2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)

Random Order Online Set Cover is as Easy as Offline

Anupam Gupta
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
anupamg@cs.cmu.edu

Gregory Kehne
School of Engineering and Applied Sciences
Harvard University
Boston, USA
gkehne@g.harvard.edu

Roie Levin
Computer Science Department
Carnegie Mellon University
Pittsburgh, USA
roiel@cs.cmu.edu

2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-6654-2055-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/FOCS52979.2021.00122

***Abstract*—We give a polynomial-time algorithm for ONLINESETCOVER with a competitive ratio of $O(\log mn)$ when the elements are revealed in random order, matching the best possible offline bound of $O(\log n)$ when the number of sets m is polynomial in the number of elements n, and circumventing the $\Omega(\log m \log n)$ lower bound known in adversarial order. We also extend the result to solving pure covering IPs when constraints arrive in random order.**

The algorithm is a multiplicative-weights-based round-and-solve approach we call LEARNORCOVER. We maintain a coarse fractional solution that is neither feasible nor monotone increasing, but can nevertheless be rounded online to achieve the claimed guarantee (in the random order model). This gives a new offline algorithm for SETCOVER that performs a single pass through the elements, which may be of independent interest.

I. INTRODUCTION

In the SETCOVER problem we are given a set system $(U, \mathcal{S})$ (where U is a ground set of size n and $\mathcal{S}$ is a collection of subsets with $|\mathcal{S}| = m$), along with a cost function $c : \mathcal{S} \to \mathbb{R}^+$. The goal is to select a minimum cost subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that the union of the sets in $\mathcal{S}'$ is U. Many algorithms have been discovered for this problem that achieve an approximation ratio of $\ln n$ (see e.g. [11], [24], [27], [33]), and this is best possible unless $\mathsf{P} = \mathsf{NP}$ [16], [14].

In the ONLINESETCOVER variant, we impose the additional restriction that the algorithm does not know U initially, nor the contents of each $S \in \mathcal{S}$. Instead, an adversary reveals the elements of U one-by-one in an arbitrary order. On the arrival of every element, it is revealed which sets of $S \in \mathcal{S}$ contain the element, and the algorithm must immediately pick one such set $S \in \mathcal{S}$ to cover it. The goal is to minimize the total cost of the sets chosen by the algorithm. In their seminal work, [7] show that despite the lack of foresight, it is possible to achieve a competitive ratio of $O(\log m \log n)$ for this version. This result has since been shown to be tight unless $\mathsf{NP} \subseteq \mathsf{BPP}$ [26], and has also been generalized significantly (e.g. [6], [9], [21], [19]).

In this paper, we answer the question:

> *What is the best competitive ratio possible for* ONLINESETCOVER *when the adversary is constrained to reveal the elements in uniformly random order (RO)?*

We call this version ROSETCOVER. Note that the element set U is still adversarially chosen and unknown, and only the arrival order is random.

A. Results

We show that with only this one additional assumption on the element arrival order, there is an efficient algorithm for ROSETCOVER with expected competitive ratio matching the best-possible offline approximation guarantee (at least in the regime where $m = \text{poly}(n)$).

Theorem I.1. LEARNORCOVER *is a polynomial-time randomized algorithm for* ROSETCOVER *achieving expected competitive ratio* $O(\log(mn))$.

When run offline, our approach gives a new asymptotically optimal algorithm for SETCOVER for $m = \text{poly}(n)$, which may be of independent interest. Indeed, given an estimate for the optimal *value* of the set cover, our algorithm makes a single pass over the elements (considered in random order), updating a fractional solution using a multiplicative-weights framework, and sampling sets as it goes. This simplicity, and the fact that it uses only $\widetilde{O}(m)$ bits of memory, may make the algorithm useful for some low-space streaming applications. (Note that previous formulations of STREAMINGSETCOVER [32], [13], [23], [15] only consider cases where *sets* arrive in a stream.)

We show next that a suitable generalization of the same algorithm achieves the same competitive ratio for the *RO*

Supported in part by NSF awards CCF-1907820, CCF1955785, and CCF-2006953.

The full version of this paper [18] is available at https://arxiv.org/abs/2111.06842.

Throughout this paper we consider the *unknown-instance model* for online set cover (see Chapter 1 of [26]). The result of [7] was presented for the *known-instance model*, but extends to the unknown setting as well. This was made explicit in subsequent work [9].

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00122

Covering Integer Program problem (ROCIP) (see Section IV for a formal description).

Theorem I.2. LEARNORCOVERCIP *is a polynomial-time randomized algorithm for* ROCIP *achieving competitive ratio* $O(\log(mn))$.

We complement our main theorem with some lower bounds. For instance, we show that the algorithms of [7], [9] have a performance of $\Theta(\log m \log n)$ even in RO, so a new algorithm is indeed needed. Moreover, we observe an $\Omega(\log n)$ lower bound on *fractional* algorithms for ROSETCOVER. This means we cannot pursue a two-phase strategy of maintaining a good monotone fractional solution and then randomly rounding it (as was done in prior works) without losing $\Omega(\log^2 n)$. Interestingly, our algorithm *does* maintain a (non-monotone) fractional solution while rounding it online, but does so in a way that avoids extra losses. We hope that our approach will be useful in other works for online problems in RO settings. (We also give other lower bounds for batched versions of the problem, and for the more general submodular cover problem.)

B. Techniques and Overview

The core contribution of this work is demonstrating that one can exploit randomness in the arrival order to *learn* about the underlying set system. What is more, this learning can be done fast enough (in terms of both sample and computational complexity) to build an $O(\log mn)$-competitive solution, even while committing to covering incoming elements immediately upon arrival. This seems like an idea with applications to other sequential decision-making problems, particularly in the RO setting.

We start in Section II with an exponential time algorithm for the unit-cost setting. This algorithm maintains a portfolio of all exponentially-many collections of cost $c(\text{OPT})$ that are feasible for the elements observed so far. When an uncovered element arrives, the algorithm takes a random collection from the portfolio, and picks a random set from it covering the element. It then prunes the portfolio to drop collections that did not cover the incoming element. We show that either the expected marginal coverage of the chosen set is large, or the expected number of solutions removed from the portfolio is large. I.e., we either make progress *covering*, or *learning*. We show that within $c(\text{OPT}) \log(mn)$ rounds, the portfolio only contains the true optimal feasible solutions, or all unseen elements are covered. (One insight that comes from our result is that a good measure of set quality is the number of times an unseen *and uncovered* element appears in it.)

We then give a polynomial-time algorithm in Section III: while it is quite different from the exponential scheme above, it is also based on this insight, and the intuition that our algorithm should make progress via learning or covering. Specifically, we maintain a distribution $\{x_S\}_{S \in \mathcal{S}}$ on sets: for each arriving uncovered element, we first sample from this distribution x, then update x via a multiplicative weights rule. If the element remains uncovered, we buy the cheapest set covering it. For the analysis, we introduce a potential function which simultaneously measures the convergence of this distribution to the optimal fractional solution, and the progress towards covering the universe. Crucially, this progress is measured in expectation over the random order, thereby circumventing lower bounds for the adversarial-order setting [26]. In Section IV we extend our method to the more general ROCIP problem: the intuitions and general proof outlines are similar, but we need to extend the algorithm to handle elements being partially covered.

Finally, we present lower bounds in Section V. Our information-theoretic lower bounds for ROSETCOVER follow from elementary combinatorial arguments. We also show a lower bound for a batched version of ROSETCOVER, following the hardness proof of [26]; we use it in turn to derive lower bounds for ROSUBMODULARCOVER. See the full version for the proofs of this section.

C. Other Related Work

There has been much recent interest in algorithms for random order online problems. Starting from the secretary problem, the RO model has been extended to include metric facility location [28], network design [29], and solving packing LPs [2], [31], [25], [20], [1], [5], load-balancing [20], [30] and scheduling [3], [4]. See [22] for a recent survey.

Our work is closely related to [17], who give an $O(\log mn)$-competitive algorithm a related stochastic model, where the elements are chosen i.i.d. from a *known distribution* over the elements. [12] generalize the result of [17] to the *prophet version*, in which elements are drawn from known but distinct distributions. Our work is a substantial strengthening, since the RO model captures the *unknown* i.i.d. setting as a special case. Moreover, the learning is an important and interesting part of our technical contribution. On the flip side, the algorithm of [17] satisfies *universality* (see their work for a definition), which our algorithm does not. We point out that [17] claim (in a note, without proof) that it is not possible to circumvent the $\Omega(\log m \log n)$ lower bound of [26] even in RO; our results show this claim is incorrect. See the full version for a discussion.

Regret bounds for online learning are also proven via the KL divergence (see, e.g., [8]). However, no reductions are known from our problem to the classic MW setting, and it is unclear how random order would play a role in the analysis: this is necessary to bypass the adversarial order lower bounds.

Finally, [26] gives an algorithm with competitive ratio $k \log(m/k)$—hence total cost $k^2 \log(m/k)$—for unweighted

ONLINESETCOVER where $k = |\text{OPT}|$. The algorithm is the same as our exponential time algorithm in Section II for the special case of $k = 1$; however, we outperform it for non-constant k, and generalize it to get polynomial-time algorithms as well.

D. Preliminaries

All logarithms in this paper are taken to be base e. In the following definitions, let $x, y \in \mathbb{R}^n_+$ be vectors. The standard dot product between x and y is denoted $\langle x, y\rangle = \sum_{i=1}^n x_i y_i$. We use a weighted generalization of KL divergence. Given a weight function c, define

$$\mathrm{KL}_c\left(x \parallel y\right) := \sum_{i=1}^n c_i \left[x_i \log\left(\frac{x_i}{y_i}\right) - x_i + y_i \right]. \quad \text{(I.1)}$$

II. WARMUP: AN EXPONENTIAL TIME ALGORITHM FOR UNIT COSTS

We begin with an exponential-time algorithm which we call SIMPLELEARNORCOVER to demonstrate some core ideas of our result. In what follows we assume that we know a number $k \in [c(\text{OPT}), 2 \cdot c(\text{OPT})]$. This assumption is easily removed by a standard guess and double procedure, at the cost of an additional factor of 2.

The algorithm is as follows. Maintain a list $\mathfrak{T} \subseteq \binom{\mathcal{S}}{k}$ of candidate k-tuples of sets. When an uncovered element v arrives, choose a k-tuple $\mathcal{T} = (T_1, \ldots, T_k)$ uniformly at random from $\mathfrak{T}$, and buy a uniformly random T from $\mathcal{T}$. Also buy an arbitrary set containing v. Finally, discard from $\mathfrak{T}$ any k-tuples that do not cover v.

Theorem II.1. SIMPLELEARNORCOVER *is a randomized algorithm for unit-cost* ROSETCOVER *with expected cost* $O(k \cdot \log(mn))$.

Proof of Theorem II.1: Consider any time step t in which a random element arrives that is *uncovered* on arrival. Let U^t be the set of elements that remain uncovered at the end of time step t. Before the algorithm takes action, there are two cases:

Case 1: At least half the tuples in $\mathfrak{T}$ cover at least $|U^{t-1}|/2$ of the $|U^{t-1}|$ as-of-yet-uncovered elements. In this case we say the algorithm performs a **Cover Step** in round t.

Case 2: At least half the tuples in $\mathfrak{T}$ cover strictly less than $|U^{t-1}|/2$ of the uncovered elements; we say the algorithm performs a **Learning Step** in round t.

Define $\mathfrak{N}(c)$ to be the number of uncovered elements remaining after c Cover Steps. Define $\mathfrak{M}(\ell)$ to be the value of $|\mathfrak{T}|$ after ℓ Learning Steps. We will show that after $10k(\log m + \log n)$ rounds, either the number of elements remaining is less than $\mathfrak{N}(10k \log n)$ or the number of tuples remaining is less than $\mathfrak{M}(10k \log m)$. In particular, we argue that both $\mathbb{E}[\mathfrak{N}(10k \log n)]$ and $\mathbb{E}[\mathfrak{M}(10k \log m)]$ are less than 1.

Claim II.2. $\mathbb{E}[\mathfrak{N}(c + 1) \mid \mathfrak{N}(c) = N] \le \left(1 - \frac{1}{4k}\right) N.$

Proof: If round t is a Cover Step, then at least half of the $\mathcal{T} \in \mathfrak{T}$ cover at least half of U^{t-1}, so $\mathbb{E}\left[\left|(\bigcup \mathcal{T}) \cap U^{t-1}\right|\right] \ge |U^{t-1}|/4$. Since T is drawn uniformly at random from the k sets in the uniformly random $\mathcal{T}$, we have $\mathbb{E}\left[\left|T \cap U^{t-1}\right|\right] \ge |U^{t-1}|/4k$. ■

Claim II.3. $\mathbb{E}[\mathfrak{M}(\ell + 1) \mid \mathfrak{M}(\ell) = M] \le \frac{3}{4} M.$

Proof: Upon the arrival of v in a Learning Step, at least half the tuples have probability at least $1/2$ of being removed from $\mathfrak{T}$, so the expected number of tuples removed from $\mathfrak{T}$ is at least $M/4$. ■

To conclude, by induction

$$\mathbb{E}[\mathfrak{N}(10k \log n)] \le n \left(1 - \frac{1}{4k}\right)^{10k \log n} \le 1$$

and

$$\mathbb{E}[\mathfrak{M}(10k \log m)] \le m^k \left(\frac{3}{4}\right)^{10k \log m} \le 1.$$

Note that if there are N remaining uncovered elements and M remaining tuples to choose from, the algorithm will pay at most $\min(k \cdot M, N)$ before all elements are covered. Thus the total expected cost of the algorithm is bounded by $10k(\log m + \log n) + \mathbb{E}[\min(k \cdot \mathfrak{M}(10k \log m), \mathfrak{N}(10k \log n))] = O(k \log mn)$. ■

Apart from the obvious challenge of modifying SIMPLELEARNORCOVER to run in polynomial time, it is unclear how to generalize it to handle non-unit costs. Still, the intuition from this algorithm will be useful for the next sections.

III. A POLYNOMIAL-TIME ALGORITHM FOR GENERAL COSTS

We build on our intuition from Section II that we can either make progress in covering or in learning about the optimal solution. To get an efficient algorithm, we directly maintain a probability distribution over sets, which we update via a multiplicative weights rule. We use a potential function that simultaneously measures progress towards learning the optimal solution, and towards covering the unseen elements. Before we present the formal details and the pseudocode, here are the main pieces of the algorithm.

1) We maintain a fractional vector x which is a (*not necessarily feasible*) guess for the LP solution of cost β to the set cover instance.
2) Every round t in which an uncovered element v^t arrives, we

a) sample every set S with probability proportional to its current LP value x_S,
b) increase the value x_S of all sets $S \ni v^t$ multiplicatively and renormalize,
c) buy a cheapest set to cover v^t if it remains uncovered.

Formally, by a guess-and-double approach, we assume we know a bound β such that $\mathrm{LP}_{\mathrm{OPT}} \leq \beta \leq 2 \cdot \mathrm{LP}_{\mathrm{OPT}}$; here $\mathrm{LP}_{\mathrm{OPT}}$ is the cost of the optimal LP solution to the final unknown instance. Define

$$\kappa_v := \min\{c_S \mid S \ni v\} \tag{III.1}$$

as the cost of the cheapest set covering v.

Algorithm 1 LEARNORCOVER

1: Let $m' \leftarrow |\{S : c(S) \leq \beta\}|$.
2: Initialize $x_S^0 \leftarrow \frac{\beta}{c_S \cdot m'} \cdot \mathbb{1}\{c(S) \leq \beta\}$, and $\mathcal{C}^0 \leftarrow \emptyset$.
3: **for** $t = 1, 2 \ldots, n$ **do**
4: $\quad v^t \leftarrow t^{th}$ element in the random order, and let $\mathcal{R}^t \leftarrow \emptyset$.
5: $\quad$ **if** v^t not already covered **then**
6: $\quad\quad$ **for each** set S, with probability
7: $\quad\quad\quad \min(\kappa_{v^t} \cdot x_S^{t-1}/\beta, 1)$ add $\mathcal{R}^t \leftarrow \mathcal{R}^t \cup \{S\}$.
8: $\quad\quad$ Update $\mathcal{C}^t \leftarrow \mathcal{C}^{t-1} \cup \mathcal{R}^t$.
9: $\quad\quad$ **if** $\sum_{S \ni v^t} x_S^{t-1} < 1$ **then**
10: $\quad\quad\quad$ For every set S, update
11: $\quad\quad\quad\quad x_S^t \leftarrow x_S^{t-1} \cdot \exp\left\{\mathbb{1}\{S \ni v^t\} \cdot \frac{\kappa_{v^t}}{c_S}\right\}$.
12: $\quad\quad\quad$ Let $Z^t = \langle c, x^t \rangle / \beta$ and normalize
13: $\quad\quad\quad\quad x^t \leftarrow x^t / Z^t$.
14: $\quad\quad$ **else**
15: $\quad\quad\quad x^t \leftarrow x^{t-1}$.
16: $\quad\quad$ Let S_{v^t} be the cheapest set containing v^t. Add $\mathcal{C}^t \leftarrow \mathcal{C}^t \cup \{S_{v^t}\}$.

The algorithm is somewhat simpler for unit costs: the x_S values are multiplied by either 1 or e, and moreover we can sample a single set for $\mathcal{R}^t$ (see the appendix of the full version for pseudocode). Because of the non-uniform set costs, we have to carefully calibrate both the learning and sampling rates. Our algorithm dynamically scales the learning and sampling rates in round t depending on κ_{v^t}, the cost of the cheapest set covering v^t. Intuitively, this ensures that all three of (a) the change in potential, (b) the cost of the sampling, and (c) the cost of the backup set, are at the same scale. Before we begin, observe that Line 13 ensures the following invariant:

Invariant 1. *For all time steps t, it holds that $\langle c, x^t \rangle = \beta$.*

Theorem III.1 (Main Theorem)**.** *LEARNORCOVER is a polynomial-time randomized algorithm for* ROSETCOVER *with expected cost $O(\beta \cdot \log(mn))$.*

Let us start by defining notation. Let x^* to be the optimal LP solution to the final, unknown set cover instance. Next let $X_v^t := \sum_{S \ni v} x_S^t$, the fractional coverage provided to v by x^t. Let U^t be the elements remaining uncovered at the end of round t (where $U^0 = U$ is the entire ground set of the set system). Define the quantity

$$\rho^t := \sum_{u \in U^t} \kappa_u. \tag{III.2}$$

With this, we are ready to define our potential function which is the central player in our analysis. Recalling that β is our guess for the value of $\mathrm{LP}_{\mathrm{OPT}}$, and the definition of KL divergence (I.1), define

$$\boxed{\Phi(t) := C_1 \cdot \mathrm{KL}_c\left(x^* \mid\mid x^t\right) + C_2 \cdot \beta \cdot \log\left(\frac{\rho^t}{\beta}\right)} \tag{III.3}$$

where C_1 and C_2 are constants to be specified later.

Lemma III.2 (Initial Potential)**.** *The initial potential is bounded as $\Phi(0) = O(\beta \cdot \log(mn))$.*

We write the proof in general language in order to reuse it for covering IPs in the following section. Recall that sets correspond to columns in the canonical formulation of SETCOVER as an integer program.

Proof: We require the following fact which we prove in the appendix.

Fact III.3. *Every pure covering LP of the form $\min_{x \geq 0}\{\langle c, x \rangle : Ax \geq 1\}$ for $c \geq 0$ and $a_{ij} \in [0, 1]$ with optimal value less than β has an optimal solution x^* which is supported only on columns j such that $c_j \leq \beta$.*

We assume WLOG that x^* is such a solution, and we first bound the KL term of $\Phi(0)$. Since $\mathrm{support}(x^*) \subseteq \mathrm{support}(x^0)$ by Fact III.3, we have

$$\begin{aligned}
&\mathrm{KL}_c\left(x^* \mid\mid x^0\right) \\
&= \sum_j c_j \cdot x_j^* \log\left(x_j^* \frac{c_j \cdot m'}{\beta}\right) + \sum_j c_j (x_j^0 - x_j^*) \\
&\leq \beta(\log(m) + 1),
\end{aligned}$$

where we used that $\langle c, x^* \rangle \leq \beta$ and that $m' < m$ is the number of columns with cost less than β.

For the second term,

$$\begin{aligned}
\beta \log(\rho^0/\beta) &= \beta \log\Big(\sum_{i \in U^0} \kappa_i/\beta\Big) \\
&\leq \beta \log(|U^0|) \\
&= \beta \log n,
\end{aligned}$$

since for all i, the cheapest cover for i costs less than β, and therefore $\kappa_u/\beta \leq 1$. The claim follows so long as C_1 and C_2 are constants. ■

The rest of the proof relates the expected decrease of potential Φ to the algorithm's cost in each round. Define the event $\Upsilon^t := \{v^t \in U^{t-1}\}$ that the element v^t is *uncovered* on arrival. Note Line 5 ensures that if event Υ^t does not hold, the algorithm takes no action and the potential does not change. So we focus on the case that event Υ^t does occur. We first analyze the change in KL divergence. Recall that $X_v^t := \sum_{S \ni v} x_S^t$.

Lemma III.4 (Change in KL)**.** *For rounds t in which Υ^t holds, the expected change in the weighted KL divergence is*

$$\begin{aligned}&\mathop{\mathbb{E}}_{v^t,\mathcal{R}^t}\left[\begin{array}{c}\mathrm{KL}_c\left(x^* \,||\, x^t\right)\\ -\;\mathrm{KL}_c\left(x^* \,||\, x^{t-1}\right)\end{array}\,\middle|\, x^{t-1}, U^{t-1}, \Upsilon^t\right]\\ &\le \mathop{\mathbb{E}}_{v\sim U^{t-1}}[(e-1)\cdot \kappa_v \min(X_v^{t-1},1) - \kappa_v].\end{aligned}$$

We emphasize that the expected change in relative entropy in the statement above depends only on the randomness of the arriving uncovered element v^t, not on the randomly chosen sets $\mathcal{R}^t$.

Proof: We break the proof into cases. If $X_v^{t-1} \ge 1$, in Line 15 we set the vector $x^t = x^{t-1}$, so the change in KL divergence is 0. This means that

$$\begin{aligned}&\mathop{\mathbb{E}}_{v^t,\mathcal{R}^t}\left[\begin{array}{c}\mathrm{KL}_c\left(x^* \,||\, x^t\right)\\ -\mathrm{KL}_c\left(x^* \,||\, x^{t-1}\right)\end{array}\,\middle|\, x^{t-1}, U^{t-1}, \Upsilon^t, X_{v^t}^{t-1} \ge 1\right]\\ &\le \mathop{\mathbb{E}}_{v\sim U^t}[(e-1)\cdot \kappa_v \min\left(X_v^{t-1},1\right) - \kappa_v \mid X_v^{t-1} \ge 1]\end{aligned} \tag{III.4}$$

trivially. Henceforth we focus on the case $X_{v^t}^{t-1} < 1$. Recall that the expected change in relative entropy depends only on the arriving uncovered element v^t. Expanding definitions,

$$\begin{aligned}&\mathop{\mathbb{E}}_{v^t,\mathcal{R}^t}\left[\begin{array}{c}\mathrm{KL}_c\left(x^* \,||\, x^t\right)\\ -\mathrm{KL}_c\left(x^* \,||\, x^{t-1}\right)\end{array}\,\middle|\, \begin{array}{c}x^{t-1}, U^{t-1},\\ \Upsilon^t, X_{v^t}^{t-1} < 1\end{array}\right]\\ &= \mathop{\mathbb{E}}_{v\sim U^{t-1}}\left[\sum_S c_S \cdot x_S^* \cdot \log \frac{x_S^{t-1}}{x_S^t}\,\middle|\, X_v^{t-1} < 1\right]\\ &= \mathop{\mathbb{E}}_{v\sim U^{t-1}}\left[\begin{array}{c}\langle c, x^*\rangle \cdot \log Z^t\\ -\sum_{S\ni v} c_S \cdot x_S^* \cdot \log e^{\kappa_v/c_S}\end{array}\,\middle|\, X_v^{t-1} < 1\right]\\ &\le \mathop{\mathbb{E}}_{v\sim U^{t-1}}\left[\begin{array}{l}\beta \log\left(\begin{array}{c}\sum_{S\ni v} \frac{c_S}{\beta} x_S^{t-1} e^{\frac{\kappa_v}{c_S}}\\ +\sum_{S\not\ni v} \frac{c_S}{\beta} x_S^{t-1}\end{array}\right)\\ -\sum_{S\ni v} \kappa_v \cdot x_S^*\end{array}\,\middle|\, X_v^{t-1} < 1\right],\end{aligned} \tag{III.5}$$

where in the last step (III.5) we expanded the definition of Z^t, and used $\langle c, x^*\rangle \le \beta$. Since x^* is a feasible set cover, which means that $\sum_{S\ni v} x_S^* \ge 1$, we can further bound (III.5) by

$$\begin{aligned}&\le \mathop{\mathbb{E}}_{v\sim U^{t-1}}\left[\begin{array}{l}\beta \log\left(\begin{array}{c}\sum_{S\ni v} \frac{c_S}{\beta} x_S^{t-1} e^{\frac{\kappa_v}{c_S}}\\ +\sum_{S\not\ni v} \frac{c_S}{\beta} x_S^{t-1}\end{array}\right)\\ -\kappa_v\end{array}\,\middle|\, X_v^{t-1} < 1\right]\\ &\le \mathop{\mathbb{E}}_{v\sim U^{t-1}}\left[\begin{array}{l}\beta \log\left(\begin{array}{c}\sum_{S} \frac{c_S}{\beta} x_S^{t-1} +\\ (e-1)\sum_{S\ni v} \frac{\kappa_v}{\beta} x_S^{t-1}\end{array}\right)\\ -\kappa_v\end{array}\,\middle|\, X_v^{t-1} < 1\right]\end{aligned} \tag{III.6}$$

where we use the approximation $e^y \le 1 + (e-1)\cdot y$ for $y \in [0,1]$ (note that κ_v is the cheapest set covering v, so for any $S \ni v$ we have $\kappa_v / c_S \le 1$). Finally, using Invariant 1, along with the approximation $\log(1+y) \le y$, we bound (III.6) by

$$\begin{aligned}&\le \mathop{\mathbb{E}}_{v\sim U^{t-1}}\left[(e-1)\cdot \kappa_v \cdot X_v^{t-1} - \kappa_v \,\middle|\, X_v^{t-1} < 1\right]\\ &\le \mathop{\mathbb{E}}_{v\sim U^{t-1}}[(e-1)\cdot \kappa_v \min(X_v^{t-1},1) - \kappa_v \mid X_v^{t-1} < 1].\end{aligned} \tag{III.7}$$

The lemma statement follows by combining (III.4) and (III.7) using the law of total expectation. ■

Next we bound the expected change in $\log \rho^t$ provided by the sampling $\mathcal{R}^t \sim \kappa_{v^t} x^{t-1}/\beta$ upon the arrival of uncovered v^t (where each $S \in \mathcal{R}^t$ independently with probability $\kappa_{v^t} x_S^{t-1}/\beta$). Recall that U^t denotes the elements uncovered at the end of round t; therefore element u is contained in $U^{t-1} \setminus U^t$ if and only if it is marginally covered by $\mathcal{R}^t$ in this round.

Lemma III.5 (Change in $\log \rho^t$)**.** *For rounds when v is uncovered on arrival, the expected change in $\log \rho^t$ is*

$$\begin{aligned}&\mathop{\mathbb{E}}_{v^t,\mathcal{R}^t}\left[\log \rho^t - \log \rho^{t-1} \mid x^{t-1}, U^{t-1}, \Upsilon^t\right]\\ &\le -\frac{1-e^{-1}}{\beta} \cdot \mathop{\mathbb{E}}_{u\sim U^{t-1}}\left[\kappa_u \cdot \min(X_u^{t-1},1)\right].\end{aligned}$$

Proof: Recall the definition of ρ^t from (III.2). Conditioned on $v^t = v$ for any fixed element v, the expected change in $\log \rho^t$ depends only on $\mathcal{R}^t$. Recall $\mathcal{R}^t$ is formed by sampling each set S with probability $\kappa_v x_S^{t-1}/\beta$.

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathcal{R}^t}\left[\log \rho^t - \log \rho^{t-1} \mid x^{t-1}, U^{t-1}, \Upsilon^t, v^t = v\right]\\ &= \mathop{\mathbb{E}}_{\mathcal{R}^t}\left[\log\left(1 - \frac{\rho^{t-1}-\rho^t}{\rho^{t-1}}\right)\,\middle|\, U^{t-1}, v^t = v\right]\\ &\le -\frac{1}{\rho^{t-1}} \cdot \mathop{\mathbb{E}}_{\mathcal{R}^t}\left[\rho^{t-1} - \rho^t \,\middle|\, U^{t-1}, v^t = v\right]\end{aligned} \tag{III.8}$$

Above, (III.8) follows from the approximation $\log(1-y) \le -y$. Expanding the definition of ρ^t from (III.2), (III.8) is bounded by

$$= -\frac{1}{\rho^{t-1}} \mathop{\mathbb{E}}_{\mathcal{R}^t}\left[\sum_{u \in U^{t-1}} \kappa_u \cdot \mathbb{1}\{u \in U^{t-1} \setminus U^t\} \;\middle|\; \begin{matrix} U^{t-1}, \\ v^t = v \end{matrix}\right]$$

$$= -\frac{1}{\rho^{t-1}} \sum_{u \in U^{t-1}} \kappa_u \cdot \mathop{\mathbb{P}}_{\mathcal{R}^t}\left(u \notin U^t \mid u \in U^{t-1}, v^t = v\right)$$

$$\leq -\frac{1 - e^{-1}}{\beta} \cdot \kappa_v \cdot \frac{1}{\rho^{t-1}} \sum_{u \in U^{t-1}} \kappa_u \cdot \min(X_u^{t-1}, 1) \quad \text{(III.9)}$$

$$= -\frac{1 - e^{-1}}{\beta} \cdot \kappa_v \cdot \frac{|U^{t-1}|}{\rho^{t-1}} \cdot \mathop{\mathbb{E}}_{u \sim U^{t-1}}\left[\kappa_u \min(X_u^{t-1}, 1)\right]. \quad \text{(III.10)}$$

Step (III.9) is due to the fact that each set $S \in \mathcal{R}^t$ is sampled independently with probability $\min(\kappa_v x_S^{t-1}/\beta, 1)$, so the probability any given element $u \in U^{t-1}$ is covered is

$$1 - \prod_{S \ni u}\left(1 - \min\left(\frac{\kappa_v x_S^{t-1}}{\beta}, 1\right)\right)$$

$$\geq 1 - \exp\left\{-\min\left(\frac{\kappa_v}{\beta} X_u^{t-1}, 1\right)\right\}$$

$$\overset{(**)}{\geq} (1 - e^{-1}) \cdot \min\left(\frac{\kappa_v}{\beta} X_u^{t-1}, 1\right)$$

Above, $(**)$ follows from convexity of the exponential. Step (III.10) follows since $\kappa_v/\beta \leq 1$. Taking the expectation of (III.10) over $v^t \sim U^{t-1}$, and using the fact that $\mathbb{E}_{v \sim U^{t-1}}[\kappa_v] = \rho^{t-1}/|U^{t-1}|$, the expected change in $\log \rho^t$ becomes

$$\mathop{\mathbb{E}}_{v^t, \mathcal{R}^t}\left[\log \rho^t - \log \rho^{t-1} \mid x^{t-1}, U^{t-1}, \Upsilon^t\right]$$

$$\leq -\frac{1 - e^{-1}}{\beta} \cdot \mathop{\mathbb{E}}_{u \sim U^{t-1}}\left[\kappa_u \cdot \min(X_u^{t-1}, 1)\right],$$

as desired. ■

Proof of Theorem III.1:

In every round t for which Υ^t holds, the expected cost of the sampled sets $\mathcal{R}^t$ is $\kappa_{v^t} \cdot \langle c, x^{t-1}\rangle/\beta = \kappa_{v^t}$ (by Invariant 1). The algorithm pays an additional κ_{v^t} in Line 16, and hence the total expected cost per round is at most $2 \cdot \kappa_{v^t}$.

Combining Lemmas III.4 and III.5, and setting the constants $C_1 = 2$ and $C_2 = 2e$, we have

$$\mathop{\mathbb{E}}_{v^t, \mathcal{R}^t}\left[\Phi(t) - \Phi(t-1) \;\middle|\; \begin{matrix} v^1, \ldots, v^{t-1}, \\ \mathcal{R}^1, \ldots, \mathcal{R}^{t-1}, \Upsilon^t \end{matrix}\right]$$

$$= \mathop{\mathbb{E}}_{v^t, \mathcal{R}^t}\left[\begin{matrix} C_1\begin{pmatrix} \mathrm{KL}_c\left(x^* \,||\, x^t\right) \\ -\mathrm{KL}_c\left(x^* \,||\, x^{t-1}\right)\end{pmatrix} \\ +C_2 \cdot \beta\begin{pmatrix} \log \rho^t \\ -\log \rho^{t-1}\end{pmatrix}\end{matrix} \;\middle|\; \begin{matrix} v^1, \ldots, v^{t-1}, \\ \mathcal{R}^1, \ldots, \mathcal{R}^{t-1}, \\ \Upsilon^t \end{matrix}\right]$$

$$\leq -\mathop{\mathbb{E}}_{v^t, \mathcal{R}^t}\left[2 \cdot \kappa_{v^t} \mid v^1, \ldots, v^{t-1}, \mathcal{R}^1, \ldots, \mathcal{R}^{t-1}, \Upsilon^t\right],$$

which cancels the expected change in the algorithm's cost. Since neither the potential Φ nor the cost paid by the algorithm change during rounds in which Υ^t does not hold, we have the inequality

$$\mathop{\mathbb{E}}_{v^t, \mathcal{R}^t}\left[\begin{matrix} \Phi(t) - \Phi(t-1) \\ +c(\textsc{Alg}(t)) \\ -c(\textsc{Alg}(t-1)) \end{matrix} \;\middle|\; \begin{matrix} v^1, \ldots, v^{t-1}, \\ \mathcal{R}^1, \ldots, \mathcal{R}^{t-1} \end{matrix}\right] \leq 0.$$

Let t^* be the last time step for which $\Phi(t^*) \geq 0$. By applying Lemma A.1 and the bound on the starting potential from Lemma III.2, we have that $\mathbb{E}[c(\textsc{Alg}(t^*))] \leq O(\beta \cdot \log(mn))$.

It remains to bound the expected cost paid by the algorithm after time t^*. Since KL divergence is a nonnegative quantity, Φ is negative only when $\rho^t \leq \beta$. The algorithm pays $O(\kappa_{v^t})$ in expectation during rounds t where $v^t \in U^{t-1}$ and 0 during rounds where $v^t \notin U^{t-1}$, and hence the expected cost paid by the algorithm after time t^* is at most $\sum_{u \in U^{t^*}} \kappa_u = \rho^{t^*} = O(\beta)$. ■

IV. Covering Integer Programs

We show how to generalize our algorithm from Section III to solve pure covering IPs when the constraints are revealed in random order, which significantly generalizes ROSetCover. Formally, the random order covering IP problem (ROCIP) is to solve

$$\begin{array}{ll} \min_z & \langle c, z\rangle \\ \text{s.t.} & Az \geq 1 \\ & z \in \mathbb{Z}_+^m, \end{array} \quad \text{(IV.1)}$$

when the rows of A are revealed in random order. Furthermore the solution z can only be incremented monotonically and must always be feasible for the subset of constraints revealed so far. (Note that we do not consider box constraints, namely upper-bound constraints of the form $z_j \leq d_j$.) We may assume without loss of generality that the entries of A are $a_{ij} \in [0, 1]$.

We describe an algorithm which guarantees that every row is covered to extent $1 - \gamma$, meaning it outputs a solution z with $Az \geq 1 - \gamma$ (this relaxation is convenient in the proof for technical reasons). With foresight, we set $\gamma = (e-1)^{-1}$. It is straightforward to wrap this algorithm in one that buys $\lceil (1-\gamma)^{-1} \rceil = 3$ copies of every column and truly satisfy the constraints, which only incurs an additional factor of 3 in the cost.

Once again, by a guess-and-double approach, we assume we know a bound β such that $\mathrm{LP}_{\mathrm{OPT}} \leq \beta \leq 2 \cdot \mathrm{LP}_{\mathrm{OPT}}$; here $\mathrm{LP}_{\mathrm{OPT}}$ is the cost of the optimal solution to the LP relaxation of (IV.1). Let z^t be the integer solution held by the algorithm at the end of round t. Define $\Delta_i^t := \max(0, 1 - \langle a_i, z^t\rangle)$ to be the extent to which i remains uncovered at the end of round t. This time we redefine $\kappa_i^t := \Delta_i^{t-1} \cdot \min_k c_k/a_{ik}$, which becomes the minimum fractional cost of covering the current deficit for i. Finally, for a vector y, denote the fractional remainder by $\widetilde{y} := y - \lfloor y \rfloor$.

The algorithm once again maintains a fractional vector x which is a guess for the (potentially infeasible) LP solution of cost β to (IV.1). This time, when the i^{th} row arrives at time t and $\Delta_i^{t-1} > \gamma$ (meaning this row is *not* already covered to extent $1-\gamma$), we (a) buy a random number of copies of every column j with probability proportional to its LP value x_j, (b) increase the value x_j multiplicatively and renormalize, and finally (c) buy a minimum cost cover for row i if necessary.

Algorithm 2 LEARNORCOVERCIP

1: $m' \leftarrow |\{j : c_j \leq \beta\}|$.
2: Initialize $x_j^0 \leftarrow \frac{\beta}{c_j \cdot m'} \cdot \mathbb{1}\{c_j \leq \beta\}$, and $z^0 \leftarrow \vec{0}$.
3: **for** $t = 1, 2 \ldots, n$ **do**
4: $\quad i \leftarrow t^{th}$ constraint in the random order.
5: $\quad$ **if** $\Delta_i^{t-1} > \gamma$ **then**
6: $\quad\quad$ Let $y := \kappa_i^t \cdot x^{t-1}/\beta$. **for each** column j, add $z_j^t \leftarrow z_j^{t-1} + \lfloor y_j \rfloor + \text{Ber}(\tilde{y}_j)$.
7: $\quad\quad$ **if** $\langle a_i, x^{t-1} \rangle < \Delta_i^{t-1}$ **then**
8: $\quad\quad\quad$ For every j, update
9: $\quad\quad\quad\quad x_j^t \leftarrow x_j^{t-1} \cdot \exp\left\{\kappa_i^t \cdot \frac{a_{ij}}{c_j}\right\}$.
10: $\quad\quad\quad$ Let $Z^t = \langle c, x^t \rangle / \beta$ and normalize
11: $\quad\quad\quad\quad x^t \leftarrow x^t / Z^t$.
12: $\quad\quad$ **else**
13: $\quad\quad\quad x^t \leftarrow x^{t-1}$
14: $\quad\quad$ Let $k^* = \text{argmin}_k \frac{c_k}{a_{ik}}$. Add
15: $\quad\quad\quad z_{k^*}^t \leftarrow z_{k^*}^t + \left\lceil \frac{\Delta_i^{t-1}}{a_{ik^*}} \right\rceil$.

Note that once again, Line 11 ensures Invariant 1 holds. The main theorem of this section is:

Theorem IV.1. LEARNORCOVERCIP *is a polynomial-time randomized algorithm for* ROCIP *which outputs a solution z with expected cost $O(\beta \cdot \log(mn))$ such that $3z$ is feasible.*

Theorem I.2 follows as a corollary, since given any intermediate solution z^t, we can buy the scaled solution $3z^t$.

We generalize the proof of Theorem III.1. Redefine x^* to be the optimal LP solution to the final, unknown instance (IV.1), and $U^t := \{i \mid \Delta_i^t > \gamma\}$ be the elements which are not covered to extent $1-\gamma$ at the end of round t. With these new versions of U^t and κ^t, the definitions of both ρ^t and the potential Φ remain the same as in (III.2) and (III.3) (except we pick the constants C_1 and C_2 differently).

Once again, we start with a bound on the initial potential.

Lemma IV.2 (Initial Potential)**.** *The initial potential is bounded as $\Phi(0) = O(\beta \cdot \log(mn))$.*

The proof is identical verbatim to that of Lemma III.2.

It remains to relate the expected decrease in Φ to the algorithm's cost in every round. For convenience, let $X_i^t := \langle a_i, x^t \rangle$ be the amount that x^t fractionally covers i. Define Υ^t to be the event that for constraint i^t arriving in round t we have $\Delta_i^{t-1} > \gamma$. The check at Line 5 ensures that if Υ^t does not hold, then neither the cost paid by the algorithm nor the potential will change. We focus on the case that Υ^t occurs, and once again start with the KL divergence.

Lemma IV.3 (Change in KL)**.** *For rounds in which Υ^t holds, the expected change in weighted KL divergence is*

$$\mathop{\mathbb{E}}_{i^t, \mathcal{R}^t}\left[\begin{array}{c} \text{KL}_c\left(x^* \mid\mid x^t\right) \\ -\text{KL}_c\left(x^* \mid\mid x^{t-1}\right)\end{array} \;\middle|\; x^{t-1}, U^{t-1}, \Upsilon^t\right] \leq \mathop{\mathbb{E}}_{i \sim U^{t-1}}[(e-1) \cdot \kappa_i^t \min(X_i^{t-1}, \Delta_i^{t-1}) - \kappa_i^t].$$

Proof: We break the proof into cases. By the check on Line 7, if $X_{i^t}^{t-1} \geq \Delta_{i^t}^{t-1}$, then the vector x^t is not updated in round t, so the change in KL divergence is 0. This means that

$$\mathop{\mathbb{E}}_{i^t, \mathcal{R}^t}\left[\begin{array}{c} \text{KL}_c\left(x^* \mid\mid x^t\right) \\ -\text{KL}_c\left(x^* \mid\mid x^{t-1}\right)\end{array} \;\middle|\; \begin{array}{l} x^{t-1}, \\ U^{t-1}, \\ \Upsilon^t, \\ X_{i^t}^{t-1} \geq \Delta_{i^t}^{t-1} \end{array}\right] \leq \mathop{\mathbb{E}}_{i \sim U^{t-1}}\left[\begin{array}{l}(e-1)\kappa_i^t \min(X_i^{t-1}, \Delta_i^{t-1}) \\ -\kappa_i^t \end{array} \;\middle|\; X_i^{t-1} \geq \Delta_i^{t-1}\right] \tag{IV.2}$$

holds trivially, since in this case $\min(X_i^{t-1}, \Delta_i^{t-1}) = \Delta_i^{t-1} > \gamma = (e-1)^{-1}$ and $\kappa_i^t \geq 0$. Henceforth we focus on the case $X_i^{t-1} < \Delta_i^{t-1}$.

The change in relative entropy depends only on the arriving uncovered constraint i^t, not on the randomly chosen columns $\mathcal{R}^t$. Expanding definitions,

$$\mathop{\mathbb{E}}_{i^t, \mathcal{R}^t}\left[\begin{array}{c} \text{KL}_c\left(x^* \mid\mid x^t\right) \\ -\text{KL}_c\left(x^* \mid\mid x^{t-1}\right)\end{array} \;\middle|\; \begin{array}{l} x^{t-1}, \\ U^{t-1}, \\ \Upsilon^t, \\ X_{i^t}^{t-1} < \Delta_{i^t}^{t-1} \end{array}\right] \tag{IV.3}$$

$$= \mathop{\mathbb{E}}_{i \sim U^{t-1}}\left[\sum_j c_j \cdot x_j^* \cdot \log \frac{x_j^{t-1}}{x_j^t} \;\middle|\; X_i^{t-1} < \Delta_i^{t-1}\right]$$

$$= \mathop{\mathbb{E}}_{i \sim U^{t-1}}\left[\begin{array}{c} \sum_j c_j \cdot x_j^* \cdot \log Z^t \\ -\sum_j c_j \cdot x_j^* \cdot \kappa_i^t \cdot \frac{a_{ij}}{c_j} \end{array} \;\middle|\; X_i^{t-1} < \Delta_i^{t-1}\right]$$

$$\leq \mathop{\mathbb{E}}_{i \sim U^{t-1}}\left[\beta \cdot \log Z^t - \kappa_i^t \cdot \sum_j a_{ij} x_j^* \;\middle|\; X_i^{t-1} < \Delta_i^{t-1}\right] \tag{IV.4}$$

where we used that $\langle c, x^* \rangle \leq \beta$. Expanding the definition of Z^t and applying the fact that x^* is a feasible solution i.e.

$\langle a_i, x^* \rangle \geq 1$, we continue to bound (IV.4) as

$$\leq \mathop{\mathbb{E}}_{i \sim U^{t-1}} \left[\begin{array}{l} \beta \log \left(\frac{1}{\beta} \sum_j c_j x_j^{t-1} e^{\kappa_i^t \cdot \frac{a_{ij}}{c_j}} \right) \\ -\kappa_i^t \end{array} \middle| \begin{array}{l} X_i^{t-1} \\ < \Delta_i^{t-1} \end{array} \right]$$
$$\leq \mathop{\mathbb{E}}_{i \sim U^{t-1}} \left[\begin{array}{l} \beta \log \left(1 + \frac{e-1}{\beta} \kappa_i^t \sum_j a_{ij} x_j^{t-1} \right) \\ -\kappa_i^t \end{array} \middle| \begin{array}{l} X_i^{t-1} \\ < \Delta_i^{t-1} \end{array} \right]. \tag{IV.5}$$

(IV.5) is derived by applying the approximation $e^y \leq 1 + (e-1)y$ for $y \in [0,1]$ and the fact that $\langle c, x^{t-1} \rangle = \beta$ by Invariant 1; the exponent lies in $[0,1]$ because by definition $\kappa_i^t \cdot a_{ij}/c_j = \Delta_i^{t-1} \cdot (a_{ij}/c_j) \cdot \min_k (c_k/a_{ik}) \leq 1$ since $\Delta_i^{t-1} \in [0,1]$. Finally, using the fact that $\log(1+y) \leq y$, we have that (IV.5) is at most

$$\leq \mathop{\mathbb{E}}_{i \sim U^{t-1}} \left[(e-1)\kappa_i^t \sum_j a_{ij} x_j^{t-1} - \kappa_i^t \middle| \begin{array}{l} X_i^{t-1} \\ < \Delta_i^{t-1} \end{array} \right]$$
$$\leq \mathop{\mathbb{E}}_{i \sim U^{t-1}} \left[\begin{array}{l} (e-1)\kappa_i^t \cdot \min\left(X_i^{t-1}, \Delta_i^{t-1}\right) \\ -\kappa_i^t \end{array} \middle| \begin{array}{l} X_i^{t-1} \\ < \Delta_i^{t-1} \end{array} \right]. \tag{IV.6}$$

The lemma statement follows by combining (IV.2) and (IV.6) using the law of total expectation. ■

We move to bounding the expected change in $\log \rho^t$ provided by updating the solution z on Line 6 on the arrival of the random row i. Recall that $U^t = \{i \mid \Delta_i^t > \gamma\}$ are the unseen elements which are at most half covered by z.

We will make use of the following lemma, which we prove in the appendix:

Fact IV.4. *Given probabilities p_j and coefficients $b_j \in [0,1]$, let $W := \sum_j b_j \operatorname{Ber}(p_j)$ be the sum of independent weighted Bernoulli random variables. Let $\Delta \geq \gamma = (e-1)^{-1}$ be some constant. Then*

$$\mathbb{E}[\min(W, \Delta)] \geq \alpha \cdot \min(\mathbb{E}[W], \Delta),$$

for a fixed constant α independent of the p_j and b_j.

We are ready to bound the expected change in $\log \rho^t$.

Lemma IV.5 (Change in $\log \rho^t$)**.** *For rounds in which Υ^t holds, the expected change in $\log \rho^t$ is*

$$\mathop{\mathbb{E}}_{i^t, \mathcal{R}^t} \left[\log \rho^t - \log \rho^{t-1} \mid x^{t-1}, U^{t-1}, \Upsilon^t \right]$$
$$\leq -\frac{\alpha}{\beta} \cdot \mathop{\mathbb{E}}_{i' \sim U^{t-1}} \left[\kappa_{i'}^t \cdot \min\left(X_{i'}^{t-1}, \Delta_{i'}^{t-1}\right) \right]$$

where α is a fixed constant.

Proof: Conditioned on $i^t = i$, the expected change in $\log \rho^t$ depends only on $\mathcal{R}^t$.

$$\mathop{\mathbb{E}}_{i^t, \mathcal{R}^t} \left[\log \rho^t - \log \rho^{t-1} \mid x^{t-1}, U^{t-1}, \Upsilon^t, i^t = i \right]$$
$$= \mathop{\mathbb{E}}_{\mathcal{R}^t} \left[\log \left(1 - \frac{\rho^{t-1} - \rho^t}{\rho^{t-1}} \right) \middle| U^{t-1}, i^t = i \right]$$
$$\leq -\frac{1}{\rho^{t-1}} \mathop{\mathbb{E}}_{\mathcal{R}^t} \left[\rho^{t-1} - \rho^t \mid U^{t-1}, i^t = i \right]. \tag{IV.7}$$

Above, follows from the approximation $\log(1-y) \leq -y$. Expanding definitions again, and using the fact that

$$\kappa_{i'}^{t-1} - \kappa_{i'}^t = \min_k (c_k/a_{i'k}) \cdot (\Delta_{i'}^{t-1} - \Delta_{i'}^t)$$
$$\geq \kappa_{i'}^t (\Delta_{i'}^{t-1} - \Delta_{i'}^t),$$

we further bound (IV.7) by

$$\leq -\frac{1}{\rho^{t-1}} \mathop{\mathbb{E}}_{\mathcal{R}^t} \left[\sum_{i' \in U^{t-1}} \kappa_{i'}^t \cdot (\Delta_{i'}^{t-1} - \Delta_{i'}^t) \right]$$
$$= -\frac{1}{\rho^{t-1}} \sum_{i' \in U^{t-1}} \kappa_{i'}^t \cdot \mathop{\mathbb{E}}_{\mathcal{R}^t} [\Delta_{i'}^{t-1} - \Delta_{i'}^t]$$
$$= -\frac{1}{\rho^{t-1}} \sum_{i' \in U^{t-1}} \kappa_{i'}^t \cdot \alpha \min\left(\frac{\kappa_i^t}{\beta} X_{i'}^{t-1}, \Delta_{i'}^{t-1} \right). \tag{IV.8}$$

To understand this last step (IV.8), note that

$$\Delta_{i'}^{t-1} - \Delta_{i'}^t$$
$$= \min\left(\sum_j a_{i'j} \lfloor y_j \rfloor + \sum_j a_{i'j} \operatorname{Ber}(\widetilde{y}),\ \Delta_{i'}^{t-1}\right).$$

By the definition of y, the first term inside the minimum has expectation $\frac{\kappa_i^t}{\beta} \cdot X_{i'}^{t-1}$, and since Υ^t holds we have $\Delta_{i'}^{t-1} > \gamma$. Therefore applying Fact IV.4 gives (IV.8) (where α is the constant given by the lemma). Since $\kappa_i^t/\beta \leq 1$, we bound (IV.8) with

$$\leq -\frac{\alpha}{\beta} \cdot \kappa_i^t \cdot \frac{1}{\rho^{t-1}} \sum_{i' \in U^{t-1}} \kappa_{i'}^t \cdot \min\left(X_{i'}^{t-1}, \Delta_{i'}^{t-1}\right)$$
$$= -\frac{\alpha}{\beta} \cdot \kappa_i^t \cdot \frac{|U^{t-1}|}{\rho^{t-1}} \cdot \mathop{\mathbb{E}}_{i' \sim U^{t-1}} \left[\kappa_{i'}^t \cdot \min\left(X_{i'}^{t-1}, \Delta_{i'}^{t-1}\right) \right]. \tag{IV.9}$$

Taking the expectation of (IV.9) over $i \sim U^{t-1}$, and using the fact that $\mathbb{E}_{i \sim U^{t-1}}[\kappa_i^t] = \rho^{t-1}/U^{t-1}$, the expected change in $\log \rho^t$ becomes

$$\mathop{\mathbb{E}}_{i^t, \mathcal{R}^t} \left[\log \rho^t - \log \rho^{t-1} \mid x^{t-1}, U^{t-1}, \Upsilon^t \right]$$
$$\leq -\frac{\alpha}{\beta} \cdot \mathop{\mathbb{E}}_{i' \sim U^{t-1}} \left[\kappa_{i'}^t \cdot \min\left(X_{i'}^{t-1}, \Delta_{i'}^{t-1}\right) \right],$$

as desired. ■

We may now combine the two previous lemmas as before.

Proof of Theorem IV.1:

In the round in which constraint i arrives, the expected cost of sampling is $\frac{\kappa_i^t}{\beta}\langle c, x^{t-1}\rangle = \kappa_i^t$ (by Invariant 1). The algorithm pays an additional

$$c_{k^*}\left\lceil \frac{\Delta_i^{t-1}}{a_{ik^*}} \right\rceil = c_{k^*}\left\lceil \frac{\kappa_i^t}{c_{k^*}} \right\rceil \leq 2\kappa_i^t$$

in Line 16, where this upper bound holds because $\frac{\kappa_i^t}{c_{k^*}} = \frac{\Delta_i^{t-1}}{a_{ik^*}} \geq 1/2$, since $\Delta_i^{t-1} \geq \gamma \geq 1/2$ and $a_{ik^*} \leq 1$. Hence the total expected cost per round is at most $3 \cdot \kappa_i^t$.

Combining Lemma IV.3 and Lemma IV.5 and choosing $C_1 = 3$ and $C_2 = 3(e-1)/\alpha$, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{i^t,\mathcal{R}^t}\left[\Phi(t) - \Phi(t-1) \mid i^1,\ldots,i^{t-1},\mathcal{R}^1,\ldots,\mathcal{R}^{t-1},\Upsilon^t\right] \\
&= \mathop{\mathbb{E}}_{i^t,\mathcal{R}^t}\left[\begin{array}{c|c} C_1\begin{pmatrix} \mathrm{KL}_c\left(x^* \mid\mid x^t\right) \\ -\mathrm{KL}_c\left(x^* \mid\mid x^{t-1}\right)\end{pmatrix} & i^1,\ldots,i^{t-1}, \\ & \mathcal{R}^1,\ldots,\mathcal{R}^{t-1}, \\ +C_2\cdot\beta\begin{pmatrix} \log\rho^t \\ -\log\rho^{t-1}\end{pmatrix} & \Upsilon^t \end{array}\right] \\
&\leq -\mathop{\mathbb{E}}_{i^t,\mathcal{R}^t}\left[3\cdot\kappa_{i^t}^t \mid i^1,\ldots,i^{t-1},\mathcal{R}^1,\ldots,\mathcal{R}^{t-1},\Upsilon^t\right],
\end{aligned}$$

which cancels the expected change in the algorithm's cost. Hence we have the inequality

$$\mathop{\mathbb{E}}_{i^t,\mathcal{R}^t}\left[\begin{array}{c|c} \Phi(t)-\Phi(t-1) & \\ +c(\textsc{Alg}(t)) & i^1,\ldots,i^{t-1}, \\ -c(\textsc{Alg}(t-1)) & \mathcal{R}^1,\ldots,\mathcal{R}^{t-1}\end{array}\right] \leq 0.$$

Let t^* be the last time step for which $\Phi(t^*) \geq 0$. By applying Lemma A.1 and the bound on the starting potential, $\Phi(0) = O(\beta \cdot \log(mn))$, we have that $\mathbb{E}[c(\textsc{Alg}(t^*))] \leq O(\beta \cdot \log(mn))$.

To bound the expected cost of the algorithm after time t^*, note that as before that KL divergence is a nonnegative quantity, and Φ is negative only when $\rho^t \leq \beta$. The algorithm pays $O(\kappa_i^t)$ in expectation during rounds t where $i \in U^{t-1}$ and 0 during rounds where $i \notin U^{t-1}$, and hence the expected cost paid by the algorithm after time t^* is at most $\sum_{i\in U^{t^*}} \kappa_i^{t^*} = \rho^{t^*} = O(\beta)$. ∎

V. Lower Bounds

We turn to showing lower bounds for SetCover and related problems in the random order model. The lower bounds for ROSetCover are proven via basic probabilistic and combinatorial arguments. We also show hardness for a batched version of ROSetCover, which has implications for related problems. We only state the theorems here; see the full version for the complete proofs.

A. Lower Bounds for ROSetCover

We start with information theoretic lower bounds for the RO setting.

Theorem V.1. *The competitive ratio of any randomized fractional or integral algorithm for* ROSetCover *is* $\Omega(\log n)$*.*

We emphasize that the set system in the construction for Theorem V.1 has a VC dimension of 2, which rules out improved algorithms for set systems of small VC dimension in this setting.

Theorem V.2. *The competitive ratio of any randomized algorithm for* ROSetCover *is* $\Omega\left(\frac{\log m}{\log\log m}\right)$ *even when* $m \gg n$*.*

Claim V.3. *There are instances on which [9] is* $\Omega(\log m \log n)$*-competitive in RO.*

B. Lower Bounds for Extensions

We study lower bounds for several extensions of ROSetCover. Our starting point is a lower bound for the batched version of the problem. Here the input is specified by a set system $(U, \mathcal{S})$ as before, along with a partition of U into batches $B_1, B_2, \ldots B_b$. For simplicity, we assume all batches have the same size s. The batches are revealed one-by-one in their entirety to the algorithm, in uniform random order. After the arrival of a batch, the algorithm must select sets to buy to cover all the elements of the batch.

Using this lower bound, we derive as a corollary a lower bound for the random order version of SubmodularCover defined in [19]. It is tempting to use the method of Theorem III.1 to improve their competitive ratio of $O(\log n \log(t \cdot f(N)/f_{\min}))$ in RO (we refer the reader to [19] for the definitions of these parameters). We show that removing a log from the bound is not possible in general.

Theorem V.4. *The competitive ratio of any polynomial-time randomized algorithm on batched* ROSetCover *with* b *batches of size* s *is* $\Omega(\log b \log s)$ *unless* NP $\subseteq$ BPP.

Corollary V.5. *The competitive ratio of any polynomial-time randomized algorithm against* ROSubmodularCover *is* $\Omega(\log n \cdot \log(f(N)/f_{\min}))$ *unless* NP $\subseteq$ BPP.

Proof: Batched ROSetCover is a special case of online SubmodularCover in which f_i is the coverage function of block i. In this case the parameter $f(N)/f_{\min} = s$, so the statement follows by applying Theorem V.4 with $b = s = \sqrt{n}$. ∎

VI. Conclusion

In this work we introduce LearnOrCover as a method for solving ROSetCover and ROCIP with competitive ratio nearly matching the best possible offline bounds. On the other hand we prove nearly tight *information theoretic* lower bounds in the RO setting. We also show lower bounds and separations for several generalizations of ROSETCOVER. We leave as an interesting open question whether it is possible to extend the technique to covering IPs *with box constraints*. We hope our method finds uses elsewhere in online algorithms for RO settings.

Acknowledgements: Roie Levin would like to thank Vidur Joshi for asking what is known about online set cover in random order while on the subway in NYC, as well as David Wajc and Ainesh Bakshi for helpful discussions.

REFERENCES

[1] S. Agrawal and N. R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the 2015 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1405–1424, 2015.

[2] S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

[3] S. Albers and M. Janke. Scheduling in the random-order model. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[4] S. Albers and M. Janke. Scheduling in the secretary model. *CoRR*, abs/2103.16340, 2021.

[5] S. Albers, A. Khan, and L. Ladewig. Improved online algorithms for knapsack and gap in the random order model. *Algorithmica*, 83(6):1750–1785, 2021.

[6] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms (TALG)*, 2(4):640–660, 2006.

[7] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.

[8] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[9] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing. *Mathematics of Operations Research*, 34(2):270–286, 2009.

[10] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.

[11] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[12] S. Dehghani, S. Ehsani, M. Hajiaghayi, V. Liaghat, and S. Seddighin. Greedy algorithms for online survivable network design. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[13] E. D. Demaine, P. Indyk, S. Mahabadi, and A. Vakilian. On streaming and communication complexity of the set cover problem. In *International Symposium on Distributed Computing*, pages 484–498. Springer, 2014.

[14] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 624–633, New York, NY, USA, 2014. ACM.

[15] Y. Emek and A. Rosén. Semi-streaming set cover. *ACM Transactions on Algorithms (TALG)*, 13(1):1–22, 2016.

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

[17] F. Grandoni, A. Gupta, S. Leonardi, P. Miettinen, P. Sankowski, and M. Singh. Set covering with our eyes closed. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 347–356. IEEE, 2008.

[18] A. Gupta, G. Kehne, and R. Levin. Random order set cover is as easy as offline. *arXiv preprint arXiv:2111.06842*, 2021.

[19] A. Gupta and R. Levin. The online submodular cover problem. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1525–1537. SIAM, 2020.

[20] A. Gupta and M. Molinaro. How the experts algorithm can help solve lps online. *Mathematics of Operations Research*, 41(4):1404–1431, 2016.

[21] A. Gupta and V. Nagarajan. Approximating sparse covering integer programs online. *Mathematics of Operations Research*, 39(4):998–1011, 2014.

[22] A. Gupta and S. Singla. Random-order models. *Beyond the Worst-Case Analysis of Algorithms*, page 234, 2020.

[23] S. Har-Peled, P. Indyk, S. Mahabadi, and A. Vakilian. Towards tight bounds for the streaming set cover problem. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 371–383, 2016.

[24] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.

[25] T. Kesselheim, A. Tönnis, K. Radke, and B. Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, pages 303–312, 2014.

[26] S. Korman. On the use of randomization in the online set cover problem. *Master's thesis, Weizmann Institute of Science, Rehovot, Israel*, 2004.

[27] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975.

[28] A. Meyerson. Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 426–431. IEEE, 2001.

[29] A. Meyerson, K. Munagala, and S. Plotkin. Designing networks incrementally. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 406–415. IEEE, 2001.

[30] M. Molinaro. Online and random-order load balancing simultaneously. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1638–1650, 2017.

[31] M. Molinaro and R. Ravi. The geometry of online packing linear programs. *Mathematics of Operations Research*, 39(1):46–59, 2014.

[32] B. Saha and L. Getoor. On maximum coverage in the streaming model & application to multi-topic blog-watch. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 697–708. SIAM, 2009.

[33] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

APPENDIX

In this paper we use several potential function arguments, and the following simple lemma.

Lemma A.1 (Expected Potential Change Lemma)**.** *Let* $\textsc{Alg}$ *be a randomized algorithm for* $\textsc{ROSetCover}$ *and let* Φ *be a potential which is a function of the state of the algorithm at time* t*. Let* $c(\textsc{Alg}(t))$ *be the cost paid by the algorithm up to and including time* t*. Let* $\mathcal{R}^t$ *and* v^t *respectively be the random variables that are the random decisions made by the algorithm in time* t*, and the random element that arrives in time* t*. Suppose that for all rounds* t *in which the algorithm has not covered the entire ground set at the beginning of the round, the inequality*

$$\mathop{\mathbb{E}}_{v^t,\mathcal{R}^t}\left[\begin{array}{l}\Phi(t)-\Phi(t-1)\\ +c(\textsc{Alg}(t))\\ -c(\textsc{Alg}(t-1))\end{array}\;\middle|\;\begin{array}{l}v^1,\ldots,v^{t-1},\\ \mathcal{R}^1,\ldots,\mathcal{R}^{t-1}\end{array}\right]\le 0.$$

holds. Let t^* *be the last time time step such that* $\Phi(t^*)\ge 0$*. Then the expected cost of the algorithm,* $c(\textsc{Alg}(t^*))$ *can be bounded by*

$$\mathbb{E}[c(\textsc{Alg}(t^*))]\le\Phi(0).$$

Proof: Let $\mathcal{T}$ be the set of rounds for which $\Phi>0$ at the beginning of the round.

Define the stochastic process:

$$X^t := \begin{cases}\Phi(t)+c(\textsc{Alg}(t)) & \text{if } t\in\mathcal{T},\\ X^{t-1} & \text{otherwise.}\end{cases}$$

By assumption, this is a supermartingale with respect to $((v^t,\mathcal{R}^t))_t$; that is,

$$\mathop{\mathbb{E}}_{v^t,\mathcal{R}^t}\left[X^{t+1}\mid v^1,\ldots,v^{t-1},\mathcal{R}^1,\ldots,\mathcal{R}^{t-1}\right]\le X^t$$

for all t. By induction we have that for all $t>0$

$$\mathop{\mathbb{E}}_{\substack{v^1,\ldots,v^t\\ \mathcal{R}^1,\ldots,\mathcal{R}^t}}[X^t]\le X^0,$$

so in particular,

$$\mathbb{E}[\Phi(t^*)]+\mathbb{E}[c(\textsc{Alg}(t^*))]\le\Phi(0).$$

The claim follows since the leftmost term is nonnegative by assumption. ∎

Fact III.3. *Every pure covering LP of the form* $\min_{x\ge 0}\{\langle c,x\rangle : Ax\ge 1\}$ *for* $c\ge 0$ *and* $a_{ij}\in[0,1]$ *with optimal value less than* β *has an optimal solution* x^* *which is supported only on columns* j *such that* $c_j\le\beta$*.*

Proof: Suppose otherwise and let x^* be an optimal LP solution. Let j' be a coordinate for which $c_{j'}>\beta$ and $x^*_{j'}>0$. Then define the vector x' by

$$x'_j = \begin{cases}0 & \text{if } j=j'\\ \frac{x^*_j}{1-x'^*_j} & \text{otherwise,}\end{cases}$$

First of all, note that $x^*_{j'}<1$ since $x^*_{j'}c_{j'}\le\sum_j c_jx^*_j\le\beta<c_{j'}$, and so $x'\ge 0$. To see that x' is feasible for each constraint $\langle a_i,x\rangle\ge 1$,

$$\langle a_i,x^*\rangle=(1-x^*_{j'})\langle a_i,x'\rangle+a_{ij'}x^*_{j'}\ge 1$$

so

$$\langle a_i,x'\rangle\ge\frac{1}{1-x^*_{j'}}(1-a_{ij'}x^*_{j'})\ge 1,$$

since $a_{ij'}\in[0,1]$. Finally, observe that x' costs strictly less than x^*, since

$$\begin{aligned}\langle c,x'\rangle &= \frac{\langle c,x^*\rangle - x^*_{j'}c_{j'}}{1-x^*_{j'}}\\ &< \frac{\langle c,x^*\rangle - x^*_{j'}\langle c,x^*\rangle}{1-x^*_{j'}}\\ &= \langle c,x^*\rangle.\end{aligned}$$

This contradicts the optimality of x^*, and so the claim holds. ■

Fact IV.4. *Given probabilities p_j and coefficients $b_j \in [0,1]$, let $W := \sum_j b_j \operatorname{Ber}(p_j)$ be the sum of independent weighted Bernoulli random variables. Let $\Delta \geq \gamma = (e-1)^{-1}$ be some constant. Then*

$$\mathbb{E}[\min(W, \Delta)] \geq \alpha \cdot \min(\mathbb{E}[W], \Delta),$$

for a fixed constant α independent of the p_j and b_j.

Proof: We first consider the case when $\mathbb{E}[W] \geq \Delta/3$. By the Paley-Zygmund inequality, noting that $\mathbb{E}[W] = \sum_j b_j p_j$ and so $\sigma^2 = \sum_j p_j(1-p_j)b_j^2 \leq \mathbb{E}[W]$, we have

$$\begin{aligned}
\mathbb{P}(W \geq \Delta/6) &\geq \mathbb{P}(W \geq \mathbb{E}[W]/2) \\
&\geq \frac{1}{4} \cdot \frac{\mathbb{E}[W]^2}{\mathbb{E}[W]^2 + \sigma^2} \\
&\geq \frac{1}{4} \cdot \frac{\mathbb{E}[W]}{1 + \mathbb{E}[W]} \\
&\geq \frac{1}{4} \cdot \frac{\Delta/3}{1 + \Delta/3} \\
&\geq 1/28,
\end{aligned}$$

since $\Delta \geq \gamma \geq 1/2$ by assumption. This implies the claim because in this case

$$\begin{aligned}
\mathbb{E}[\min(W, \Delta)] &\geq \frac{\Delta}{6} \cdot \mathbb{P}(W \geq \Delta/6) \\
&\geq \frac{\Delta}{168} \\
&\geq \frac{1}{168} \cdot \min(\mathbb{E}[W], \Delta).
\end{aligned}$$

Otherwise $\mathbb{E}[W] < \Delta/3$. Let $\mathcal{R}$ denote the random subset of j for which $\operatorname{Ber}(p_j) = 1$ in a given realization of W, and let $\mathcal{K}$ be the random subset of the j which is output by the $(1/3, 1/3)$-contention resolution scheme for knapsack constraints when given $\mathcal{R}$ as input, as defined in [10, Lemma 4.15]. The set $\mathcal{K}$ has the properties that (1) (over the randomness in $\mathcal{R}$) every j appears in $\mathcal{K}$ with probability at least $p_j/3$, (2) $\sum_{j \in \mathcal{K}} a_{ij} < \Delta$, and (3) $\mathcal{K} \subseteq \mathcal{R}$. Hence in this case

$$\begin{aligned}
\mathbb{E}[\min(W, \Delta)] &\geq \mathop{\mathbb{E}}_{\mathcal{K}}\left[\sum_{j \in \mathcal{K}} a_{ij}\right] \\
&\geq \frac{\mathbb{E}[W]}{3} = \frac{1}{3} \cdot \min(\mathbb{E}[W], \Delta).
\end{aligned}$$

Therefore the claim holds with $\alpha = 1/168$. ■