determining the value of the original instance). To achieve this property, the labeling family is allowed to vary based on the specific instance sampled from the distribution  $\mathcal{D}_r$ . However, we need to limit the dependence of the labeling family to the underlying instance, which is captured through the definition of *obliviousness* below.

**DEFINITION 4.5.** *We say that the labeling family  $\Phi_r$  is oblivious iff it satisfies the following properties:*

- (i) *The only variable in  $\mathcal{D}_r$  which  $\Phi_r$  can depend on is  $j^* \in [w_r]$  (it can depend arbitrarily on the constants in  $\mathcal{D}_r$ ).*
- (ii) *For any player  $q \in P$ , the local mapping  $\phi_q$  and  $j^*$  are independent of each other in  $\mathcal{D}_r$ .*

Intuitively speaking, Condition (i) above implies that a function  $\phi \in \Phi_r$  can “prioritize” the special instances based on the index  $j^*$ , but it cannot use any further knowledge about the special or fooling instances. For example, one may be able to use  $\phi$  to distinguish special instances from other instances, i.e., determine  $j^*$ , but would not be able to infer whether the special instance is a **Yes** instance or a **No** one only based on  $\phi$ . Condition (ii) on the other hand implies that for each player  $q$ , no information about the special instance is revealed by the local mapping  $\phi_q$ . This means that given the function  $\phi_q$  (and not  $\phi$  as a whole), one is not able to determine  $j^*$ .

Finally, we say that the family of distributions  $\{\mathcal{D}_r\}$  is a  $\gamma$ -hard recursive family, iff (i) it is parameterized by increasing sequences  $\{p_r\}$  and  $\{s_r\}$ , and non-increasing sequence  $\{w_r\}$ , and (ii), the packing and labeling functions in the family are locally computable,  $\gamma$ -preserving, and oblivious. We are now ready to present our main theorem of this section.

**THEOREM 4.1.** *Let  $R \geq 1$  be an integer and suppose  $\{\mathcal{D}_r\}_{r=0}^R$  is a  $\gamma$ -hard recursive family for some  $\gamma \in (0, 1)$ ; for any  $r \leq R$ , any  $r$ -round protocol for  $\mathcal{P}_{s_r}$  on  $\mathcal{D}_r$  which errs w.p. at most  $1/3 - r \cdot \gamma$  requires  $\Omega(w_r/r^4)$  total communication.*

### 4.3 Correctness of the Framework: Proof of

**Theorem 4.1** We first set up some notation. For any  $r$ -round protocol  $\pi$  and any  $\ell \in [r]$ , we use  $\Pi_\ell := (\Pi_{\ell,1}, \dots, \Pi_{\ell,p_r})$  to denote the random variable for the transcript of the message communicated by each player in round  $\ell$  of  $\pi$ . We further use  $\Phi$  (resp.  $\Phi_q$ ) to denote the random variable for  $\phi$  (resp. local mapping  $\phi_q$ ) and  $J$  to denote the random variable for the index  $j^*$ . Finally, for any  $i \in [g_r]$  and  $j \in [w_r]$ ,  $I_j^i$  denotes the random variable for the instance  $I_j^i$ .

We start by stating a simple property of oblivious mapping functions.

**PROPOSITION 4.1.** *For any  $i \in [g_r]$  and any player  $q \in P_i$ , conditioned on input  $(I^i(q), \phi_q)$  to player  $q$ , the index  $j^* \in [w_r]$  is chosen uniformly at random.*

*Proof.* By Condition (ii) of obliviousness in Definition 4.5,  $\Phi_q \perp J$ , and hence  $J \perp \Phi_q = \phi_q$ . Moreover, by Condition (i) of Definition 4.5,  $\Phi_q$  cannot depend on  $I^i(q)$  and hence  $I^i(q) \perp \Phi_q = \phi_q$  also. Now notice

that while the distribution of  $I_j^i$  and  $I_{j^*}^i$  for  $j \neq j^*$ , i.e.,  $\mathcal{D}_{r-1}^\otimes$  and  $\mathcal{D}_{r-1}$  are different, the distribution of  $I_j^i(q)$  and  $I_{j^*}^i(q)$  are identical by definition of  $\mathcal{D}_{r-1}^\otimes$ . As such,  $I^i(q)$  and  $j^*$  are also independent of each other conditioned on  $\Phi_q = \phi_q$ , finalizing the proof.

We show that any protocol with a small communication cost cannot learn essentially any useful information about the special instance  $I_r^*$  in its first round.

**LEMMA 4.6.** *For any deterministic protocol  $\pi$  for  $\mathcal{D}_r$ ,  $\mathbb{I}(I_r^*; \Pi_1 \mid \Phi, J) \leq |\Pi_1|/w_r$ .*

*Proof.* The first step is to show that the information revealed about  $I_r^*$  via  $\Pi_1$  can be partitioned over the messages sent by each individual player about their own input in their special instance.

**CLAIM 4.7.**  $\mathbb{I}(I_r^*; \Pi_1 \mid \Phi, J) \leq \sum_{q \in P} \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi, J)$ .

*Proof.* Intuitively, the claim is true because after conditioning on  $\Phi$  and  $J$ , the input of players become independent of each other on all fooling instances, i.e., every instance except for their copy of  $I_r^*$ . As a result, the messages communicated by one player do not add extra information to messages of another one about  $I_r^*$ . Moreover, since each player  $q$  is observing  $I_r^*(q)$ , the information revealed by this player can only be about  $I_r^*(q)$  and not  $I_r^*$ . We now provide the formal proof.

Recall that  $\Pi_1 = (\Pi_{1,1}, \dots, \Pi_{1,p_r})$ . By chain rule of mutual information,

$$\mathbb{I}(I_r^*; \Pi_1 \mid \Phi, J) \stackrel{\text{Fact 2.1-(4)}}{=} \sum_{q \in P} \mathbb{I}(I_r^*; \Pi_{1,q} \mid \Pi_1^{<q}, \Phi, J).$$

We first show that for each  $q \in P$ ,

$$(4.1) \quad \mathbb{I}(I_r^*; \Pi_{1,q} \mid \Pi_1^{<q}, \Phi, J) \leq \mathbb{I}(I_r^*; \Pi_{1,q} \mid \Phi, J).$$

Recall that, for any player  $q$ ,  $I(q)$  denotes the input to player  $q$  in all instances in which  $q$  is participating, and define  $I(-q)$  as the collection of the inputs to all other players across all instances. We argue that  $I(q) \perp I(-q) \mid I_r^*, \Phi, J$ . The reason is simply because after conditioning on  $I_r^*$ , the only variables in  $I(q)$  and  $I(-q)$  are fooling instances that are sampled from  $\mathcal{D}_{r-1}^\otimes$  which is a product distribution across players. This implies that  $\mathbb{I}(I(q); I(-q) \mid I_r^*, \Phi, J) = 0$  (by Fact 2.1-(2)). Now, notice that the input to each player  $q$  is uniquely identified by  $(I(q), \Phi)$  (by locally computable property in Definition 4.3) and hence conditioned on  $I_r^*, \Phi, J$ , the message  $\Pi_{1,q}$  is a deterministic function of  $I(q)$ . As such, by the data processing inequality (Fact 2.1-(5)), we have that  $\mathbb{I}(\Pi_{1,q}; \Pi_1^{<q} \mid I_r^*, \Phi, J) = 0$ ; by Proposition 2.2, this

implies Eq (4.1) (here, conditioning on  $\Pi_1^{<q}$  in RHS of Eq (4.1) can only decrease the mutual information).

Define  $I_r^*(-q)$  as the input to all players in  $I_r^*$  except for player  $q$ ; hence  $I_r^* = (I_r^*(q), I_r^*(-q))$ . By chain rule of mutual information (Fact 2.1-(4)),

$$\begin{aligned} \mathbb{I}(I_r^*; \Pi_{1,q} \mid \Phi, J) &= \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi, J) \\ &\quad + \mathbb{I}(I_r^*(-q); \Pi_{1,q} \mid I_r^*(q), \Phi, J) \\ &= \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi, J), \end{aligned}$$

since  $\mathbb{I}(I_r^*(-q); \Pi_{1,q} \mid I_r^*(q), \Phi, J) = 0$  as  $\Pi_{1,q}$  is independent of  $I_r^*(-q)$  after conditioning on  $I_r^*(q)$  (and Fact 2.1-(2)). The claim now follows from Eq (4.1) and above equation.  $\blacksquare$

Next, we use a direct-sum style argument to show that as each player is oblivious to the identity of the special instance in the input, the message sent by this player cannot reveal much information about the special instance, unless it is too large.

**CLAIM 4.8.** *For any group  $P_i$  and player  $q \in P_i$ ,  $\mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi, J) \leq |\Pi_{1,q}|/w_r$ .*

*Proof.* We first argue that,

$$(4.2) \quad \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi, J) \leq \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi_q, J).$$

Let  $\Phi = (\Phi_q, \Phi^{-q})$  where  $\Phi^{-q}$  denotes the rest of the mapping function  $\Phi$  beyond  $\Phi_q$ . We have,  $\Pi_{1,q} \perp \Phi^{-q} \mid \Phi_q, J, I_r^*(q)$  since after conditioning on  $J$ ,  $\Phi$  does not depend on any other variable in  $\mathcal{D}_r$  (by obliviousness property in Definition 4.5), and hence the input to player  $q$  and as a result  $\Pi_{1,q}$  are independent of  $\Phi^{-q}$  after conditioning on both  $\Phi_q$  and  $J$ . Eq (4.2) now follows from the independence of  $\Pi_{1,q}$  and  $\Phi^{-q}$  and Proposition 2.2 (as conditioning on  $\Phi^{<q}$  in RHS of Eq (4.2) can only decrease the mutual information).

We can bound the RHS of Eq (4.2) as follows,

$$\begin{aligned} \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi_q, J) &= \mathbb{E}_{j \in [w_r]} \left[ \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi_q, J = j) \right] \\ &= \frac{1}{w_r} \sum_{j=1}^{w_r} \mathbb{I}(I_j^i(q); \Pi_{1,q} \mid \Phi_q, J = j). \end{aligned}$$

Recall that  $j^*$  is chosen uniformly at random from  $[w_r]$  and  $I_r^* = I_j^i$  conditioned on  $J = j$ . Our goal now is to drop the conditioning on the event  $J = j$ . By Definition 4.5,  $\Phi_q$  is independent of  $J = j$ . Moreover,  $I_j^i(q)$  is sampled from  $\mathcal{D}_{r-1}(q)$  (both in  $\mathcal{D}_{r-1}$  and in  $\mathcal{D}_{r-1}^\otimes$ ) and hence is independent of  $J = j$ , even conditioned on  $\Phi_q$ . Finally, by Proposition 4.1, the input to player  $q$  is independent of  $J = j$  and as  $\Pi_{1,q}$  is a deterministic function of the input to player  $q$ ,  $\Pi_{1,q}$

is also independent of  $J = j$ , even conditioned on  $\Phi_q$  and  $I_j^i(q)$ . This means that the joint distribution of  $I_j^i(q), \Pi_{1,q}$ , and  $\Phi_q$  is independent of the event  $J = j$  and hence we can drop this conditioning in the above term, and obtain that,

$$\begin{aligned} & \frac{1}{w_r} \sum_{j=1}^{w_r} \mathbb{I}(I_j^i(q); \Pi_{1,q} \mid \Phi_i, J = j) \\ &= \frac{1}{w_r} \sum_{j=1}^{w_r} \mathbb{I}(I_j^i(q); \Pi_{1,q} \mid \Phi_i) \\ &\leq \frac{1}{w_r} \sum_{j=1}^{w_r} \mathbb{I}(I_j^i(q); \Pi_{1,q} \mid I^{i,<j}(q), \Phi_i) \\ &= \frac{1}{w_r} \cdot \mathbb{I}(I^i(q); \Pi_{1,q} \mid \Phi_i), \end{aligned}$$

where the inequality holds since  $I_j^i(q) \perp I^{i,<j}(q) \mid \Phi_i$  and hence conditioning on  $I^{i,<j}(q)$  can only increase the mutual information by Proposition 2.1. Finally,

$$\begin{aligned} \frac{1}{w_r} \cdot \mathbb{I}(I^i(q); \Pi_{1,q} \mid \Phi_i) &\stackrel{\text{Fact 2.1-(1)}}{\leq} \frac{1}{w_1} \cdot \mathbb{H}(\Pi_{1,q} \mid \Phi_i) \\ &\stackrel{\text{Fact 2.1-(3)}}{\leq} \frac{1}{w_r} \cdot \mathbb{H}(\Pi_{1,q}) \\ &\stackrel{\text{Fact 2.1-(1)}}{\leq} \frac{1}{w_r} \cdot |\Pi_{1,q}|. \quad \blacksquare \end{aligned}$$

Lemma 4.6 now follows from the previous two claims:

$$\begin{aligned} \mathbb{I}(I_r^* \mid \Pi_1 \mid \Phi, J) &\stackrel{\text{Claim 4.7}}{\leq} \sum_{q \in P} \mathbb{I}(I_r^*(q); \Pi_{1,q} \mid \Phi, J) \\ &\stackrel{\text{Claim 4.8}}{\leq} \frac{1}{w_r} \cdot \sum_{q \in P} |\Pi_{1,q}| = \frac{1}{w_r} \cdot |\Pi_1|. \quad \blacksquare \end{aligned}$$

For any tuple  $(\Pi_1, \phi, j)$ , we define the distribution  $\psi(\Pi_1, \phi, j)$  as the distribution of  $I_r^*$  in  $\mathcal{D}_r$  conditioned on  $\Pi_1 = \Pi_1, \Phi = \phi$ , and  $J = j$ . Recall that the original distribution of  $I_r^*$  is  $\mathcal{D}_{r-1}$ . In the following, we show that if the first message sent by the players is not too large, and hence does not reveal much information by about  $I_r^*$  by Lemma 4.6, even after the aforementioned conditioning, distribution of  $I_r^*$  does not change by much in average. Formally,

LEMMA 4.9. *If  $|\Pi_1| = o(w_r/r^4)$ , then*

$$\mathbb{E}_{(\Pi_1, \phi, j)} \left[ \|\psi(\Pi_1, \phi, j) - \mathcal{D}_{r-1}\| \right] = o(1/r^2).$$

*Proof.* [Proof Sketch] Since  $I_r^*$  is independent of  $\phi$  and  $j^*$  in  $\mathcal{D}_r$ , we have  $\mathcal{D}_{r-1} = \text{dist}(I_r^*) = \text{dist}(I_r^* \mid \Phi, J)$ . As such, it suffices to show that  $\text{dist}(I_r^* \mid \Phi, J)$  is close to the distribution of  $\text{dist}(I_r^* \mid \Pi_1, \Phi, J)$ . By Lemma 4.6 and the assumption  $|\Pi_1| = o(w_r/r^4)$ , we know that the information revealed about  $I_r^*$  by  $\Pi_1$ , conditioned on  $\Phi, J$  is quite small, i.e.,  $o(1/r^4)$ . This intuitively means that having an extra knowledge of  $\Pi_1$  would not be able to change the distribution of  $I_r^*$  by much. To make this formal, we use the connection between mutual information and KL-divergence (Fact 2.2) between the two distributions above and use Pinsker's inequality (Fact 2.3) to relate the KL-divergence to total variation distance and obtain the final bound. We defer this simple calculation to the full version of the paper.  $\blacksquare$

Define the recursive function  $\delta(r) := \delta(r-1) - o(1/r^2) - \gamma$  with base  $\delta(0) = 1/2$ . We have,

LEMMA 4.10. *For any deterministic  $\delta(r)$ -error  $r$ -round protocol  $\pi$  for  $\mathcal{D}_r$ , we have  $\|\pi\| = \Omega(w_r/r^4)$ .*

*Proof.* Proof is by induction on the number of rounds.

**Base case:** The base case refers to 0-round protocols for  $\mathcal{D}_0$ , i.e., protocols that are not allowed any communication. As in the distribution  $\mathcal{D}_0$ , Yes and No instances happen w.p. 1/2 each and the coordinator has no input, any 0-round protocol can only output the correct answer w.p. 1/2, proving the induction base.

**Induction step:** Suppose the lemma holds for all integers up to  $r$  and we prove it for  $r$ -round protocols. The proof is by contradiction. Given an  $r$ -round protocol  $\pi_r$  violating the induction hypothesis, we create an  $(r-1)$ -round protocol  $\pi_{r-1}$  which also violates the induction hypothesis, a contradiction. Given an instance  $I_{r-1}$  of  $\mathcal{P}_{s_{r-1}}$  over players  $P^{r-1}$  and domain  $\mathcal{D}^{r-1} = \{0, 1\}^{s_{r-1}}$ , the protocol  $\pi_{r-1}$  works as follows:

1. Let  $P^r = [p_r]$  and partition  $P^r$  into  $g_r$  equal-size groups  $P_1, \dots, P_{g_r}$  as is done in  $\mathcal{D}_r$ . Create an instance  $I_r$  of  $\mathcal{D}_r$  as follows:
2. Using *public randomness*, the players in  $P^{r-1}$  sample  $R := (\Pi_1, \phi, j^*) \sim (\text{dist}(\pi_r), \mathcal{D}_r)$ , i.e., from the (joint) distribution of protocol  $\pi_r$  over distribution  $\mathcal{D}_r$ .
3. The  $q$ -th player in  $P^{r-1}$  (in instance  $I_{r-1}$ ) mimics the role of the  $q$ -th player in each group  $P_i$  for  $i \in [g_r]$  in  $I_r$ , denoted by player  $(i, q)$ , as follows:
  - (a) Set the input for  $(i, q)$  in the special instance  $I_{j^*}^i(q)$  of  $I_r$  as the original input of  $q$  in  $I_{r-1}$ , i.e.,  $I_{r-1}(q)$  mapped via  $\sigma_r$  and  $\phi$  to  $I$  (as

is done in  $I_r$  to the domain  $D_{j^*}^i$ ). This is possible by the locally computable property of  $\sigma_r$  and  $\phi$  in Definition 4.3.

- (b) Sample the input for  $(i, q)$  in all the fooling instances  $I_j^i(q)$  of  $I_r$  for any  $j \neq j^*$  using *private randomness* from the *correlated* distribution  $\mathcal{D}_r \mid (I_r^* = I_{r-1}, (\Pi_1, \Phi, \mathbf{J}) = R)$ . This sampling is possible by Proposition 4.2 below.

4. Run the protocol  $\pi_r$  from the second round onwards on  $I_r$  assuming that in the first round the communicated message was  $\Pi_1$  and output the same answer as  $\pi_r$ .

In Line (3b), the distribution the players are sampling from depends on  $\Pi_1, \phi, j^*$  which are public knowledge (through sampling via public randomness), as well as  $I_r^*$  which is *not* a public information as each player  $q$  only knows  $I_r^*(q)$  and not all of  $I_r^*$ . Moreover, while random variables  $I_j^i(q)$  (for  $j \neq j^*$ ) are originally independent across different players  $q$  (as they are sampled from the product distribution  $\mathcal{D}_{r-1}^{\otimes}$ ), conditioning on the first message of the protocol, i.e.,  $\Pi_1$  correlates them, and hence a-priori it is not clear whether the sampling in Line (3b) can be done without any further communication. Nevertheless, we prove that this is the case and to sample from the distribution in Line (3b), each player only needs to know  $I_r^*(q)$  and not  $I_r^*$ .

**PROPOSITION 4.2.** *Suppose  $\mathsf{l}$  is the collection of all instances in the distribution  $\mathcal{D}_r$  and  $\mathsf{l}(q)$  is the input to player  $q$  in instances in which  $q$  participates; then,*

$$\begin{aligned} \text{dist}(\mathsf{l} \mid I_r^* = I_{r-1}, (\Pi_1, \Phi, \mathbf{J}) = R) \\ = \mathbb{X}_{q \in P} \text{dist}(\mathsf{l}(q) \mid I_r^*(q) = I_{r-1}(q), (\Pi_1, \Phi, \mathbf{J}) = R). \end{aligned}$$

*Proof.* Fix any player  $q \in P$ , and recall that  $\mathsf{l}(-q)$  is the collection of the inputs to all players other than  $q$  across all instances (special and fooling). We prove that  $\mathsf{l}(q) \perp \mathsf{l}(-q) \mid (I_r^*(q), \Pi_1, \Phi, \mathbf{J})$  in  $\mathcal{D}_r$ , which immediately implies the result. To prove this claim, by Fact 2.1-(2), it suffices to show that  $\mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Pi_1, \Phi, \mathbf{J}) = 0$ . Define  $\Pi_1^{-q}$  as the set of all messages in  $\Pi_1$  except for the message of player  $q$ , i.e.,  $\Pi_{1,q}$ . We have,

$$\begin{aligned} \mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Pi_1, \Phi, \mathbf{J}) \\ \leq \mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Pi_{1,q}, \Phi, \mathbf{J}), \end{aligned}$$

since  $\mathsf{l}(q) \perp \Pi_1^{-q} \mid \mathsf{l}(-q), I_r^*(q), \Pi_{1,q}, \Phi, \mathbf{J}$  as the input to players  $P \setminus \{q\}$  is uniquely determined by  $\mathsf{l}(-q), \Phi$  (by the locally computable property in Definition 4.3) and hence  $\Pi_1^{-q}$  is deterministic after the conditioning; this

independence means that conditioning on  $\Pi_1^{-q}$  in the RHS above can only decrease the mutual information by Proposition 2.2. We can further bound the RHS above by,

$$\begin{aligned} \mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Pi_{1,q}, \Phi, \mathbf{J}) \\ \leq \mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Phi, \mathbf{J}), \end{aligned}$$

since  $\mathsf{l}(-q) \perp \Pi_{1,q} \mid \mathsf{l}(q), I_r^*(q), \Phi, \mathbf{J}$  as the input to player  $q$  is uniquely determined by  $\mathsf{l}(q), \Phi$  (again by Definition 4.3) and hence after the conditioning,  $\Pi_{1,q}$  is deterministic; this implies that conditioning on  $\Pi_{1,q}$  in RHS above can only decrease the mutual information by Proposition 2.2. Finally, observe that  $\mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Phi, \mathbf{J}) = 0$  by Fact 2.1-(2), since after conditioning on  $I_r^*(q)$ , the only remaining instances in  $\mathsf{l}(q)$  are fooling instances which are sampled from the distribution  $\mathcal{D}_{r-1}^{\otimes}$  which is independent across the players. This implies that  $\mathbb{I}(\mathsf{l}(q); \mathsf{l}(-q) \mid I_r^*(q), \Pi_1, \Phi, \mathbf{J}) = 0$  also which finalizes the proof. ■

Having proved Proposition 4.2, it is now easy to see that  $\pi_{r-1}$  is indeed a valid  $r - 1$  round protocol for distribution  $\mathcal{D}_{r-1}$ : each player  $q$  can perform the sampling in Line (3b) without any communication as  $(I^*(q), \Pi_1, \Phi, \mathbf{J})$  are all known to  $q$ ; this allows the players to simulate the first round of protocol  $\pi_r$  without any communication and hence only need  $r - 1$  rounds of communication to compute the answer of  $\pi_r$ . We can now prove that,

**CLAIM 4.11.** *Assuming  $\pi_r$  is a  $\delta$ -error protocol on  $\mathcal{D}_r$ ,  $\pi_{r-1}$  would be a  $(\delta + \gamma + o(\frac{1}{r^2}))$ -error protocol on  $\mathcal{D}_{r-1}$ .*

*Proof.* [Proof Sketch] Our goal is to calculate the probability that  $\pi_{r-1}$  errs on an instance  $I_{r-1} \sim \mathcal{D}_{r-1}$ . For the sake of analysis, suppose that  $I_{r-1}$  is instead sampled from the distribution  $\psi$  for a randomly chosen tuple  $(\Pi_1, \phi, j^*)$  (defined before Lemma 4.9). Notice that by Lemma 4.9, these two distributions are quite close to each other in total variation distance, and hence if  $\pi_{r-1}$  has a small error on distribution  $\psi$  it would necessarily have a small error on  $\mathcal{D}_{r-1}$  as well (by Fact 2.4).

Using Proposition 4.2, it is easy to verify that if  $I_{r-1}$  is sampled from  $\psi$ , then the instance  $I_r$  constructed by  $\pi_{r-1}$  is sampled from  $\mathcal{D}_r$  and moreover  $I_r^* = I_{r-1}$ . As such, since (i)  $\pi_r$  is a  $\delta$ -error protocol for  $\mathcal{D}_r$ , (ii) the answer to  $I_r$  and  $I_r^* = I_{r-1}$  are the same w.p.  $1 - \gamma$  (by  $\gamma$ -preserving property in Definition 4.4), and (iii)  $\pi_{r-1}$  outputs the same answer as  $\pi_r$ , protocol  $\pi_{r-1}$  is a  $(\delta + \gamma)$ -error protocol for  $\psi$ . We defer the formal proof and detailed calculation of this probability of error to the full version of the paper. ■

We are now ready to finalize the proof of Lemma 4.10. Suppose  $\pi_r$  is a deterministic  $\delta(r)$ -error protocol for  $\mathcal{D}_r$  with communication cost  $\|\pi_r\| = o(w_r/r^4)$ . By Claim 4.11,  $\pi_{r-1}$  would be a randomized  $\delta(r-1)$ -error protocol for  $\mathcal{D}_{r-1}$  with  $\|\pi_{r-1}\| \leq \|\pi_r\|$  (as  $\delta(r-1) = \delta(r) + \gamma + o(1/r^2)$ ). By an averaging argument, we can fix the randomness in  $\pi_{r-1}$  to obtain a deterministic protocol  $\pi'_{r-1}$  over the distribution  $\mathcal{D}_{r-1}$  with the same error  $\delta(r-1)$  and communication of  $\|\pi'_{r-1}\| = o(w_r/r^4) = o(w_{r-1}/r^4)$  (as  $\{w_r\}_{r \geq 0}$  is a non-increasing sequence). But such a protocol contradicts the induction hypothesis for  $(r-1)$ -round protocols, finalizing the proof.

*Proof.* [Proof of Theorem 4.1] By Lemma 4.10, any deterministic  $\delta(r)$ -error  $r$ -round protocol for  $\mathcal{D}_r$  requires  $\Omega(w_r/r^4)$  total communication. This immediately extends to randomized protocols by an averaging argument, i.e., the easy direction of Yao's minimax principle [62]. The statement in the theorem now follows from this since for any  $r \geq 0$ ,  $\delta(r) = \delta(r-1) - \gamma - o(1/r^2) = \delta(0) - r \cdot \gamma - \sum_{\ell=1}^r o(1/\ell^2) = 1/2 - r \cdot \gamma - o(1) > 1/3 - r \cdot \gamma$  (as  $\delta(0) = 1/2$  and  $\sum_{\ell=1}^r 1/\ell^2$  is a converging series and hence is bounded by some absolute constant independent of  $r$ ).

## 5 Distributed Lower Bound for Max-Coverage

We prove our main lower bound for maximum coverage in this section, formalizing Result 1.

**THEOREM 5.1.** *For integers  $1 \leq r, c \leq o\left(\frac{\log k}{\log \log k}\right)$  with  $c \geq 4r$ , any  $r$ -round protocol for the maximum coverage problem that can approximate the value of optimal solution to within a factor of better than  $\left(\frac{1}{2c} \cdot \frac{k^{1/2r}}{\log k}\right)$  w.p. at least  $3/4$  requires  $\Omega\left(\frac{k}{r^4} \cdot m^{\frac{c}{(c+2) \cdot 4r}}\right)$  communication per machine. The lower bound applies to instances with  $m$  sets,  $n = m^{1/\Theta(c)}$  elements, and  $k = \Theta(n^{2r/(2r+1)})$ .*

The proof is based on an application of Theorem 4.1. In the following, let  $c \geq 1$  be any integer (as in Theorem 5.1) and  $N \geq 12c^2$  be a sufficiently large integer which we use to define the main parameters for our problem. To invoke Theorem 4.1, we need to instantiate the recursive family of distributions  $\{\mathcal{D}_r\}_{r=0}^c$  in Section 4 with appropriate sequences and gadgets for the maximum coverage problem. We first define sequences (for all  $0 \leq r \leq c$ ):

$$\begin{aligned} k_r &= p_r = (N^2 - N)^r, & n_r &= N^{2r+1}, \\ m_r &= (N^c \cdot (N^2 - N))^r, & w_r &= N^c, \\ g_r &= (N^2 - N). \end{aligned}$$

Here,  $m_r$ ,  $n_r$ , and  $k_r$ , respectively represent the number of sets and elements and the parameter  $k$  in the maximum coverage problem in the instances of each distribution  $\mathcal{D}_r$  and together can identify the size of each instance (i.e., the parameter  $s_r$  defined in Section 4 for the distribution  $\mathcal{D}_r$ ). Moreover,  $p_r$ ,  $w_r$  and  $g_r$  represent the number of players, the width parameter, and the number of groups in  $\mathcal{D}_r$ , respectively (notice that  $g_r = p_r/p_{r-1}$  as needed in distribution  $\mathcal{D}_r$ ).

Using the sequences above, we define:

**coverage( $N, r$ ):** the problem of deciding whether the optimal  $k_r$  cover of universe  $[n_r]$  with  $m_r$  input sets is at least  $(k_r \cdot N)$  (Yes case), or at most  $(k_r \cdot 2c \cdot \log(N^{2r}))$  (No case).

Notice that there is a gap of roughly  $N \approx k_r^{1/2r}$  (ignoring the lower order terms) between the value of the optimal solution in Yes and No cases of coverage( $N, r$ ). We prove a lower bound for deciding between Yes and No instances of coverage( $N, r$ ), when the input sets are partitioned between the players, which implies an identical lower bound for algorithms that can approximate the value of optimal solution in maximum coverage to within a factor smaller than (roughly)  $k_r^{1/2r}$ .

Recall that to use the framework introduced in Section 4, one needs to define two problem-specific gadgets, i.e., a packing function, and a labeling family. In the following section, we design a crucial building block for our packing function.

**RND Set-Systems.** Our packing function is based on the following set-system.

**DEFINITION 5.1.** *For integers  $N, r, c \geq 1$ , an  $(N, r, c)$ -randomly nearly disjoint (RND) set-system over a universe  $\mathcal{X}$  of  $N^{2r}$  elements, is a collection  $\mathcal{S}$  of subsets of  $\mathcal{X}$  satisfying the following properties:*

- (i) *Each set  $A \in \mathcal{S}$  is of size  $N^{2r-1}$ .*
- (ii) *Fix any set  $B \in \mathcal{S}$  and suppose  $\mathcal{C}_B$  is a collection of  $N^{c \cdot r}$  subsets of  $\mathcal{X}$  whereby each set in  $\mathcal{C}_B$  is chosen by picking an arbitrary set  $A \neq B$  in  $\mathcal{S}$ , and then picking an  $N$ -subset uniformly at random from  $A$  (we do not assume independence between the sets in  $\mathcal{C}_B$ ). Then,*

$$\begin{aligned} \Pr\left(\exists S \in \mathcal{C}_B \text{ s.t. } |S \cap B| \geq 2c \cdot r \cdot \log N\right) \\ = o(1/N^3). \end{aligned}$$

*Intuitively, this means that any random  $N$ -subset of some set  $A \in \mathcal{S}$  is essentially disjoint from any other set  $B \in \mathcal{S}$  w.h.p.*

We prove an existence of large RND set-systems in the full version of the paper.

LEMMA 5.2. *For integers  $1 \leq r \leq c$  and sufficiently large integer  $N \geq c$ , there exists an  $(N, r, c)$ -RND set-system  $\mathcal{S}$  of size  $N^c$  over any universe  $\mathcal{X}$  of size  $N^{2r}$ .*

**5.1 Proof of Theorem 5.1** To prove Theorem 5.1 using our framework in Section 4, we parameterize the recursive family of distributions  $\{\mathcal{D}_r\}_{r=0}^c$  for the coverage problem, i.e.,  $\text{coverage}(N, r)$ , with the aforementioned sequences plus the packing and labeling functions which we define below.

**Packing function  $\sigma_r$ :** Mapping instances  $I_1^i, \dots, I_{w_r}^i$  each over  $n_{r-1} = N^{2r-1}$  elements and  $m_{r-1}$  sets for any group  $i \in [g_r]$  to a single instance  $I^i$  on  $N^{2r}$  elements and  $w_r \cdot m_{r-1}$  sets.

1. Let  $\mathcal{A} = \{A_1, \dots, A_{w_r}\}$  be an  $(N, r, c)$ -RND system with  $w_r = N^c$  sets over some universe  $\mathcal{X}_i$  of  $N^{2r}$  elements (guaranteed to exist by Lemma 5.2 since  $c < N$ ). By definition of  $\mathcal{A}$ , for any set  $A_j \in \mathcal{A}$ ,  $|A_j| = N^{2r-1} = n_{r-1}$ .
2. Return the instance  $I$  over the universe  $\mathcal{X}_i$  with the collection of all sets in  $I_1^i, \dots, I_{w_r}^i$  after mapping the elements in  $I_j^i$  to  $A_j$  arbitrarily.

We now define the labeling family  $\Phi_r$  as a function of the index  $j^* \in [w_r]$  of special instances.

**Labeling family  $\Phi_r$ :** Mapping instances  $I^1, \dots, I^{g_r}$  over  $N^{2r}$  elements to a single instance  $I$  on  $n_r = N^{2r+1}$  elements and  $m_r$  sets.

1. Let  $j^* \in [w_r]$  be the index of the special instance in the distribution  $\mathcal{D}_r$ . For each permutation  $\pi$  of  $[N^{2r+1}]$  we have a unique function  $\phi(j^*, \pi)$  in the family.
2. For any instance  $I^i$  for  $i \in [g_r]$ , map the elements in  $\mathcal{X}_i \setminus A_{j^*}$  to  $\pi(1, \dots, N^{2r} - N^{2r-1})$  and the elements in  $A_{j^*}$  to  $\pi(N^{2r} + (g_r - 1) \cdot N^{2r-1}) \dots \pi(N^{2r} + g_r \cdot N^{2r-1} - 1)$ .
3. Return the instance  $I$  over the universe  $[N^{2r+1}]$  which consists of the collection of all sets in  $I^1, \dots, I^{g_r}$  after the mapping above.

Finally, we define the base case distribution  $\mathcal{D}_0$  of the recursive family  $\{\mathcal{D}_r\}_{r=0}^c$ . By definition of our sequences, this distribution is over  $p_0 = 1$  player,  $n_0 = N$  elements, and  $m_0 = 1$  set.

**Distribution  $\mathcal{D}_0$ :** The base case of the recursive family of distributions  $\{\mathcal{D}_r\}_{r=0}^c$ .

1. W.p. 1/2, the player has a single set of size  $N$  covering the universe (the **Yes** case).
2. W.p. 1/2, the player has a single set  $\{\emptyset\}$ , i.e., a set that covers no elements (the **No** case).

To invoke Theorem 4.1, we prove that this family is a  $\gamma$ -hard recursive family for the parameter  $\gamma = o(r/N)$ . The sequences clearly satisfy the required monotonicity properties. It is also straightforward to verify that  $\sigma_r$  and functions  $\phi \in \Phi_r$  are *locally computable* (Definition 4.3): both functions are specifying a mapping of elements to the new instance and hence each player can compute its final input by simply mapping the original input sets according to  $\sigma_r$  and  $\phi$  to the new universe. In other words, the local mapping of each player  $q \in P_i$  only specifies which element in the instance  $I$  corresponds to which element in  $I_j^i(q)$  for  $j \in [w_r]$ . It thus remains to prove the *preserving* and *obliviousness* property of the packing and labeling functions.

We start by showing that the labeling family  $\Phi_r$  is oblivious. The first property of Definition 4.5 is immediate to see as  $\Phi_r$  is only a function of  $j^*$  and  $\sigma_r$ . For the second property, consider any group  $P_i$  and instance  $I^i$ ; the labeling function never maps two elements belonging to a *single* instance  $I^i$  to the same element in the final instance (there are however overlaps between the elements across different groups). Moreover, picking a uniformly at random labeling function  $\phi$  from  $\Phi_r$  (as is done in  $\mathcal{D}_r$ ) results in mapping the elements in  $I^i$  according to a *random permutation*; as such, the set of elements in instance  $I^i$  is mapped to a uniformly at random chosen subset of the elements in  $I$ , *independent* of the choice of  $j^*$ . As the local mapping  $\phi_q$  of each player  $q \in P_i$  is only a function of the set of elements to which elements in  $I^i$  are mapped to,  $\phi_q$  is also independent of  $j^*$ , proving that  $\Phi_r$  is indeed oblivious.

The rest of this section is devoted to the proof of the preserving property of the packing and labeling functions defined for maximum coverage. We first make some observations about the instances created in  $\mathcal{D}_r$ . Recall that the special instances in the distribution are  $I_{j^*}^1, \dots, I_{j^*}^{g_r}$ . After applying the packing function, each instance  $I_{j^*}^i$  is supported on the set of elements  $A_{j^*}$ . After additionally applying the labeling function,  $A_{j^*}$  is mapped to a unique set of elements in  $I$  (according to the underlying permutation  $\pi$  in  $\phi$ ); as a result,

OBSERVATION 5.3. *The elements in the special instances  $I_{j^*}^1, \dots, I_{j^*}^{g_r}$  are mapped to disjoint set of ele-*

ments in the final instance.

The input to each player  $q \in P_i$  in an instance of  $\mathcal{D}_r$  is created by mapping the sets in instances  $I_1^i, \dots, I_{w_r}^i$  (which are all sampled from distributions  $\mathcal{D}_{r-1}$  or  $\mathcal{D}_{r-1}^\otimes$ ) to the final instance  $I$ . As the packing and labeling functions, by construction, never map two elements belonging to the same instance  $I_j^i$  to the same element in the final instance, the size of each set in the input to player  $q$  is equal across any two distributions  $\mathcal{D}_r$  and  $\mathcal{D}_{r'}$  for  $r \neq r'$ , and thus is  $N$  by definition of  $\mathcal{D}_0$  (we ignore empty sets in  $\mathcal{D}_0$  as one can consider them as not giving any set to the player instead; these sets are only added to simplify that math). Moreover, as argued earlier, the elements are being mapped to the final instance according to a random permutation.

**OBSERVATION 5.4.** *For any group  $P_i$ , any player  $q \in P_i$ , the distribution of any single input set to player  $q$  in the final instance  $I \sim \mathcal{D}_r$  is uniform over all  $N$ -subsets of the universe. This also holds for an instance  $I \sim \mathcal{D}_r^\otimes$  as marginal distribution of a player input is identical.*

We now prove the preserving property in the following two lemmas. The proof of the first lemma is quite simple and is deferred to the full version.

**LEMMA 5.5.** *For any instance  $I \sim \mathcal{D}_r$ ; if  $I_r^*$  is a Yes instance, then  $I$  is also a Yes instance.*

We now prove the more involved case.

**LEMMA 5.6.** *For any instance  $I \sim \mathcal{D}_r$ ; if  $I_r^*$  is a No instance, then w.p. at least  $1 - 1/N$ ,  $I$  is also a No instance.*

*Proof.* Let  $U$  be the universe of elements in  $I$  and  $U^* \subseteq U$  be the set of elements to which the elements in special instances  $I_{j^*}^1, \dots, I_{j^*}^{g_r}$  are mapped to (these are all elements in  $U$  except for the first  $N^{2r}$  elements according to the permutation  $\pi$  in the labeling function  $\phi$ ). In the following, we bound the contribution of each set in players inputs in covering  $U^*$  and then use the fact that  $|U \setminus U^*|$  is rather small to finalize the proof.

For any group  $P_i$  for  $i \in [g_r]$ , let  $U_i$  be the set of all elements across instances in which the players in  $P_i$  are participating in. Moreover, define  $U_i^* := U^* \cap U_i$ ; notice that  $U_i^*$  is precisely the set of elements in the special instance  $I_{j^*}^i$ . We first bound the contribution of special instances. The proof is deferred to the full version.

**CLAIM 5.7.** *If  $I_r^*$  is a No instance, then for any integer  $\ell \geq 0$ , any collection of  $\ell$  sets from the special instances  $I_{j^*}^1, \dots, I_{j^*}^{g_r}$  can cover at most  $k_r + \ell \cdot (2c \cdot \log N^{2r-2})$  elements in  $U^*$ .*

We also bound the contribution of fooling instances using the RND set-systems properties. We defer the proof of this claim the full version of the paper.

**CLAIM 5.8.** *With probability  $1 - o(1/N)$  in the instance  $I$ , simultaneously for all integers  $\ell \geq 0$ , any collection of  $\ell$  sets from the fooling instances  $\{I_j^i \mid i \in [g_r], j \in [w_r] \setminus \{j^*\}\}$  can cover at most  $\ell \cdot r \cdot (2c \cdot \log N)$  elements in  $U^*$ .*

In the following, we condition on the event in Claim 5.8, which happens w.p. at least  $1 - 1/N$ . Let  $\mathcal{C} = \mathcal{C}_s \cup \mathcal{C}_f$  be any collection of  $k_r$  sets (i.e., a potential  $k_r$ -cover) in the input instance  $I$  such that  $\mathcal{C}_s$  are  $\mathcal{C}_f$  are chosen from the special instances and fooling instances, respectively. Let  $\ell_s = |\mathcal{C}_s|$  and  $\ell_f = |\mathcal{C}_f|$ ; we have,

$$\begin{aligned} |c(\mathcal{C})| &= |c(\mathcal{C}) \cap U^*| + |c(\mathcal{C}) \cap (U \setminus U^*)| \\ &\leq |c(\mathcal{C}_s) \cap U^*| + |c(\mathcal{C}_f) \cap U^*| + |U \setminus U^*| \\ &\leq k_r + \ell_s \cdot (2r - 2) \cdot 2c \cdot \log N + \ell_f \cdot r \cdot 2c \cdot \log N + N^{2r} \end{aligned}$$

(by Claim 5.7 for the first term and Claim 5.8 for the second term)

$$\begin{aligned} &\leq 4k_r + k_r \cdot (2r - 2) \cdot 2c \cdot \log N \\ (2k_r \geq N^{2r}) & \\ &\leq k_r \cdot 2r \cdot 2c \cdot \log N \leq k_r \cdot 2c \cdot \log N^{2r}. \end{aligned}$$

This means that w.p. at least  $1 - 1/N$ ,  $I$  is also a No instance. ■

Lemmas 5.5 and 5.6 prove that the packing function  $\sigma_r$  and labeling family  $\Phi_r$  are  $\gamma$ -preserving for  $\gamma = 1/N$ .

*Proof.* [Proof of Theorem 5.1] The results in this section imply that the family of distributions  $\{\mathcal{D}_r\}_{r=0}^c$  for the  $\text{coverage}(N, r)$  are  $\gamma$ -hard for the parameter  $\gamma = 1/N$ , as long as  $r \leq 4c \leq 4\sqrt{N}/12$ . Consequently, by Theorem 4.1, any  $r$ -round protocol that can compute the value of  $\text{coverage}(N, r)$  on  $\mathcal{D}_r$  w.p. at least  $2/3 + r \cdot \gamma = 2/3 + r/N < 3/4$  requires  $\Omega(w_r/r^4) = \Omega(N^c/r^4)$  total communication. Recall that the gap between the value of optimal solution between Yes and No instances of  $\text{coverage}(N, r)$  is at least  $N / (2c \cdot \log(N^{2r})) \geq \left(\frac{k_r^{1/2r}}{2c \cdot \log k_r}\right)$ . As such, any  $r$ -round distributed algorithm that can approximate the value of optimal solution to within a factor better than this w.p. at least  $3/4$  can distinguish between Yes and No cases of this distribution, and hence requires  $\Omega(N^{c-2r}/r^4) = \Omega\left(\frac{k_r}{r^4} \cdot m^{\frac{c}{(c+2) \cdot 4r}}\right)$  per player communication. Finally, since  $N \leq 2k_r^{1/2r}$ , the condition  $c \leq \sqrt{N}/12$  holds as long as  $c = o\left(\frac{\log k_r}{\log \log k_r}\right)$ , finalizing the proof. ■

## 6 Distributed Algorithms for Max-Coverage

In this section, we show that both the *round-approximation* tradeoff and the *round-communication*

tradeoff achieved by our lower bound in Theorem 5.1 are essentially tight, formalizing Result 2.

**6.1 An  $O(r \cdot k^{1/r})$ -Approximation Algorithm** Recall that Theorem 5.1 shows that getting better than  $k^{\Omega(1/r)}$  approximation in  $r$  rounds requires a relatively large communication of  $m^{\Omega(1/r)}$ , (potentially) larger than any  $\text{poly}(n)$ . In this section, we prove that this round-approximation tradeoff is essentially tight by showing that one can always obtain a  $k^{O(1/r)}$  approximation (with a slightly larger constant in the exponent) in  $r$  rounds using a limited communication of nearly linear in  $n$ .

**THEOREM 6.1.** *There exists a deterministic distributed algorithm for the maximum coverage problem that for any integer  $r \geq 1$  computes an  $O(r \cdot k^{1/r+1})$  approximation in  $r$  rounds and  $\tilde{O}(n)$  communication per each machine.*

On a high level, our algorithm follows an iterative sketching method: in each round, each machine computes a small collection  $\mathcal{C}_i$  of its input sets  $\mathcal{S}_i$  as a sketch and sends it to the coordinator. The coordinator is maintaining a collection of sets  $\mathcal{X}$  and updates it by iterating over the received sketches and picking any set that still has a relatively large contribution to this partial solution. The coordinator then communicates the set of elements covered by  $\mathcal{X}$  to the machines and the machines update their inputs accordingly and repeat this process. At the end, the coordinator returns (a constant approximation to) the optimal  $k$ -cover over the collection of *all* received sets across different rounds.

In the following, we assume that our algorithm is given a value  $\widetilde{\text{opt}}$  such that  $\text{opt} \leq \widetilde{\text{opt}} \leq 2 \cdot \text{opt}$ . We can remove this assumption by guessing the value of  $\widetilde{\text{opt}}$  in powers of two (up to  $n$ ) and solve the problem simultaneously for all of them and return the best solution, which increases the communication cost by only an  $O(\log n)$  factor.

We first introduce the algorithm for computing the sketch on each machine.

**GreedySketch**( $U, \mathcal{S}, \tau$ ). An algorithm for computing the sketch of each machine's input.

**Input:** A collection  $\mathcal{S}$  of sets from  $[n]$ , a target universe  $U \subseteq [n]$ , and a threshold  $\tau$ .

**Output:** A collection  $\mathcal{C}$  of subsets of  $U$ .

1. Let  $\mathcal{C} = \emptyset$  initially.
2. Iterate over the sets in  $\mathcal{S}$  in an arbitrary order and for each set  $S \in \mathcal{S}$ , if  $|(S \cap U) \setminus c(\mathcal{C})| \geq \tau$ , then add  $(S \cap U) \setminus c(\mathcal{C})$  to  $\mathcal{C}$ .

3. Return  $\mathcal{C}$  as the answer.

Notice that in the Line (2) of GreedySketch, we are adding the *new contribution* of the set  $S$  and not the complete set itself. This way, we can bound the total representation size of the output collection  $\mathcal{C}$  by  $\tilde{O}(n)$  (as each element in  $U$  appears in at most one set). We now present our algorithm in Theorem 6.1.

**Iterative Sketching Greedy Algorithm (ISGreedy):**

**Input:** A collection  $\mathcal{S}_i$  of subsets of  $[n]$  for each machine  $i \in [p]$  and a value  $\widetilde{\text{opt}} \in [\text{opt}, 2 \cdot \text{opt}]$ .

**Output:** A  $k$ -cover from the sets in  $\mathcal{S} := \bigcup_{i \in [p]} \mathcal{S}_i$ .

1. Let  $\mathcal{X}^0 = \emptyset$  and  $U_i^0 = [n]$ , for each  $i \in [p]$  initially. Define  $\tau := \widetilde{\text{opt}}/4r \cdot k$ .
2. For  $j = 1$  to  $r$  rounds:
  - (a) Each machine  $i$  computes  $\mathcal{C}_i^j = \text{GreedySketch}(U_i^{j-1}, \mathcal{S}_i, \tau)$  and sends it to coordinator.
  - (b) The coordinator sets  $\mathcal{X}^j = \mathcal{X}^{j-1}$  initially and iterates over the sets in  $\bigcup_{i \in [p]} \mathcal{C}_i^j$ , in decreasing order of  $|c(\mathcal{C}_i^j)|$  over  $i$  (and consistent with the order in GreedySketch for each particular  $i$ ), and adds each set  $S$  to  $\mathcal{X}^j$  if  $|S \setminus c(\mathcal{X}^j)| \geq \frac{1}{k^{1/r+1}} \cdot |S|$ .
  - (c) The coordinator communicates  $c(\mathcal{X}^j)$  to each machine  $i$  and the machine updates its input by setting  $U_i^j = c(\mathcal{C}_i^j) \setminus c(\mathcal{X}^j)$ .
3. At the end, the coordinator returns the best  $k$ -cover among all sets in  $\mathcal{C} := \bigcup_{i \in [p], j \in [r]} \mathcal{C}_i^j$  sent by the machines over all rounds.

The round complexity of ISGreedy is trivially  $r$ . For its communication cost, notice that at each round, each machine is communicating at most  $\tilde{O}(n)$  bits and the coordinator communicates  $\tilde{O}(n)$  bits back to each machine. As the number of rounds never needs to be more than  $O(\log k)$ , we obtain that ISGreedy requires  $\tilde{O}(n)$  communication per each machine. Therefore, it only remains to analyze the approximation guarantee of this algorithm. To do so, it suffices to show that,

**LEMMA 6.1.** *Define  $\mathcal{C} := \bigcup_{i \in [p], j \in [r]} \mathcal{C}_i^j$ . The optimal  $k$ -cover of  $\mathcal{C}$  covers  $(\frac{\text{opt}}{4r \cdot k^{1/r+1}})$  elements.*

We defer the proof of this key lemma to the full version of the paper. Theorem 6.1 follows from Lemma 6.1



as the coordinator can simply run any constant factor approximation algorithm for maximum coverage on the collection  $\mathcal{C}$  and obtains the final result.

**6.2 An  $(\frac{e}{e-1})$ -Approximation Algorithm** We now prove that the *round-communication* tradeoff for the distributed maximum coverage problem proven in Theorem 5.1 is essentially tight. Theorem 5.1 shows that using  $k \cdot m^{O(1/r)}$  communication in  $r$  rounds only allows for a relatively large approximation factor of  $k^{\Omega(1/r)}$ . Here, we show that we can always obtain an (almost)  $(\frac{e}{e-1})$ -approximation (the optimal approximation ratio with sublinear in  $m$  communication) in  $r$  rounds using  $k \cdot m^{\Omega(1/r)}$  (for some larger constant in the exponent).

As stated in the introduction, our algorithm in this part is quite general and works for maximizing any monotone submodular function subject to a cardinality constraint. Hence, in the following, we present our results in this more general form.

**THEOREM 6.2.** *There exists a randomized distributed algorithm for submodular maximization subject to cardinality constraint that for any ground set  $V$  of size  $m$ , any monotone submodular function  $f : 2^V \rightarrow \mathbb{R}^+$ , and any  $r \geq 1$  and parameter  $\varepsilon \in (0, 1)$ , with high probability computes an  $(\frac{e}{e-1} + \varepsilon)$ -approximation in  $r$  rounds while communicating  $O(k \cdot m^{O(1/\varepsilon \cdot r)})$  items from  $V$ .*

Our algorithm follows the sample-and-prune technique of [48]. At each round, we sample a set of items from the machines and send them to the coordinator. The coordinator then computes a greedy solution  $X$  over the received sets and reports  $X$  back to the machines. The machines then prune any item that cannot be added to this partial greedy solution  $X$  and continue this process in the next rounds. At the end, the coordinator outputs  $X$ . By using a thresholding greedy algorithm and a more careful analysis, we show that the dependence of the number of rounds on  $\Omega(\log \Delta)$  (where  $\Delta$  is the ratio of maximum value of  $f$  on any singleton set to its minimum value) in [48] can be completely avoided, resulting in an algorithm with only constant number of rounds.

We assume that the algorithm is given a value  $\widetilde{\text{opt}}$  such that  $\text{opt} \leq \widetilde{\text{opt}} \leq 2 \cdot \text{opt}$ . In general, one can guess  $\text{opt}$  in powers of two in the range  $\Delta$  to  $k \cdot \Delta$  in parallel and solve the problem for all of them and return the best solution. This would increase the communication cost by only a factor of  $\Theta(\log k)$  (and one extra round of communication just to communicate  $\Delta$  if it is unknown). We now present our algorithm.

Sample and Prune Greedy Algorithm (SPGreedy).

**Input:** A collection  $V_i \subseteq V$  of items for each machine  $i \in [p]$  and a value  $\widetilde{\text{opt}} \in [\text{opt}, 2 \cdot \text{opt}]$ .

**Output:** A collection of  $k$  items from  $V$ .

1. Define the parameters  $\ell := \lceil \lg_{(1+\varepsilon)}(2e) \rceil (= \Theta(1/\varepsilon))$  and  $s = \lceil r/\ell \rceil$ . The algorithm consists of  $\ell$  iterations each with  $s$  steps.
2. For  $j = 1$  to  $\ell$  iterations:
  - (a) Let  $\tau_j = \frac{\widetilde{\text{opt}}}{k} \cdot \left(\frac{1}{1+\varepsilon}\right)^{j-1}$  and  $X^{j,0} = X^{j-1,s}$  initially (we assume  $X^{0,*} = \emptyset$ ).
  - (b) For  $t = 1$  to  $s$  steps:
    - (i) Define  $V^{j,t} = \{a \in V \mid f_{X^{j,(t-1)}}(a) \geq \tau_j\}$ .
    - (ii) Each machine  $i \in [p]$  samples each item in  $V^{j,t} \cap V_i$  independently and with probability  $q_t := \begin{cases} \frac{4k \log m}{m^{1-(t/s)}} & \text{if } t < s \\ 1 & \text{if } t = s \end{cases}$ , and sends them to the coordinator.
    - (iii) The coordinator iterates over each received item  $a$  (in an arbitrary order) and adds  $a$  to  $X^{j,t}$  iff  $f_{X^{j,t}}(a) \geq \tau_j$ .
    - (iv) The coordinator communicates the set  $X^{j,t}$  to the machines.
3. The coordinator returns  $X^{\ell,s}$  in the last step (if at any earlier point of the algorithm size of some  $X^{*,*}$  is already  $k$ , the coordinator terminates the algorithm and outputs this set as the answer).

SPGreedy requires  $\ell = \Theta(1/\varepsilon)$  iterations each consists of  $s = \lceil r/\ell \rceil$  steps. Moreover, each step can be implemented in one round of communication. As such, the round complexity of this algorithm is simply  $O(r)$ . The bounds on the communication cost of this algorithm and its approximation ratio are proven in the following two lemma which we defer to the full version of the paper.

It is now easy to bound the communication cost of this protocol.

**LEMMA 6.2.** *SPGreedy communicates at most  $O(r \cdot k \cdot m^{1/s} \cdot \log m)$  items w.p. at least  $1 - 1/m^k$ .*

**LEMMA 6.3.** *Suppose  $X$  is the set returned by SPSGreedy; then,  $f(X) \geq (1 - 1/e - \varepsilon) \cdot \text{opt}$ .*

Theorem 6.2 now follows immediately from Lemma 6.2 and Lemma 6.3.

We conclude this section by stating the following corollary of Theorem 6.2 for the maximum coverage problem, which formalizes the first part of Result 2. The proof is a direct application of Theorem 6.2 plus the known sketching methods for coverage functions in [51, 20] to further optimize the communication cost. We defer the proof to the full version.

**COROLLARY 6.1.** *There exists a randomized distributed algorithm for the maximum coverage problem that for any integer  $r \geq 1$ , and any parameter  $\varepsilon \in (0, 1)$ , with high probability computes an  $\left(\frac{\varepsilon}{\varepsilon-1} + \varepsilon\right)$ -approximation in  $r$  rounds and  $\tilde{O}\left(\frac{k}{\varepsilon^4} \cdot m^{O(1/\varepsilon \cdot r)} + n\right)$  communication.*

**Acknowledgements** The first author is grateful to Alessandro Epasto for bringing [36] to his attention, to David Woodruff for a helpful discussion on the implication of the results in [5] for proving multi-pass dynamic streaming lower bounds, and to Qin Zhang for useful comments about our framework in this paper for proving distributed lower bounds. We also thank the anonymous reviewers of SODA for many valuable comments and suggestions.

## References

- [1] F. N. Afrati, A. D. Sarma, S. Salihoglu, and J. D. Ullman. Upper and lower bounds on the cost of a map-reduce computation. *PVLDB*, 6(4):277–288, 2013.
- [2] K. J. Ahn and S. Guha. Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 202–211, 2015.
- [3] K. J. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12*, pages 459–467. SIAM, 2012.
- [4] K. J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 5–14, 2012.
- [5] Y. Ai, W. Hu, Y. Li, and D. P. Woodruff. New characterizations in turnstile streams with applications. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 20:1–20:22, 2016.
- [6] N. Alon, A. Moitra, and B. Sudakov. Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1079–1090, 2012.
- [7] 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*, pages 1499–1512, 2015.
- [8] S. Assadi. Combinatorial auctions do need modest interaction. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, pages 145–162, 2017.
- [9] S. Assadi. Tight space-approximation tradeoff for the multi-pass streaming set cover problem. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, pages 321–335, 2017.
- [10] S. Assadi and S. Khanna. Randomized composable coresets for matching and vertex cover. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2017, Washington DC, USA, July 24-26, 2017*, pages 3–12, 2017.
- [11] S. Assadi, S. Khanna, and Y. Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 698–711, 2016.
- [12] S. Assadi, S. Khanna, and Y. Li. On estimating maximum matching size in graph streams. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1723–1742, 2017.
- [13] S. Assadi, S. Khanna, Y. Li, and G. Yaroslavtsev. Maximum matchings in dynamic graph streams and the simultaneous communication model. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364, 2016.
- [14] G. Ausiello, N. Boria, A. Giannakos, G. Lucarelli, and V. T. Paschos. Online maximum k-coverage. *Discrete Applied Mathematics*, 160(13-14):1901–1913, 2012.
- [15] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: massive data summarization on the fly. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 671–680, 2014.
- [16] A. Badanidiyuru and J. Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1497–1514, 2014.
- [17] M. Balcan, S. Ehrlich, and Y. Liang. Distributed k-means and k-median clustering on general communication topologies. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference*

- on *Neural Information Processing Systems 2013, Lake Tahoe, Nevada, United States.*, pages 1995–2003, 2013.
- [18] M. Bateni, A. Bhaskara, S. Lattanzi, and V. S. Mirrokni. Distributed balanced clustering via mapping coresets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2591–2599, 2014.
- [19] M. Bateni, H. Esfandiari, and V. S. Mirrokni. Distributed coverage maximization via sketching. *CoRR*, abs/1612.02327, 2016.
- [20] M. Bateni, H. Esfandiari, and V. S. Mirrokni. Almost optimal streaming algorithms for coverage problems. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2017, Washington DC, USA, July 24-26, 2017*, pages 13–23, 2017.
- [21] P. Beame, P. Koutris, and D. Suciu. Communication steps for parallel query processing. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013*, pages 273–284, 2013.
- [22] G. E. Blelloch, R. Peng, and K. Tangwongsan. Linear-work greedy parallel approximate set cover and variants. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Collocated with FCRC 2011)*, pages 23–32, 2011.
- [23] 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*, pages 668–677, 2013.
- [24] A. Chakrabarti and S. Kale. Submodular maximization meets streaming: Matchings, matroids, and more. In *Integer Programming and Combinatorial Optimization - 17th International Conference, IPCO 2014, Bonn, Germany, June 23-25, 2014. Proceedings*, pages 210–221, 2014.
- [25] A. Chakrabarti and A. Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1365–1373, 2016.
- [26] J. Chen, H. L. Nguyen, and Q. Zhang. Submodular maximization over sliding windows. *CoRR*, abs/1611.00129, 2016.
- [27] F. Chierichetti, R. Kumar, and A. Tomkins. Max-cover in map-reduce. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 231–240, 2010.
- [28] G. Cormode, H. J. Karloff, and A. Wirth. Set cover algorithms for very large datasets. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 479–488, 2010.
- [29] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.
- [30] R. da Ponte Barbosa, A. Ene, H. L. Nguyen, and J. Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1236–1244, 2015.
- [31] R. da Ponte Barbosa, A. Ene, H. L. Nguyen, and J. Ward. A new framework for distributed submodular maximization. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, New Brunswick, New Jersey, USA*, pages 645–654, 2016.
- [32] E. D. Demaine, P. Indyk, S. Mahabadi, and A. Vakilian. On streaming and communication complexity of the set cover problem. In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, pages 484–498, 2014.
- [33] 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*, pages 233–242, 2014.
- [34] D. Dolev and T. Feder. Determinism vs. nondeterminism in multiparty communication complexity. *SIAM J. Comput.*, 21(5):889–895, 1992.
- [35] Y. Emek and A. Rosén. Semi-streaming set cover - (extended abstract). In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 453–464, 2014.
- [36] A. Epasto, S. Lattanzi, S. Vassilvitskii, and M. Zadimoghaddam. Submodular optimization over sliding windows. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 421–430, 2017.
- [37] D. V. Gucht, R. Williams, D. P. Woodruff, and Q. Zhang. The communication complexity of distributed set-joins with applications to matrix multiplication. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 199–212, 2015.
- [38] S. Guha, Y. Li, and Q. Zhang. Distributed partial clustering. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2017, Washington DC, USA, July 24-26, 2017*, pages 143–152, 2017.
- [39] S. Har-Peled, P. Indyk, S. Mahabadi, and A. Vakilian. Towards tight bounds for the streaming set cover problem. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 371–383, 2016.
- [40] S. Im and B. Moseley. Brief announcement: Fast and better distributed mapreduce algorithms for k-center

- clustering. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 65–67, 2015.
- [41] P. Indyk, S. Mahabadi, M. Mahdian, and V. S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS’14, Snowbird, UT, USA, June 22-27, 2014*, pages 100–108, 2014.
- [42] R. Jacob, T. Lieber, and N. Sitchinava. On the complexity of list ranking in the parallel external memory model. In *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, pages 384–395, 2014.
- [43] M. Kapralov and D. P. Woodruff. Spanners and sparsifiers in dynamic streams. In *ACM Symposium on Principles of Distributed Computing, PODC ’14, Paris, France, July 15-18, 2014*, pages 272–281, 2014.
- [44] H. J. Karloff, S. Suri, and S. Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 938–948, 2010.
- [45] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pages 137–146, 2003.
- [46] C. Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015.
- [47] A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1650–1654, 2007.
- [48] R. Kumar, B. Moseley, S. Vassilvitskii, and A. Vattani. Fast greedy algorithms in mapreduce and streaming. In *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA ’13, Montreal, QC, Canada - July 23 - 25, 2013*, pages 1–10, 2013.
- [49] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.
- [50] S. Lattanzi, B. Moseley, S. Suri, and S. Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 85–94, 2011.
- [51] A. McGregor and H. T. Vu. Better streaming algorithms for the maximum coverage problem. In *20th International Conference on Database Theory, ICDT 2017, March 21-24, 2017, Venice, Italy*, pages 22:1–22:18, 2017.
- [52] V. S. Mirrokni and M. Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 153–162, 2015.
- [53] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, Nevada, United States.*, pages 2049–2057, 2013.
- [54] 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, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 486–501, 2012.
- [55] A. Pietracaprina, G. Pucci, M. Riondato, F. Silvestri, and E. Upfal. Space-round tradeoffs for mapreduce computations. In *International Conference on Supercomputing, ICS’12, Italy*, pages 235–244, 2012.
- [56] T. Roughgarden, S. Vassilvitskii, and J. R. Wang. Shuffles and circuits: (on lower bounds for modern parallel computation). In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016*, pages 1–12, 2016.
- [57] I. Z. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978.
- [58] B. Saha and L. Getoor. On maximum coverage in the streaming model & application to multi-topic blog-watch. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, Sparks, Nevada, USA*, pages 697–708, 2009.
- [59] 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, New York, NY, USA, May 19 - 22, 2012*, pages 941–960, 2012.
- [60] D. P. Woodruff and Q. Zhang. When distributed computation is communication expensive. In *Distributed Computing - 27th International Symposium, DISC 2013, Jerusalem, Israel, October 14-18, 2013. Proceedings*, pages 16–30, 2013.
- [61] D. P. Woodruff and Q. Zhang. 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*, pages 718–733, 2014.
- [62] A. C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 209–213, 1979.