Tight Bounds for Single-Pass Streaming Complexity of the Set Cover Problem[*] [†]

Sepehr Assadi
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA, USA
sassadi@cis.upenn.edu

Sanjeev Khanna
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA, USA
sanjeev@cis.upenn.edu

Yang Li
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA, USA
yangli2@cis.upenn.edu

ABSTRACT

We resolve the space complexity of *single-pass* streaming algorithms for approximating the classic set cover problem. For finding an α-approximate set cover (for $\alpha = o(\sqrt{n})$) via a single-pass streaming algorithm, we show that $\Theta(mn/\alpha)$ space is both sufficient and necessary (up to an $O(\log n)$ factor); here m denotes number of the sets and n denotes size of the universe. This provides a strong negative answer to the open question posed by Indyk *et al.* (2015) regarding the possibility of having a single-pass algorithm with a small approximation factor that uses sub-linear space.

We further study the problem of *estimating* the *size* of a minimum set cover (as opposed to finding the actual sets), and establish that an additional factor of α saving in the space is achievable in this case and that this is the best possible. In other words, we show that $\Theta(mn/\alpha^2)$ space is both sufficient and necessary (up to logarithmic factors) for estimating the size of a minimum set cover to within a factor of α. Our algorithm in fact works for the more general problem of estimating the optimal value of a *covering integer program*. On the other hand, our lower bound holds even for set cover instances where the sets are presented in a *random order*.

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Theory, Algorithms

Keywords

Streaming algorithms, communication complexity, set cover, covering integer programs

1. INTRODUCTION

The *set cover* problem is a fundamental optimization problem with many applications in computer science and related disciplines. The input is a universe $[n]$ and a collection of m subsets of $[n]$, $\mathcal{S} = \langle S_1, \ldots, S_m \rangle$, and the goal is to find a subset of $\mathcal{S}$ with the *smallest* cardinality that *covers* $[n]$, i.e., whose union is $[n]$; we call such a collection of sets a *minimum set cover* and throughout the paper denote its cardinality by $opt := opt(\mathcal{S})$.

The set cover problem can be formulated in the well-established streaming model [1, 24], whereby the sets in $\mathcal{S}$ are presented one by one in a stream and the goal is to solve the set cover problem using a *space-efficient* algorithm. The streaming setting for the set cover problem has been studied in several recent work, including [28, 13, 11, 18, 8]. We refer the interested reader to these references for many applications of the set cover problem in the streaming model. In this paper, we focus on algorithms that make only *one pass* over the stream (i.e., single-pass streaming algorithms), and our goal is to settle the space complexity of single-pass streaming algorithms that *approximate* the set cover problem.

Two versions of the set cover problem are considered in this paper: (*i*) computing a minimum set cover, and (*ii*) computing the size of a minimum set cover. Formally,

Definition 1 (α-approximation). *An algorithm $\mathcal{A}$ is said to* α-approximate *the set cover problem iff on every input instance $\mathcal{S}$, $\mathcal{A}$ outputs a collection of (the indices of) at most $\alpha \cdot opt$ sets that covers $[n]$, along with a* certificate of covering *which, for each element $e \in [n]$, specifies the set used for covering e. If $\mathcal{A}$ is a randomized algorithm, we require that the certificate corresponds to a valid set cover w.p.*[1] *at least* $2/3$.

We remark that the requirement of returning a certificate of covering is standard in the literature (see, e.g., [13, 8]).

We are also interested in algorithms that only compute the size of a minimum set cover, referred to as estimation algorithms.

[*] Supported in part by National Science Foundation grants CCF-1116961, CCF-1552909, and IIS-1447470 and an Adobe research award.

[†] A full version of this paper is available as an online preprint at [2].

[1] Throughout, we use *w.p.* and *w.h.p.* to abbreviate "with probability" and "with high probability", respectively.

Definition 2 (α-estimation). *An algorithm $\mathcal{A}$ is said to* α-estimate *the set cover problem iff on every input instance $\mathcal{S}$, $\mathcal{A}$ outputs an estimate for the cardinality of a minimum set cover in the range $[opt, \alpha \cdot opt]$. If $\mathcal{A}$ is a randomized algorithm, we require that:*

$$\Pr\Big(\mathcal{A}(\mathcal{S}) \in [opt, \alpha \cdot opt]\Big) \geq 2/3$$

1.1 Our Results

We resolve the space complexities of both versions of the set cover problem. Specifically, we show that for any $\alpha = o(\sqrt{n}/\log n)$ and any $m = \text{poly}(n)$,

- There is a *deterministic* single-pass streaming algorithm that *α-approximates* the set cover problem using space $\widetilde{O}(mn/\alpha)$ bits and moreover, any single-pass streaming algorithm (possibly *randomized*) that α-approximates the set cover problem must use space of $\Omega(mn/\alpha)$ bits.

- There is a *randomized* single-pass streaming algorithm that *α-estimates* the set cover problem using space $\widetilde{O}(mn/\alpha^2)$ bits and moreover, any single-pass streaming algorithm (possibly *randomized*) that α-estimates the set cover problem must use $\widetilde{\Omega}(mn/\alpha^2)$ bits of space.

We should point out right away that in this paper, we are *not* concerned with poly-time computability, though our algorithms for set cover can be made computationally efficient for any $\alpha \geq \log n$ by allowing an extra $\log n$ factor in the space requirement[2].

We establish our upper bound result for α-estimation for a much more general problem: estimating the optimal value of a *covering integer linear program* (see Section 4 for a formal definition). Moreover, the space lower bound for α-estimation (for the original set cover problem) holds even if the sets are presented in a *random order*. We now describe each of these two sets of results in more details.

Approximating Set Cover. There is a very simple deterministic α-approximation algorithm for the set cover problem using space $\widetilde{O}(mn/\alpha)$ bits which we mention in Section 1.2 for completeness. Perhaps surprisingly, we establish that this simple algorithm is essentially the best possible; any α-approximation algorithm for the set cover problem requires $\widetilde{\Omega}(mn/\alpha)$ bits of space (see Theorem 1 for a formal statement).

Prior to our work, the best known lower bounds for single-pass streams ruled out $(3/2-\epsilon)$-approximation using $o(mn)$ space [18] (see also [16]), $o(\sqrt{n})$-approximation in $o(m)$ space [13, 8], and $O(1)$-approximation in $o(mn)$ space [11] (only for *deterministic* algorithms); see Section 1.3 for more detail on different existing lower bounds. Note that these lower bound results leave open the possibility of a single-pass randomized 3/2-approximation or even a deterministic $O(\log n)$-approximation algorithm for the set cover problem using only $\widetilde{O}(m)$ space. Our result on the other hand, rules out the possibility of having any *non-trivial* trade-off between the approximation factor and the space requirement, answering an open question raised by Indyk *et al.* [18] in the strongest sense possible.

We also point out that the bound of $\alpha = o(\sqrt{n}/\log n)$ in our lower bound is tight up to an $O(\log n)$ factor since an $O(\sqrt{n})$-approximation is known to be achievable in $\widetilde{O}(n)$ space (independent of m for $m = \text{poly}(n)$) [13, 8].

Estimating Set Cover Size. We present an $\widetilde{O}(mn/\alpha^2)$ space algorithm for α-estimating the set cover problem, and in general any covering integer program (see Theorem 3 for a formal statement). Our upper bound suggests that if one is only interested in α-estimating the size of a minimum set cover (instead of knowing the actual sets), then an additional α factor saving in the space (compare to the best possible α-approximation algorithm) is possible. To the best of our knowledge, this is the first non-trivial *gap* between the space complexity of α-approximation and α-estimation for the set cover problem.

We further show that space complexity of our $\widetilde{O}(mn/\alpha^2)$ space α-estimation algorithm for the set cover problem is essentially tight (up to logarithmic factors). In other words, any α-estimation algorithm for set cover (possibly randomized) requires $\widetilde{\Omega}(mn/\alpha^2)$ space (see Theorem 4 for a formal statement).

There are examples of classic optimization problems in the streaming literature for which size estimation seems to be distinctly easier in the *random arrival streams*[3] compare to the *adversarial streams* (see, e.g., [21]). However, we show that this is *not* the case for the set cover problem, i.e., the lower bound of $\widetilde{\Omega}(mn/\alpha^2)$ for α-estimation continues to hold even for random arrival streams.

We note in passing two other results also: (i) our bounds for α-approximation/estimation also prove *tight* bounds on the *one-way communication complexity* of the *two-player* communication model of the set cover problem (see Theorem 2 and Theorem 5), previously studied in [25, 11, 8]; (ii) the use of randomization in our α-estimation algorithm is inevitable: any *deterministic* α-estimation algorithm for the set cover requires $\Omega(mn/\alpha)$ bits of space (see the full version of the paper [2]).

1.2 Our Techniques

Upper bounds. An α-approximation using $\widetilde{O}(mn/\alpha)$ bits of space can be simply achieved as follows. Merge (i.e., take the union of) every α sets in the stream into a single set; at the end of the stream, solve the set cover problem over the merged sets. To recover a certificate of covering, we also record for each element e in each merged set, any set in the merge that covers e. It is an easy exercise to verify that this algorithm indeed achieves an α-approximation and can be implemented in space $\widetilde{O}(mn/\alpha)$ bits.

Our $\widetilde{O}(mn/\alpha^2)$-space α-estimation algorithm is more involved and in fact works for any covering integer program (henceforth, a *covering ILP* for short). We follow the line of work in [11] and [18] by performing "dimensionality reduction" over the sets (in our case, columns of the constraint matrix A) and storing their projection over a randomly sampled subset of the universe (here, constraints) during the stream. However, the goal of our *constraint sampling* approach is

[2] Set cover admits a classic $\log n$-approximation algorithm [20, 29], and unless $\mathsf{P} = \mathsf{NP}$, there is no polynomial time α-approximation for the set cover problem for $\alpha < (1-\epsilon)\log n$ (for any constant $\epsilon > 0$) [12, 14, 23].

[3] In random arrival streams, the input (in our case, the collection of sets) is randomly permuted before being presented to the algorithm

entirely different from the ones in [11, 18]. The *element sampling* approach of [11, 18] aims to find a "small" cover of the sampled universe which also covers the vast majority of the elements in the original universe. This allows the algorithm to find a small set cover of the sampled universe in one pass while reducing the number of remaining uncovered elements for the next pass; hence, applying this approach repeatedly over *multiple* passes on the input allows one to obtain a complete cover.

On the other hand, the goal of our constraint sampling approach is to create a smaller instance of set cover (in general, covering ILP) with the property that the minimum set cover *size* of the sampled instance is a "proxy" for the minimum set cover size of the original instance. We crucially use the fact that the algorithm does *not* need to identify the actual cover and hence it can estimate the size of the solution based on the optimum set cover size in the sampled universe.

At the core of our approach is a simple yet very general lemma, referred to as the *constraint sampling lemma* (Lemma 4.1) which may be of independent interest. Informally, this lemma states that for any covering ILP instance $\mathcal{I}$, the optimal value of a sub-sampled instance $\mathcal{I}_{\mathsf{R}}$, obtained by picking roughly $1/\alpha$ fraction of the constraints uniformly at random from $\mathcal{I}$, is an α estimator of the optimum value of $\mathcal{I}$ whenever no constraint is "too hard" to satisfy.

Nevertheless, the constraint sampling is not enough for reducing the space to meet the desired $\widetilde{O}(mn/\alpha^2)$ bound (see Theorem 3). Hence, we combine it with a *pruning* step, similar to the "set filtering" step of [18] for (unweighted) set cover (see also "GreedyPass" algorithm of [8]) to sparsify the columns in the input matrix A before performing the sampling. We point out that as the variables in $\mathcal{I}$ can have different weights in the objective function (e.g. for weighted set cover), our pruning step needs to be sensitive to the weights.

Lower bounds. As is typical in the streaming literature, our lower bounds are obtained by establishing *communication complexity* lower bounds; in particular, in the *one-way two-player* communication model. To prove these bounds, we use the *information complexity* paradigm, which allows one to reduce the problem, via a direct sum type argument, to multiple instances of a simpler problem. For our lower bound for α-estimation, this simpler problem turned out to be a variant of the well-known *Set Disjointness* problem. However, for the lower bound of α-approximation algorithms, we introduce and analyze a new intermediate problem, called the *Trap* problem.

The Trap problem is a *non-boolean* problem defined as follows: Alice is given a set S, Bob is given a set E such that all elements of E belong to S except for a *special element* e^*, and the goal of the players is to "trap" this special element, i.e., to find a *small* subset of E which contains e^*. For our purpose, Bob only needs to trap e^* in a set of cardinality $|E|/2$. To prove a lower bound for the Trap problem, we design a novel reduction from the well-known *Index* problem, which requires Alice and Bob to use the protocol for the Trap problem over non-legal inputs (i.e., the ones for which the Trap problem is not well-defined), while ensuring that they are not being "fooled" by the output of the Trap protocol over these inputs.

To prove our lower bound for α-estimation in random arrival streams, we follow the approach of [6] in proving the communication complexity lower bounds when the input data is *randomly allocated* between the players (as opposed to adversarial partitions). However, the distributions and the problem considered in this paper are different from the ones in [6].

1.3 Related Work

Communication complexity of the set cover problem was first studied by Nisan [25]. Among other things, Nisan showed that the two-player communication complexity of $(\frac{1}{2} - \epsilon)\log n$-estimating the set cover is $\Omega(m)$. In particular, this implies that any constant-pass streaming algorithm that $(\frac{1}{2} - \epsilon)\log n$-estimates the set cover must use $\Omega(m)$ bits of space.

Saha and Getoor [28] initiated the study of set cover in the *semi-streaming* model [15] where the sets are arriving in a stream and the algorithms are required to use $\widetilde{O}(n)$ space, and obtained an $O(\log n)$-approximation via an $O(\log n)$-pass algorithm that uses $O(n \log n)$ space. A similar performance was also achieved by [9] in the context of "disk friendly" algorithms. As designed, algorithm of [9] achieves $(1 + \beta \ln n)$-approximation by making $O(\log_\beta n)$ passes over the stream using $O(n \log n)$ space.

The *single-pass semi-streaming* setting for set cover was initially and throughly studied by Emek and Rosén [13]. They provided an $O(\sqrt{n})$-approximation using $\widetilde{O}(n)$ space (also extends to the weighted set cover problem) and a lower bound that states that no semi-streaming algorithm (i.e., an algorithm using only $\widetilde{O}(n)$ space) that $O(n^{1/2-\epsilon})$-estimates set cover exists. Recently, Chakrabarti and Wirth [8] generalized the space bounds in [13] to *multi-pass* algorithms, providing an almost complete understanding of the pass vs approximation tradeoff for semi-streaming algorithms. In particular, they developed a deterministic p-pass $(p + 1) \cdot n^{1/(p+1)}$-approximation algorithm in $\widetilde{O}(n)$ space and prove that any p-pass $n^{1/(p+1)}/(c \cdot (p + 1)^2)$-estimation algorithm requires $\Omega(n^c/p^3)$ space for some constant $c > 1$ (m in their "hard instances" is $\Theta(n^{cp})$). This, in particular, implies that any *single-pass* $o(\sqrt{n})$-estimation algorithm requires $\Omega(m)$ space.

Demaine *et al.* [11] studied the trade-off between the number of passes, the approximation ratio, and the space requirement of general streaming algorithms (i.e., not necessarily semi-streaming) for the set cover problem and developed an algorithm that for any $\delta = \Omega(1/\log n)$, makes $O(4^{1/\delta})$ passes over the stream and achieves an $O(4^{1/\delta}\rho)$-approximation using $\widetilde{O}(mn^\delta)$ space; here ρ is the approximation factor of the *off-line* algorithm for solving the set cover problem. The authors further showed that any *constant-pass deterministic* $O(1)$-estimation algorithm for the set cover requires $\Omega(mn)$ space. Very recently, Indyk *et al.* [18] (see also [16]) made a significant improvement on the trade-off achieved by [11]: they presented an algorithm that for any $\delta > 0$, makes $O(1/\delta)$ passes over the stream while achieving an $O(\rho/\delta)$-approximation using $\widetilde{O}(mn^\delta)$ space. The authors also established two lower bounds: for multi-pass algorithms, any algorithm that computes an *optimal* set cover solution while making only $(\frac{1}{2\delta} - 1)$ passes must use $\widetilde{\Omega}(mn^\delta)$ space. More relevant to our paper, they also showed that any *single-pass* streaming algorithm (possibly randomized) that can distinguish between the instances with set cover size of 2 and 3 with error probability $1/\text{poly}(m)$, must use

$\Omega(mn)$ bits.

Organization. We introduce in Section 2 some preliminaries needed for the rest of the paper. In Section 3, we present our $\Omega(mn/\alpha)$ space lower bound for computing an α-approximate set cover in a single-pass. In Section 4, we present a single-pass streaming algorithm for estimating the optimal value of a covering integer program and prove an $\widetilde{O}(mn/\alpha^2)$ upper bound on space complexity of α-estimating the (weighted) set cover problem. In Section 5, we present our $\Omega(mn/\alpha^2)$ space lower bound for α-estimating set cover in a single-pass. Omitted details are deferred to the full version of the paper [2].

2. PRELIMINARIES

Notation. We use bold face letters to represent random variables. For any random variable $\boldsymbol{X}$, $\textsc{supp}(\boldsymbol{X})$ denotes its support set. We define $|\boldsymbol{X}| := \log|\textsc{supp}(\boldsymbol{X})|$. For any k-dimensional tuple $X = (X_1, \ldots, X_k)$ and any $i \in [k]$, we define $X^{<i} := (X_1, \ldots, X_{i-1})$, and similarly $X^{-i} := (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_k)$. The notation "$X \in_R U$" indicates that X is chosen uniformly at random from a set U. Finally, we use upper case letters (e.g. M) to represent matrices and lower case letter (e.g. v) to represent vectors.

Concentration Bounds. In this paper, we use an extension of the Chernoff bound for *negatively correlated* random variables. Random variables $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ are negatively correlated if for every set $S \subseteq [n]$, $\Pr(\wedge_{i\in S}\boldsymbol{X}_i = 1) \leq \prod_{i\in S}\Pr(\boldsymbol{X}_i = 1)$. It was first proved in [26] that the Chernoff bound continues to hold for the case of random variables that satisfy this generalized version of negative correlation (see also [17]).

2.1 Tools from Information Theory

We briefly review some basic concepts from information theory needed for establishing our lower bounds. For a broader introduction to the field, we refer the reader to the excellent text by Cover and Thomas [10].

In the following, we denote the *Shannon Entropy* of a random variable $\boldsymbol{A}$ by $H(\boldsymbol{A})$ and the *mutual information* of two random variables $\boldsymbol{A}$ and $\boldsymbol{B}$ by $I(\boldsymbol{A};\boldsymbol{B}) = H(\boldsymbol{A}) - H(\boldsymbol{A} \mid \boldsymbol{B}) = H(\boldsymbol{B}) - H(\boldsymbol{B} \mid \boldsymbol{A})$. If the distribution $\mathcal{D}$ of the random variables is not clear from the context, we use $H_{\mathcal{D}}(\boldsymbol{A})$ (resp. $I_{\mathcal{D}}(\boldsymbol{A};\boldsymbol{B})$).

We use the following basic properties of entropy and mutual information (proofs can be found in [10], Chapter 2).

Claim 2.1. *Let $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ be three random variables.*

1. $0 \leq H(\boldsymbol{A}) \leq |\boldsymbol{A}|$. $H(\boldsymbol{A}) = |\boldsymbol{A}|$ *iff $\boldsymbol{A}$ is uniformly distributed over its support.*
2. $I(\boldsymbol{A};\boldsymbol{B}) \geq 0$. *The equality holds iff $\boldsymbol{A}$ and $\boldsymbol{B}$ are* independent.
3. Conditioning on a random variable reduces entropy*: $H(\boldsymbol{A} \mid \boldsymbol{B}, \boldsymbol{C}) \leq H(\boldsymbol{A} \mid \boldsymbol{B})$. The equality holds iff $\boldsymbol{A}$ and $\boldsymbol{C}$ are independent conditioned on $\boldsymbol{B}$.*
4. Subadditivity of entropy*: $H(\boldsymbol{A}, \boldsymbol{B} \mid \boldsymbol{C}) \leq H(\boldsymbol{A} \mid \boldsymbol{C}) + H(\boldsymbol{B} \mid \boldsymbol{C})$.*
5. The chain rule for mutual information*: $I(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{C}) = I(\boldsymbol{A};\boldsymbol{C}) + I(\boldsymbol{B};\boldsymbol{C} \mid \boldsymbol{A})$.*
6. *For any event E independent of $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$, $I(\boldsymbol{A};\boldsymbol{B} \mid \boldsymbol{C}, E) = I(\boldsymbol{A};\boldsymbol{B} \mid \boldsymbol{C})$.*

We also use the following simple claim, which states that conditioning on independent random variables can only increase the mutual information (see the full version [2] for a proof).

Claim 2.2. *For any random variables $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, and $\boldsymbol{D}$, if $\boldsymbol{A}$ and $\boldsymbol{D}$ are independent conditioned on $\boldsymbol{C}$, then $I(\boldsymbol{A};\boldsymbol{B} \mid \boldsymbol{C}) \leq I(\boldsymbol{A};\boldsymbol{B} \mid \boldsymbol{C}, \boldsymbol{D})$.*

2.2 Communication Complexity and Information Complexity

Communication complexity and information complexity play an important role in our lower bound proofs. We now provide necessary definitions for completeness.

Communication complexity. We prove our lower bounds for single-pass streaming algorithms are through communication complexity lower bounds. Here, we briefly provide some context necessary for our purpose; for a more detailed treatment of communication complexity, we refer the reader to the excellent text by Kushilevitz and Nisan [22].

In this paper, we focus on the *two-player one-way communication* model. Let P be a relation with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$. Alice receives an input $X \in \mathcal{X}$ and Bob receives $Y \in \mathcal{Y}$, where (X, Y) are chosen from a joint distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. In addition to private randomness, the players also have an access to a shared public tape of random bits R. Alice sends a single message $M(X, R)$ and Bob needs to output an answer $Z := Z(M(X, R), Y, R)$ such that $(X, Y, Z) \in P$.

We use Π to denote a protocol used by the players. Unless specified otherwise, we always assume that the protocol Π can be randomized (using both public and private randomness), *even against a prior distribution $\mathcal{D}$ of inputs.* For any $0 < \delta < 1$, we say Π is a δ-error protocol for P over a distribution $\mathcal{D}$, if the probability that for an input (X, Y), Bob outputs some Z where $(X, Y, Z) \notin P$ is at most δ (the probability is taken over the randomness of both the distribution and the protocol).

Definition 3. *The* communication cost *of a protocol Π for a problem P on an input distribution $\mathcal{D}$, denoted by $\|\Pi\|$, is the worst-case size of the message sent from Alice to Bob in the protocol Π, when the inputs are chosen from the distribution $\mathcal{D}$.*
The communication complexity $\mathsf{CC}^{\delta}_{\mathcal{D}}(P)$ *of a problem P with respect to a distribution $\mathcal{D}$ is the minimum communication cost of a δ-error protocol Π over $\mathcal{D}$.*

Information complexity. There are several possible definitions of information complexity of a communication problem that have been considered depending on the application (see, e.g., [4, 5, 7, 3]). Our definition is tuned specifically for *one-way protocols*, similar in the spirit of [4] (see also [19]).

Definition 4. *Consider an input distribution $\mathcal{D}$ and a protocol Π (for some problem P). Let $\boldsymbol{X}$ be the random variable for the input of Alice drawn from $\mathcal{D}$, and let $\boldsymbol{\Pi} := \boldsymbol{\Pi}(\boldsymbol{X})$ be the random variable denoting the message sent from Alice to Bob* concatenated *with the public randomness $\boldsymbol{R}$ used by* Π. *The information cost $\mathsf{ICost}_{\mathcal{D}}(\Pi)$ of a one-way protocol Π with respect to $\mathcal{D}$ is $I_{\mathcal{D}}(\boldsymbol{\Pi};\boldsymbol{X})$.*
The information complexity $\mathsf{IC}^{\delta}_{\mathcal{D}}(P)$ *of P with respect to a*

distribution $\mathcal{D}$ is the minimum $\mathsf{ICost}_{\mathcal{D}}(\Pi)$ *taken over all one-way δ-error protocols Π for P over $\mathcal{D}$.*

Note that any public coin protocol is a distribution over private coins protocols, run by first using public randomness to sample a random string $\boldsymbol{R} = R$ and then running the corresponding private coin protocol Π^R. We also use $\boldsymbol{\Pi}^R$ to denote the random variable of the message sent from Alice to Bob, assuming that the public randomness is $\boldsymbol{R} = R$. We have the following well-known claim (see the full version [2] for a proof).

CLAIM 2.3. *For any distribution $\mathcal{D}$ and any protocol Π, let $\boldsymbol{R}$ denote the public randomness used in Π; then,*

$$\mathsf{ICost}_{\mathcal{D}}(\Pi) = \underset{R\sim\boldsymbol{R}}{\mathrm{E}}\left[I_{\mathcal{D}}(\boldsymbol{\Pi}^R; \boldsymbol{X} \mid \boldsymbol{R} = R)\right]$$

The following well-known proposition (see, e.g., [7]) relates communication complexity and information complexity. A short proof is provided in the full version of the paper [2].

PROPOSITION 2.4. *For every $0 < \delta < 1$ and every distribution $\mathcal{D}$: $\mathsf{CC}^{\delta}_{\mathcal{D}}(P) \geq \mathsf{IC}^{\delta}_{\mathcal{D}}(P)$.*

3. A LOWER BOUND FOR APPROXIMATING SET COVER

In this section, we prove that the simple α-approximation algorithm described in Section 1.2 is in fact optimal in terms of the space requirement. Formally,

THEOREM 1. *For any $\alpha = o(\frac{\sqrt{n}}{\log n})$ and $m = poly(n)$, any* randomized *single-pass streaming algorithm (possibly randomized) that α-approximates the set cover problem with probability at least $2/3$ requires $\Omega(mn/\alpha)$ bits of space.*

Fix a (sufficiently large) value for n, $m = \text{poly}(n)$ (also $m = \Omega(\alpha \log n)$), and $\alpha = o(\frac{\sqrt{n}}{\log n})$; throughout this section, $\mathsf{SetCover}_{\mathsf{apx}}$ refers to the problem of α-approximating the set cover problem for instances with $m+1$ sets[4] defined over the universe $[n]$ in the one-way communication model, whereby the sets are partitioned between Alice and Bob.

Overview. We design a hard input distribution $\mathcal{D}_{\mathsf{apx}}$ for $\mathsf{SetCover}_{\mathsf{apx}}$, whereby Alice is provided with a collection of m sets $S_1, \ldots, S_m$, each of size (roughly) n/α and Bob is given a *single* set T of size (roughly) $n - 2\alpha$. The input to the players are *correlated* such that there exists a set S_{i^*} in Alice's collection (i^* is unknown to Alice), such that $S_{i^*} \cup T$ covers all elements in $[n]$ except for a single *special element.* This in particular ensures that the optimal set cover size in this distribution is at most 3 w.h.p.

On the other hand, we "hide" this special element among the 2α elements in $\overline{T}$ in a way that if Bob does not have (essentially) full information about Alice's collection, he cannot even identify a set of α elements from $\overline{T}$ that contain this special element (w.p strictly more than half). This implies that in order for Bob to be sure that he returns a valid set cover, he should additionally cover a majority of $\overline{T}$ with sets *other than S_{i^*}*. We design the distribution in a way that the sets in Alice's collection are "far" from each other and hence Bob is forced to use a *distinct* set for (roughly) each element in $\overline{T}$ that he needs to cover with sets other than S_{i^*}. This implies that Bob needs to output a set cover of size α (i.e., an $(\alpha/3)$-approximation) to ensure that every element in $[n]$ is covered.

3.1 A Hard Input Distribution for $\mathsf{SetCover}_{\mathsf{apx}}$

Consider the following distribution.

Distribution $\mathcal{D}_{\mathsf{apx}}$. A hard input distribution for $\mathsf{SetCover}_{\mathsf{apx}}$.

Notation. Let $\mathcal{F}$ be the collection of all subsets of $[n]$ with cardinality $\frac{n}{10\alpha}$, and $\ell := 2\alpha \log m$.

- **Alice.** The input of Alice is a collection of m sets $\mathcal{S} = (S_1, \ldots, S_m)$, where for any $i \in [m]$, S_i is a set chosen independently and uniformly at random from $\mathcal{F}$.
- **Bob.** Pick an $i^* \in [m]$ (called the *special index*) uniformly at random; the input to Bob is a set $T = [n] \setminus E$, where E is chosen uniformly at random from all subsets of $[n]$ with $|E| = \ell$ and $|E \setminus S_{i^*}| = 1$.[a]

[a] Since $\alpha = o(\sqrt{n}/\log n)$ and $m = \text{poly}(n)$, the size of E is strictly smaller than the size of S_{i^*}.

The claims below summarize some useful properties of the distribution $\mathcal{D}_{\mathsf{apx}}$.

CLAIM 3.1. *For any instance $(\mathcal{S}, T) \sim \mathcal{D}_{\mathsf{apx}}$, with probability $1 - o(1)$, $opt(\mathcal{S}, T) \leq 3$.*

PROOF. Let e^* denote the element in $E \setminus S_{i^*}$. $\mathcal{S}^{-i^*}$ contains $m - 1$ random subsets of $[n]$ of size $n/10\alpha$, drawn independent of the choice of e^*. Therefore, each set in $\mathcal{S}^{-i^*}$ covers e^* with probability $1/10\alpha$. The probability that none of these $m - 1$ sets covers e^* is at most

$$\begin{aligned}(1 - 1/10\alpha)^{m-1} &\leq (1 - 1/10\alpha)^{\Omega(\alpha \log n)} \\ &\leq \exp(-\Omega(\alpha \log n)/10\alpha) = o(1)\end{aligned}$$

Hence, with probability $1 - o(1)$, there is at least one set $S \in \mathcal{S}^{-i^*}$ that covers e^*. Now, it is straightforward to verify that (S_{i^*}, T, S) form a valid set cover. □

LEMMA 3.2. *With probability $1 - o(1)$, no collection of 3α sets from $\mathcal{S}^{-i^*}$ covers more than $\ell/2$ elements of E.*

PROOF. Recall that the sets in $\mathcal{S}^{-i^*}$ and the set E are chosen independent of each other. For each set $S \in \mathcal{S}^{-i^*}$ and for each element $e \in E$, we define an indicator binary random variable $\boldsymbol{X}_e$, where $\boldsymbol{X}_e = 1$ iff $e \in S$. Let $\boldsymbol{X} := \sum_e \boldsymbol{X}_e$, which is the number of elements in E covered by S. We have,

$$\mathrm{E}[\boldsymbol{X}] = \sum_e \mathrm{E}[\boldsymbol{X}_e] = \frac{|E|}{10\alpha} = \frac{\log m}{5}$$

Moreover, the variables $\boldsymbol{X}_e$ are negatively correlated since

[4] To simplify the exposition, we use $m + 1$ instead of m as the number of sets.

for any set $S' \subseteq E$,

$$\Pr\left(\bigwedge_{e \in S'} \boldsymbol{X}_e = 1\right) = \frac{\binom{n-|S'|}{\frac{n}{10\alpha}-|S'|}}{\binom{n}{\frac{n}{10\alpha}}}$$

$$= \frac{\left(\frac{n}{10\alpha}\right)\cdot\left(\frac{n}{10\alpha}-1\right)\ldots\left(\frac{n}{10\alpha}-|S'|+1\right)}{(n)\cdot(n-1)\ldots(n-|S'|+1)}$$

$$\leq \left(\frac{1}{10\alpha}\right)^{|S'|} = \prod_{e \in S'} \Pr\left(\boldsymbol{X}_e = 1\right)$$

Hence, by the extended Chernoff bound (see Section 2),

$$\Pr\left(\boldsymbol{X} \geq \frac{\log m}{3}\right) = o(\frac{1}{m})$$

Therefore, using a union bound over all $m-1$ sets in $\mathcal{S}^{-i^*}$, with probability $1-o(1)$, no set in $\mathcal{S}^{-i^*}$ covers more than $\log m/3$ elements in E, which implies that any collection of 3α sets can only cover up to $3\alpha \cdot \log m/3 = \ell/2$ elements in the set E. □

3.2 The Lower Bound for the Distribution $\mathcal{D}_{\mathsf{apx}}$

In order to prove our lower bound for $\mathsf{SetCover}_{\mathsf{apx}}$ on $\mathcal{D}_{\mathsf{apx}}$, we define an intermediate communication problem which we call the *Trap* problem.

Problem 1 (*Trap* problem). *Alice is given a set $S \subseteq [n]$ and Bob is given a set $E \subseteq [n]$ such that $E \setminus S = \{e^*\}$; Bob needs to output a set $L \subseteq E$ with $|L| \leq |E|/2$ such that $e^* \in L$.*

In the following, we use Trap to refer to the trap problem with $|S| = n/10\alpha$ and $|E| = \ell = 2\alpha \log m$ (notice the similarity to the parameters in $\mathcal{D}_{\mathsf{apx}}$). We define the following distribution $\mathcal{D}_{\mathsf{Trap}}$ for Trap. Alice is given a set $S \in_R \mathcal{F}$ (recall that $\mathcal{F}$ is the collection of all subsets of $[n]$ of size $n/10\alpha$) and Bob is given a set E chosen uniformly at random from all sets that satisfy $|E \setminus S| = 1$ and $|E| = 2\alpha \log m$. We first use a direct sum style argument to prove that under the distributions $\mathcal{D}_{\mathsf{apx}}$ and $\mathcal{D}_{\mathsf{Trap}}$, information complexity of solving $\mathsf{SetCover}_{\mathsf{apx}}$ is essentially equivalent to solving m copies of Trap. Formally,

Lemma 3.3. *For any constant $\delta < 1/2$,*

$$\mathsf{IC}^{\delta}_{\mathcal{D}_{\mathsf{apx}}}(\mathsf{SetCover}_{\mathsf{apx}}) \geq m \cdot \mathsf{IC}^{\delta+o(1)}_{\mathcal{D}_{\mathsf{Trap}}}(\mathsf{Trap}).$$

Proof. Let Π_{SC} be a δ-error protocol for $\mathsf{SetCover}_{\mathsf{apx}}$; we design a δ'-error protocol Π_{Trap} for solving Trap over $\mathcal{D}_{\mathsf{Trap}}$ with parameter $\delta' = \delta + o(1)$ such that the information cost of Π_{Trap} on $\mathcal{D}_{\mathsf{Trap}}$ is at most $\frac{1}{m} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{apx}}}(\Pi_{\mathsf{SC}})$. The protocol Π_{Trap} is as follows.

Protocol Π_{Trap}. The protocol for solving Trap using a protocol Π_{SC} for $\mathsf{SetCover}_{\mathsf{apx}}$.

Input: An instance $(S, E) \sim \mathcal{D}_{\mathsf{Trap}}$. **Output:** A set L with $|L| \leq |E|/2$, such that $e^* \in L$.

1. Using *public randomness*, the players sample an index $i^* \in [m]$ uniformly at random.
2. Alice creates a tuple $\mathcal{S} = (S_1, \ldots, S_m)$ by assigning $S_{i^*} = S$ and sampling each remaining set uniformly at random from $\mathcal{F}$ using *private randomness*. Bob creates a set $T := \overline{E}$.
3. The players run the protocol Π_{SC} over the input $(\mathcal{S}, T)$.
4. Bob computes the set L of all elements in $E = \overline{T}$ whose certificate (i.e., the set used to cover them) is not S_{i^*}, and outputs L.

We first argue the correctness of Π_{Trap} and then bound its information cost. To argue the correctness, notice that the distribution of instances of $\mathsf{SetCover}_{\mathsf{apx}}$ constructed in the reduction is exactly $\mathcal{D}_{\mathsf{apx}}$. Consequently, it follows from Claim 3.1 that, with probability $1-o(1)$, any α-approximate set cover can have at most 3α sets. Let $\widehat{\mathcal{S}}$ be the set cover computed by Bob minus the sets S_{i^*} and T. As $e^* \in E = \overline{T}$ and moreover is *not* in S_{i^*}, it follows that e^* should be covered by some set in $\widehat{\mathcal{S}}$. This means that the set L that is output by Bob contains e^*. Moreover, by Lemma 3.2, the number of elements in E covered by the sets in $\widehat{\mathcal{S}}$ is at most $\ell/2$ w.p. $1-o(1)$. Hence, $|L| \leq \ell/2 = |E|/2$. This implies that:

$$\Pr_{\mathcal{D}_{\mathsf{Trap}}}\left(\Pi_{\mathsf{Trap}} \text{ errs}\right) \leq \Pr_{\mathcal{D}_{\mathsf{apx}}}\left(\Pi_{\mathsf{SC}} \text{ errs}\right) + o(1) \leq \delta + o(1)$$

We now bound the information cost of Π_{Trap}. Let $\boldsymbol{I}$ be the random variable for the choice of $i^* \in [m]$ in the protocol Π_{Trap} (which is uniform in $[m]$). Using Claim 2.3, we have,

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{Trap}}}(\Pi_{\mathsf{Trap}}) = \mathop{\mathrm{E}}_{i \sim \boldsymbol{I}}\left[I_{\mathcal{D}_{\mathsf{Trap}}}\left(\boldsymbol{\Pi}^{i}_{\mathsf{Trap}}; \boldsymbol{S} \mid \boldsymbol{I} = i\right)\right]$$

$$= \frac{1}{m} \cdot \sum_{i=1}^{m} I_{\mathcal{D}_{\mathsf{Trap}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{S}_i \mid \boldsymbol{I} = i\right)$$

$$= \frac{1}{m} \cdot \sum_{i=1}^{m} I_{\mathcal{D}_{\mathsf{apx}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{S}_i \mid \boldsymbol{I} = i\right)$$

$$= \frac{1}{m} \cdot \sum_{i=1}^{m} I_{\mathcal{D}_{\mathsf{apx}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{S}_i\right)$$

where the last two equalities hold since (i) the joint distribution of $\boldsymbol{\Pi}_{\mathsf{SC}}$ and $\boldsymbol{S}_i$ conditioned on $\boldsymbol{I} = i$ under $\mathcal{D}_{\mathsf{Trap}}$ is equivalent to the one under $\mathcal{D}_{\mathsf{apx}}$, and (ii) the random variables $\boldsymbol{\Pi}_{\mathsf{SC}}$ and $\boldsymbol{S}_i$ are independent of the event $\boldsymbol{I} = i$ (by the definition of $\mathcal{D}_{\mathsf{apx}}$) and hence we can "drop" the conditioning on this event (by Claim 2.1-(6)).

We can further derive,

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{Trap}}}(\Pi_{\mathsf{Trap}}) = \frac{1}{m} \cdot \sum_{i=1}^{m} I_{\mathcal{D}_{\mathsf{apx}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{S}_i\right)$$

$$\leq \frac{1}{m} \cdot \sum_{i=1}^{m} I_{\mathcal{D}_{\mathsf{apx}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{S}_i \mid \boldsymbol{\mathcal{S}}^{<i}\right)$$

The inequality holds since $\boldsymbol{S}_i$ and $\boldsymbol{\mathcal{S}}^{<i}$ are independent and conditioning on independent variables can only increase the mutual information (i.e., Claim 2.2). Finally,

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{Trap}}}(\Pi_{\mathsf{Trap}}) \leq \frac{1}{m} \cdot \sum_{i=1}^{m} I_{\mathcal{D}_{\mathsf{apx}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{S}_i \mid \boldsymbol{\mathcal{S}}^{<i}\right)$$

$$= \frac{1}{m} \cdot I_{\mathcal{D}_{\mathsf{apx}}}\left(\boldsymbol{\Pi}_{\mathsf{SC}}; \boldsymbol{\mathcal{S}}\right) = \frac{1}{m} \cdot \mathsf{ICost}_{\mathcal{D}_{\mathsf{apx}}}(\Pi_{\mathsf{SC}})$$

where the first equality is by the chain rule for mutual information (see Claim 2.1-(5)). □

Having established Lemma 3.3, our task now is to lower bound the information complexity of Trap over the distribution $\mathcal{D}_{\mathsf{Trap}}$. We prove this lower bound using a novel reduction from the well-known *Index* problem, denoted by Index^n_k. In Index^n_k over the distribution $\mathcal{D}_{\mathsf{Index}}$, Alice is given a set $A \subseteq [n]$ of size k chosen uniformly at random and Bob is given an element a such that w.p. $1/2$ $a \in_R A$ and w.p. $1/2$ $a \in_R [n] \setminus A$; Bob needs to determine whether $a \in A$ (the YES case) or not (the NO case).

We remark that similar distributions for Index^n_k have been previously studied (see, e.g., [27], Section 3.3). For the sake of completeness, a self-contained proof of the following lemma is provided in the full version [2].

Lemma 3.4. *For any $k < n/2$, and any constant $\delta' < 1/2$, $\mathsf{IC}^{\delta'}_{\mathcal{D}_{\mathsf{Index}}}(\mathsf{Index}^n_k) = \Omega(k)$.*

Using Lemma 3.4, we prove the following lemma, which is the key part of the proof.

Lemma 3.5. *For any constant $\delta < 1/2$, $\mathsf{IC}^{\delta}_{\mathcal{D}_{\mathsf{Trap}}}(\mathsf{Trap}) = \Omega(n/\alpha)$.*

Proof of Lemma 3.5. Let $k = n/10\alpha$; we design a δ'-error protocol Π_{Index} for Index^n_k using any δ-error protocol Π_{Trap} (over $\mathcal{D}_{\mathsf{Trap}}$) as a subroutine, for some constant $\delta' < 1/2$.

Protocol Π_{Index}. The protocol for reducing Index^n_k to Trap.

Input: An instance $(A, a) \sim \mathcal{D}_{\mathsf{Index}}$. **Output:** YES if $a \in A$ and NO otherwise.

1. Alice picks a set $B \subseteq A$ with $|B| = \ell - 1$ uniformly at random using *private randomness.*
2. To invoke the protocol Π_{Trap}, Alice creates a set $S := A$ and sends the message $\Pi_{\mathsf{Trap}}(S)$, along with the set B to Bob.
3. If $a \in B$, Bob outputs YES and terminates the protocol.
4. Otherwise, Bob constructs a set $E = B \cup \{a\}$ and computes $L := \Pi_{\mathsf{Trap}}(S, E)$ using the message received from Alice.
5. If $a \in L$, Bob outputs NO, and otherwise outputs YES.

We should note right away that the distribution of instances for Trap defined in the previous reduction does *not* match $\mathcal{D}_{\mathsf{Trap}}$. Therefore, we need a more careful argument to establish the correctness of the reduction.

We prove this lemma in two claims; the first claim establishes the correctness of the reduction and the second one proves an upper bound on the information cost of Π_{Index} based on the information cost of Π_{Trap}.

Claim 3.6. *Π_{Index} is a δ'-error protocol for Index^n_k over $\mathcal{D}_{\mathsf{Index}}$ for the parameter $k = n/10\alpha$ and a constant $\delta' < 1/2$.*

Proof. Let $\boldsymbol{R}$ denote the private coins used by Alice to construct the set B. Also, define $\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}$ (resp. $\mathcal{D}^{\mathsf{N}}_{\mathsf{Index}}$) as the distribution of YES instances (resp. NO instances) of $\mathcal{D}_{\mathsf{Index}}$. We have,

$$\Pr_{\mathcal{D}_{\mathsf{Index}}, \boldsymbol{R}}\Big(\Pi_{\mathsf{Index}} \text{ errs}\Big) = \frac{1}{2} \cdot \Pr_{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}, \boldsymbol{R}}\Big(\Pi_{\mathsf{Index}} \text{ errs}\Big) + \frac{1}{2} \cdot \Pr_{\mathcal{D}^{\mathsf{N}}_{\mathsf{Index}}, \boldsymbol{R}}\Big(\Pi_{\mathsf{Index}} \text{ errs}\Big) \quad (1)$$

Note that we do not consider the randomness of the protocol Π_{Trap} (used in construction of Π_{Index}) as it is independent of the randomness of the distribution $\mathcal{D}_{\mathsf{Index}}$ and the private coins $\boldsymbol{R}$. We now bound each term in Equation (1) separately. We first start with the easier case which is the second term.

The distribution of instances (S, E) for Trap created in the reduction by the choice of $(A, a) \sim \mathcal{D}^{\mathsf{N}}_{\mathsf{Index}}$ and the randomness of $\boldsymbol{R}$, is the same as the distribution $\mathcal{D}_{\mathsf{Trap}}$. Moreover, in this case, the output of Π_{Index} would be wrong iff $a \in E \setminus S$ (corresponding to the element e^* in Trap) does not belong to the set L output by Π_{Trap}. Hence,

$$\Pr_{\mathcal{D}^{\mathsf{N}}_{\mathsf{Index}}, \boldsymbol{R}}\Big(\Pi_{\mathsf{Index}} \text{ errs}\Big) = \Pr_{\mathcal{D}_{\mathsf{Trap}}}\Big(\Pi_{\mathsf{Trap}} \text{ errs}\Big) \leq \delta \quad (2)$$

We now bound the first term in Equation (1). Note that when $(A, a) \sim \mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}$, there is a small chance that Π_{Index} is "lucky" and a belongs to the set B (see Line (3) of the protocol). Let this event be $\mathcal{E}$. Conditioned on $\mathcal{E}$, Bob outputs the correct answer with probability 1; however note that probability of $\mathcal{E}$ happening is only $o(1)$. Now suppose $\mathcal{E}$ does not happen. In this case, the distribution of instances (S, E) created by the choice of $(A, a) \sim \mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}$ (and randomness of $\boldsymbol{R}$) does *not* match the distribution $\mathcal{D}_{\mathsf{Trap}}$. However, we have the following important property: Given that (S, E) is the instance of Trap created by choosing (A, a) from $\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}$ and sampling $\ell - 1$ random elements of A (using $\boldsymbol{R}$), the element a is *uniform* over the set E. In other words, knowing (S, E) does not reveal any information about the element a.

Note that since (S, E) is not chosen according to the distribution $\mathcal{D}_{\mathsf{Trap}}$ (actually it is not even a "legal" input for Trap), it is possible that Π_{Trap} terminates, outputs a non-valid set, or outputs a set $L \subseteq E$. Unless $L \subseteq E$ (and satisfies the cardinality constraint), Bob is always able to determine that Π_{Trap} is not functioning correctly and hence outputs YES (and errs with probability at most $\delta < 1/2$). However, if $L \subseteq E$, Bob would not know whether the input to Π_{Trap} is legal or not. In the following, we explicitly analyze this case.

In this case, L is a subset of E chosen by the (inner) randomness of Π_{Trap} for a fixed S and E and moreover $|L| \leq |E|/2$ (by definition of Trap). The probability that Π_{Index} errs in this case is exactly equal to the probability that $a \in L$. However, as stated before, for a fixed (S, E), the choice of L is independent of the choice of a and moreover, a is uniform over E; hence $a \in L$ happens with probability at most $1/2$. Formally,

$$\Pr_{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}, \boldsymbol{R}}(\Pi_{\mathsf{Index}} \text{ errs} \mid \overline{\mathcal{E}}) = \Pr_{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}}, \boldsymbol{R}}\Big(a \in L = \Pi_{\mathsf{Trap}}(\boldsymbol{S}, \boldsymbol{E}) \mid \overline{\mathcal{E}}\Big)$$

Let $\boldsymbol{R}^{\mathsf{Trap}}$ denote the inner randomness of Π_{Trap}. For brevity, we define $(S, E, R^{\mathsf{Trap}})$ as the event that $\boldsymbol{S} = S, \boldsymbol{E} = E$ and $\boldsymbol{R}^{\mathsf{Trap}} = R^{\mathsf{Trap}}$. Using this notation, we can write the RHS above as,

$$\underset{(S,E)\sim(\boldsymbol{S},\boldsymbol{E})\mid\overline{\mathcal{E}}}{\mathrm{E}}\ \underset{R^{\mathsf{Trap}}\sim\boldsymbol{R}^{\mathsf{Trap}}}{\mathrm{E}}\left[\underset{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}},\boldsymbol{R}}{\Pr}\left(a\in L\mid(S,E,R^{\mathsf{Trap}}),\overline{\mathcal{E}}\right)\right]$$

$$(L=\Pi_{\mathsf{Trap}}(S,E)\text{ is a fixed set conditioned on }(S,E,R^{\mathsf{Trap}}))$$

$$=\underset{(S,E)\sim(\boldsymbol{S},\boldsymbol{E})\mid\overline{\mathcal{E}}}{\mathrm{E}}\ \underset{R^{\mathsf{Trap}}\sim\boldsymbol{R}^{\mathsf{Trap}}}{\mathrm{E}}\left[\frac{|L|}{|E|}\right]$$

$$(a\text{ is uniform on }E\text{ conditioned on }(S,E,R^{\mathsf{Trap}})\text{ and }\overline{\mathcal{E}})$$

Hence, we have, $\Pr_{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}},\boldsymbol{R}}(\Pi_{\mathsf{Index}}\text{ errs}\mid\overline{\mathcal{E}})\leq\frac{1}{2}$, since by definition, for any output set L, $|L|\leq|E|/2$.

As stated earlier, whenever $\mathcal{E}$ happens, Π_{Index} makes no error; hence,

$$\underset{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}},\boldsymbol{R}}{\Pr}(\Pi_{\mathsf{Index}}\text{ errs})=\underset{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}},\boldsymbol{R}}{\Pr}(\overline{\mathcal{E}})\cdot\underset{\mathcal{D}^{\mathsf{Y}}_{\mathsf{Index}},\boldsymbol{R}}{\Pr}(\Pi_{\mathsf{Index}}\text{ errs}\mid\overline{\mathcal{E}})$$
$$\leq\frac{1-o(1)}{2}\qquad(3)$$

Finally, by plugging the bounds in Equations (2,3) in Equation (1) and assuming δ is bounded away from $1/2$, we have,

$$\underset{\mathcal{D}_{\mathsf{Index}},\boldsymbol{R}}{\Pr}(\Pi_{\mathsf{Index}}\text{ errs})\leq\frac{1}{2}\cdot\frac{1-o(1)}{2}+\frac{1}{2}\cdot\delta$$
$$=\frac{1-o(1)}{4}+\frac{\delta}{2}\leq\frac{1}{2}-\epsilon$$

for some constant ϵ bounded away from 0. □

We now bound the information cost of Π_{Index} under $\mathcal{D}_{\mathsf{Index}}$.

Claim 3.7. *We have,*

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{Index}}}(\Pi_{\mathsf{Index}})\leq\mathsf{ICost}_{\mathcal{D}_{\mathsf{Trap}}}(\Pi_{\mathsf{Trap}})+O(\ell\log n)$$

Proof.

$$\mathsf{ICost}_{\mathcal{D}_{\mathsf{Index}}}(\Pi_{\mathsf{Index}})$$
$$=I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{\Pi}_{\mathsf{Index}}(\boldsymbol{A});\boldsymbol{A}\Big)$$
$$=I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S}),\boldsymbol{B};\boldsymbol{A}\Big)$$
$$=I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S});\boldsymbol{A}\Big)+I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{B};\boldsymbol{A}\mid\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S})\Big)$$
$$(\text{the chain rule for mutual information, Claim 2.1-(5)})$$
$$\leq I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S});\boldsymbol{A}\Big)+H_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{B}\mid\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S})\Big)$$
$$\leq I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S});\boldsymbol{A}\Big)+O(\ell\log n)$$
$$(|\boldsymbol{B}|=O(\ell\log n)\text{ and Claim 2.1-(1)})$$
$$=I_{\mathcal{D}_{\mathsf{Index}}}\Big(\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S});\boldsymbol{S}\Big)+O(\ell\log n)$$
$$=I_{\mathcal{D}_{\mathsf{Trap}}}\Big(\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S});\boldsymbol{S}\Big)+O(\ell\log n)$$
$$((\boldsymbol{\Pi}_{\mathsf{Trap}}(\boldsymbol{S}),\boldsymbol{S})\text{ is distributed the same under }\mathcal{D}_{\mathsf{Index}},\mathcal{D}_{\mathsf{Trap}})$$
$$=\mathsf{ICost}_{\mathcal{D}_{\mathsf{Trap}}}(\Pi_{\mathsf{Trap}})+O(\ell\log n)$$

□

The lower bound now follows from Claims 3.6 and 3.7, and Lemma 3.4 for the parameters $k=|S|=\frac{n}{10\alpha}$ and $\delta'<1/2$, and using the fact that $\alpha=o(\sqrt{n}/\log n)$, $\ell=2\alpha\log m$, and $m=\mathrm{poly}(n)$, and hence $\Omega(n/\alpha)=\omega(\ell\log n)$.

To conclude, by Lemma 3.3 and Lemma 3.5, for any set of parameters $\delta<1/2$, $\alpha=o(\frac{\sqrt{n}}{\log n})$, and $m=\mathrm{poly}(n)$,

$$\mathsf{IC}^{\delta}_{\mathcal{D}_{\mathsf{apx}}}(\mathsf{SetCover}_{\mathsf{apx}})\geq m\cdot\Big(\Omega(n/\alpha)\Big)=\Omega(mn/\alpha)$$

Since the information complexity is a lower bound on the communication complexity (Proposition 2.4), we have,

Theorem 2. *For any constant $\delta<1/2$, $\alpha=o(\frac{\sqrt{n}}{\log n})$, and $m=poly(n)$,*

$$\mathsf{CC}^{\delta}_{\mathcal{D}_{\mathsf{apx}}}(\mathsf{SetCover}_{\mathsf{apx}})=\Omega(mn/\alpha).$$

Finally, since one-way communication complexity is also a lower bound on the space complexity of single-pass streaming algorithms, we obtain Theorem 1 as a corollary of Theorem 2.

4. AN UPPER BOUND FOR ESTIMATING COVERING ILPS

In this section, we show that if we are only interested in estimating the *size* of a minimum set cover (instead of finding the actual sets), we can bypass the $\Omega(mn/\alpha)$ lower bound established in Section 3. In fact, we prove this upper bound for the more general problem of estimating the optimal solution of a *covering integer program* (henceforth, *covering ILP*) in the streaming setting.

A covering ILP can be formally defined as follows.

$$\min c\cdot x\quad\text{s.t.}\quad Ax\geq b$$

where A is a matrix with dimension $n\times m$, b is a vector of dimension n, c is a vector of dimension m, and x is an m-dimensional vector of non-negative integer variables. Moreover, all coefficients in A,b, and c are also non-negative integers. We denote this linear program by $\mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$. We use a_{max}, b_{max}, and c_{max}, to denote the *largest* entry of, respectively, the matrix A, the vector b, and the vector c. Finally, we define the *optimal value* of the $\mathcal{I}:=\mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$ as $c\cdot x^*$ where x^* is the *optimal* solution to $\mathcal{I}$, and denote it by $opt:=opt(\mathcal{I})$.

We consider the following streaming setting for covering ILPs. The input to a streaming algorithm for an instance $\mathcal{I}:=\mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$ is the n-dimensional vector b, and a stream of the m columns of A presented one by one, where the i-th column of A, A_i, is presented along with the i-th entry of c, denoted by c_i (we will refer to c_i as the weight of the i-th column). It is easy to see that this streaming setting for covering ILPs captures, as special cases, the *set cover* problem, the *weighted set cover* problem, and the *set multi-cover* problem. We prove the following theorem for α-estimating the optimal value of a covering ILPs in the streaming setting.

Theorem 3. *There is a randomized algorithm that given a parameter $\alpha\geq 1$, for any instance $\mathcal{I}:=\mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$ with poly(n)-bounded entries, makes a single pass over a stream of columns of A (presented in an arbitrary order), and outputs an α-estimation to $opt(\mathcal{I})$ w.h.p. using space $\widetilde{O}\left((mn/\alpha^2)\cdot b_{\mathsf{max}}+m+nb_{\mathsf{max}}\right)$ bits.*

In particular, for the weighted set cover *problem with poly(n) bounded weights and $\alpha\leq\sqrt{n}$, the space complexity of this algorithm is $\widetilde{O}(mn/\alpha^2+n)$.*[5]

To prove Theorem 3, we design a general approach based on sampling constraints of a covering ILP instance. The

[5] Note that $\Omega(n)$ space is necessary to even determine whether or not a given instance is feasible.

goal is to show that if we sample (roughly) $1/\alpha$ fraction of the constraints from an instance $\mathcal{I} := \mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$, then the optimum value of the resulting covering ILP, denoted by $\mathcal{I}_{\mathsf{R}}$, is a good estimator of $opt(\mathcal{I})$. Note that in general, this may not be the case; simply consider a weighted set cover instance that contains an element e which is only covered by a singleton set of weight W (for $W \gg m$) and all the remaining sets are of weight 1 only. Clearly, $opt(\mathcal{I}_{\mathsf{R}}) \ll opt(\mathcal{I})$ as long as e is not sampled in $\mathcal{I}_{\mathsf{R}}$, which happens w.p. $1 - 1/\alpha$.

To circumvent this issue, we define a notion of *cost* for covering ILPs which, informally, is the minimum value of the objective function if the goal is to only satisfy a single constraint (in the above example, the cost of that weighted set cover instance is W). This allows us to bound the loss incurred in the process of estimation by sampling based on the cost of the covering ILP.

Constraint sampling alone can only reduce the space requirement by a factor of α, which is not enough to meet the bounds given in Theorem 3. Hence, we combine it with a *pruning* step to sparsify the columns in A before performing the sampling. We should point out that as columns are weighted, the pruning step needs to be sensitive to the weights.

In the rest of this section, we first introduce our *constraint sampling lemma* (Lemma 4.1) and prove its correctness, and then provide our algorithm for Theorem 3.

4.1 Covering ILPs and Constraint Sampling Lemma

In this section, we provide a general result for estimating the optimal value of a Covering ILP using a sampling based approach. For a vector v, we will use v_i to denote the i-th dimension of v. For a matrix A, we will use A_i to denote the i-th column of A, and use $a_{j,i}$ to denote the entry of A at the i-th column and the j-th row (to match the notation with the set cover problem, we use $a_{j,i}$ instead of the standard notation $a_{i,j}$).

For each constraint $j \in [n]$ (i.e., the j-th constraint) of a covering ILP instance $\mathcal{I} := \mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$, we define the *cost* of the constraint j, denoted by $Cost(j)$, as,

$$Cost(j) := \min_{x} c \cdot x \quad \text{s.t} \ \sum_{i=1}^{m} a_{j,i} \cdot x_i \geq b_j$$

which is the *minimum solution value* of the objective function for satisfying the constraint j. Furthermore, the *cost* of $\mathcal{I}$, denoted by $Cost(\mathcal{I})$, is defined to be

$$Cost(\mathcal{I}) := \max_{j \in [n]} Cost(j)$$

Clearly, $Cost(\mathcal{I})$ is a lower bound on $opt(\mathcal{I})$.

Constraint Sampling. Given any instance of covering ILP $\mathcal{I} := \mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$, let $\mathcal{I}_{\mathsf{R}}$ be a covering ILP instance $\mathsf{ILP}_{\mathsf{Cover}}(A,\widetilde{b},c)$ obtained by setting $\widetilde{b}_j := b_j$ with probability p, and $\widetilde{b}_j := 0$ with probability $1-p$, for each dimension $j \in [n]$ of b independently. Note that setting $\widetilde{b}_j := 0$ in $\mathcal{I}_{\mathsf{R}}$ is equivalent to removing the j-th constraint from $\mathcal{I}$, since all entries in $\mathcal{I}$ are non-negative. Therefore, intuitively, $\mathcal{I}_{\mathsf{R}}$ is a covering ILP obtained by sampling (and keeping) the constraints of $\mathcal{I}$ with a sampling rate of p.

We establish the following lemma that asserts that $opt(\mathcal{I}_{\mathsf{R}})$ is a good estimator of $opt(\mathcal{I})$ (under certain conditions). As $opt(\mathcal{I}_{\mathsf{R}}) \leq opt(\mathcal{I})$ trivially holds (removing constraints can only decrease the optimal value), it suffices to give a lower bound on $opt(\mathcal{I}_{\mathsf{R}})$.

Lemma 4.1 (Constraint Sampling Lemma). *Fix an $\alpha \geq 32 \ln n$; for any covering ILP $\mathcal{I}$ with n constraints, suppose $\mathcal{I}_R$ is obtained from $\mathcal{I}$ by sampling each constraint with probability $p := \frac{4 \ln n}{\alpha}$; then*

$$\Pr\left(opt(\mathcal{I}_R) + Cost(\mathcal{I}) \geq \frac{opt(\mathcal{I})}{8\alpha}\right) \geq \frac{3}{4}$$

Proof. Suppose by contradiction that the lemma statement is false and throughout the proof let $\mathcal{I}$ be any instance where w.p. at least $1/4$, $opt(\mathcal{I}_{\mathsf{R}}) + Cost(\mathcal{I}) < \frac{opt(\mathcal{I})}{8\alpha}$ (we denote this event by $\mathcal{E}_1(\mathcal{I}_{\mathsf{R}})$, or shortly $\mathcal{E}_1$). We will show that in this case, $\mathcal{I}$ has a feasible solution with a value smaller than $opt(\mathcal{I})$. To continue, define $\mathcal{E}_2(\mathcal{I}_{\mathsf{R}})$ (or $\mathcal{E}_2$ in short) as the event that $opt(\mathcal{I}_{\mathsf{R}}) < \frac{opt(\mathcal{I})}{8\alpha}$. Note that whenever $\mathcal{E}_1$ happens, then $\mathcal{E}_2$ also happens, hence $\mathcal{E}_2$ happens w.p. at least $1/4$.

For the sake of analysis, suppose we repeat, for 32α times, the procedure of sampling each constraint of $\mathcal{I}$ independently with probability p, and obtain 32α covering ILP instances $S := \left\{\mathcal{I}_{\mathsf{R}}^1, \ldots, \mathcal{I}_{\mathsf{R}}^{32\alpha}\right\}$. Since $\mathcal{E}_2$ happens with probability at least $1/4$ on each instance $\mathcal{I}_{\mathsf{R}}$, the expected number of times that $\mathcal{E}_2$ happens for instances in S is at least $8\alpha > 12 \ln n$. Hence, by the Chernoff bound, with probability at least $1 - 1/n$, $\mathcal{E}_2$ happens on at least 4α of instances in S. Let $T \subseteq S$ be a set of 4α instances for which $\mathcal{E}_2$ happens. In the following, we show that if $\mathcal{I}$ has the property that $\Pr(\mathcal{E}_1(\mathcal{I}_R)) \geq 1/4$, then w.p. at least $1 - 1/n$, every constraint in $\mathcal{I}$ appears in at least one of the instances in T. Since each of these 4α sampled instances admits a solution of value at most $\frac{opt(\mathcal{I})}{8\alpha}$ (by the definition of $\mathcal{E}_2$), the "max" of their solutions, i.e., the vector obtained by setting the i-th entry to be the largest value of x_i among all these solutions, gives a feasible solution to $\mathcal{I}$ with value at most $4\alpha \cdot \frac{opt(\mathcal{I})}{8\alpha} = \frac{opt(\mathcal{I})}{2}$; a contradiction.

We use "$j \in \mathcal{I}_{\mathsf{R}}$" to denote the event that the constraint j of $\mathcal{I}$ is sampled in $\mathcal{I}_{\mathsf{R}}$, and we need to show that w.h.p. for all j, there exists an instance $\mathcal{I}_{\mathsf{R}} \in T$ where $j \in \mathcal{I}_{\mathsf{R}}$. We establish the following claim.

Claim 4.2. *For any $j \in [n]$, $\Pr\left(j \in \mathcal{I}_R \mid \mathcal{E}_2(\mathcal{I}_R)\right) \geq \frac{\ln n}{2\alpha}$.*

Before proving Claim 4.2, we show how this claim would imply the lemma. By Claim 4.2, for each of the 4α instances $\mathcal{I}_{\mathsf{R}} \in T$, and for any $j \in [n]$, the probability that the constraint j is sampled in $\mathcal{I}_{\mathsf{R}}$ is at least $\frac{\ln n}{2\alpha}$. Then, the probability that j is sampled in none of the 4α instances of T is at most:

$$\left(1 - \frac{\ln n}{2\alpha}\right)^{4\alpha} \leq \exp(-2 \ln n) = \frac{1}{n^2}$$

Hence, by union bound, w.p. at least $1 - 1/n$, every constraint appears in at least one of the instances in T, and this will complete the proof. It remains to prove Claim 4.2.

Proof of Claim 4.2. Fix any $j \in [n]$; by Bayes rule,

$$\Pr\left(j \in \mathcal{I}_{\mathsf{R}} \mid \mathcal{E}_2(\mathcal{I}_{\mathsf{R}})\right) = \frac{\Pr\left(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}\right) \cdot \Pr\left(j \in \mathcal{I}_{\mathsf{R}}\right)}{\Pr\left(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}})\right)}$$

Since $\Pr\left(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}})\right) \leq 1$ and $\Pr(j \in \mathcal{I}_{\mathsf{R}}) = p = \frac{4\ln n}{\alpha}$, we have,

$$\Pr\left(j \in \mathcal{I}_{\mathsf{R}} \mid \mathcal{E}_2(\mathcal{I}_{\mathsf{R}})\right) \geq \Pr\left(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}\right) \cdot \frac{4\ln n}{\alpha} \qquad (4)$$

and hence it suffices to establish a lower bound of $1/8$ for $\Pr\left(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}\right)$.

Consider the following probabilistic process (for a fixed $j \in [n]$): we first remove the constraint j from $\mathcal{I}$ (w.p. 1) and then sample each of the remaining constraints of $\mathcal{I}$ w.p. p. Let $\mathcal{I}'_{\mathsf{R}}$ be an instance created by this process. We prove $\Pr(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}) \geq 1/8$ in two steps by first showing that the probability that $\mathcal{E}_1$ happens to $\mathcal{I}'_{\mathsf{R}}$ (i.e., $\Pr(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}}))$) is at least $1/8$, and then use a coupling argument to prove that $\Pr(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}) \geq \Pr(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}}))$.

We first show that $\Pr(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}}))$ (which by definition is the probability that $opt(\mathcal{I}'_{\mathsf{R}}) + Cost(\mathcal{I}) \leq \frac{opt(\mathcal{I})}{16\alpha}$) is at least $1/8$. To see this, note that the probability that $\mathcal{E}_1$ happens to $\mathcal{I}'_{\mathsf{R}}$ is equal to the probability that $\mathcal{E}_1$ happens to $\mathcal{I}_{\mathsf{R}}$ conditioned on j not being sampled (i.e., $\Pr(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}}) \mid j \notin \mathcal{I}_{\mathsf{R}})$). Now, if we expand $\Pr(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}}))$,

$$\begin{aligned}\Pr\left(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}})\right) &= \Pr(j \in \mathcal{I}_{\mathsf{R}}) \Pr\left(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}\right) \\ &\qquad + \Pr(j \notin \mathcal{I}_{\mathsf{R}}) \Pr\left(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}}) \mid j \notin \mathcal{I}_{\mathsf{R}}\right) \\ &\leq \Pr(j \in \mathcal{I}_{\mathsf{R}}) + \Pr\left(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}}) \mid j \notin \mathcal{I}_{\mathsf{R}}\right) \\ &= p + \Pr\left(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}})\right)\end{aligned}$$

As $\Pr(\mathcal{E}_1(\mathcal{I}_{\mathsf{R}})) \geq 1/4$ and $p = \frac{4\ln n}{\alpha} \leq 1/8$ (since $\alpha \geq 32\ln n$), we have,

$$1/4 \leq 1/8 + \Pr\left(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}})\right)$$

and therefore, $\Pr(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}})) \geq 1/8$.

It remains to show that $\Pr(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}) \geq \Pr(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}}))$. To see this, note that conditioned on $j \in \mathcal{I}_{\mathsf{R}}$, the distribution of sampling all constraints other than j is exactly the same as the distribution of $\mathcal{I}'_{\mathsf{R}}$. Therefore, for any instance $\mathcal{I}'_{\mathsf{R}}$ drawn from this distribution, there is a unique instance $\mathcal{I}_{\mathsf{R}}$ sampled from the original constraint sampling distribution *conditioned* on $j \in \mathcal{I}_{\mathsf{R}}$. For any such $(\mathcal{I}'_{\mathsf{R}}, \mathcal{I}_{\mathsf{R}})$ pair, we have $opt(\mathcal{I}_{\mathsf{R}}) \leq opt(\mathcal{I}'_{\mathsf{R}}) + Cost(j)$ $(\leq opt(\mathcal{I}'_{\mathsf{R}}) + Cost(\mathcal{I}))$ since satisfying the constraint $j \in \mathcal{I}_{\mathsf{R}}$ requires increasing the value of the objective function in $\mathcal{I}'_{\mathsf{R}}$ by at most $Cost(j)$. Therefore if $opt(\mathcal{I}'_{\mathsf{R}}) + Cost(\mathcal{I}) \leq \frac{opt}{8\alpha}$ (i.e., $\mathcal{E}_1$ happens to $\mathcal{I}'_{\mathsf{R}}$), then $opt(\mathcal{I}_{\mathsf{R}}) \leq \frac{opt}{8\alpha}$ (i.e., $\mathcal{E}_2$ happens to $\mathcal{I}_{\mathsf{R}}$ conditioned on $j \in \mathcal{I}_{\mathsf{R}}$). Hence,

$$\Pr\left(\mathcal{E}_2(\mathcal{I}_{\mathsf{R}}) \mid j \in \mathcal{I}_{\mathsf{R}}\right) \geq \Pr\left(\mathcal{E}_1(\mathcal{I}'_{\mathsf{R}})\right) \geq 1/8$$

Plugging in this bound in Equation (4), we obtain that $\Pr(j \in \mathcal{I}_{\mathsf{R}} \mid \mathcal{E}_2) \geq \frac{\ln n}{2\alpha}$. □

4.2 An α-estimation of Covering ILPs in the Streaming Setting

We now prove Theorem 3. Throughout this section, for simplicity of exposition, we assume that $\alpha \geq 32\ln n$ (otherwise the space bound in Theorem 3 is enough to store the whole input and solve the problem optimally), the value of c_{max} is provided to the algorithm, and x is a vector of *binary* variables, i.e., $x \in \{0,1\}^m$ (hence covering ILP instances are always referring to covering ILP instances with binary variables). In the full version of the paper [2], we describe how to eliminate the later two assumptions.

Algorithm overview. For any covering ILP instance $\mathcal{I} := \mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$, our algorithm estimates $opt := opt(\mathcal{I})$ in two parts running in parallel. In the first part, the goal is simply to compute $Cost(\mathcal{I})$ (see Claim 4.3). For the second part, we design a tester algorithm (henceforth, *Tester*) that given any "guess" k of the value of opt, if $k \geq opt$, *Tester* accepts k w.p. 1 and for any k where $Cost(\mathcal{I}) \leq k \leq \frac{opt}{32\alpha}$, w.h.p. *Tester* rejects k.

Let $K := \{2^\gamma\}_{\gamma \in [\lceil \log(mc_{\mathsf{max}}) \rceil]}$; for each $k \in K$ (in parallel), we run *Tester*(k). At the end of the stream, the algorithm knows $Cost(\mathcal{I})$ (using the output of the part one), and hence it can identify among all guesses that are *at least* $Cost(\mathcal{I})$, the smallest guess accepted by *Tester* (denoted by k^*). On one hand, $k^* \leq opt$ since for any guess $k \geq opt$, $k \geq Cost(\mathcal{I})$ also (since $opt \geq Cost(\mathcal{I})$) and *Tester* accepts k. On the other hand, $k^* \geq \frac{opt}{32\alpha}$ w.h.p. since (*i*) if $Cost(\mathcal{I}) \geq \frac{opt}{32\alpha}$, $k^* \geq Cost(\mathcal{I}) \geq \frac{opt}{32\alpha}$ and (*ii*) if $Cost(\mathcal{I}) < \frac{opt}{32\alpha}$, the guess $\frac{opt}{32\alpha}$ will be rejected w.h.p. by *Tester*. Consequently, $32\alpha \cdot k^*$ is an $O(\alpha)$-estimation of $opt(\mathcal{I})$.

There is a simple *dynamic programming* algorithm that can compute the *Cost* of a covering ILP presented in a stream (see the full version [2] for a proof).

Claim 4.3. *For any $\mathcal{I} := \mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$ presented in a stream, $Cost(\mathcal{I})$ can be computed in space $O(nb_{\mathsf{max}} \log c_{\mathsf{max}})$ bits.*

To continue, we need the following notation. For any vector v with dimension d and any set $S \subseteq [d]$, $v(S)$ denotes the projection of v onto the dimensions indexed by S. For any two vectors u and v, let $\min(u,v)$ denote a vector w where at the i-th dimension: $w_i = \min(u_i, v_i)$, i.e., the *coordinate-wise minimum*. We now provide the aforementioned *Tester* algorithm.

Tester(k)**:** An algorithm for testing a guess k of the optimal value of a covering ILP.

Input: An instance $\mathcal{I} := \mathsf{ILP}_{\mathsf{Cover}}(A,b,c)$ presented as a stream $\langle A_1, c_1 \rangle, \ldots, \langle A_m, c_m \rangle$, a parameter $\alpha \geq 32\ln n$, and a guess $k \in K$.

Output: ACCEPT if $k \geq opt$ and REJECT if $Cost(\mathcal{I}) \leq k \leq \frac{opt}{32\alpha}$. The answer could be either ACCEPT or REJECT if $\frac{opt}{32\alpha} < k < opt$.

```
1. Preprocessing:
   (i) Maintain an $n$-dimensional vector $b_{\mathsf{res}} \leftarrow b$, an $m$-dimensional vector $\widetilde{c} \leftarrow 0^m$, and an $n \times m$ dimensional matrix $\widetilde{A} \leftarrow 0^{n \times m}$.
   (ii) Let $V$ be a subset of $[n]$ obtained by sampling each element in $[n]$ independently with probability $p := 4\ln n/\alpha$.
2. Streaming: when a pair $\langle A_i, c_i \rangle$ arrives:
   (i) If $c_i > k$, directly continue to the next input pair of the stream. Otherwise:
```

(ii) *Prune step*: Let $u_i := \min(b_{\mathsf{res}}, A_i)$ (the coordinate-wise minimum). If $\|u_i\|_1 \geq \frac{n\cdot b_{\mathsf{max}}}{\alpha}$, update $b_{\mathsf{res}} \leftarrow b_{\mathsf{res}} - u_i$ (we say $\langle A_i, c_i\rangle$ is *pruned* by *Tester* in this case). Otherwise, assign $\widetilde{A}_i \leftarrow u_i(V)$, and $\widetilde{c}_i \leftarrow c_i$.

3. At the end of the stream, solve the following covering ILP (denoted by $\mathcal{I}_{tester}$):

$$\min \widetilde{c}\cdot x \quad \text{s.t.} \quad \widetilde{A}x \geq b_{\mathsf{res}}(V)$$

If $opt(\mathcal{I}_{tester})$ is at most k, ACCEPT; otherwise REJECT.

We first make the following observation. In the prune step of *Tester*, if we replace $\widetilde{A}_i \leftarrow u_i(V)$ by $\widetilde{A}_i \leftarrow A_i(V)$, the solution of the resulting covering ILP instance (denoted by $\mathcal{I}'_{tester}$) has the property that $opt(\mathcal{I}'_{tester}) = opt(\mathcal{I}_{tester})$ (we use $\mathcal{I}_{tester}$ only to control the space requirement). To see this, let ${b_{\mathsf{res}}}^i$ denotes the content of the vector b_{res} when $\langle A_i, c_i\rangle$ arrives. By construction, $(u_i)_j := \min(({b_{\mathsf{res}}}^i)_j, a_{j,i})$, and hence if $(u_i)_j \neq a_{j,i}$, then both $(u_i)_j$ and $a_{j,i}$ are at least $({b_{\mathsf{res}}}^i)_j$, which is at least $(b_{\mathsf{res}})_j$ (since every dimension of b_{res} is monotonically decreasing). However, for any *integer* program $\mathsf{ILP}_{\mathsf{Cover}}(A, b, c)$, changing any entry $a_{j,i}$ of A between two values that are at least b_j does not change the optimal value, and hence $opt(\mathcal{I}'_{tester}) = opt(\mathcal{I}_{tester})$. To simplify the proof, in the following, when concerning $opt(\mathcal{I}_{tester})$, we redefine $\mathcal{I}_{tester}$ to be $\mathcal{I}'_{tester}$.

We now prove the correctness of *Tester* in the following two lemmas.

Lemma 4.4. *For any guess $k \geq opt$,* $\Pr\Big(\mathit{Tester}(k) = \mathsf{ACCEPT}\Big) = 1.$

Proof. Fix any optimal solution x^* of $\mathcal{I}$; we will show that x^* is a feasible solution for $\mathcal{I}_{tester}$, and since by the construction of $\widetilde{c}$, we have $\widetilde{c}\cdot x^* \leq c\cdot x^* \leq opt$, this will show that $opt(\mathcal{I}_{tester}) \leq opt \leq k$ and hence $\mathit{Tester}(k) = \mathsf{ACCEPT}$.

Fix a constraint j in $\mathcal{I}_{tester}$: $\Big\{\sum_{i\in[m]} \widetilde{a}_{j,i}x_i \geq b_{\mathsf{res}}(V)_j\Big\}$. If $j \notin V$, $b_{\mathsf{res}}(V)_j = 0$ and hence the constraint is trivially satisfied for any solution x^*. Suppose $j \in V$ and let P denote the set of (indices of) pairs that are pruned. By construction of the *Tester*, $b_{\mathsf{res}}(V)_j = \max(b_j - \sum_{i\in P} a_{j,i}, 0)$. If $b_{\mathsf{res}}(V)_j = 0$, again the constraint is trivially satisfied. Suppose $b_{\mathsf{res}}(V)_j = b_j - \sum_{i\in P} a_{j,i}$. The constraint j can be written as $\big\{\sum_i \widetilde{a}_{j,i}x_i \geq b_j - \sum_{i\in P} a_{j,i}\big\}$. By construction of the tester, $\widetilde{a}_{j,i} = 0$ for all i that are pruned and otherwise $\widetilde{a}_{j,i} = a_{j,i}$. Hence, we can further write the constraint j as $\big\{\sum_{i\notin P} a_{j,i}x_i \geq b_j - \sum_{i\in P} a_{j,i}\big\}$.

Now, since x^* satisfies the constraint j in $\mathcal{I}$,

$$\begin{aligned}\sum_{i\in[m]} a_{j,i}x^*_i &\geq b_j \\ \sum_{i\notin P} a_{j,i}x^* &\geq b_j - \sum_{i\in P} a_{j,i}x^*_i \\ &\geq b_j - \sum_{i\in P} a_{j,i} \qquad\qquad (x^*_i \leq 1)\end{aligned}$$

and the constraint j is satisfied by x^* in $\mathcal{I}_{tester}$ as well. Therefore, x^* is a feasible solution of $\mathcal{I}_{tester}$; this completes the proof. □

We now show that *Tester* will reject guesses that are smaller than $\frac{opt}{32\alpha}$. We will only prove that the rejection happens with probability 3/4; however, the probability of error can be reduced to any $\delta < 1$ by running $O(\log 1/\delta)$ parallel instances of the *Tester* and for each guess, REJECT if any one of the instances outputs REJECT and otherwise ACCEPT. In our case $\delta = O(|K|^{-1})$ so we can apply union bound for all different guesses.

Lemma 4.5. *For any guess k where $Cost(\mathcal{I}) \leq k < \frac{opt}{32\alpha}$,* $\Pr\Big(\mathit{Tester}(k) = \mathsf{REJECT}\Big) \geq 3/4.$

Proof. By construction of $\mathit{Tester}(k)$, we need to prove that $\Pr\Big(opt(\mathcal{I}_{tester}) > k\Big) \geq 3/4$. Define the following covering ILP $\mathcal{I}'$:

$$\min \widetilde{c}\cdot x \quad \text{s.t.} \quad \widehat{A}x \geq b_{\mathsf{res}}(V)$$

where $\widehat{A}_i = A_i$ if $\langle A_i, c_i\rangle$ is not pruned by *Tester*, and $\widehat{A}_i = 0^n$ otherwise. In $\mathit{Tester}(k)$, for each pair $\langle A_i, c_i\rangle$ that is not pruned, instead of storing the entire vector A_i, we store the projection of A_i onto dimensions indexed by V (which is the definition of $\widetilde{A}_i$ in $\mathcal{I}_{tester}$). This is equivalent to performing constraint sampling on $\mathcal{I}'$ with a sampling rate of $p = 4\ln n/\alpha$. Therefore, by Lemma 4.1, with probability at least 3/4, $opt(\mathcal{I}_{tester}) + Cost(\mathcal{I}') \geq \frac{opt(\mathcal{I}')}{8\alpha}$. Since $Cost(\mathcal{I}') \leq Cost(\mathcal{I}) \leq k < \frac{opt(\mathcal{I})}{32\alpha}$, this implies that

$$opt(\mathcal{I}_{tester}) \geq \frac{opt(\mathcal{I}')}{8\alpha} - Cost(\mathcal{I}') > \frac{opt(\mathcal{I}')}{8\alpha} - \frac{opt(\mathcal{I})}{32\alpha}.$$

Therefore, we only need to show that $opt(\mathcal{I}') \geq \frac{opt(\mathcal{I})}{2}$ since then $opt(\mathcal{I}_{tester}) > \frac{opt(\mathcal{I})}{16\alpha} - \frac{opt(\mathcal{I})}{32\alpha} = \frac{opt(\mathcal{I})}{32\alpha} > k$ and *Tester* will reject k.

To show that $opt(\mathcal{I}') \geq \frac{opt(\mathcal{I})}{2}$, we first note that for any optimal solution x^* of $\mathcal{I}'$, if we further set $x^*_i = 1$ for any pair $\langle A_i, c_i\rangle$ that are pruned, the resulting x^*_i is a feasible solution for $\mathcal{I}$. Therefore, if we show that the total weight of the $\langle A_i, c_i\rangle$ pairs that are pruned is at most $\frac{opt}{2}$, $opt(\mathcal{I}')$ must be at least $\frac{opt}{2}$ or we will have a solution for $\mathcal{I}$ better than $opt(\mathcal{I})$.

To see that the total weight of the pruned pairs is at most $opt/2$, since only pairs with $c_i \leq k$ $(\leq \frac{opt}{32\alpha})$ will be considered, we only need to show that at most 16α pairs can be pruned. By the construction of the prune step, each pruned pair reduces the ℓ_1-norm of the vector b_{res} by an additive factor of at least $\frac{nb_{\mathsf{max}}}{\alpha}$. Since b_{res} is initialized to be b and $\|b\|_1 \leq nb_{\mathsf{max}}$, at most α $(\leq 16\alpha)$ pairs can be pruned. This completes the proof. □

We now finalize the proof of Theorem 3.

Proof of Theorem 3. We run the algorithm described in the beginning of this section. The correctness of the algorithm follows from Claim 4.3 and Lemmas 4.4 and 4.5. We now analyze the space complexity of this algorithm. We need to run the algorithm in Claim 4.3 to compute $Cost(\mathcal{I})$, which require $\widetilde{O}(nb_{\mathsf{max}})$ space. We also need to run *Tester* for $O(\log(m\cdot c_{\mathsf{max}}))$ different guesses of k.

In $\mathit{Tester}(k)$, we need $O(n\log b_{\mathsf{max}})$ bits to store the vector b_{res} and $O(m\log c_{\mathsf{max}})$ bits to maintain the vector $\widetilde{c}$. Finally, the matrix $\widetilde{A}$ requires $O(mnb_{\mathsf{max}}/\alpha\cdot(\log n/\alpha)\cdot(\log a_{\mathsf{max}}\log n))$ bits to store. This is because each column $\widetilde{A}_i$ of $\widetilde{A}$ is either 0^n

or $u_i(V)$ where $\|u_i\|_1 < \frac{n \cdot b_{\mathsf{max}}}{\alpha}$. Since $\|u_i\|_1 < \frac{n \cdot b_{\mathsf{max}}}{\alpha}$, there are at most $\frac{n \cdot b_{\mathsf{max}}}{\alpha}$ non-zero entries in u_i. Therefore, after projecting u_i to V (to obtain $\widetilde{A}_i$) in expectation the number of non-zero entries in $\widetilde{A}_i$ is at most $\widetilde{O}(nb_{\mathsf{max}}/\alpha^2)$. Using the Chernoff bound w.h.p at most $O(\frac{nb_{\mathsf{max}}}{\alpha^2})$ non-zero entries of u_i remain in each $\widetilde{A}_i$, where each entry needs $O(\log a_{\mathsf{max}} \log n)$ bits to store. Note that the space complexity of the algorithm can be made *deterministic* by simply terminating the execution when at least one set $\widetilde{A}_i$ has $(c \cdot \frac{nb_{\mathsf{max}}}{\alpha^2})$ non-zero entries (for a sufficiently large constant $c > 1$); as this event happens with $o(1)$ probability, the error probability of the algorithm increases only by $o(1)$. Finally, as all entries in (A, b, c) are poly(n)-bounded, the total space requirement of the algorithm is $\widetilde{O}((mn/\alpha^2) \cdot b_{\mathsf{max}} + m + nb_{\mathsf{max}})$. □

We also make the following remark about α-approximating covering ILPs.

REMARK 4.6. *The algorithm described in Section 1.2 for α-approximating set cover can also be extended to obtain an α-*approximation algorithm *for covering ILPs in space $\widetilde{O}(mnb_{\mathsf{max}}/\alpha)$: Group the columns by the weights and merge every α sets for each group independently.*

5. A LOWER BOUND FOR ESTIMATING SET COVER

Our algorithm in Theorem 3 establish a factor α gap on the space requirement of α-approximation and α-estimation algorithms for the set cover problem. We now show that this gap is the best possible. In other words, the space complexity of our algorithm in Theorem 3 for the original set cover problem is tight (up to logarithmic factors) *even for random arrival streams.* Formally,

THEOREM 4. *Let $\mathcal{S}$ be a collection of m subsets of $[n]$ presented one by one in a* random order*. For any $\alpha = o(\sqrt{\frac{n}{\log n}})$ and any $m =$ poly(n), any* randomized *algorithm that makes a single pass over $\mathcal{S}$ and outputs an α-estimation of the set cover problem with probability* 0.9 *(over the randomness of both the stream order and the algorithm) must use $\widetilde{\Omega}(\frac{mn}{\alpha^2})$ bits of space.*

Fix a (sufficiently large) value for n, $m = \text{poly}(n)$, and $\alpha = o(\sqrt{\frac{n}{\log n}})$; throughout this section, $\mathsf{SetCover}_{\mathsf{est}}$ refers to the problem of α-estimating the set cover problem with $m + 1$ sets (see footnote 4) defined over the universe $[n]$ in the one-way communication model, whereby the sets are partitioned between Alice and Bob.

Overview. We start by introducing a hard distribution $\mathcal{D}_{\mathsf{est}}$ for $\mathsf{SetCover}_{\mathsf{est}}$ in the spirit of the distribution $\mathcal{D}_{\mathsf{apx}}$ in Section 3. However, since in $\mathsf{SetCover}_{\mathsf{est}}$ the goal is only to estimate the *size* of the optimal cover, "hiding" one single element (as was done in $\mathcal{D}_{\mathsf{apx}}$) is not enough for the lower bound. Here, instead of trying to hide a single element, we give Bob a "block" of elements and his goal would be to decide whether this block appeared in one set of Alice as a whole or was it partitioned across many different sets[6]. Similar to $\mathcal{D}_{\mathsf{apx}}$, distribution $\mathcal{D}_{\mathsf{est}}$ is also not a product distribution; however, we introduce a way of decomposing $\mathcal{D}_{\mathsf{est}}$ into a convex combination of *product distributions* and then exploit the simplicity of product distributions to prove the lower bound.

Nevertheless, the distribution $\mathcal{D}_{\mathsf{est}}$ is still "adversarial" and hence is not suitable for proving the lower bound for random arrival streams. Therefore, we define an extension to the original hard distribution as $\mathcal{D}_{\mathsf{ext}}$ which *randomly* partitions the sets of distribution $\mathcal{D}_{\mathsf{est}}$ between Alice and Bob. We prove a lower bound for this distribution using a reduction from protocols over $\mathcal{D}_{\mathsf{est}}$. Finally, we show how an algorithm for set cover over random arrival streams would be able to solve instances of $\mathsf{SetCover}_{\mathsf{est}}$ over $\mathcal{D}_{\mathsf{ext}}$ and establish Theorem 4.

5.1 A Hard Input Distribution for $\mathsf{SetCover}_{\mathsf{est}}$

Consider the following distribution $\mathcal{D}_{\mathsf{est}}$ for $\mathsf{SetCover}_{\mathsf{est}}$.

Distribution $\mathcal{D}_{\mathsf{est}}$. A hard input distribution for $\mathsf{SetCover}_{\mathsf{est}}$.

Notation. Let $\mathcal{F}$ be the collection of all subsets of $[n]$ with cardinality $\frac{n}{10\alpha}$.

- **Alice.** The input of Alice is a collection of m sets $\mathcal{S} = (S_1, \ldots, S_m)$, where for any $i \in [m]$, S_i is a set chosen independently and uniformly at random from $\mathcal{F}$.
- **Bob.** Pick $\theta \in \{0, 1\}$ and $i^* \in [m]$ independently and uniformly at random; the input of Bob is a single set T defined as follows.
  - If $\theta = 0$, then $\overline{T}$ is a set of size $\alpha \log m$ chosen uniformly at random from all subsets of S_{i^*} with size $\alpha \log m$.[a]
  - If $\theta = 1$, then $\overline{T}$ is a set of size $\alpha \log m$ chosen uniformly at random from all subsets of $[n] \setminus S_{i^*}$ with size $\alpha \log m$.

[a] Since $\alpha = o(\sqrt{n/\log n})$ and $m = \text{poly}(n)$, the size of $\overline{T}$ is strictly smaller than the size of S_{i^*}.

Recall that $opt(\mathcal{S}, T)$ denotes the *set cover size* of the input instance $(\mathcal{S}, T)$. The following lemma can be proven analogous to Lemma 3.2 and its proof is deferred to the full version of the paper [2].

LEMMA 5.1. *For $(\mathcal{S}, T) \sim \mathcal{D}_{est}$:*

(i) $\Pr(opt(\mathcal{S}, T) = 2 \mid \theta = 0) = 1$.

(ii) $\Pr(opt(\mathcal{S}, T) > 2\alpha \mid \theta = 1) = 1 - o(1)$.

Furthermore, using similar techniques as in the proof of Theorem 2, we can prove the following theorem. The proof is deferred to the full version of the paper [2].

THEOREM 5. *For any constant $\delta < 1/2$, $\alpha = o(\sqrt{\frac{n}{\log n}})$, and $m =$ poly(n),*

$$\mathsf{CC}^{\delta}_{\mathcal{D}_{est}}(\mathsf{SetCover}_{\mathsf{est}}) = \widetilde{\Omega}(mn/\alpha^2).$$

As a corollary of this result, we have that the space complexity of single-pass streaming algorithms for the set cover problem on *adversarial streams* is $\widetilde{\Omega}(mn/\alpha^2)$.

[6] The actual set given to Bob is the complement of this block; hence the optimal set cover size varies significantly between the two cases.

5.2 Extension to Random Arrival Streams

We now show that the lower bound established in Theorem 5 can be further strengthened to prove a lower bound on the space complexity of single-pass streaming algorithms in the random arrival model. To do so, we first define an extension of the distribution $\mathcal{D}_{\mathsf{est}}$, denoted by $\mathcal{D}_{\mathsf{ext}}$, prove a lower bound for $\mathcal{D}_{\mathsf{ext}}$, and then show that how to use this lower bound on the one-way communication complexity to establish a lower bound for the random arrival model.

We define the distribution $\mathcal{D}_{\mathsf{ext}}$ as follows.

Distribution $\mathcal{D}_{\mathsf{ext}}$. An extension of the hard distribution $\mathcal{D}_{\mathsf{est}}$ for $\mathsf{SetCover}_{\mathsf{est}}$.

1. Sample the sets $\mathcal{S} = \{S_1, \ldots, S_m, T\}$ in the same way as in the distribution $\mathcal{D}_{\mathsf{est}}$.
2. Assign each set in $\mathcal{S}$ to Alice with probability $1/2$, and the remaining sets are assigned to Bob.

We prove that the distribution $\mathcal{D}_{\mathsf{ext}}$ is still a hard distribution for $\mathsf{SetCover}_{\mathsf{est}}$.

Lemma 5.2. *For any constant $\delta < 1/8$, $\alpha = o(\sqrt{\frac{n}{\log n}})$, and $m = poly(n)$,*

$$\mathsf{CC}^{\delta}_{\mathcal{D}_{ext}}(\mathsf{SetCover}_{\mathsf{est}}) = \widetilde{\Omega}(mn/\alpha^2).$$

Proof. We design a reduction from $\mathsf{SetCover}_{\mathsf{est}}$ over the distribution $\mathcal{D}_{\mathsf{est}}$ to prove this lemma. Let Π_{Ext} be a δ-error protocol over the distribution $\mathcal{D}_{\mathsf{ext}}$. Let $\delta' = 3/8 + \delta$; in the following, we create a δ'-error protocol Π_{SC} for the distribution $\mathcal{D}_{\mathsf{est}}$ (using Π_{Ext} as a subroutine).

Consider an instance of the problem from the distribution $\mathcal{D}_{\mathsf{est}}$. Define a mapping $\sigma : [m+1] \mapsto \mathcal{S}$ such that for $i \leq m$, $\sigma(i) = S_i$ and $\sigma(m+1) = T$. Alice and Bob use public randomness to partition the set of integers $[m+1]$ between each other, assigning each number in $[m+1]$ to Alice (resp. to Bob) with probability $1/2$. As shown in the full version of the paper [2], in Theorem 5, we may assume that Bob additionally knows the special index i^*.

Consider the random partitioning of $[m+1]$ done by the players. If $i^* = \sigma^{-1}(S_{i^*})$ is assigned to Bob, or $m+1 = \sigma^{-1}(T)$ is assigned to Alice, Bob always outputs 2. Otherwise, Bob samples one set from $\mathcal{F}$ for each j assigned to him independently and uniformly at random and treat these sets plus the set T as his "new input". Moreover, Alice discards the sets $S_j = \sigma(j)$, where j is assigned to Bob and similarly treat the remaining set as her new input. The players now run the protocol Π_{Ext} over this distribution and Bob outputs the estimate returned by Π_{Ext} as his estimate of the set cover size.

Let $\boldsymbol{R}$ denote the randomness of the reduction (*excluding* the inner randomness of Π_{Ext}). Define $\mathcal{E}$ as the event that in the described reduction, i^* is assigned to Alice and $m+1$ is assigned to Bob. Let $\mathcal{D}_{\mathsf{new}}$ be the distribution of the instances over the *new inputs* of Alice and Bob (i.e., the input in which Alice drops the sets assigned to Bob, and Bob randomly generates the sets assigned to Alice) when $\mathcal{E}$ happens. Similarly, we define $\widehat{\mathcal{E}}$ to be the event that in the distribution $\mathcal{D}_{\mathsf{ext}}$, S_{i^*} is assigned to Alice and T is assigned to Bob. It is straightforward to verify that $\mathcal{D}_{\mathsf{new}} = (\mathcal{D}_{\mathsf{ext}} \mid \widehat{\mathcal{E}})$.

We now have,

$$\begin{aligned}
\Pr_{\mathcal{D}_{\mathsf{est}},\boldsymbol{R}}\left(\Pi_{\mathsf{SC}} \text{ errs}\right) &= \frac{1}{2} \cdot \Pr_{\boldsymbol{R}}\left(\overline{\mathcal{E}}\right) + \Pr_{\boldsymbol{R}}\left(\mathcal{E}\right) \cdot \Pr_{\mathcal{D}_{\mathsf{new}}}\left(\Pi_{\mathsf{Ext}} \text{ errs}\right) \\
&\qquad\qquad (\Pr_{\mathcal{D}_{\mathsf{est}},\boldsymbol{R}}\left(\Pi_{\mathsf{SC}} \text{ errs} \mid \overline{\mathcal{E}}\right) = 1/2) \\
&= \frac{1}{2} \cdot \Pr_{\boldsymbol{R}}\left(\overline{\mathcal{E}}\right) + \Pr_{\boldsymbol{R}}\left(\mathcal{E}\right) \cdot \Pr_{\mathcal{D}_{\mathsf{ext}}}\left(\Pi_{\mathsf{Ext}} \text{ errs} \mid \widehat{\mathcal{E}}\right) \\
&\qquad\qquad (\mathcal{D}_{\mathsf{new}} = (\mathcal{D}_{\mathsf{ext}} \mid \widehat{\mathcal{E}})) \\
&\leq \frac{1}{2} \cdot \Pr_{\boldsymbol{R}}\left(\overline{\mathcal{E}}\right) + \Pr_{\boldsymbol{R}}\left(\mathcal{E}\right) \cdot \frac{\Pr_{\mathcal{D}_{\mathsf{ext}}}\left(\Pi_{\mathsf{Ext}} \text{ errs}\right)}{\Pr_{\mathcal{D}_{\mathsf{ext}}}\left(\widehat{\mathcal{E}}\right)} \\
&= \frac{1}{2} \cdot \Pr_{\boldsymbol{R}}\left(\overline{\mathcal{E}}\right) + \Pr_{\mathcal{D}_{\mathsf{ext}}}\left(\Pi_{\mathsf{Ext}} \text{ errs}\right) \\
&\qquad\qquad (\Pr_{\boldsymbol{R}}(\mathcal{E}) = \Pr_{\mathcal{D}_{\mathsf{ext}}}(\widehat{\mathcal{E}})) \\
&\leq \frac{3}{8} + \delta \qquad\qquad (\Pr_{\boldsymbol{R}}(\overline{\mathcal{E}}) = 3/4)
\end{aligned}$$

Finally, since $\delta < 1/8$, we obtain a $(1/2 - \epsilon)$-error protocol (for some constant ϵ bounded away from 0) for the distribution $\mathcal{D}_{\mathsf{est}}$. The lower bound now follows from Theorem 5. □

We can now prove the lower bound for the random arrival model.

Proof of Theorem 4. Suppose $\mathcal{A}$ is a single-pass algorithm satisfying the conditions in the theorem statement. We use $\mathcal{A}$ to create a δ-error protocol $\mathsf{SetCover}_{\mathsf{est}}$ over the distribution $\mathcal{D}_{\mathsf{ext}}$ with parameter $\delta = 0.1 < 1/8$.

Consider any input $\mathcal{S}$ in the distribution $\mathcal{D}_{\mathsf{ext}}$ and denote the sets given to Alice by $\mathcal{S}_A$ and the sets given to Bob by $\mathcal{S}_B$. Alice creates a stream created by a random permutation of $\mathcal{S}_A$ denoted by s_A, and Bob does the same for $\mathcal{S}_B$ and obtains s_B. The players can now compute $\mathcal{A}\left(\langle s_A, s_B \rangle\right)$ to estimate the set cover size and the communication complexity of this protocol equals the space complexity of $\mathcal{A}$. Moreover, partitioning made in the distribution $\mathcal{D}_{\mathsf{ext}}$ together with the choice of random permutations made by the players, ensures that $\langle s_A, s_B \rangle$ is a random permutation of the original set $\mathcal{S}$. Hence, the probability that $\mathcal{A}$ fails to output an α-estimate of the set cover problem is at most $\delta = 0.1$. The lower bound now follows from Lemma 5.2. □

Acknowledgements

We thank Piotr Indyk for introducing us to the problem of determining single-pass streaming complexity of set cover, and the organizers of the DIMACS Workshop on Big Data through the Lens of Sublinear Algorithms (August 2015) where this conversation happened. We are also thankful to anonymous reviewers for many valuable comments.

6. REFERENCES

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29. ACM, 1996.

[2] S. Assadi, S. Khanna, and Y. Li. Tight bounds for single-pass streaming complexity of the set cover problem. *CoRR*, abs/1603.05715, 2016.

[3] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *43rd Symposium on Foundations of Computer Science*

(FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings, pages 209–218, 2002.

[4] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002*, pages 93–102, 2002.

[5] B. Barak, M. Braverman, X. Chen, and A. Rao. How to compress interactive communication. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 67–76, 2010.

[6] A. Chakrabarti, G. Cormode, and A. McGregor. Robust lower bounds for communication and stream computation. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 641–650, 2008.

[7] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 270–278, 2001.

[8] A. Chakrabarti and T. Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. *CoRR, abs/1507.04645. To appear in Symposium on Discrete Algorithms (SODA)*, 2016.

[9] 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.

[10] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[11] 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.

[12] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014.

[13] 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.

[14] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[15] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005.

[16] S. Har-Peled, P. Indyk, S. Mahabadi, and A. Vakilian. Towards tight bounds for the streaming set cover problem. *To appear in PODS*, 2016.

[17] R. Impagliazzo and V. Kabanets. Constructive proofs of concentration bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, pages 617–631, 2010.

[18] P. Indyk, S. Mahabadi, and A. Vakilian. Towards tight bounds for the streaming set cover problem. *CoRR*, abs/1509.00118, 2015.

[19] T. S. Jayram and D. P. Woodruff. Optimal bounds for johnson-lindenstrauss transforms and streaming problems with sub-constant error. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1–10, 2011.

[20] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9(3):256–278, 1974.

[21] M. Kapralov, S. Khanna, and M. Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751, 2014.

[22] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.

[23] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[24] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005.

[25] N. Nisan. The communication complexity of approximate set packing and covering. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, pages 868–875, 2002.

[26] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997.

[27] M. Saglam. *Tight bounds for data stream algorithms and communication problems*. PhD thesis, Simon Fraser University, 2011.

[28] 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, April 30 - May 2, 2009, Sparks, Nevada, USA*, pages 697–708, 2009.

[29] P. Slavík. A tight analysis of the greedy algorithm for set cover. *J. Algorithms*, 25(2):237–254, 1997.