

Tight Bounds for Single-Pass Streaming Complexity of the Set Cover Problem

Sepehr Assadi*

Sanjeev Khanna*

Yang Li*

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 any $\alpha = o(\sqrt{n})$) using 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.* [17] 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*.

*Department of Computer and Information Science, University of Pennsylvania. Supported in part by National Science Foundation grants CCF-1116961, CCF-1552909, and IIS-1447470 and an Adobe research award.
Email: {sassadi,sanjeev,yangli2}@cis.upenn.edu.

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 $[n]$, $\mathcal{S} = \langle S_1, \dots, S_m \rangle$, 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 $\text{opt} := \text{opt}(\mathcal{S})$.

The set cover problem can be formulated in the well-established streaming model [1, 23], 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 [7, 10, 12, 17, 27]. 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 $\alpha \cdot \text{opt}$ 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., [7, 12]).

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 $[\text{opt}, \alpha \cdot \text{opt}]$. If \mathcal{A} is a randomized algorithm, we require that:*

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

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 $\tilde{O}(mn/\alpha)$ bits and moreover, any single-pass streaming algorithm (possibly *randomized*) that α -approximates the set cover problem must use $\Omega(mn/\alpha)$ bits of space.
- There is a *randomized* single-pass streaming algorithm that α -estimates the set cover problem using space $\tilde{O}(mn/\alpha^2)$ bits and moreover, any single-pass streaming algorithm (possibly *randomized*) that α -estimates the set cover problem must use $\tilde{\Omega}(mn/\alpha^2)$ 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 requirement².

¹Throughout, we use *w.p.* and *w.h.p.* to abbreviate “with probability” and “with high probability”, respectively.

²Set cover admits a classic $\log n$ -approximation algorithm [19, 28], and unless $P = NP$, there is no polynomial time α -approximation for the set cover problem for $\alpha < (1 - \epsilon) \log n$ (for any constant $\epsilon > 0$) [11, 13, 22].

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 $\tilde{O}(mn/\alpha)$ 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 $\tilde{\Omega}(mn/\alpha)$ 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 [17] (see also [15]), $o(\sqrt{n})$ -approximation in $o(m)$ space [7, 12], and $O(1)$ -approximation in $o(mn)$ space [10] (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 $\tilde{O}(m)$ 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.* [17] in the strongest sense possible.

We should 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 $\tilde{O}(n)$ space (essentially independent of m for $m = \text{poly}(n)$) [7, 12].

Estimating Set Cover Size. We present an $\tilde{O}(mn/\alpha^2)$ 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 the space complexity of our $\tilde{O}(mn/\alpha^2)$ 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 $\tilde{\Omega}(mn/\alpha^2)$ 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., [20]). However, we show that this is *not* the case for the set cover problem, i.e., the lower bound of $\tilde{\Omega}(mn/\alpha^2)$ 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 and α -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 [7, 10, 24]; (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 Theorem 6).

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

1.2 Our Techniques

Upper bounds. An α -approximation using $\tilde{O}(mn/\alpha)$ 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 $\tilde{O}(mn/\alpha)$ bits.

Our $\tilde{O}(mn/\alpha^2)$ -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 [10] and [17] 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 entirely different from the ones in [10,17]. The *element sampling* approach of [10,17] 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}_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 $\tilde{O}(mn/\alpha^2)$ bound (see Theorem 3). Hence, we combine it with a *pruning* step, similar to the “set filtering” step of [17] for (unweighted) set cover (see also “GreedyPass” algorithm of [7]) 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 [5] 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 [5].

1.3 Related Work

Communication complexity of the set cover problem was first studied by Nisan [24]. 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 [27] initiated the study of set cover in the *semi-streaming* model [14] where the sets are arriving in a stream and the algorithms are required to use $\tilde{O}(n)$ 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 [8] in the context of “disk friendly” algorithms. As designed, the algorithm of [8] 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 thoroughly studied by Emek and Rosén [12]. They provided an $O(\sqrt{n})$ -approximation using $\tilde{O}(n)$ space (which 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 $\tilde{O}(n)$ space) that $O(n^{1/2-\epsilon})$ -estimates set cover exists. Recently, Chakrabarti and Wirth [7] generalized the space bounds in [12] to *multi-pass* algorithms, providing an almost complete understanding of the pass/approximation tradeoff for semi-streaming algorithms. In particular, they developed a deterministic p -pass $(p + 1) \cdot n^{1/(p+1)}$ -approximation algorithm in $\tilde{O}(n)$ 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.* [10] 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 $\tilde{O}(mn^\delta)$ 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.* [17] (see also [15]) made a significant improvement on the trade-off achieved by [10]: they presented an algorithm that for any $\delta > 0$, makes $O(1/\delta)$ passes over the stream and achieves an $O(\rho/\delta)$ -approximation using $\tilde{O}(mn^\delta)$ 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 $\tilde{\Omega}(mn^\delta)$ 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 w.h.p., 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 $\tilde{O}(mn/\alpha^2)$ upper bound on the 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.

Finally, Section 6 contains our $\Omega(mn/\alpha)$ space lower bound for *deterministic* α -estimation algorithms.

2 Preliminaries

Notation. We use bold face letters to represent random variables. For any random variable X , $\text{SUPP}(X)$ denotes its support set. We define $|X| := \log |\text{SUPP}(X)|$. For any k -dimensional tuple $X = (X_1, \dots, X_k)$ and any $i \in [k]$, we define $X^{<i} := (X_1, \dots, X_{i-1})$, and $X^{-i} := (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_k)$. 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. We use an extension of the Chernoff-Hoeffding bound for *negatively correlated* random variables. Random variables X_1, \dots, X_n are negatively correlated if for every set $S \subseteq [n]$, $\Pr(\bigwedge_{i \in S} X_i = 1) \leq \prod_{i \in S} \Pr(X_i = 1)$. It was first proved in [25] that the Chernoff-Hoeffding bound continues to hold for the case of random variables that satisfy this generalized version of negative correlation (see also [16]).

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 [9].

In the following, we denote the *Shannon Entropy* of a random variable A by $H(A)$ and the *mutual information* of two random variables A and B by $I(A; B) = H(A) - H(A | B) = H(B) - H(B | A)$. If the distribution \mathcal{D} of the random variables is not clear from the context, we use $H_{\mathcal{D}}(A)$ (resp. $I_{\mathcal{D}}(A; B)$). We use H_2 to denote the binary entropy function where for any real number $0 < \delta < 1$, $H_2(\delta) = \delta \log \frac{1}{\delta} + (1 - \delta) \log \frac{1}{1-\delta}$.

We use the following basic properties of entropy and mutual information (proofs can be found in [9], Chapter 2).

Claim 2.1. Let A, B , and C be three random variables.

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

The following claim (Fano’s inequality) states that if a random variable A can be used to estimate the value of another random variable B , then A should “consume” most of the entropy of B .

Claim 2.2 (Fano’s inequality). For any binary random variable B and any (possibly randomized) function f that predicts B based on A , if $\Pr(f(A) \neq B) = \delta$, then $H(B | A) \leq H_2(\delta)$.

We also use the following simple claim, which states that conditioning on independent random variables can only increase the mutual information.

Claim 2.3. *For any random variables A, B, C , and D , if A and D are independent conditioned on C , then $I(A; B \mid C) \leq I(A; B \mid C, D)$.*

Proof. Since A and D are independent conditioned on C , by Claim 2.1-(3), $H(A \mid C) = H(A \mid C, D)$ and $H(A \mid C, B) \geq H(A \mid C, B, D)$. We have,

$$\begin{aligned} I(A; B \mid C) &= H(A \mid C) - H(A \mid C, B) = H(A \mid C, D) - H(A \mid C, B) \\ &\leq H(A \mid C, D) - H(A \mid C, B, D) = I(A; B \mid C, D) \end{aligned}$$

■

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. Our lower bounds for single-pass streaming algorithms are established 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 [21].

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 $CC_{\mathcal{D}}^{\delta}(P)$ 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., [2–4, 6]). Our definition is tuned specifically for *one-way protocols*, similar in the spirit of [3] (see also [18]).

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

The information complexity $IC_{\mathcal{D}}^{\delta}(P)$ of P with respect to a distribution \mathcal{D} is the minimum $\text{ICost}_{\mathcal{D}}(\Pi)$ 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 $R = R$ and then running the corresponding private coin protocol Π^R . We also use Π^R to denote the random variable of the message sent from Alice to Bob, assuming that the public randomness is $R = R$. We have the following well-known claim.

Claim 2.4. *For any distribution \mathcal{D} and any protocol Π , let R denote the public randomness used in Π ; then, $\text{ICost}_{\mathcal{D}}(\Pi) = \mathbb{E}_{R \sim R} [I_{\mathcal{D}}(\Pi^R; X \mid R = R)]$.*

Proof. Let $\Pi = (M, R)$, where M denotes the message sent by Alice and R is the public randomness. We have,

$$\begin{aligned} \text{ICost}_{\mathcal{D}}(\Pi) &= I(\Pi; X) = I(M, R; X) = I(R; X) + I(M; X \mid R) \\ &\quad \text{(the chain rule for mutual information, Claim 2.1-(5))} \\ &= \mathbb{E}_{R \sim R} [I_{\mathcal{D}}(\Pi^R; X \mid R = R)] \\ &\quad (M = \Pi^R \text{ whenever } R = R \text{ and } I(R; X) = 0 \text{ by Claim 2.1-(2)}) \end{aligned}$$

The following well-known proposition (see, e.g., [6]) relates communication complexity and information complexity. ■

Proposition 2.5. *For every $0 < \delta < 1$ and every distribution \mathcal{D} : $\text{CC}_{\mathcal{D}}^{\delta}(P) \geq \text{IC}_{\mathcal{D}}^{\delta}(P)$.*

Proof. Let Π be a protocol with the minimum communication complexity for P on \mathcal{D} and R denotes the public randomness of Π ; using Claim 2.4, we can write,

$$\begin{aligned} \text{IC}_{\mathcal{D}}^{\delta}(P) &= \mathbb{E}_{R \sim R} [I_{\mathcal{D}}(\Pi^R; X \mid R = R)] \leq \mathbb{E}_{R \sim R} [H_{\mathcal{D}}(\Pi^R \mid R = R)] \\ &\leq \mathbb{E}_{R \sim R} [\|\Pi^R\|] \leq \|\Pi\| = \text{CC}_{\mathcal{D}}^{\delta}(P) \end{aligned}$$
■

3 An $\Omega(mn/\alpha)$ -Space Lower bound for α -Approximate 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 $m = \text{poly}(n)$, any randomized single-pass streaming algorithm 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, $\text{SetCover}_{\text{apx}}$ refers to the problem of α -approximating the set cover problem for instances with $m + 1$ sets⁴ defined over the universe $[n]$ in the one-way communication model, whereby the sets are partitioned between Alice and Bob.

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

Overview. We design a hard input distribution \mathcal{D}_{apx} for $\text{SetCover}_{\text{apx}}$, whereby Alice is provided with a collection of m sets S_1, \dots, 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 \bar{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 \bar{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 \bar{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 \bar{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 $\text{SetCover}_{\text{apx}}$

Consider the following distribution.

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

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 $\mathcal{S} = (S_1, \dots, S_m)$, 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

^aSince $\alpha = o(\sqrt{n}/\log n)$ 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}_{apx} .

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

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

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

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 X_e , where $X_e = 1$ iff $e \in S$. Let $X := \sum_e X_e$, which is the number of elements in E covered by S . We have,

$$\mathbb{E}[X] = \sum_e \mathbb{E}[X_e] = \frac{|E|}{10\alpha} = \frac{\log m}{5}$$

Moreover, the variables X_e are negatively correlated since for any set $S' \subseteq E$,

$$\begin{aligned} \Pr\left(\bigwedge_{e \in S'} X_e = 1\right) &= \frac{\binom{\frac{n}{10\alpha} - |S'|}{\frac{n}{10\alpha}}}{\binom{\frac{n}{10\alpha}}{\frac{n}{10\alpha}}} = \frac{\left(\frac{n}{10\alpha}\right) \cdot \left(\frac{n}{10\alpha} - 1\right) \cdots \left(\frac{n}{10\alpha} - |S'| + 1\right)}{(n) \cdot (n-1) \cdots (n - |S'| + 1)} \\ &\leq \left(\frac{1}{10\alpha}\right)^{|S'|} = \prod_{e \in S'} \Pr(X_e = 1) \end{aligned}$$

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

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

Therefore, using 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 E . \blacksquare

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

In order to prove our lower bound for $\text{SetCover}_{\text{apx}}$ on \mathcal{D}_{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}_{apx}). We define the following distribution $\mathcal{D}_{\text{Trap}}$ 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}_{apx} and $\mathcal{D}_{\text{Trap}}$, information complexity of solving $\text{SetCover}_{\text{apx}}$ is essentially equivalent to solving m copies of *Trap*. Formally,

Lemma 3.3. For any constant $\delta < 1/2$, $\text{IC}_{\mathcal{D}_{\text{apx}}}^\delta(\text{SetCover}_{\text{apx}}) \geq m \cdot \text{IC}_{\mathcal{D}_{\text{Trap}}}^{\delta+o(1)}(\text{Trap})$.

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

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

Input: An instance $(S, E) \sim \mathcal{D}_{\text{Trap}}$. **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 $\mathcal{S} = (S_1, \dots, S_m)$ 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 $\text{SetCover}_{\text{apx}}$ constructed in the reduction is exactly \mathcal{D}_{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 $\hat{\mathcal{S}}$ be the set cover computed by Bob minus the sets S_{i^*} and T . As $e^* \in E = \bar{T}$ and moreover is *not* in S_{i^*} , it follows that e^* should be covered by some set in $\hat{\mathcal{S}}$. 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 $\hat{\mathcal{S}}$ is at most $\ell/2$ w.p. $1 - o(1)$. Hence, $|L| \leq \ell/2 = |E|/2$. This implies that:

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

We now bound the information cost of Π_{Trap} . Let I be the random variable for the choice of $i^* \in [m]$ in the protocol Π_{Trap} (which is uniform in $[m]$). Using Claim 2.4, we have,

$$\begin{aligned} \text{ICost}_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}) &= \mathbb{E}_{i \sim I} \left[I_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}^i; \mathcal{S} \mid I = i) \right] = \frac{1}{m} \cdot \sum_{i=1}^m I_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{SC}}; S_i \mid I = i) \\ &= \frac{1}{m} \cdot \sum_{i=1}^m I_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}}; S_i \mid I = i) = \frac{1}{m} \cdot \sum_{i=1}^m I_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}}; S_i) \end{aligned}$$

where the last two equalities hold since (i) the joint distribution of Π_{SC} and S_i conditioned on $I = i$ under $\mathcal{D}_{\text{Trap}}$ is equivalent to the one under \mathcal{D}_{apx} , and (ii) the random variables Π_{SC} and S_i are independent of the event $I = i$ (by the definition of \mathcal{D}_{apx}) and hence we can “drop” the conditioning on this event (by Claim 2.1-(7)).

We can further derive,

$$\text{ICost}_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}) = \frac{1}{m} \cdot \sum_{i=1}^m I_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}}; S_i) \leq \frac{1}{m} \cdot \sum_{i=1}^m I_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}}; S_i \mid \mathcal{S}^{<i})$$

The inequality holds since S_i and $\mathcal{S}^{<i}$ are independent and conditioning on independent variables can only increase the mutual information (i.e., Claim 2.3). Finally,

$$\text{ICost}_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}) \leq \frac{1}{m} \cdot \sum_{i=1}^m I_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}}; S_i \mid \mathcal{S}^{<i}) = \frac{1}{m} \cdot I_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}}; \mathcal{S}) = \frac{1}{m} \cdot \text{ICost}_{\mathcal{D}_{\text{apx}}}(\Pi_{\text{SC}})$$

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 $\mathcal{D}_{\text{Trap}}$. We prove this lower bound using a novel reduction from the well-known *Index* problem, denoted by Index_k^n . In Index_k^n over the distribution $\mathcal{D}_{\text{Index}}$, 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_k^n have been previously studied in the literature (see, e.g., [26], Section 3.3). For the sake of completeness, we provide a self-contained proof of the following lemma in Appendix A.

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

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

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

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

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

Input: An instance $(A, a) \sim \mathcal{D}_{\text{Index}}$. **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 $\Pi_{\text{Trap}}(S)$, 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 $L := \Pi_{\text{Trap}}(S, E)$ 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 $\mathcal{D}_{\text{Trap}}$. 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_k^n over $\mathcal{D}_{\text{Index}}$ for the parameter $k = n/10\alpha$ and a constant $\delta' < 1/2$.

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

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

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 $\mathcal{D}_{\text{Index}}$ and the private coins 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 $(A, a) \sim \mathcal{D}_{\text{Index}}^N$ and the randomness of R , is the same as the distribution $\mathcal{D}_{\text{Trap}}$. 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,

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

We now bound the first term in Equation (1). Note that when $(A, a) \sim \mathcal{D}_{\text{Index}}^Y$, 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 $(A, a) \sim \mathcal{D}_{\text{Index}}^Y$ (and randomness of R) does *not* match the distribution $\mathcal{D}_{\text{Trap}}$. However, we have the following

important property: Given that (S, E) is the instance of Trap created by choosing (A, a) from $\mathcal{D}_{\text{Index}}^Y$ and sampling $\ell - 1$ random elements of A (using 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 $\mathcal{D}_{\text{Trap}}$ (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, (here, R^{Trap} denotes the inner randomness of Π_{Trap})

$$\begin{aligned} \Pr_{\mathcal{D}_{\text{Index}}^Y, R}(\Pi_{\text{Index}} \text{ errs} \mid \bar{\mathcal{E}}) &= \Pr_{\mathcal{D}_{\text{Index}}^Y, R} \left(a \in L = \Pi_{\text{Trap}}(S, E) \mid \bar{\mathcal{E}} \right) \\ &= \mathbb{E}_{(S, E) \sim (S, E) \mid \bar{\mathcal{E}}} \mathbb{E}_{R^{\text{Trap}} \sim R^{\text{Trap}}} \left[\Pr_{\mathcal{D}_{\text{Index}}^Y, R} \left(a \in L \mid S = S, E = E, R^{\text{Trap}} = R^{\text{Trap}}, \bar{\mathcal{E}} \right) \right] \\ &\quad (L = \Pi_{\text{Trap}}(S, E) \text{ is a fixed set conditioned on } (S, E, R^{\text{Trap}})) \\ &= \mathbb{E}_{(S, E) \sim (S, E) \mid \bar{\mathcal{E}}} \mathbb{E}_{R^{\text{Trap}} \sim R^{\text{Trap}}} \left[\frac{|L|}{|E|} \right] \\ &\quad (a \text{ is uniform on } E \text{ conditioned on } (S, E, R^{\text{Trap}}) \text{ and } \bar{\mathcal{E}}) \end{aligned}$$

Hence, we have, $\Pr_{\mathcal{D}_{\text{Index}}^Y, R}(\Pi_{\text{Index}} \text{ errs} \mid \bar{\mathcal{E}}) \leq \frac{1}{2}$, 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,

$$\Pr_{\mathcal{D}_{\text{Index}}^Y, R}(\Pi_{\text{Index}} \text{ errs}) = \Pr_{\mathcal{D}_{\text{Index}}^Y, R}(\bar{\mathcal{E}}) \cdot \Pr_{\mathcal{D}_{\text{Index}}^Y, R}(\Pi_{\text{Index}} \text{ errs} \mid \bar{\mathcal{E}}) \leq \frac{1 - o(1)}{2} \quad (3)$$

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

$$\Pr_{\mathcal{D}_{\text{Index}}^Y, R}(\Pi_{\text{Index}} \text{ errs}) \leq \frac{1}{2} \cdot \frac{1 - o(1)}{2} + \frac{1}{2} \cdot \delta = \frac{1 - o(1)}{4} + \frac{\delta}{2} \leq \frac{1}{2} - \epsilon$$

for some constant ϵ bounded away from 0. ■

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

Claim 3.7. $\text{ICost}_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Index}}) \leq \text{ICost}_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}) + O(\ell \log n)$.

Proof. We have,

$$\begin{aligned} \text{ICost}_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Index}}) &= I_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Index}}(A); A) \\ &= I_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Trap}}(S), B; A) \\ &= I_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Trap}}(S); A) + I_{\mathcal{D}_{\text{Index}}}(B; A \mid \Pi_{\text{Trap}}(S)) \\ &\quad (\text{the chain rule for mutual information, Claim 2.1-(5)}) \\ &\leq I_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Trap}}(S); A) + H_{\mathcal{D}_{\text{Index}}}(B \mid \Pi_{\text{Trap}}(S)) \end{aligned}$$

$$\begin{aligned}
&\leq I_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Trap}}(S); A) + O(\ell \log n) \quad (|B| = O(\ell \log n) \text{ and Claim 2.1-(1)}) \\
&= I_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Trap}}(S); S) + O(\ell \log n) \quad (A = S \text{ as defined in } \Pi_{\text{Index}}) \\
&= I_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}(S); S) + O(\ell \log n) \\
&\quad (\text{the joint distribution of } (\Pi_{\text{Trap}}(S), S) \text{ is identical under } \mathcal{D}_{\text{Index}} \text{ and } \mathcal{D}_{\text{Trap}}) \\
&= \text{ICost}_{\mathcal{D}_{\text{Trap}}}(\Pi_{\text{Trap}}) + O(\ell \log n)
\end{aligned}$$

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 $m = \text{poly}(n)$, 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 $m = \text{poly}(n)$, ■

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

Since the information complexity is a lower bound on the communication complexity (Proposition 2.5), we have,

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

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

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 $\tilde{O}(mn/\alpha^2)$ -Space Upper Bound for α -Estimation

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 $\text{ILP}_{\text{Cover}}(A, b, c)$. 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 $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$ as $c \cdot x^*$ where x^* is the *optimal* solution to \mathcal{I} , and denote it by $\text{opt} := \text{opt}(\mathcal{I})$.

We consider the following streaming setting for covering ILPs. The input to a streaming algorithm for an instance $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$ 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 $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$ with $\text{poly}(n)$ -bounded entries, makes a single pass over a stream of columns of A (presented in an arbitrary order), and outputs an α -estimation to $\text{opt}(\mathcal{I})$ w.h.p. using space $\tilde{O}((mn/\alpha^2) \cdot b_{\max} + m + nb_{\max})$ bits.*

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

To prove Theorem 3, we design a general approach based on sampling constraints of a covering ILP instance. The goal is to show that if we sample (roughly) $1/\alpha$ fraction of the constraints from an instance $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$, then the optimum value of the resulting covering ILP, denoted by \mathcal{I}_R , is a good estimator of $\text{opt}(\mathcal{I})$. 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, $\text{opt}(\mathcal{I}_R) \ll \text{opt}(\mathcal{I})$ as long as e is not sampled in \mathcal{I}_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 $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$, we define the *cost* of the constraint j , denoted by $\text{Cost}(j)$, as,

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

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

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

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

Constraint Sampling. Given any instance of covering ILP $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$, let \mathcal{I}_R be a covering ILP instance $\text{ILP}_{\text{Cover}}(A, \tilde{b}, c)$ obtained by setting $\tilde{b}_j := b_j$ with probability p , and $\tilde{b}_j := 0$ with probability $1 - p$, for each dimension $j \in [n]$ of b independently. Note that setting

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

$\tilde{b}_j := 0$ in \mathcal{I}_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}_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 which asserts that $\text{opt}(\mathcal{I}_R)$ is a good estimator of $\text{opt}(\mathcal{I})$ (under certain conditions). As $\text{opt}(\mathcal{I}_R) \leq \text{opt}(\mathcal{I})$ trivially holds (removing constraints can only decrease the optimal value), it suffices to give a lower bound on $\text{opt}(\mathcal{I}_R)$.

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*

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

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 $1/4$, $\text{opt}(\mathcal{I}_R) + \text{Cost}(\mathcal{I}) < \frac{\text{opt}(\mathcal{I})}{8\alpha}$ (we denote this event by $\mathcal{E}_1(\mathcal{I}_R)$, or shortly \mathcal{E}_1). We will show that in this case, \mathcal{I} has a feasible solution with a value smaller than $\text{opt}(\mathcal{I})$. To continue, define $\mathcal{E}_2(\mathcal{I}_R)$ (or \mathcal{E}_2 in short) as the event that $\text{opt}(\mathcal{I}_R) < \frac{\text{opt}(\mathcal{I})}{8\alpha}$. 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 $S := \{\mathcal{I}_R^1, \dots, \mathcal{I}_R^{32\alpha}\}$. Since \mathcal{E}_2 happens with probability at least $1/4$ on each instance \mathcal{I}_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 sampled 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 instances admits a solution of value at most $\frac{\text{opt}(\mathcal{I})}{8\alpha}$ (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 $4\alpha \cdot \frac{\text{opt}(\mathcal{I})}{8\alpha} = \frac{\text{opt}(\mathcal{I})}{2}$; a contradiction.

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

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

Before proving Claim 4.2, we show how this claim would imply the lemma. By Claim 4.2, for each of the 4α instances $\mathcal{I}_R \in T$, and for any $j \in [n]$, the probability that the constraint j is sampled in \mathcal{I}_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,

$$\Pr(j \in \mathcal{I}_R \mid \mathcal{E}_2(\mathcal{I}_R)) = \frac{\Pr(\mathcal{E}_2(\mathcal{I}_R) \mid j \in \mathcal{I}_R) \cdot \Pr(j \in \mathcal{I}_R)}{\Pr(\mathcal{E}_2(\mathcal{I}_R))}$$

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

$$\Pr(j \in \mathcal{I}_R \mid \mathcal{E}_2(\mathcal{I}_R)) \geq \Pr(\mathcal{E}_2(\mathcal{I}_R) \mid j \in \mathcal{I}_R) \cdot \frac{4 \ln n}{\alpha} \quad (4)$$

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

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}'_R be an instance created by this process. We prove $\Pr(\mathcal{E}_2(\mathcal{I}_R) \mid j \in \mathcal{I}_R) \geq 1/8$ in two steps by first showing that the probability that \mathcal{E}_1 happens to \mathcal{I}'_R (i.e., $\Pr(\mathcal{E}_1(\mathcal{I}'_R))$) is at least $1/8$, and then use a coupling argument to prove that $\Pr(\mathcal{E}_2(\mathcal{I}_R) \mid j \in \mathcal{I}_R) \geq \Pr(\mathcal{E}_1(\mathcal{I}'_R))$.

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

$$\begin{aligned} \Pr(\mathcal{E}_1(\mathcal{I}_R)) &= \Pr(j \in \mathcal{I}_R) \Pr(\mathcal{E}_1(\mathcal{I}_R) \mid j \in \mathcal{I}_R) + \Pr(j \notin \mathcal{I}_R) \Pr(\mathcal{E}_1(\mathcal{I}_R) \mid j \notin \mathcal{I}_R) \\ &\leq \Pr(j \in \mathcal{I}_R) + \Pr(\mathcal{E}_1(\mathcal{I}_R) \mid j \notin \mathcal{I}_R) = p + \Pr(\mathcal{E}_1(\mathcal{I}'_R)) \end{aligned}$$

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

$$1/4 \leq 1/8 + \Pr(\mathcal{E}_1(\mathcal{I}'_R))$$

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

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

$$\Pr(\mathcal{E}_2(\mathcal{I}_R) \mid j \in \mathcal{I}_R) \geq \Pr(\mathcal{E}_1(\mathcal{I}'_R)) \geq 1/8$$

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

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 Section 4.3, we describe how to eliminate the later two assumptions.

Algorithm overview. For any covering ILP instance $\mathcal{I} := \text{ILP}_{\text{Cover}}(A, b, c)$, our algorithm estimates $\text{opt} := \text{opt}(\mathcal{I})$ in two parts running in parallel. In the first part, the goal is simply to

compute $\text{Cost}(\mathcal{I})$ (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 $k \geq \text{opt}$, *Tester* accepts k w.p. 1 and for any k where $\text{Cost}(\mathcal{I}) \leq k \leq \frac{\text{opt}}{32\alpha}$, w.h.p. *Tester* rejects k .

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

We first show that one can compute the *Cost* of a covering ILP presented in a stream using a simple *dynamic programming* algorithm.

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

Proof. We maintain arrays $\text{Cost}_j[y] \leftarrow +\infty$ for any $j \in [n]$ and $y \in [b_{\max}]$: $\text{Cost}_j[y]$ is the minimum cost for achieving a coverage of y for the j -th constraint. Define $\text{Cost}_j[y] = 0$ for $y \leq 0$. Upon arrival of $\langle A_i, c_i \rangle$, compute $\text{Cost}_j[y] = \min(\text{Cost}_j[y], \text{Cost}_j[y - a_{j,i}] + c_i)$, for every y from b_j down to 1. We then have $\text{Cost}(\mathcal{I}) = \max_j \text{Cost}_j[b_j]$. ■

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

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

1. *Preprocessing:*

- (i) Maintain an n -dimensional vector $b_{\text{res}} \leftarrow b$, an m -dimensional vector $\tilde{c} \leftarrow 0^m$, and an $n \times m$ dimensional matrix $\tilde{A} \leftarrow 0^{n \times m}$.
- (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 $u_i := \min(b_{\text{res}}, A_i)$ (the coordinate-wise minimum). If $\|u_i\|_1 \geq \frac{n \cdot b_{\max}}{\alpha}$, update $b_{\text{res}} \leftarrow b_{\text{res}} - u_i$ (we say $\langle A_i, c_i \rangle$ is *pruned* by *Tester* in this case). Otherwise, assign $\tilde{A}_i \leftarrow u_i(V)$, and $\tilde{c}_i \leftarrow c_i$.

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

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

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

We first make the following observation. In the prune step of *Tester*, if we replace $\tilde{A}_i \leftarrow u_i(V)$ by $\tilde{A}_i \leftarrow A_i(V)$, the solution of the resulting covering ILP instance (denoted by \mathcal{I}'_{tester}) has the property that $\text{opt}(\mathcal{I}'_{tester}) = \text{opt}(\mathcal{I}_{tester})$ (we use \mathcal{I}_{tester} only to control the space requirement). To see this, let b_{res}^i denotes the content of the vector b_{res} when $\langle A_i, c_i \rangle$ arrives. By construction, $(u_i)_j := \min((b_{\text{res}}^i)_j, a_{j,i})$, and hence if $(u_i)_j \neq a_{j,i}$, then both $(u_i)_j$ and $a_{j,i}$ are at least $(b_{\text{res}}^i)_j$, which is at least $(b_{\text{res}})_j$ (since every dimension of b_{res} is monotonically decreasing). However, for any integer program $\text{ILP}_{\text{Cover}}(A, b, c)$, 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 $\text{opt}(\mathcal{I}'_{tester}) = \text{opt}(\mathcal{I}_{tester})$. To simplify the proof, in the following, when concerning $\text{opt}(\mathcal{I}_{tester})$, 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 $k \geq \text{opt}$, $\Pr(\text{Tester}(k) = \text{ACCEPT}) = 1$.*

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 \tilde{c} , we have $\tilde{c} \cdot x^* \leq c \cdot x^* \leq \text{opt}$, this will show that $\text{opt}(\mathcal{I}_{tester}) \leq \text{opt} \leq k$ and hence $\text{Tester}(k) = \text{ACCEPT}$.

Fix a constraint j in \mathcal{I}_{tester} as follows:

$$\sum_{i \in [m]} \tilde{a}_{j,i} x_i \geq b_{\text{res}}(V)_j$$

If $j \notin V$, $b_{\text{res}}(V)_j = 0$ and 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*, $b_{\text{res}}(V)_j = \max(b_j - \sum_{i \in P} a_{j,i}, 0)$. If $b_{\text{res}}(V)_j = 0$, again the constraint is trivially satisfied. Suppose $b_{\text{res}}(V)_j = b_j - \sum_{i \in P} a_{j,i}$. The constraint j can be written as

$$\sum_i \tilde{a}_{j,i} x_i \geq b_j - \sum_{i \in P} a_{j,i}$$

By construction of the tester, $\tilde{a}_{j,i} = 0$ for all i that are pruned and otherwise $\tilde{a}_{j,i} = a_{j,i}$. Hence, we can further write the constraint j as

$$\sum_{i \notin P} a_{j,i} x_i \geq b_j - \sum_{i \in P} a_{j,i}$$

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

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

and the constraint j is satisfied. Therefore, x^* is a feasible solution of \mathcal{I}_{tester} ; this completes the proof. \blacksquare

We now show that *Tester* will reject guesses that are smaller than $\frac{\text{opt}}{32\alpha}$. 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 $\text{Cost}(\mathcal{I}) \leq k < \frac{\text{opt}}{32\alpha}$, $\Pr(\text{Tester}(k) = \text{REJECT}) \geq 3/4$.*

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

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

where $\hat{A}_i = A_i$ if $\langle A_i, c_i \rangle$ is not pruned by Tester , and $\hat{A}_i = 0^n$ otherwise. In $\text{Tester}(k)$, 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 \tilde{A}_i in $\mathcal{I}_{\text{tester}}$). 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 $3/4$, $\text{opt}(\mathcal{I}_{\text{tester}}) + \text{Cost}(\mathcal{I}') \geq \frac{\text{opt}(\mathcal{I}')}{8\alpha}$. Since $\text{Cost}(\mathcal{I}') \leq \text{Cost}(\mathcal{I}) \leq k < \frac{\text{opt}(\mathcal{I})}{32\alpha}$, this implies that

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

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

To show that $\text{opt}(\mathcal{I}') \geq \frac{\text{opt}(\mathcal{I})}{2}$, 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 $\frac{\text{opt}}{2}$, $\text{opt}(\mathcal{I}')$ must be at least $\frac{\text{opt}}{2}$ or we will have a solution for \mathcal{I} better than $\text{opt}(\mathcal{I})$.

To see that the total weight of the pruned pairs is at most $\text{opt}/2$, since only pairs with $c_i \leq k$ ($\leq \frac{\text{opt}}{32\alpha}$) 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 $\frac{nb_{\text{max}}}{\alpha}$. Since b_{res} is initialized to be b and $\|b\|_1 \leq nb_{\text{max}}$, at most α ($\leq 16\alpha$) pairs can be pruned. This completes the proof. \blacksquare

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 $\text{Cost}(\mathcal{I})$, which require $\tilde{O}(nb_{\text{max}})$ space. We also need to run Tester for $O(\log(m \cdot c_{\text{max}}))$ different guesses of k .

In $\text{Tester}(k)$, we need $O(n \log b_{\text{max}})$ bits to store the vector b_{res} and $O(m \log c_{\text{max}})$ bits to maintain the vector \tilde{c} . Finally, the matrix \tilde{A} requires $O(mnb_{\text{max}}/\alpha \cdot (\log n/\alpha) \cdot (\log a_{\text{max}} \log n))$ bits to store. This is because each column \tilde{A}_i of \tilde{A} is either 0^n or $u_i(V)$ where $\|u_i\|_1 < \frac{n \cdot b_{\text{max}}}{\alpha}$. Since $\|u_i\|_1 < \frac{n \cdot b_{\text{max}}}{\alpha}$, there are at most $\frac{n \cdot b_{\text{max}}}{\alpha}$ non-zero entries in u_i . Therefore, after projecting u_i to V (to obtain \tilde{A}_i) in expectation the number of non-zero entries in \tilde{A}_i is at most $\tilde{O}(nb_{\text{max}}/\alpha^2)$. Using the Chernoff bound w.h.p at most $O(\frac{nb_{\text{max}}}{\alpha^2})$ non-zero entries of u_i remain in each \tilde{A}_i , where each entry needs $O(\log a_{\text{max}} \log n)$ 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 \tilde{A}_i has $(c \cdot \frac{nb_{\text{max}}}{\alpha^2})$ 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 $\text{poly}(n)$ -bounded, the total space requirement of the algorithm is $\tilde{O}((mn/\alpha^2) \cdot b_{\text{max}} + m + nb_{\text{max}})$. \blacksquare

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

Remark 4.6. The simple 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 $\tilde{O}(mnb_{\text{max}}/\alpha)$: Group the columns by the weights and merge every α sets for each group independently.

4.3 Further Remarks

We now briefly describe how to eliminate the assumptions that variables are binary and c_{\max} is given to the algorithm.

Binary variables versus integer variables. We point out that any algorithm that solves covering ILP with *binary* variables in the streaming setting, can also be used to solve covering ILP with *non-negative integer* variables (or shortly, integer variables), while increasing the number of columns by a factor of $O(\log b_{\max})$.

To see this, given any covering ILP instance $\text{ILP}_{\text{Cover}}(A, b, c)$ (with integer variables), we replace each $\langle A_i, c_i \rangle$ pair with $O(\log b_{\max})$ pairs (or columns) where the j -th pair ($j \in [\log b_{\max}]$) is defined to be $\langle 2^{j-1}A_i, 2^{j-1}c_i \rangle$. Every combination of the corresponding j binary variables maps to a unique binary representation of the variable x_i in $\text{ILP}_{\text{Cover}}(A, b, c)$ for $x_i = O(b_{\max})$. Since no variable in $\text{ILP}_{\text{Cover}}(A, b, c)$ needs to be more than b_{\max} , the correctness of this reduction follows.

Knowing c_{\max} versus not knowing c_{\max} . As we pointed out earlier, the assumption that the value of c_{\max} is provided to the algorithm is only for simplicity; here we briefly describe how to eliminate this assumption. Our algorithm uses c_{\max} to determine the range of opt , which determines the set of guesses K that *Tester* needs to run. If c_{\max} is not provided, one natural fix is to use the same randomness (i.e., the same set V) for all testers, and whenever a larger c_i arrives, create a tester for each new guess by duplicating the state of *Tester* for the current largest guess and continue from there (the correctness is provided by the design of our *Tester*).

However, if we use this approach, since we do not know the total number of guesses upfront, we cannot boost the probability of success to *ensure* that w.h.p. no tester fails (if c_{\max} is $\text{poly}(n)$ -bounded, the number of guesses is $O(\log(nc_{\max})) = O(\log n)$, $O(\log \log n)$ parallel copies suffice, and though we do not know the value of the constant, $\log n$ copies always suffice; but this will not be the case for general c_{\max}). To address this issue, we point out that it suffices to ensure that the copy of *Tester* that runs a specific guess k' succeed w.h.p., where k' is the largest power of 2 with $k' \leq \frac{\text{opt}}{32\alpha}$. To see this, we change the algorithm to pick the largest rejected guess k^* and return $64\alpha k^*$ (previously, it picks the smallest accepted guess k and returns $32\alpha k$), if k' is correctly rejected by the tester, $k^* \geq k' \geq \frac{\text{opt}}{64\alpha}$. On the other hand, for any guess no less than opt , *Tester* accepts it w.p. 1, and hence $k^* \leq \text{opt}$. Therefore, as long as *Tester* rejects k' correctly, the output of the algorithm is always an $O(\alpha)$ -estimation and hence we do not need the knowledge of c_{\max} upfront.

The set multi-cover problem. For the set multi-cover problem, Theorem 3 gives an α -estimation algorithm using space $\tilde{O}(\frac{mn b_{\max}}{\alpha^2} + m + nb_{\max})$; here, b_{\max} is the maximum demand of the elements. However, we remark that if sets are unweighted, a better space requirement of $\tilde{O}(\frac{mn}{\alpha^2} + m + n)$ is achievable for this problem: Instead of running the tester for multiple guesses, just run it once with the following changes. In the prune step, change from " $\|u_i\|_1 \geq \frac{nb_{\max}}{\alpha}$ " to " $\|u_i\|_1 \geq \frac{n}{\alpha}$ "; and output $\text{est} := \#_{\text{pruned}} + 8\alpha(\text{opt}(\mathcal{I}_{\text{tester}}) + b_{\max})$, where $\#_{\text{pruned}}$ is the number of sets that are pruned. One one hand, $\text{est} \geq \text{opt}$ w.h.p since $\#_{\text{pruned}}$ plus the optimal solution of the residual covering ILP (denoted by opt_{res}) is a feasible solution of \mathcal{I} , and by Lemma 4.1, w.h.p. $8\alpha(\text{opt}(\mathcal{I}_{\text{tester}}) + b_{\max}) \geq \text{opt}_{\text{res}}$. One the other hand, $\text{est} = O(\alpha \cdot \text{opt})$ since $\#_{\text{pruned}} \leq \alpha b_{\max} \leq \alpha \cdot \text{opt}$ (b_{\max} sets is needed even for covering one element); $\text{opt}(\mathcal{I}_{\text{tester}}) \leq \text{opt}$ and $b_{\max} \leq \text{opt}$, which implies $8\alpha(\text{opt}(\mathcal{I}_{\text{tester}}) + b_{\max}) \leq 8\alpha(\text{opt} + \text{opt}) = O(\alpha \cdot \text{opt})$.

5 An $\Omega(mn/\alpha^2)$ -Space Lower Bound for α -Estimate Set Cover

Our algorithm in Theorem 3 suggests that there exists at least 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 $m = \text{poly}(n)$, 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 $\tilde{\Omega}(\frac{mn}{\alpha^2})$ 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, $\text{SetCover}_{\text{est}}$ 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}_{est} for $\text{SetCover}_{\text{est}}$ in the spirit of the distribution \mathcal{D}_{apx} in Section 3. However, since in $\text{SetCover}_{\text{est}}$ the goal is only to estimate the size of the optimal cover, “hiding” one single element (as was done in \mathcal{D}_{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 a single set of Alice as a whole or was it partitioned across many different sets⁶. Similar to \mathcal{D}_{apx} , distribution \mathcal{D}_{est} is also not a product distribution; however, we introduce a way of decomposing \mathcal{D}_{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}_{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}_{ext} which *randomly* partitions the sets of distribution \mathcal{D}_{est} between Alice and Bob. We prove a lower bound for this distribution using a reduction from protocols over \mathcal{D}_{est} . Finally, we show how an algorithm for set cover over random arrival streams would be able to solve instances of $\text{SetCover}_{\text{est}}$ over \mathcal{D}_{ext} and establish Theorem 4.

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

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

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

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 $\mathcal{S} = (S_1, \dots, S_m)$, 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 \bar{T} is a set of size $\alpha \log m$ chosen uniformly at random from S_{i^*} .^a
 - If $\theta = 1$, then \bar{T} is a set of size $\alpha \log m$ chosen uniformly at random from $[n] \setminus S_{i^*}$.

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

Recall that $\text{opt}(\mathcal{S}, T)$ denotes the *set cover size* of the input instance (\mathcal{S}, T) . We first establish the following lemma regarding the parameter θ and $\text{opt}(\mathcal{S}, T)$ in the distribution \mathcal{D}_{est} .

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

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

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

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

Proof. Part (i) is immediate since by construction, when $\theta = 0$, $T \cup S_{i^*} = [n]$. We now prove part (ii).

Since a valid set cover must cover \bar{T} , it suffices for us to show that w.h.p. no 2α sets from \mathcal{S} can cover \bar{T} . By construction, neither T nor S_{i^*} contains any element in \bar{T} , hence \bar{T} must be covered by at most 2α sets in $\mathcal{S} \setminus \{S_{i^*}\}$.

Fix a collection $\hat{\mathcal{S}}$ of 2α sets in $\mathcal{S} \setminus \{S_{i^*}\}$; we first analyze the probability that $\hat{\mathcal{S}}$ covers \bar{T} and then take union bound over all choices of 2α sets from $\mathcal{S} \setminus \{S_{i^*}\}$. Note that according to the distribution \mathcal{D}_{est} , the sets in $\hat{\mathcal{S}}$ are drawn independent of \bar{T} . Fix any choice of \bar{T} ; for each element $k \in \bar{T}$, and for each set $S_j \in \hat{\mathcal{S}}$, define an indicator random variable $X_k^j \in \{0, 1\}$, where $X_k^j = 1$ iff $k \in S_j$. Let $\mathbf{X} := \sum_j \sum_k X_k^j$ and notice that:

$$\mathbb{E}[\mathbf{X}] = \sum_j \sum_k \mathbb{E}[X_k^j] = (2\alpha) \cdot (\alpha \log m) \cdot \left(\frac{1}{10\alpha}\right) = \alpha \log m / 5$$

We have,

$$\Pr(\hat{\mathcal{S}} \text{ covers } \bar{T}) \leq \Pr(\mathbf{X} \geq \alpha \log m) = \Pr(\mathbf{X} \geq 5\mathbb{E}[\mathbf{X}]) \leq \exp(-3\alpha \log m)$$

where the last equality uses the fact that X_k^j variables are negatively correlated (which can be proven analogous to Lemma 3.2) and applies the extended Chernoff bound (see Section 2). Finally, by union bound,

$$\begin{aligned} \Pr(\text{opt}(\mathcal{S}, T) \leq 2\alpha) &\leq \Pr(\exists \hat{\mathcal{S}} \text{ covers } \bar{T}) \leq \binom{m}{2\alpha} \cdot \exp(-3\alpha \log m) \\ &\leq \exp(2\alpha \cdot \log m - 3\alpha \log m) = o(1) \end{aligned}$$

Notice that distribution \mathcal{D}_{est} is not a product distribution due to the correlation between the input given to Alice and Bob. However, we can express the distribution as a convex combination of a relatively small set of product distributions; this significantly simplifies the proof of the lower bound. To do so, we need the following definition. For integers k, t and n , a collection P of t subsets of $[n]$ is called a *random (k, t) -partition* iff the t sets in P are constructed as follows: Pick k elements from $[n]$, denoted by S , uniformly at random, and partition S randomly into t sets of equal size. We refer to each set in P as a *block*.

An alternative definition of the distribution \mathcal{D}_{est} .

Parameters: $k = \frac{n}{5\alpha}$ $p = \alpha \log m$ $t = k/p$

1. For any $i \in [m]$, let P_i be a random (k, t) -partition in $[n]$ (chosen independently).
2. The input to Alice is $\mathcal{S} = (S_1, \dots, S_m)$, where each S_i is created by picking $t/2$ blocks from P_i uniformly at random.
3. The input to Bob is a set T where \bar{T} is created by first picking an $i^* \in [m]$ uniformly at random, and then picking a block from P_{i^*} uniformly at random.

To see that the two formulations of the distribution \mathcal{D}_{est} are indeed equivalent, notice that (i) the input given to Alice in the new formulation is a collection of sets of size $n/10\alpha$

chosen independently and uniformly at random (by the independence of P_i 's), and (ii) the complement of the set given to Bob is a set of size $\alpha \log m$ which, for $i^* \in_R [m]$, with probability half, is chosen uniformly at random from S_{i^*} , and with probability half, is chosen from $[n] \setminus S_{i^*}$ (by the randomness in the choice of each block in P_{i^*}).

Fix any δ -error protocol Π_{SC} ($\delta < 1/2$) for $\text{SetCover}_{\text{est}}$ on the distribution \mathcal{D}_{est} . Recall that Π_{SC} denotes the random variable for the concatenation of the message of Alice with the public randomness used in the protocol Π_{SC} . We further use $\mathcal{P} := (P_1, \dots, P_t)$ to denote the random partitions (P_1, \dots, P_t) , \mathbf{I} for the choice of the special index i^* , and θ for the parameter $\theta \in \{0, 1\}$, whereby $\theta = 0$ iff $\bar{T} \subseteq S_{i^*}$.

We make the following simple observations about the distribution \mathcal{D}_{est} . The proofs are straightforward.

Remark 5.2. In the distribution \mathcal{D}_{est} ,

1. The random variables \mathcal{S} , \mathcal{P} , and $\Pi_{SC}(\mathcal{S})$ are all independent of the random variable \mathbf{I} .
2. For any $i \in [m]$, conditioned on $P_i = P$, and $\mathbf{I} = i$, the random variables S_i and \bar{T} are independent of each other. Moreover, $\text{SUPP}(S_i)$ and $\text{SUPP}(\bar{T})$ contain, respectively, $\binom{t}{2}$ and t elements and both S_i and \bar{T} are uniform over their support.
3. For any $i \in [m]$, the random variable S_i is independent of both \mathcal{S}^{-i} and \mathcal{P}^{-i} .

5.2 The Lower Bound for the Distribution \mathcal{D}_{est}

Our goal in this section is to lower bound $\text{ICost}_{\mathcal{D}_{\text{est}}}(\Pi_{SC})$ and ultimately $\|\Pi_{SC}\|$. We start by simplifying the expression for $\text{ICost}_{\mathcal{D}_{\text{est}}}(\Pi_{SC})$.

Lemma 5.3. $\text{ICost}_{\mathcal{D}_{\text{est}}}(\Pi_{SC}) \geq \sum_{i=1}^m I(\Pi_{SC}; S_i \mid P_i)$

Proof. We have,

$$\text{ICost}_{\mathcal{D}_{\text{est}}}(\Pi_{SC}) = I(\Pi_{SC}; \mathcal{S}) \geq I(\Pi_{SC}; \mathcal{S} \mid \mathcal{P})$$

where the inequality holds since (i) $H(\Pi_{SC}) \geq H(\Pi_{SC} \mid \mathcal{P})$ and (ii) $H(\Pi_{SC} \mid \mathcal{S}) = H(\Pi_{SC} \mid \mathcal{S}, \mathcal{P})$ as Π_{SC} is independent of \mathcal{P} conditioned on \mathcal{S} . We now bound the conditional mutual information term in the above equation.

$$\begin{aligned} I(\Pi_{SC}; \mathcal{S} \mid \mathcal{P}) &= \sum_{i=1}^m I(S_i; \Pi_{SC} \mid \mathcal{P}, \mathcal{S}^{<i}) \\ &\quad \text{(the chain rule for the mutual information, Claim 2.1-(5))} \\ &= \sum_{i=1}^m H(S_i \mid \mathcal{P}, \mathcal{S}^{<i}) - H(S_i \mid \Pi_{SC}, \mathcal{P}, \mathcal{S}^{<i}) \\ &\geq \sum_{i=1}^m H(S_i \mid P_i) - H(S_i \mid \Pi_{SC}, P_i) \\ &= \sum_{i=1}^m I(S_i; \Pi_{SC} \mid P_i) \end{aligned}$$

The inequality holds since:

- (i) $H(S_i \mid P_i) = H(S_i \mid P_i, \mathcal{P}^{-i}, \mathcal{S}^{<i}) = H(S_i \mid \mathcal{P}, \mathcal{S}^{<i})$ because conditioned on P_i , S_i is independent of \mathcal{P}^{-i} and $\mathcal{S}^{<i}$ (Remark 5.2-(3)), hence the equality holds by Claim 2.1-(3).
- (ii) $H(S_i \mid \Pi_{SC}, P_i) \geq H(S_i \mid \Pi_{SC}, P_i, \mathcal{P}^{-i}, \mathcal{S}^{<i}) = H(S_i \mid \Pi_{SC}, \mathcal{P}, \mathcal{S}^{<i})$ since conditioning reduces the entropy, i.e., Claim 2.1-(3).

Equipped with Lemma 5.3, we only need to bound $\sum_{i \in [m]} I(\Pi_{SC}; S_i | P_i)$. Note that, ■

$$\sum_{i=1}^m I(\Pi_{SC}; S_i | P_i) = \sum_{i=1}^m H(S_i | P_i) - \sum_{i=1}^m H(S_i | \Pi_{SC}, P_i) \quad (5)$$

Furthermore, for each $i \in [m]$, $|\text{SUPP}(S_i | P_i)| = \binom{t}{\frac{t}{2}}$ and S_i is uniform over its support (Remark 5.2-(2)); hence, by Claim 2.1-(1),

$$\sum_{i=1}^m H(S_i | P_i) = \sum_{i=1}^m \log \left(\binom{t}{\frac{t}{2}} \right) = m \cdot \log \left(2^{t - \Theta(\log t)} \right) = m \cdot t - \Theta(m \log t) \quad (6)$$

Consequently, we only need to bound $\sum_{i=1}^m H(S_i | \Pi_{SC}, P_i)$. In order to do so, we show that Π_{SC} can be used to estimate the value of the parameter θ , and hence we only need to establish a lower bound for the problem of estimating θ .

Lemma 5.4. *Any δ -error protocol Π_{SC} over the distribution \mathcal{D}_{est} can be used to determine the value of θ with error probability $\delta + o(1)$.*

Proof. Alice sends the message $\Pi_{SC}(\mathcal{S})$ as before. Using this message, Bob can compute an α -estimation of the set cover problem using $\Pi_{SC}(\mathcal{S})$ and his input. If the estimation is less than 2α , we output $\theta = 0$ and otherwise we output $\theta = 1$. The bound on the error probability follows from Lemma 5.1. ■

Before continuing, we make the following remark which would be useful in the next section.

Remark 5.5. *We assume that in $\text{SetCover}_{\text{est}}$ over the distribution \mathcal{D}_{est} , Bob is additionally provided with the special index i^* .*

Note that this assumption can only make our lower bound stronger since Bob can always ignore this information and solves the original $\text{SetCover}_{\text{est}}$.

Let β be the function that estimates θ used in Lemma 5.4; the input to β is the message given from Alice, the public coins used by the players, the set \bar{T} , and (by Remark 5.5) the special index i^* . We have,

$$\Pr(\beta(\Pi_{SC}, \bar{T}, I) \neq \theta) \leq \delta + o(1)$$

Hence, by Fano's inequality (Claim 2.2),

$$\begin{aligned} H_2(\delta + o(1)) &\geq H(\theta | \Pi_{SC}, \bar{T}, I) \\ &= \mathbb{E}_{i \sim I} \left[H(\theta | \Pi_{SC}, \bar{T}, I = i) \right] \\ &= \frac{1}{m} \sum_{i=1}^m H(\theta | \Pi_{SC}, \bar{T}, I = i) \end{aligned} \quad (7)$$

We now show that each term above is lower bounded by $H(S_i | \Pi_{SC}, P_i)/t$ and hence we obtain the desired upper bound on $H(S_i | \Pi_{SC}, P_i)$ in Equation (5).

Lemma 5.6. *For any $i \in [m]$, $H(\theta | \Pi_{SC}, \bar{T}, I = i) \geq H(S_i | \Pi_{SC}, P_i)/t$.*

Proof. We have,

$$\begin{aligned}
H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, I = i) &\geq H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, P_i, I = i) \\
&\quad (\text{conditioning on random variables reduces entropy, Claim 2.1-(3)}) \\
&= \mathbb{E}_{P \sim P_i \mid I=i} \left[H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, P_i = P, I = i) \right]
\end{aligned}$$

For brevity, let E denote the event $(P_i = P, I = i)$. We can write the above equation as,

$$H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, P_i, I = i) = \mathbb{E}_{P \sim P_i \mid I=i} \mathbb{E}_{\bar{\mathbf{T}} \sim \bar{\mathbf{T}} \mid E} \left[H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}} = \bar{\mathbf{T}}, E) \right]$$

Note that by Remark 5.2-(2), conditioned on the event E , $\bar{\mathbf{T}}$ is chosen to be one of the blocks of $P = (B_1, \dots, B_t)$ uniformly at random. Hence,

$$H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, P_i, I = i) = \mathbb{E}_{P \sim P_i \mid I=i} \left[\sum_{j=1}^t \frac{H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}} = B_j, E)}{t} \right]$$

Define a random variable $\mathbf{X} := (X_1, \dots, X_t)$, where each $X_j \in \{0, 1\}$ and $X_j = 1$ iff S_i contains the block B_j . Note that conditioned on E , \mathbf{X} uniquely determines the set S_i . Moreover, notice that conditioned on $\bar{\mathbf{T}} = B_j$ and E , $\theta = 0$ iff $X_j = 1$. Hence,

$$H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, P_i, I = i) = \mathbb{E}_{P \sim P_i \mid I=i} \left[\sum_{j=1}^t \frac{H(\mathbf{X}_j \mid \Pi_{SC}, \bar{\mathbf{T}} = B_j, E)}{t} \right]$$

Now notice that X_j is independent of the event $\bar{\mathbf{T}} = B_j$ since S_i is chosen independent of $\bar{\mathbf{T}}$ conditioned on E (Remark 5.2-(2)). Similarly, since Π_{SC} is only a function of \mathbf{S} and \mathbf{S} is independent of $\bar{\mathbf{T}}$ conditioned on E , Π_{SC} is also independent of the event $\bar{\mathbf{T}} = B_j$. Consequently, by Claim 2.1-(6), we can “drop” the conditioning on $\bar{\mathbf{T}} = B_j$,

$$\begin{aligned}
H(\theta \mid \Pi_{SC}, \bar{\mathbf{T}}, P_i, I = i) &= \mathbb{E}_{P \sim P_i \mid I=i} \left[\sum_{j=1}^t \frac{H(\mathbf{X}_j \mid \Pi_{SC}, E)}{t} \right] \\
&\geq \mathbb{E}_{P \sim P_i \mid I=i} \left[\frac{H(\mathbf{X} \mid \Pi_{SC}, E)}{t} \right] \\
&\quad (\text{sub-additivity of the entropy, Claim 2.1-(4)}) \\
&= \mathbb{E}_{P \sim P_i \mid I=i} \left[\frac{H(S_i \mid \Pi_{SC}, E)}{t} \right] \\
&\quad (S_i \text{ and } \mathbf{X} \text{ uniquely define each other conditioned on } E) \\
&= \mathbb{E}_{P \sim P_i \mid I=i} \left[\frac{H(S_i \mid \Pi_{SC}, P_i = P, I = i)}{t} \right] \\
&\quad (E \text{ is defined as } (P_i = P, I = i)) \\
&= \frac{H(S_i \mid \Pi_{SC}, P_i, I = i)}{t}
\end{aligned}$$

Finally, by Remark 5.2-(1), S_i , Π_{SC} , and P_i are all independent of the event $I = i$, and hence by Claim 2.1-(6), $H(S_i \mid \Pi_{SC}, P_i, I = i) = H(S_i \mid \Pi_{SC}, P_i)$, which concludes the proof. \blacksquare

By plugging in the bound from Lemma 5.6 in Equation (7) we have,

$$\sum_{i=1}^m H(S_i \mid \Pi_{SC}, P_i) \leq H_2(\delta + o(1)) \cdot (mt)$$

Finally, by plugging in this bound together with the bound from Equation (6) in Equation (5), we get,

$$\begin{aligned} \sum_{i=1}^m I(\Pi_{\text{SC}}; S_i \mid P_i) &\geq mt - \Theta(m \log t) - H_2(\delta + o(1)) \cdot (mt) \\ &= \left(1 - H_2(\delta + o(1))\right) \cdot (mt) - \Theta(m \log t) \end{aligned}$$

Recall that $t = k/p = \tilde{\Omega}(n/\alpha^2)$ and $\delta < 1/2$, hence $H_2(\delta + o(1)) = 1 - \epsilon$ for some constant ϵ bounded away from 0. By Lemma 5.3,

$$\text{IC}_{\mathcal{D}_{\text{est}}}^{\delta}(\text{SetCover}_{\text{est}}) = \min_{\Pi_{\text{SC}}} \left(\text{ICost}_{\mathcal{D}_{\text{est}}}(\Pi_{\text{SC}}) \right) = \tilde{\Omega}(mn/\alpha^2)$$

To conclude, since the information complexity is a lower bound on the communication complexity (Proposition 2.5), we obtain the following theorem.

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

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

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 $\tilde{\Omega}(mn/\alpha^2)$.

5.3 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}_{est} , denoted by \mathcal{D}_{ext} , prove a lower bound for \mathcal{D}_{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}_{ext} as follows.

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

1. Sample the sets $\mathcal{S} = \{S_1, \dots, S_m, T\}$ in the same way as in the distribution \mathcal{D}_{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}_{ext} is still a hard distribution for $\text{SetCover}_{\text{est}}$.

Lemma 5.7. *For any constant $\delta < 1/8$, $\alpha = o(\sqrt{n/\log n})$, and $m = \text{poly}(n)$,*

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

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

Consider an instance of the problem from the distribution \mathcal{D}_{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 $1/2$. Note that by Remark 5.5, we may assume that Bob 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 \mathcal{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}_{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 $\hat{\mathcal{E}}$ to be the event that in the distribution \mathcal{D}_{ext} , S_{i^*} is assigned to Alice and T is assigned to Bob. It is straightforward to verify that $\mathcal{D}_{\text{new}} = (\mathcal{D}_{\text{ext}} \mid \hat{\mathcal{E}})$. We now have,

$$\begin{aligned}
\Pr_{\mathcal{D}_{\text{est}}, \mathcal{R}}(\Pi_{\text{SC}} \text{ errs}) &= \frac{1}{2} \cdot \Pr_{\mathcal{R}}(\bar{\mathcal{E}}) + \Pr_{\mathcal{R}}(\mathcal{E}) \cdot \Pr_{\mathcal{D}_{\text{new}}}(\Pi_{\text{Ext}} \text{ errs}) \\
&\quad (\text{since Bob outputs 2 when } \bar{\mathcal{E}}, \text{ he will succeed with probability } 1/2) \\
&= \frac{1}{2} \cdot \Pr_{\mathcal{R}}(\bar{\mathcal{E}}) + \Pr_{\mathcal{R}}(\mathcal{E}) \cdot \Pr_{\mathcal{D}_{\text{ext}}}(\Pi_{\text{Ext}} \text{ errs} \mid \hat{\mathcal{E}}) \quad (\mathcal{D}_{\text{new}} = (\mathcal{D}_{\text{ext}} \mid \hat{\mathcal{E}})) \\
&\leq \frac{1}{2} \cdot \Pr_{\mathcal{R}}(\bar{\mathcal{E}}) + \Pr_{\mathcal{R}}(\mathcal{E}) \cdot \frac{\Pr_{\mathcal{D}_{\text{ext}}}(\Pi_{\text{Ext}} \text{ errs})}{\Pr_{\mathcal{D}_{\text{ext}}}(\hat{\mathcal{E}})} \\
&= \frac{1}{2} \cdot \Pr_{\mathcal{R}}(\bar{\mathcal{E}}) + \Pr_{\mathcal{D}_{\text{ext}}}(\Pi_{\text{Ext}} \text{ errs}) \quad (\Pr_{\mathcal{R}}(\mathcal{E}) = \Pr_{\mathcal{D}_{\text{ext}}}(\hat{\mathcal{E}})) \\
&\leq \frac{3}{8} + \delta \quad (\Pr_{\mathcal{R}}(\bar{\mathcal{E}}) = 3/4)
\end{aligned}$$

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}_{est} . The lower bound now follows from Theorem 5. \blacksquare

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

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

Consider any input \mathcal{S} in the distribution \mathcal{D}_{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 $\mathcal{A}(\langle s_A, s_B \rangle)$ 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}_{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 5.7. \blacksquare

6 An $\Omega(mn/\alpha)$ -Space Lower Bound for Deterministic α -Estimation

In Section 4, we provided a *randomized* algorithm for α -estimating the set cover problem, with an (essentially) optimal space bound (as we proved in Section 5); here, we establish that randomization is crucial for achieving better space bound for α -estimation: any *deterministic* α -estimation algorithm for the set cover problem requires $\Omega(\frac{mn}{\alpha})$ bits of space. In other words, α -approximation and α -estimation are as *hard* as each other for deterministic algorithms. Formally,

Theorem 6. *For any $\alpha = o(\sqrt{\frac{n}{\log n}})$ and $m = \text{poly}(n)$, any deterministic α -estimation single-pass streaming algorithm for the set cover problem requires $\Omega(mn/\alpha)$ bits of space.*

Before continuing, we make the following remark. The previous best lower bound for *deterministic* algorithm asserts that any $O(1)$ -estimation algorithm requires $\Omega(mn)$ bits of space even allowing *constant number of passes* [10]. This lower bound can be stated more generally in terms of its dependence on α : for any $\alpha \leq \log n - O(\log \log n)$, any deterministic α -estimation algorithm requires space of $\Omega(mn/\alpha \cdot 2^\alpha)$ bits (see Lemma 11 and Theorem 3 in [10]). Our bound in Theorem 6 provide an exponential improvement over this bound for *single-pass* algorithms.

Recall the definition of $\text{SetCover}_{\text{est}}$ from Section 5 for a fixed choice of parameters n, m , and α . We define the following hard input distribution for deterministic protocols of $\text{SetCover}_{\text{est}}$.

Distribution \mathcal{D}_{det} . A hard input distribution for *deterministic* protocols of $\text{SetCover}_{\text{est}}$.

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

- **Alice.** The input to Alice is an m -subset $S \subseteq \mathcal{F}$ chosen uniformly at random.
- **Bob.** Toss a coin $\theta \in_R \{0, 1\}$; let the input to Bob be the set T , where if $\theta = 0$, $\bar{T} \in_R S$ and otherwise, $\bar{T} \in_R \mathcal{F} \setminus S$.

We stress that distribution \mathcal{D}_{det} can *only* be hard for deterministic protocols; Alice can simply run a *randomized* protocol for checking the equality of each S_i with \bar{T} (independently for each $S_i \in S$) and since the one-way communication complexity of the *Equality* problem with public randomness is $O(1)$ (see, e.g., [21]), the total communication complexity of this protocol is $O(m)$. However, such approach is not possible for deterministic protocols since deterministic communication complexity of the *Equality* problem is linear in the input size (again, see [21]).

The following lemma can be proven similar to Lemma 5.1 (assuming that $\alpha = o(\sqrt{\frac{n}{\log n}})$).

Lemma 6.1. *For $(S, T) \sim \mathcal{D}_{\text{det}}$:*

- (i) $\Pr(\text{opt}(S, T) = 2 \mid \theta = 0) = 1$.
- (ii) $\Pr(\text{opt}(S, T) > 2\alpha \mid \theta = 1) = 1 - 2^{-\Omega(n/\alpha)}$.

To prove Theorem 6, we use a reduction from the *Sparse Indexing* problem of [26]. In $\text{SparseIndexing}_k^N$, Alice is given a set S of k elements from a universe $[N]$ and Bob is given an element $e \in [N]$; Bob needs to output whether or not $e \in S$. The crucial property of this problem that we need in our proof is that the communication complexity of *Sparse Indexing* depends on the success probability required from the protocol.

The following lemma is a restatement of Theorem 3.2 from [26].

Lemma 6.2 ([26]). *Consider the following distribution \mathcal{D}_{SI} for $\text{SparseIndexing}_k^N$:*

- Alice is given a k -subset $S \subseteq [N]$ chosen uniformly at random.
- With probability half Bob is given an element $e \in S$ uniformly at random, and with probability half Bob receives a random element $e \in [N] \setminus S$.

For any $\delta < 1/3$, the communication complexity of any deterministic δ -error protocol over distribution \mathcal{D}_{SI} is $\Omega(\min \{k \log(1/\delta), k \log(N/k)\})$ bits.

We now show that how a deterministic protocol for $\text{SetCover}_{\text{est}}$ can be used to solve the hard distribution of $\text{SparselIndexing}_k^N$ described in the previous lemma.

Notice that there is a simple one-to-one mapping ϕ from an instance (S, e) of $\text{SparselIndexing}_k^N$ to an instance (S, T) of $\text{SetCover}_{\text{est}}$ with parameters n and m where $N = \binom{n}{\frac{n}{10\alpha}}$ and $k = m$. Fix any arbitrary bijection $\sigma : [N] \mapsto \mathcal{F}$; we map S to the collection $\mathcal{S} = \{\sigma(j) \mid j \in S\}$ and map e to the set $T = [n] \setminus \sigma(e)$. Under this transformation, if we choose $(S, e) \sim \mathcal{D}_{\text{SI}}$ the $\text{SetCover}_{\text{est}}$ distribution induced by $\phi(S, e)$ is equivalent to the distribution \mathcal{D}_{det} . Moreover, the answer of (S, e) for the $\text{SparselIndexing}_k^N$ problem is YES iff $\theta = 0$ in (S, T) . We now proceed to the proof of Theorem 6.

Proof of Theorem 6. Let Π_{SC} be a deterministic protocol for the $\text{SetCover}_{\text{est}}$ problem; we use Π_{SC} to design a δ -error protocol Π_{SI} for $\text{SparselIndexing}_k^N$ on the distribution \mathcal{D}_{SI} for $\delta = 2^{-\Omega(n/\alpha)}$. The lower bound on message size of Π_{SC} then follows from Lemma 6.2.

Protocol Π_{SI} works as follows. Consider the mapping ϕ from (S, e) to (S, T) defined before. Given an instance $(S, e) \sim \mathcal{D}_{\text{SI}}$, Alice and Bob can each compute their input in $\phi(S, e)$ without any communication. Next, they run the protocol $\Pi_{\text{SC}}(S, T)$ and if the set cover size is less than 2α Bob outputs YES and otherwise outputs NO.

We argue that Π_{SI} is a δ -error protocol for $\text{SparselIndexing}_k^N$ on the distribution \mathcal{D}_{SI} . To see this, recall that if $(S, e) \sim \mathcal{D}_{\text{SI}}$ then $(S, T) \sim \mathcal{D}_{\text{det}}$. Let \mathcal{E} be an event in the distribution \mathcal{D}_{det} , where $\theta = 1$ but $\text{opt} < 2\alpha$. Note that since Π_{SC} is a deterministic α -approximation protocol and never errs, the answer for Π_{SI} would be wrong only if \mathcal{E} happens. We have,

$$\Pr_{(S,e) \sim \mathcal{D}_{\text{SI}}} (\Pi_{\text{SI}}(S, e) \text{ errs}) \leq \Pr_{(S,T) \sim \mathcal{D}_{\text{det}}} (\mathcal{E}) = 2^{-\Omega(n/\alpha)}$$

where the equality is by Lemma 6.1. This implies that Π_{SI} is a δ -error protocol for the distribution \mathcal{D}_{SI} . By Lemma 6.2,

$$\|\Pi_{\text{SC}}\| = \|\Pi_{\text{SI}}\| = \Omega(\min(k \cdot \log 1/\delta, k \log(N/k))) = \Omega(mn/\alpha)$$

As one-way communication complexity is a lower bound on the space complexity of single-pass streaming algorithm we obtain the final result. \blacksquare

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.

References

- [1] ALON, N., MATIAS, Y., AND SZEGEDY, M. The space complexity of approximating the frequency moments. In *STOC (1996)*, ACM, pp. 20–29.
- [2] BAR-YOSSEF, Z., JAYRAM, T. S., KUMAR, R., AND SIVAKUMAR, D. 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 (2002)*, pp. 209–218.
- [3] BAR-YOSSEF, Z., JAYRAM, T. S., KUMAR, R., AND SIVAKUMAR, D. Information theory methods in communication complexity. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002 (2002)*, pp. 93–102.
- [4] BARAK, B., BRAVERMAN, M., CHEN, X., AND RAO, A. 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 (2010)*, pp. 67–76.
- [5] CHAKRABARTI, A., CORMODE, G., AND MCGREGOR, A. 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 (2008)*, pp. 641–650.
- [6] CHAKRABARTI, A., SHI, Y., WIRTH, A., AND YAO, A. C. 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 (2001)*, pp. 270–278.
- [7] CHAKRABARTI, A., AND WIRTH, T. Incidence geometries and the pass complexity of semi-streaming set cover. *CoRR*, abs/1507.04645. To appear in *Symposium on Discrete Algorithms (SODA) (2016)*.
- [8] CORMODE, G., KARLOFF, H. J., AND WIRTH, A. 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 (2010)*, pp. 479–488.
- [9] COVER, T. M., AND THOMAS, J. A. *Elements of information theory (2. ed.)*. Wiley, 2006.
- [10] DEMAINE, E. D., INDYK, P., MAHABADI, S., AND VAKILIAN, A. 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 (2014)*, pp. 484–498.
- [11] DINUR, I., AND STEURER, D. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014 (2014)*, pp. 624–633.
- [12] EMEK, Y., AND ROSÉN, A. 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 (2014)*, pp. 453–464.
- [13] FEIGE, U. A threshold of $\ln n$ for approximating set cover. *J. ACM* 45, 4 (1998), 634–652.
- [14] FEIGENBAUM, J., KANNAN, S., MCGREGOR, A., SURI, S., AND ZHANG, J. On graph problems in a semi-streaming model. *Theor. Comput. Sci.* 348, 2-3 (2005), 207–216.

- [15] HAR-PELED, S., INDYK, P., MAHABADI, S., AND VAKILIAN, A. Towards tight bounds for the streaming set cover problem. *To appear in PODS* (2016).
- [16] IMPAGLIAZZO, R., AND KABANETS, V. 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* (2010), pp. 617–631.
- [17] INDYK, P., MAHABADI, S., AND VAKILIAN, A. Towards tight bounds for the streaming set cover problem. *CoRR abs/1509.00118* (2015).
- [18] JAYRAM, T. S., AND WOODRUFF, D. P. 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* (2011), pp. 1–10.
- [19] JOHNSON, D. S. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.* 9, 3 (1974), 256–278.
- [20] KAPRALOV, M., KHANNA, S., AND SUDAN, M. 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* (2014), pp. 734–751.
- [21] KUSHILEVITZ, E., AND NISAN, N. *Communication complexity*. Cambridge University Press, 1997.
- [22] LUND, C., AND YANNAKAKIS, M. On the hardness of approximating minimization problems. *J. ACM* 41, 5 (1994), 960–981.
- [23] MUTHUKRISHNAN, S. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science* 1, 2 (2005).
- [24] NISAN, N. 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* (2002), pp. 868–875.
- [25] PANCONESI, A., AND SRINIVASAN, A. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.* 26, 2 (1997), 350–368.
- [26] SAGLAM, M. *Tight bounds for data stream algorithms and communication problems*. PhD thesis, Simon Fraser University, 2011.
- [27] SAHA, B., AND GETOOR, L. 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* (2009), pp. 697–708.
- [28] SLAVÍK, P. A tight analysis of the greedy algorithm for set cover. *J. Algorithms* 25, 2 (1997), 237–254.

A A Lower Bound for $\text{ICost}_{\mathcal{D}_{\text{Index}}}^\delta(\text{Index}_k^n)$

In this section we prove Lemma 3.4. As stated earlier, this lemma is well-known and is proved here only for the sake of completeness.

Proof of Lemma 3.4. Consider the following “modification” to the distribution $\mathcal{D}_{\text{Index}}$. Suppose we first sample a $2k$ -set $P \subseteq [n]$, and then let the input to Alice be a k -set $A \subset P$ chosen uniformly at random, and the input to Bob be an element $a \in P$ chosen uniformly at random. It is an easy exercise to verify that this modification does not change the distribution $\mathcal{D}_{\text{Index}}$. However, this allows us to analyze the information cost of any protocol Π_{Index} over $\mathcal{D}_{\text{Index}}$ easier. In particular,

$$\begin{aligned} \text{ICost}_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Index}}) &= I(A; \Pi_{\text{Index}}) \\ &\geq I(A; \Pi_{\text{Index}} \mid P) \\ &= H(A \mid P) - H(A \mid \Pi_{\text{Index}}, P) \\ &= 2k - \Theta(\log k) - H(A \mid \Pi_{\text{Index}}, P) \end{aligned}$$

where the inequality holds since (i) $H(\Pi_{\text{Index}}) \geq H(\Pi_{\text{Index}} \mid P)$ and (ii) $H(\Pi_{\text{Index}} \mid A) = H(\Pi_{\text{Index}} \mid A, P)$ as Π_{Index} is independent of P conditioned on A .

We now bound $H(A \mid \Pi_{\text{Index}}, P)$. Define $\theta \in \{0, 1\}$ where $\theta = 1$ iff $a \in A$. Note that a δ' -error protocol for Index is also a δ' -error (randomized) estimator for θ (given the message Π_{Index} , the public randomness used along with the message, and the element a). Hence, using Fano’s inequality (Claim 2.2),

$$\begin{aligned} H_2(\delta') &\geq H(\theta \mid \Pi_{\text{Index}}, a) \\ &\geq H(\theta \mid \Pi_{\text{Index}}, a, P) \quad (\text{conditioning decreases entropy, Claim 2.1-(3)}) \\ &= \mathbb{E}_{P \sim \mathcal{P}} \mathbb{E}_{a \sim a \mid P=P} \left[H(\theta \mid \Pi_{\text{Index}}, a = a, P = P) \right] \end{aligned}$$

Conditioned on $P = P$, a is chosen uniformly at random from P . Define $X := (X_1, \dots, X_{2k})$, where $X_i = 1$ if i -th element in P belongs to A and $X_i = 0$ otherwise. Using this notation, we have $\theta = X_i$ conditioned on $P = P$, and $a = a$ being the i -th element of P . Hence,

$$\begin{aligned} H_2(\delta') &\geq \mathbb{E}_{P \sim \mathcal{P}} \left[\sum_{i=1}^{2k} \frac{1}{2k} \cdot H(X_i \mid \Pi_{\text{Index}}, a = a, P = P) \right] \\ &= \mathbb{E}_{P \sim \mathcal{P}} \left[\sum_{i=1}^{2k} \frac{1}{2k} \cdot H(X_i \mid \Pi_{\text{Index}}, P = P) \right] \\ &\quad (X_i \text{ and } \Pi_{\text{Index}} \text{ are independent of } a = a \text{ conditioned on } P = P, \text{ and Claim 2.1-(6)}) \\ &\geq \frac{1}{2k} \cdot \mathbb{E}_{P \sim \mathcal{P}} \left[H(A \mid \Pi_{\text{Index}}, P = P) \right] \quad (\text{sub-additivity of the entropy, Claim 2.1-(4)}) \\ &= \frac{1}{2k} H(A \mid \Pi_{\text{Index}}, P) \end{aligned}$$

Consequently,

$$H(A \mid \Pi_{\text{Index}}, P) \leq 2k \cdot H_2(\delta')$$

Finally, since $\delta' < 1/2$ and hence $H_2(\delta') = (1 - \epsilon)$ for some ϵ bounded away from 0, we have,

$$\text{ICost}_{\mathcal{D}_{\text{Index}}}(\Pi_{\text{Index}}) \geq 2k \cdot (1 - H_2(\delta')) - \Theta(\log k) = \Omega(k)$$

■