SET COVERING WITH OUR EYES CLOSED*

FABRIZIO GRANDONI†, ANUPAM GUPTA‡, STEFANO LEONARDI§, PAULI MIETTINEN¶, PIOTR SANKOWSKI§ ‖, AND MOHIT SINGH**

Abstract. Given a universe U of n elements and a weighted collection $\mathscr{S}$ of m subsets of U, the *universal set cover* problem is to a-priori map each element $u \in U$ to a set $\mathbf{S}(u) \in \mathscr{S}$ containing u, such that any set $X \subseteq U$ is covered by $\mathbf{S}(X) = \cup_{u \in X} \mathbf{S}(u)$. The aim is to find a mapping such that the cost of $\mathbf{S}(X)$ is as close as possible to the optimal set-cover cost for X. (Such problems are also called *oblivious* or *a-priori* optimization problems.) Unfortunately, for every universal mapping, the cost of $\mathbf{S}(X)$ can be $\Omega(\sqrt{n})$ times larger than optimal if the set X is adversarially chosen.

In this paper we study the *performance on average*, when X is a set of randomly chosen elements from the universe: we show how to efficiently find a universal map whose expected cost is $O(\log mn)$ times the expected optimal cost. In fact, we give a slightly improved analysis and show that this is the best possible. We generalize these ideas to weighted set cover and show similar guarantees to (non-metric) facility location, where we have to balance the facility opening cost with the cost of connecting clients to the facilities. We show applications of our results to universal multi-cut and disc-covering problems, and show how all these universal mappings give us algorithms for the *stochastic online* variants of the problems with the same competitive factors.

Key words. approximation algorithms, universal algorithms, online algorithms, set cover, facility location

AMS subject classifications. 68W05, 68W25, 68W27, 68W40

1. Introduction.

In the classical *set cover* problem we are given a set X, taken from a *universe* U of n elements, and a collection $\mathscr{S} \subseteq 2^U$ of m subsets of U, with a cost function $c\colon \mathscr{S} \to \mathbb{R}_{\geq 0}$. (The pair $(U, \mathscr{S})$ is sometimes called a *set system*). The aim is to compute a sub-collection $\mathscr{S}' \subseteq \mathscr{S}$ which covers X, i.e., $X \subseteq \cup_{S \in \mathscr{S}'} S$, with minimum cost $c(\mathscr{S}') := \sum_{S \in \mathscr{S}'} c(S)$. For our purposes it is more convenient to interpret each feasible solution as a mapping $\mathbf{S} : U \to \mathscr{S}$ which defines, for each $u \in X$, a subset $\mathbf{S}(u)$ which covers u (breaking ties in an arbitrary way). In particular, $\mathbf{S}(X) := \cup_{u \in X} \mathbf{S}(u)$ provides the desired sub-collection $\mathscr{S}'$, of cost $c(\mathbf{S}(X)) := \sum_{S \in \mathbf{S}(X)} c(S)$. In the *cardinality* (or *unweighted*) version of the problem, all the set costs are 1, and the goal is to minimize the number $|\mathbf{S}(X)|$ of subsets used to cover X.

---

*A preliminary version of this paper appeared in FOCS'08 [21]. Part of this work was done when the non-Roman authors were visiting the Sapienza Università di Roma.

†IDSIA, University of Lugano, Galleria 1, 6928 Manno-Lugano, Switzerland, fabrizio@idsia.ch. Partially supported by the ERC Starting Grant NEWNET 279352.

‡Department of Computer Science, Carnegie Mellon University, 7203 Gates Building Pittsburgh PA 15213, anupamg@cs.cmu.edu. Supported by NSF awards CCF-0448095 and CCF-0729022, and an Alfred P. Sloan Fellowship.

§Dipartimento di Informatica e Sistemistica, Sapienza Università di Roma, Via Ariosto 25, 00185 Rome, Italy, stefano.leonardi@dis.uniroma1.it. Partially supported by the EU within the 6th Framework Programme under contract no. 001907 "Dynamically Evolving, Large Scale Information Systems" (DELIS)

¶Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany, pmiettin@mpi-inf.mpg.de. Part of this work was done when the author was with Helsinki Institute for Information Technology, University of Helsinki, Finland.

‖Institute of Informatics, University of Warsaw, ul. Banacha 2, 02097 Warsaw, Poland, sank@mimuw.edu.pl. Partially supported by the Polish Ministry of Science grant N N206 355636.

**School of Computer Science, McGill University, McConnell Engineering Building, Room 324, 3480 University Street, Montreal, Quebec, Canada H3A 2A7, mohit@cs.mcgill.ca. Part of this work was done when the author was at Carnegie Mellon University.

In their seminal work, Jia et al. [33] define, among other problems, a *universal* variant of the set cover problem. Here the mapping $\mathbf{S}$ has to be provided a-priori, i.e., without knowing the actual value of $X \subseteq U$. The problem now is to find a mapping which minimizes the worst-case ratio $\max_{X \subseteq U}\{c(\mathbf{S}(X))/c(\mathsf{opt}(X))\}$ between the cost of the set cover given by $\mathbf{S}$ (which is computed without knowing X), and the cost of the optimal "offline" solution $\mathsf{opt}(X)$ (which is based on the knowledge of X). A universal algorithm is α-competitive if the ratio above is at most α.

Universal algorithms are useful for applications in distributed environments, where decisions have to be taken locally, with little communication overhead. Similarly, in critical or real-time applications we might not have enough time to run any approximation algorithm once the actual instance of the problem shows up. Hence we need to perform most of the computation beforehand, even if this might imply worse competitive factors and higher preprocessing time. Indeed, we might also think of applications where the solution computed a-priori is wired on a circuit. Eventually, universal problems have strong implications to online problems (where the instance is revealed gradually, and the solution is computed step-by-step). In particular, any universal algorithm provides an online algorithm with the same competitive ratio.

The standard competitive analysis for universal (and online) algorithms assumes that the input is chosen adversarially. This is often too pessimistic: indeed, for universal set cover, Jia et al. [33] gave $\tilde{\Theta}(\sqrt{n})$ bounds. (The $\tilde{\Theta}$ notation suppresses poly-logarithmic factors). In many situations it is reasonable to assume that the input is sampled according to some probability distribution. In other words, what if we are competing against nature and the lack of information about the future, and not against a malicious adversary out to get us? Can we give algorithms with a better performance in that case?

1.1. Our Results and Techniques. We formalize the questions above by defining a *stochastic* variant of the universal set cover problem. Here the input X is obtained by sampling k times a given probability distribution $\pi : U \to \mathbb{Q}_{\geq 0}$, $\sum_{u \in U} \pi(u) = 1$. Let $\omega \in U^k$ be the random sequence of elements obtained, possibly with repetitions. Sometimes we consider ω as a multi-set. In that case, $|\omega|$ denotes the cardinality of ω (in particular, $|\omega| = k$). We use a similar notation for any subsequence of ω. The aim is minimizing the ratio $\mathbf{E}_\omega[c(\mathbf{S}(\omega))]/\mathbf{E}_\omega[c(\mathsf{opt}(\omega))]$ between the expected cost of the solution computed w.r.t. $\mathbf{S}$ and the expected optimal cost. We sometimes omit ω when the meaning will be clear from the context. We call an algorithm for the universal stochastic set cover problem (and related problems) *length-aware* if it is given the length k of the sequence in input, and *length-oblivious* otherwise. As we will see, this distinction is crucial.

As a warm up for the reader, we present a lower bound on the quality of the mapping obtained by running on the set system $(U, \mathscr{S})$ the standard greedy algorithm, which selects in each step the subset with the best ratio of cost to number of uncovered elements. This algorithm defines an order on the selected sets: let each element be mapped to the first set in the order covering it. Consider a set $S_{all} = U$ covering the whole universe, of cost $c(S_{all}) = \sqrt{n}$, and singleton sets $S_u = \{u\}$ for each $u \in U$, each of unit cost $c(S_u) = 1$. The greedy set cover algorithm maps all the elements into S_{all}. For a uniform distribution π and $k = 1$, the cost of this mapping is $\sqrt{n}$, while the optimal mapping (assigning each $u \in U$ to the corresponding singleton set S_u) has always cost one. Note that, for $k \simeq n$, the situation changes drastically: now the greedy algorithm produces the optimal mapping with high probability. Indeed, essentially the same example shows that any length-oblivious universal algorithm for the

(weighted) stochastic set cover problem must be $\Omega(\sqrt{n})$-competitive (see Section 3).

Motivated by the example above, we developed an algorithm based on the interleaving of standard greedy with a second, *even more myopic*, greedy algorithm that selects the min-cost set which covers at least *one* uncovered element (disregarding the actual number of covered elements). In each selection step we trust the *min-ratio* greedy algorithm if a subset with a sufficiently small ratio exists, and the *min-cost* one otherwise. The threshold ratio is derived from the length k of the sequence. The main result of this paper can be stated as follows (see Section 3):

THEOREM 1.1. *There exists a polynomial-time length-aware algorithm that returns a universal mapping* $\mathbf{S}$ *to the (weighted) universal stochastic set cover problem with* $\mathbf{E}[c(\mathbf{S})] = O(\log mn)\mathbf{E}[c(\mathsf{opt})]$.

Above, and elsewhere in this paper, log denotes logarithm of base 2. When m is polynomial in n, this is asymptotically the best possible due to the $o(\log n)$-inapproximability of set cover (which extends to the universal stochastic case by choosing $k \gg n$). For values of $m \gg n$, the competitive factor can be improved to $O\left(\frac{\log m}{\log\log m - \log\log n}\right)$, and this bound is tight (see Section 4).

The crux of our analysis is bounding the cost of the min-cost sets selected by the algorithm when it cannot find a good min-ratio set. Here we use a counting argument to show that the number of sampled elements among the still-uncovered elements is sufficiently *small* compared to the number of sets used by the optimal solution to cover those elements. We then translate this into a convenient lower bound on the cost paid by the optimum solution to cover the mentioned elements.

In the unweighted case we can do better: here the standard greedy algorithm provides a *length-oblivious* universal algorithm with the same competitive ratio. However, its analysis requires some new ideas.

THEOREM 1.2. *There exists a polynomial-time length-oblivious algorithm that returns a universal mapping* $\mathbf{S}$ *to the unweighted universal stochastic set cover problem with* $\mathbf{E}[|\mathbf{S}|] = O(\log mn)\,\mathbf{E}[|\mathsf{opt}|]$.

Based on the proof of Theorem 1.2, we also show that the dependence on n in the competitive factor can be removed if exponential time is allowed, and can possibly be reduced when the set system has a small VC-dimension. The latter result is especially suited for applications where $m \ll n$, one of which we highlight in Section 9.2. Additionally, it should be noted that, due to concentration bounds, our length-aware mappings can be used to construct solutions for the independent activation model introduced in [31,35] as well. This is shortly discussed in Section 8.

Our results naturally extend to the stochastic version of the *online set cover* problem. Here the random sequence ω of elements is presented to the algorithm one element at a time, and, each time a new element u is given, the algorithm is forced to define a set $\mathbf{S}(u) \ni u$. In other words, the mapping $\mathbf{S}$ is constructed in an online fashion. We remark that, once the value $\mathbf{S}(u)$ is chosen, it cannot be modified in the following steps. Moreover, the length k of the sequence is not given to the algorithm. Similarly to the universal stochastic case, the aim is to minimize $\mathbf{E}_\omega[c(\mathbf{S}(\omega))]/\mathbf{E}_\omega[c(\mathsf{opt}(\omega))]$.

A length-oblivious universal algorithm would immediately imply an online algorithm with the same competitive factor. However, as there is no such algorithm (for the weighted case), we achieve the same task by combining a family of universal mappings, computed via our (length-aware) universal algorithm for carefully-chosen sequence lengths (see Section 5):

THEOREM 1.3. *There exists a polynomial-time* $O(\log mn)$*-competitive algorithm*

for the online (weighted) stochastic set cover problem.

Our techniques are fairly flexible, and can be applied to other covering-like problems. In Sections 6 and 9 we describe universal algorithms for the stochastic versions of (non-metric) *facility location*, *multi-cut*, and *disc covering* in the plane.

In the next sections, we implicitly assume that π is a uniform distribution; this assumption is without loss of generality using the standard reduction described in Section 7.

1.2. Related Work.

Universal, Oblivious and A-Priori Problems. These are problems where a single solution is constructed which is evaluated given multiple inputs—and either the worst-case or the average-case performance is considered. For instance, the universal TSP problem, where one computes a permutation that is used for all possible inputs, has been studied both in the worst-case scenario for the Euclidean plane [6,42] and general metrics [22, 25, 33], as well as in the average-case [7, 20, 32, 46, 48]. (For the related problem of universal Steiner tree, see [20, 22, 33, 35].) For *universal set cover* and *facility location*, the previous results are in the worst-case: Jia et al. [33] introduced the problems, show that the adversary is very powerful in such models, and give nearly-matching $\Omega(\sqrt{n})$ and $O(\sqrt{n \log n})$ bounds on the competitive factor. For *oblivious routing* [8, 28, 43] (see, e.g., [50, 51] for special cases), a tight logarithmic competitive result as well as a polynomial-time algorithm to compute the best routing is known in the worst case for undirected graphs [5, 44]. For *oblivious routing on directed graphs* the situation is similar to our problem: in the worst case the lower bound of $\Omega(\sqrt{n})$ [5] nearly matches upper bounds [26] but for the average case, [23] give an $O(\log^2 n)$-competitive oblivious routing algorithm when demands are chosen randomly from a known demand-distribution; they also use "demand-dependent" routings and show that these are necessary.

Online Problems. Online problems have a long history (see, e.g., [9,17]), and there have been many attempts to relax the strict worst-case notion of competitive analysis: see, e.g., [1, 13, 20] and the references therein. Online problems with stochastic inputs (either i.i.d. draws from some distribution, or inputs arriving in random order) have been studied, e.g., in the context of optimization problems [4, 20, 39, 40], secretary problems [19], mechanism design [24], and matching problems in Ad-auctions [38].

Alon et al. [2] gave the first online algorithm for set cover with a competitive ratio of $O(\log m \log n)$; they used an elegant primal-dual-style approach that has subsequently found many applications (e.g., [3, 11]). This ratio is the best possible under complexity-theoretic assumptions [15]; even unconditionally, no deterministic online algorithm can do much better than this [2]. Online versions of *metric* facility location are studied in both the worst case [18, 39], the average case [20], as well as in the stronger *random permutation model* [39], where the adversary chooses a set of clients unknown to the algorithm, and the clients are presented to us in a random order. It is easy to show that for our problems, the random permutation model (and hence any model where elements are drawn from an *unknown* distribution) are as hard as the worst case.

Offline Problems: Set Cover and (Non-Metric) Facility Location. The set cover problem is one of the foster-children for approximation algorithms: a $\Theta(\log n)$-approximation has been long known for it [34,37], and this is the best possible [14,45]. For the special case of set systems with small VC-dimension, a better algorithm is given in [10]. Other objective functions have also been used, e.g., min-latency [16] and min-entropy [12,27]. The $O(\log n)$ approximation for non-metric facility location is due to Hochbaum [29].

Stochastic Optimization. Research in (offline) stochastic optimization gives results for k-stage stochastic set cover; however, the approximation in most papers [31, 47] is dependent on the number of stages k. Srinivasan [49] shows how to round an LP-relaxation of the k-stage set cover problem with only an $O(\log n)$ loss, independent of k; this can be used to obtain an $O(\log n)$ approximation to the expected cost of the *best online* algorithm for stochastic set cover in poly(mn) time. In contrast to this, our results get within $O(\log nm)$ of the *best expected offline* cost.

2. The Unweighted Set Cover Problem. In this section, we present a $O(\log mn)$-competitive algorithm for the universal stochastic set cover problem in the unweighted case (i.e., $c(S) = 1$ for all sets $S \in \mathscr{S}$). Moreover, the proof will introduce ideas and arguments which we will extend upon for the case of weighted set cover in the following section.

Our algorithm is the natural adaptation of the standard greedy algorithm for the set cover problem (see Algorithm 1). However, its analysis is different from the one for the classical offline greedy algorithm. We remark that our algorithm is length-oblivious, i.e., the mapping S computed by the algorithm works for any sequence length k.

Algorithm 1: Mapping for unweighted set cover.

Data: Set system $(U, \mathscr{S})$.
while $U \neq \emptyset$ **do**
  let $S \leftarrow$ set in $\mathscr{S}$ maximizing $|S \cap U|$;
  $\mathsf{S}(v) \leftarrow S$ for each $v \in S \cap U$;
  $U \leftarrow U \setminus S$;

For the analysis, fix some sequence length k and let $\mu = \mathbf{E}_{\omega \in U^k}[|\mathsf{opt}(\omega)|]$ be the expected optimal cost. We first show that there are 2μ sets which cover all but δn elements from U, where $\delta = \mu \frac{3 \ln 2m}{k}$.

LEMMA 2.1 (Existence of Small Almost-Cover). *Let $(U, \mathscr{S})$ be any set system with n elements and m sets. There exist 2μ sets in $\mathscr{S}$ which cover all but δn elements from U, for $\delta = \mu \frac{3 \ln 2m}{k}$.*

Proof. Let d denote the *median* of opt, i.e., in at least half of the scenarios from U^k, the optimal solution uses at most d sets to cover all the k elements occurring in that scenario. By Markov's inequality, $d \leq 2\mu$.

There are at most $p := \sum_{j=0}^{d} \binom{m}{j} \leq \binom{m}{d} 2^d \leq (2m)^d$ collections of at most d sets from $\mathscr{S}$: let these collections be $\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_p$. In order to lighten the notation, we use $\cup\mathscr{C}_i$ to denote the union $\cup_{S \in \mathscr{C}_i} S$ of the sets in $\mathscr{C}_i$. In particular, $\cup\mathscr{C}_i$ is not a collection of sets. We now show that $|\cup\mathscr{C}_i| \geq n(1-\delta)$ for some i.

Suppose by contradiction that $|\cup\mathscr{C}_i| < n(1-\delta) \leq n\mathrm{e}^{-\delta}$ for each $1 \leq i \leq p$. Since half of the n^k scenarios have a cover with at most d sets, the k elements for any such scenario can be picked from some collection $\mathscr{C}_i$. Hence,

$$\sum_{i=1}^{p} |\cup\mathscr{C}_i|^k \geq \frac{1}{2} n^k.$$

Plugging in $p \leq (2m)^d = \mathrm{e}^{d \ln 2m} \leq \mathrm{e}^{2\mu \ln 2m}$ and $|\cup\mathscr{C}_i| < n\mathrm{e}^{-\delta}$, we get

$$p(n\mathrm{e}^{-\delta})^k > \frac{1}{2} n^k \implies \mathrm{e}^{(2\mu \ln 2m) - k\delta} > \frac{1}{2} \implies \mathrm{e}^{-\mu \ln 2m} > \frac{1}{2}.$$

Since $m \geq 1$ and $\mu \geq 1$, we also get $e^{-\mu \ln 2m} \leq \frac{1}{2}$, which gives the desired contradiction. □

The greedy algorithm is a $O(\log n)$-approximation [36, Thm 5.15] for the *partial coverage problem* (pick the fewest sets to cover some $(1-\delta)$ fraction of the elements). This implies the following corollary.

COROLLARY 2.2. *Algorithm 1 covers at least $n(1-\delta)$ elements using the first $O(\mu \log n)$ sets.*

Finally, we can complete the analysis of Algorithm 1. (A slightly improved result will be described in Section 4.)

Proof. (Theorem 1.2) The first $O(\mu \log n)$ sets picked by the greedy algorithm cover all except δn elements of U, by Corollary 2.2. We count all these sets as contributing to $\mathbf{E}[|\mathbf{S}|]$; note that this is fairly pessimistic.

From the remaining elements, we expect to see at most $\frac{k}{n} \cdot \delta n = 3\mu \ln 2m$ elements in a random sequence of length k. Whenever one of those elements appears, we use at most one new set to cover it. Hence, in expectation, we use at most $3\mu \ln 2m$ sets for covering the elements which show up from the δn remaining elements, making the total expected number of sets $O(\mu(\log n + \log m))$ as claimed. □

2.1. An Exponential-Time Variant. Surprisingly, we can trade off the $O(\log n)$ factor in the approximation for a worse running time; this is quite unusual for competitive analysis where the lack of information rather than lack of computational resources is typically the deciding factor. Instead of running the greedy algorithm to find the first $O(\mu \log n)$ sets which cover $(1-\delta)n$ elements, we can run an exponential-time algorithm which finds 2μ sets which cover $(1-\delta)n$ elements (whose existence is shown in Lemma 2.1). Thus we obtain an exponential-time universal algorithm whose expected competitive factor is $O(\log m)$.

In Section 9.2 we give improved algorithms when the set system admits small "ϵ-nets" (e.g., when it has small VC-dimension), and also describe an application of this result to the disc-cover problem.

3. The Weighted Set Cover Problem. We now consider the general (weighted) version of the universal stochastic set cover problem. As mentioned in the introduction, and in contrast to the unweighted case where we could get a length-oblivious universal mapping $\mathbf{S}$, in the weighted case there is no mapping $\mathbf{S}$ that is good for all sequence lengths k.

THEOREM 3.1. *Any length-oblivious algorithm for the (weighted) universal stochastic set cover problem has a competitive ratio of $\Omega(\sqrt{n})$.*

Proof. Consider a set $S_{all} = U$ covering the whole universe, of cost $c(S_{all}) = \sqrt{n}$, and singleton sets $S_u = \{u\}$ for each $u \in U$, each of unit cost $c(S_u) = 1$. Take any length-oblivious algorithm. If this algorithm maps more than half the elements to S_{all} then the adversary can choose $k = 1$ and the algorithm pays in expectation $\Omega(\sqrt{n})$ while the optimum is 1. Otherwise, the algorithm maps less than half the elements to S_{all} and the adversary chooses $k = n$. In this case the algorithm pays, in expectation, $\Omega(n)$ while the optimum is at most $\sqrt{n}$. □

Hence, we do the next best thing: we give a $O(\log mn)$-competitive universal algorithm, which is aware of the input length k. We first present an algorithm for computing a universal mapping $\mathbf{S}$ when given the value of $\mathbf{E}[c(\mathsf{opt})]$. This assumption will be relaxed later, by showing that indeed the value of k is sufficient.

Consider Algorithm 2 in the figure. In each iteration of the algorithm, we either choose a set with the best ratio of cost to number of uncovered elements (*Type I* sets), or simply take the cheapest set which covers at least one uncovered element

Algorithm 2: Mapping for weighted set cover.

Data: Set system $(U, \mathscr{S})$, $c\colon \mathscr{S} \to \mathbb{R}_{\geq 0}$, $\mathbf{E}[c(\mathsf{opt})]$.
while $U \neq \emptyset$ **do**
  let $S \leftarrow$ set in $\mathscr{S}$ minimizing $\frac{c(S)}{|S \cap U|}$;
  if $\frac{c(S)}{|S \cap U|} > \frac{64\mathbf{E}[c(\mathsf{opt})]}{|U|}$ then let $S \leftarrow$ set in $\mathscr{S}$ minimizing $c(S)$;
  $\mathbf{S}(u) \leftarrow S$ for each $u \in S \cap U$;
  $U \leftarrow U \setminus S$ and $\mathscr{S} \leftarrow$ all sets covering at least one element remaining in U ;

(*Type II* sets). We remark that since the set U is updated at each step, we may alternate between picking sets of Type I and II in an arbitrary way. We also observe that both types of sets are needed in general, as the proof of Theorem 3.1 shows. As already mentioned in the introduction, and if not differently specified, $\mathbf{E}[c(\mathsf{opt})] = \mathbf{E}_{\omega \in U^k}[c(\mathsf{opt})]$ and $\mathbf{E}[|\mathsf{opt}|] = \mathbf{E}_{\omega \in U^k}[|\mathsf{opt}|]$.

We bound the cost of sets of Type I and II separately. The following lemma shows that the total cost of Type I sets is small, even in the fairly pessimistic assumption that we use all such sets to cover the random sequence ω. Since Type I sets are *min-ratio* sets, their cost can be bounded using the standard greedy analysis of set cover.

LEMMA 3.2 (Type I Set Cost). *The cost of Type I sets selected by Algorithm 2 is* $O(\log n) \cdot \mathbf{E}[c(\mathsf{opt})]$.

Proof. Let $R_1, \ldots, R_h$ be the Type I sets picked by the algorithm in this order. Moreover, let U_i denote the set of uncovered elements just before R_i was picked. Since the algorithm picked a Type I set, $c(R_i) \leq |R_i \cap U_i| \frac{64\ \mathbf{E}[c(\mathsf{opt})]}{|U_i|}$. Hence, the total cost of the sets R_i can be bounded by

$$\sum_{i=1}^{h} c(R_i) \leq \sum_{i=1}^{h} \frac{64|R_i \cap U_i| \times \mathbf{E}[c(\mathsf{opt})]}{|U_i|} \leq 64\mathbf{E}[c(\mathsf{opt})] \sum_{t=1}^{n} \frac{1}{t} \leq 64\mathbf{E}[c(\mathsf{opt})] \ln n.$$

□

It remains to bound the expected cost of the Type II sets, which is also the technical heart of our argument. Let $S_1, \ldots, S_\ell$ be the Type II sets selected by Algorithm 2 in this order. Observe that, since Type II sets are picked on the basis of their cost alone, $c(S_i) \leq c(S_{i+1})$ for each $1 \leq i \leq \ell - 1$. Let U_i denote the set of uncovered elements just before S_i was picked. Define $n_i = |U_i|$ and let $k_i = n_i \frac{k}{n}$ be the expected number of elements sampled from U_i (with repetitions). Denote by ω_i the subsequence (or multi-set) of the input sequence ω obtained by taking only elements belonging to U_i, and let $\mathsf{opt}|_{\omega_i}$ be the subcover obtained by taking for each $u \in \omega_i$ the set in $\mathsf{opt} = \mathsf{opt}(\omega)$ covering u. (Note that this is *not necessarily* the optimal set cover for ω_i.) With the usual notation, $c(\mathsf{opt}|_{\omega_i})$ and $|\mathsf{opt}|_{\omega_i}|$ denote the cost and number of the sets in $\mathsf{opt}|_{\omega_i}$, respectively. For any positive integer q, let Ω_i^q be the set of scenarios ω_i such that $|\omega_i| = q$.

The bound on the expected cost of the Type II sets is given in Lemma 3.6. The basic idea behind our proof is as follows. It is easy to prove a $O(\log n \cdot \mathbf{E}[c(\mathsf{opt})])$ bound on the expected cost incurred by using sets S_i with $k_i = \Omega(\log n)$. For the remaining sets $S_1, \ldots, S_j$, $k_j = O(\log n)$, we can naturally express the expected cost incurred by the algorithm as a function of $c(S_i)$'s and k_i's. Lemma 3.5 provides a bound on each k_i in terms of $\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]$ (in its proof, we exploit the technical Lemma

3.3). The resulting bound on the total cost in terms of $c(S_i)$ and $\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]$ is then converted into a bound in terms of $\mathbf{E}[c(\mathsf{opt}|_{\omega_i})]$ by means of Lemma 3.4. We next proceed with the proof of the mentioned claims.

LEMMA 3.3. *For every $i \in \{1,\ldots,\ell\}$, if $k_i \geq 8\log 2n$ then there exists $q \geq \frac{k_i}{2}$ such that*

$$\Pr\left[c(\mathsf{opt}) \leq 8\mathbf{E}[c(\mathsf{opt})] \;\; \textit{and} \;\; |\mathsf{opt}| \leq 8\mathbf{E}[|\mathsf{opt}|] \mid \omega \in \Omega_i^q\right] \geq \frac{1}{2}.$$

Proof. We restrict our attention to scenarios in $\Omega_i^{\geq \frac{k_i}{2}} := \uplus_{p \geq \frac{k_i}{2}} \Omega_i^p$, i.e., scenarios where the sampled k elements contain at least $\frac{k_i}{2}$ elements from U_i. Chernoff's bound implies $\Pr[|\omega_i| < \frac{k_i}{2}] \leq \exp\left(-\frac{(1/2)^2\, 8\log 2n}{2}\right) \leq \frac{1}{2n}$, and hence

$$\Pr[\omega \in \Omega_i^{\geq \frac{k_i}{2}}] \geq 1 - \frac{1}{2n} \geq \frac{1}{2}. \tag{3.1}$$

Observe that

$$\mathbf{E}[|\mathsf{opt}|] \geq \Pr\left[\omega \in \Omega_i^{\geq \frac{k_i}{2}}\right] \cdot \mathbf{E}\left[|\mathsf{opt}| \mid \omega \in \Omega_i^{\geq \frac{k_i}{2}}\right] \overset{(3.1)}{\geq} \frac{1}{2}\mathbf{E}\left[|\mathsf{opt}| \mid \omega \in \Omega_i^{\geq \frac{k_i}{2}}\right]. \tag{3.2}$$

Let d_i be the upper quartile of $|\mathsf{opt}| = |\mathsf{opt}(\omega)|$ restricted to $\omega \in \Omega_i^{\geq \frac{k_i}{2}}$. In other terms, in three-quarters of the scenarios in $\Omega_i^{\geq \frac{k_i}{2}}$, the optimal solution $\mathsf{opt} = \mathsf{opt}(\omega)$ uses at most d_i sets to cover the elements in the scenario. By Markov's inequality and the definition of d_i,

$$\frac{1}{d_i}\mathbf{E}\left[|\mathsf{opt}| \mid \omega \in \Omega_i^{\geq \frac{k_i}{2}}\right] \geq \Pr[|\mathsf{opt}| \geq d_i \mid \omega \in \Omega_i^{\geq \frac{k_i}{2}}] \geq \frac{1}{4}. \tag{3.3}$$

Altogether,

$$d_i \overset{(3.3)}{\leq} 4\mathbf{E}\left[|\mathsf{opt}| \mid \omega \in \Omega_i^{\geq \frac{k_i}{2}}\right] \overset{(3.2)}{\leq} 8\mathbf{E}\left[|\mathsf{opt}|\right].$$

In words, in at least three quarters of the scenarios with $\omega \in \Omega_i^{\geq \frac{k_i}{2}}$ the cardinality of the optimum solution is at most 8 times the expected cardinality of the optimum solution (with no restriction on ω).

Essentially the same argument shows that the cost $c(\mathsf{opt})$ is at most $8\mathbf{E}[c(\mathsf{opt})]$ with probability at least 3/4 given that $\omega \in \Omega_i^{\geq \frac{k_i}{2}}$. Hence, by the union bound,

$$\Pr\left[c(\mathsf{opt}) \leq 8\mathbf{E}[c(\mathsf{opt})] \;\text{ and }\; |\mathsf{opt}| \leq 8\mathbf{E}[|\mathsf{opt}|] \;\mid\; \omega \in \Omega_i^{\geq \frac{k_i}{2}}\right] \geq \frac{1}{2}.$$

Since $\Omega_i^{\geq \frac{k_i}{2}} = \uplus_{p \geq \frac{k_i}{2}} \Omega_i^p$, an averaging argument implies that some $q \geq \frac{k_i}{2}$ satisfies the claim of the lemma. ∎

LEMMA 3.4. *For all $1 \leq i \leq \ell$,*

(a) $c(S_i)\mathbf{E}\left[|\mathsf{opt}|_{\omega_{i+1}}|\right] \leq \mathbf{E}\left[c(\mathsf{opt}|_{\omega_{i+1}})\right]$;

(b) $c(S_i)\left(\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] - \mathbf{E}\left[|\mathsf{opt}|_{\omega_{i+1}}|\right]\right) \leq \mathbf{E}[c(\mathsf{opt}|_{\omega_i})] - \mathbf{E}\left[c(\mathsf{opt}|_{\omega_{i+1}})\right]$.

Proof. The set S_{i+1} is the cheapest set covering any element of U_{i+1}, and hence $c(S_{i+1})$ is a lower bound on the cost of the sets in $\mathsf{opt}|_{\omega_{i+1}}$. Since by construction $c(S_i) \le c(S_{i+1})$,

$$c(S_i)|\mathsf{opt}|_{\omega_{i+1}}| \le c(S_{i+1})|\mathsf{opt}|_{\omega_{i+1}}| \le c(\mathsf{opt}|_{\omega_{i+1}}).$$

Analogously, the number of sets that opt uses to cover the elements $U_i \setminus U_{i+1}$ covered by S_i is $|\mathsf{opt}|_{\omega_i}| - |\mathsf{opt}|_{\omega_{i+1}}|$, and for each such set opt pays at least $c(S_i)$. Thus,

$$c(S_i)(|\mathsf{opt}|_{\omega_i}| - |\mathsf{opt}|_{\omega_{i+1}}|) \le c(\mathsf{opt}|_{\omega_i}) - c(\mathsf{opt}|_{\omega_{i+1}}).$$

Taking expectations on the inequalities gives the lemma. □

The next lemma proves that, if k_i is large enough, the optimal solution uses many sets to cover the remaining elements. The observation here is similar to Lemma 2.1, but now the number of sets in the set cover is not equal to its cost. This is why we needed a careful restriction of the optimal solution to subproblems given by $\mathsf{opt}|_{\omega_i}$. We recall that we are focusing on iterations where the condition $\frac{c(S)}{|S\cap U_i|} \le \frac{64\mathbf{E}[c(opt)]}{|U_i|}$ is not satisfied by any set S, where U_i is the set of uncovered elements at the beginning of the iteration.

LEMMA 3.5. *For every $i \in \{1,\dots,\ell\}$, if $k_i \ge 8\log 2n$ then $k_i \le 16\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]\log m$.*

Proof. For a contradiction, assume that $k_i > 16\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]\log m$, and use Lemma 3.3 to define q. There are exactly n_i^q equally likely different sequences ω_i corresponding to sequences in Ω_i^q. In at least one half of these scenarios, opt (and hence $\mathsf{opt}|_{\omega_i}$) uses at most $8\mathbf{E}[|\mathsf{opt}|]$ sets of cost at most $8\mathbf{E}[c(\mathsf{opt})]$.

Let $\mathscr{S}_i$ be the multi-set of sets $\{S \cap U_i \mid S \in \mathscr{S}\}$, and denote by $\mathscr{C}_1, \mathscr{C}_2, \dots, \mathscr{C}_p$ the collections of at most $8\mathbf{E}[|\mathsf{opt}|]$ sets from $\mathscr{S}_i$ with total cost at most $8\mathbf{E}[c(\mathsf{opt})]$; there are at most $(2m)^{8\mathbf{E}[|\mathsf{opt}|]}$ of these collections. As previously, let $\cup\mathscr{C}_j$ denote the union of the sets from $\mathscr{C}_j$. Analogously to the proof of Lemma 2.1,

$$\sum_{j=1}^{p} |\cup\mathscr{C}_j|^q \ge \frac{1}{2} n_i^q.$$

Hence there is a collection $\mathscr{C}_j$ with

$$|\cup\mathscr{C}_j| \ge \frac{n_i}{2^{1/q} p^{1/q}} \ge \frac{n_i}{2(2m)^{8\mathbf{E}[|\mathsf{opt}|]/q}} \ge \frac{n_i}{2(2m)^{1/\log m}} \ge \frac{n_i}{8},$$

where we use the assumption $q \ge k_i/2 > 8\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]\log m$. Since the total cost of sets in $\mathscr{C}_j$ is at most $8\mathbf{E}[c(\mathsf{opt})]$ and they cover $n_i/8$ elements from U_i, there is a set $S \in \mathscr{C}_j$ with

$$\min_{S\in\mathscr{C}_j} \frac{c(S)}{|S\cap U_i|} \le \frac{\sum_{S\in\mathscr{C}_j} c(S)}{\sum_{S\in\mathscr{C}_j} |S\cap U_i|} \le \frac{8\mathbf{E}[\,c(\mathsf{opt})]}{n_i/8} = \frac{64\mathbf{E}[c(\mathsf{opt})]}{n_i}.$$

However, the Type II set S_i was picked by the algorithm because there were no set for which $\frac{c(S)}{|S\cap U_i|} < \frac{64\mathbf{E}[c(\mathsf{opt})]}{|U_i|}$, so we get a contradiction and the lemma follows. □

Finally, we can bound the expected cost of Type II sets: recall that we incur the cost of some set S_i only if one of the corresponding elements $S_i \cap U_i$ is sampled.

LEMMA 3.6 (Type II Set Cost). *The expected cost of Type II sets selected by Algorithm 2 is $O(\log mn)\,\mathbf{E}[c(\mathsf{opt})]$.*

Proof. Recall that the Type II sets were $S_1, S_2, \ldots, S_\ell$. Set $k_{\ell+1} = 0$ and $c(S_0) = 0$ for notational convenience. Moreover, let j be such that $k_j \geq 8 \log 2n$ but $k_{j+1} < 8 \log 2n$. Hence, in expectation we see at most $8 \log 2n$ elements from U_{j+1}, and since each of these elements is covered by a set that does not cost more than the one covering it in opt, the cost incurred by using the sets $S_{j+1}, \ldots, S_\ell$ is bounded by $8 \log 2n \, \mathbf{E}[c(\mathsf{opt})]$.

By Lemmas 3.4 and 3.5, the expected cost incurred by using the remaining sets $S_1, \ldots, S_j$ satisfies

$$\begin{aligned}
&\sum_{i=1}^{j} c(S_i) \mathbf{Pr}[\omega \cap (S_i \cap U_i) \neq \emptyset] \\
&\quad \leq \sum_{i=1}^{j} c(S_i) \mathbf{E}[|\omega \cap (S_i \cap U_i)|] \overset{U_{i+1} \subseteq U_i \setminus S_i}{\leq} \sum_{i=1}^{j} c(S_i) \mathbf{E}[|\omega \cap (U_i \setminus U_{i+1})|] \\
&\quad \leq \sum_{i=1}^{j} c(S_i) \left(k_i - k_{i+1}\right) \overset{c(S_0)=0}{\leq} \sum_{i=1}^{j} k_i \left(c(S_i) - c(S_{i-1})\right) \\
&\quad \overset{\text{Lem. 3.5}}{\leq} \sum_{i=1}^{j} 16 \mathbf{E}[|\mathsf{opt}|_{\omega_i}|] \log m \cdot \left(c(S_i) - c(S_{i-1})\right) \\
&\quad = 16 \log m \cdot \Big(c(S_j) \mathbf{E}\big[|\mathsf{opt}|_{\omega_{j+1}}|\big] + \sum_{i=1}^{j} c(S_i) \big(\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] - \mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big] \big) \Big) \\
&\quad \overset{\text{Lem. 3.4}}{\leq} 16 \log m \cdot \Big(\mathbf{E}\big[c(\mathsf{opt}|_{\omega_{j+1}})\big] + \sum_{i=1}^{j} \big(\mathbf{E}[c(\mathsf{opt}|_{\omega_i})] - \mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big] \big) \Big) \\
&\quad = 16 \log m \cdot \mathbf{E}[c(\mathsf{opt}|_{\omega_1})] \leq 16 \log m \cdot \mathbf{E}[c(\mathsf{opt})].
\end{aligned}$$

This concludes the proof of the lemma. □

We have all the ingredients to prove the main result of this section.

Proof. (Theorem 1.1) Lemmas 3.2 and 3.6 together imply that Algorithm 2 is $O(\log mn)$-competitive. We now show how to adapt the result to the case when we are given as input the sequence length k, instead of $\mathbf{E}[c(\mathsf{opt})]$.

Algorithm 2 uses the value of $\mathbf{E}[c(\mathsf{opt})]$ only in comparison with $\frac{c(S) \cdot |U|}{|S \cap U|}$ for different sets S. This fraction can take at most mn^2 different values, and thus the algorithm can generate at most $mn^2 + 1$ different mappings $\{\mathbf{S}_i\}_{i=1}^{mn^2+1}$. For any such map $\mathbf{S}$, computing the expected cost $\mathbf{E}[c(\mathbf{S})]$ is easy: indeed, if $\mathbf{S}^{-1}(S)$ is the pre-image of $S \in \mathscr{S}$, then

$$\mathbf{E}[c(\mathbf{S})] = \sum_{S \in \mathscr{S}} c(S) \cdot \Pr[\omega \cap \mathbf{S}^{-1}(S) \neq \emptyset].$$

The value of k is sufficient (and necessary) to compute the probabilities above. Hence, we can select the mapping $\mathbf{S}_i$ with the minimum expected cost for the considered value of k; this cost is at most the cost of the mapping generated with the knowledge of $\mathbf{E}[c(\mathsf{opt})]$. □

4. Matching Bounds. In this section we present slightly refined upper bounds and matching lower bounds for universal stochastic set cover.

If we stay within polynomial time, and if $m = \text{poly}(n)$, then the resulting $O(\log mn) = O(\log n)$ competitive factor is asymptotically the best possible given

suitable complexity-theoretic assumptions. However, for the cases when $m \gg n$, we can show a better dependence on the parameters.

THEOREM 4.1. *For $m > n$, there exists a polynomial-time length-aware (resp. length-oblivious) $O\left(\frac{\log m}{\log\log m - \log\log n}\right)$-competitive algorithm for the weighted (resp. unweighted) universal stochastic set cover problem.*

Proof. Let us slightly modify the universal algorithm for weighted set cover as follows: fixing a value $0 < x \leq \log m$, we increase the threshold value for $c(S)/|S \cap U|$ to $2^x \cdot 64\,\mathbf{E}[c(\mathsf{opt})]/|U|$. Let us adapt the analysis. The same argument as in Lemma 3.2 shows that the cost of Type I sets is bounded by $O(2^x \log n)\mathbf{E}[c(\mathsf{opt})]$. Lemmas 3.3 and 3.4 hold unchanged. Lemma 3.5 holds with 16 replaced by $16/x$. As a consequence, the expected cost of Type II sets becomes $O(\log n + \frac{\log m}{x})\mathbf{E}[c(\mathsf{opt})]$. Choosing $x = \log\log m - \log\log n$ gives the claim for the weighted case.

A similar result can be shown for Algorithm 1, in the unweighted (length-oblivious) case. □

The following theorem (which extends directly to *online* stochastic set cover) shows that the bounds above are tight.

THEOREM 4.2. *There are values of m and n such that any mapping $\mathbf{S}$ for the (unweighted) universal stochastic set cover problem satisfies*

$$\mathbf{E}[|\mathbf{S}|] = \Omega\left(\frac{\log m}{\log\log m - \log\log n}\right)\mathbf{E}[|\mathsf{opt}|].$$

Proof. Consider an n element universe $U = \{1, \ldots, n\}$ with the uniform distribution over the elements, and $\mathscr{S}$ consisting of all $m = \binom{n}{\sqrt{n}}$ subsets of U of size $\sqrt{n}$; hence $\log m = \Theta(\sqrt{n}\log n)$ and $\log\log m - \log\log n = \Theta(\log n)$. The sequence length is $k = \sqrt{n}/2$.

Let $\mathscr{S}_i$ be the collection of sets in $\mathbf{S}$ which are associated to the first i elements in the input sequence $\omega = (\omega_1, \ldots, \omega_k)$. Since the sets in $\mathscr{S}_i$, $|\mathscr{S}_i| \leq i$, cover at most $i\sqrt{n} \leq n/2$ elements altogether, with probability at least $1/2$ one has $\mathbf{S}(\omega_{i+1}) \notin \mathscr{S}_i$, and consequently $|\mathscr{S}_{i+1}| = |\mathscr{S}_i| + 1$. Hence, $E[|\mathscr{S}_k|] \geq \frac{k}{2} = \frac{\sqrt{n}}{4}$, i.e. $\mathbf{S}$ uses at least $\frac{\sqrt{n}}{4}$ sets in expectation. In contrast, the optimum solution uses one set deterministically. The claim follows. □

5. Online Stochastic Set Cover. The length-oblivious universal algorithm for unweighted stochastic set cover immediately gives an online algorithm with the same competitive factor: it is sufficient to compute the universal mapping $\mathbf{S}$ beforehand, and use $\mathbf{S}(v)$ to cover each new element v which arrives as input.

The same approach does not work in the weighted case, since our universal algorithm is length-aware in that case (for online algorithms the final sequence length is typically unspecified, and the competitive ratio must hold at any point of time). However, we are still able to design an online algorithm for weighted stochastic set cover with the same $O(\log mn)$ competitive ratio. The basic idea is using the universal mapping from Section 3 to cover each new element, and update the mapping from time to time. The main difficulty is choosing the update points properly: indeed, the standard approach of updating the mapping each time the number of elements doubles does not work here.

Let ω^i denote a random sequence of i elements, and let $\mathbf{S}_i$ be the mapping produced by the universal algorithm from Section 3 for a sequence of length $k = i$. We recall that the number of distinct mappings $\mathbf{S}_i$ is polynomially bounded. Our algorithm works as follows. Let k be the current number of samplings performed. The

algorithm maintains a variable k', initially set to 1, which is larger than k at any time. For a given value of k', the mapping used by the online algorithm is the universal mapping $\mathbf{S}_{k'}$. When $k = k'$, we proceed as follows. We compute a value k'' such that $\mathbf{E}[c(\mathbf{S}_{k''}(\omega^{k''}))] > 2\mathbf{E}[c(\mathbf{S}_{k'}(\omega^{k'}))]$ and $\mathbf{E}[c(\mathbf{S}_{k''-1}(\omega^{k''-1}))] \leq 2\mathbf{E}[c(\mathbf{S}_{k'}(\omega^{k'}))]$. Then we choose $\tilde{k} \in \{k''-1, k''\}$ randomly, so that $\mathbf{E}[c(\mathbf{S}_{\tilde{k}}(\omega^{\tilde{k}}))] = 2\mathbf{E}[c(\mathbf{S}_{k'}(\omega^{k'}))]$. We set $\tilde{k} = \infty$ if there is no value of k'' satisfying this property: this happens when $\mathbf{E}[c(\mathbf{S}_{k'}(\omega^{k'}))]$ is at least one half of the minimum over the (polynomially-many) mappings $\mathbf{S}_i$ of the cost of $\mathbf{S}_i$ given that all elements are sampled. Eventually, we set $k' = \max\{\tilde{k}, k'+1\}$, and modify the mapping consequently. We remark that the algorithm above takes polynomial time per sample, and does not assume any knowledge of the final number of samplings.

Proof. (Theorem 1.3) Consider the algorithm above. Let $k \geq 1$ be any sequence length, and $\mathbf{S}$ be the corresponding mapping computed by the algorithm. Let moreover $1 = k_1, k_2, \ldots, k_h > k$ be the sequence of different values of k' computed by the algorithm. The analysis is trivial for $h = 1$, so assume $h \geq 2$ and hence $k_h \geq 2$. Trivially, for any mapping $\mathbf{S}_i$, increasing the sequence length can only increase the expected cost. Furthermore, for $i \leq h-2$, one has $\mathbf{E}\big[c(\mathbf{S}_{k_i}(\omega^{k_i}))\big] \leq \frac{1}{2}\mathbf{E}\big[c(\mathbf{S}_{k_{i+1}}(\omega^{k_{i+1}}))\big]$. Combining these two observations, one obtains that the expected cost of the solution is bounded by

$$\begin{aligned}\mathbf{E}\big[c(\mathbf{S}(\omega^k))\big] &= \mathbf{E}\big[c(\mathbf{S}_{k_1}(\omega^{k_1}))\big] + \mathbf{E}\big[c(\mathbf{S}_{k_2}(\omega^{k_2-k_1}))\big] + \ldots + \mathbf{E}\big[c(\mathbf{S}_{k_h}(\omega^{k-k_{h-1}}))\big] \\ &\leq \mathbf{E}\big[c(\mathbf{S}_{k_1}(\omega^{k_1}))\big] + \mathbf{E}\big[c(\mathbf{S}_{k_2}(\omega^{k_2}))\big] + \ldots + \mathbf{E}\big[c(\mathbf{S}_{k_h}(\omega^{k_h}))\big] \\ &\leq 2\mathbf{E}\big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\big] + \mathbf{E}\big[c(\mathbf{S}_{k_h}(\omega^{k_h}))\big].\end{aligned}$$

Observe that, if $k_h = \infty$, by construction $\mathbf{E}\big[c(\mathbf{S}_{k_h}(\omega^{k_h}))\big] \leq 2{\cdot}\mathbf{E}\big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\big]$. Similarly, if $k_h < \infty$, and the associated value of k'' satisfies $k''-1 > k_{h-1}$, then $\mathbf{E}\big[c(\mathbf{S}_{k_h}(\omega^{k_h}))\big] = 2 \cdot \mathbf{E}\big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\big]$. Hence, in the two mentioned cases

$$\begin{aligned}\mathbf{E}\big[c(\mathbf{S}(\omega^k))\big] &\leq 4 \cdot \mathbf{E}\big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\big] \\ &\leq O(\log mn) \cdot \mathbf{E}\big[c(\mathsf{opt}(\omega^{k_{h-1}}))\big] \leq O(\log mn) \cdot \mathbf{E}\big[c(\mathsf{opt}(\omega^k))\big].\end{aligned}$$

It remains to consider the case $k_h < \infty$ and $k''-1 = k_{h-1}$. Observe that it must hold that $k_h = k_{h-1}+1 = k+1$. By subadditivity of the optimum solution, $\mathbf{E}\big[c(\mathsf{opt}(\omega^{k_h}))\big] = \mathbf{E}\big[c(\mathsf{opt}(\omega^{k+1}))\big] \leq \frac{k+1}{k}\mathbf{E}\big[c(\mathsf{opt}(\omega^k))\big] = O(\mathbf{E}\big[c(\mathsf{opt}(\omega^k))\big])$. Thus

$$\mathbf{E}\big[c(\mathbf{S}_{k_h}(\omega^{k_h}))\big] = O(\log mn) \cdot \mathbf{E}\big[c(\mathsf{opt}(\omega^{k_h}))\big] = O(\log mn) \cdot \mathbf{E}\big[c(\mathsf{opt}(\omega^k))\big].$$

Also in this case, we can conclude that $\mathbf{E}\big[c(\mathbf{S}(\omega^k))\big] = O(\log mn) \cdot \mathbf{E}\big[c(\mathsf{opt}(\omega^k))\big]$. □

6. Universal Stochastic Facility Location. In this section we consider the universal stochastic version of (non-metric) facility location, a generalization of universal stochastic set cover. For this problem, we provide a $O(\log n)$-competitive length-aware algorithm, where n is the total number of clients and facilities.

The *universal stochastic facility location* problem is defined as follows. An instance of the problem is a set of clients C and a set of facilities F, with a (possibly non-metric) *distance* function $d\colon C \times F \to \mathbb{R}_{\geq 0}$. Each facility $f \in F$ has an opening cost $c(f) \geq 0$. We let $n = |F| + |C|$. Given a mapping $\mathbf{S}\colon C \to F$ of clients into facilities, and a subset $X \subseteq C$, we define $c(\mathbf{S}(X))$ as the total cost of opening facilities in $\mathbf{S}(X) = \cup_{u \in X}\mathbf{S}(u)$ plus the total distance from each $u \in X$ to $\mathbf{S}(u)$. We also denote by $|\mathbf{S}(X)|$ the number of facilities in $\mathbf{S}(X)$. With the usual notation, the aim is

Algorithm 3: Algorithm for the (weighted) stochastic facility location problem.

Data: C, F, $d: C \times F \to \mathbb{R}_{\geq 0}$, $c: F \to \mathbb{R}_{\geq 0}$, k, $\mathsf{apx} \in [\mathbf{E}[c(\mathsf{opt})], 2\mathbf{E}[c(\mathsf{opt})]]$.
while $C \neq \emptyset$ **do**
  let $f \in F$ and $S \subseteq C$ minimize $\mathsf{avg} := \frac{c(f)+\left(1-\left(1-\frac{1}{n}\right)^k\right)\cdot\sum_{v\in S} d(v,f)}{|S|}$;
  if $\mathsf{avg} > \frac{1280\mathrm{e}\cdot\mathsf{apx}}{|C|}$ **then** let $f \in F$ and $S = \{v\} \subseteq C$ minimize $c(f) + d(v, f)$;
  $\mathbf{S}(u) \leftarrow S$ for each $u \in S$;
  $C \leftarrow C \setminus S$;

finding a mapping which minimizes $\mathbf{E}_\omega[c(\mathbf{S}(\omega))]/\mathbf{E}_\omega[c(\mathsf{opt}(\omega))]$, where ω is a random sequence of k clients. As for weighted set cover, we first assume that the algorithm is given as input $\mathbf{E}[c(\mathsf{opt})]$. More precisely, it is sufficient to know a 2 approximation apx of $\mathbf{E}[c(\mathsf{opt})]$. We later show how to remove this assumption[1].

Algorithm 3 in the Figure is an extension of the algorithm from Section 3, where the new challenge is to handle the connection costs for clients. In each iteration, we select a facility and map a subset of clients to it. We first look for a facility f and a subset S of clients such that the average expected cost of opening f and connecting sampled clients in S to f is less than some threshold[2]. If no such set can be found, then the algorithm chooses a facility f and a client v for which the cost of opening f plus the cost of connecting v to f is minimized. Observe that if the connection cost for each client-facility pair is zero then the algorithm reduces (essentially) to Algorithm 2.

We remark that the first step in the while loop can be implemented in polynomial time even if the number of candidate sets S is exponential. In fact, it suffices to consider, for each facility f, the closest i clients still in C, for every $i = 1, \ldots, |C|$.

Analogously to Section 3, we partition the pairs (f, S) computed by the algorithm in two subsets: The pairs computed in the first step of the while loop are of *Type I*, and the remaining pairs of *Type II*. The cost paid by the solution for a pair (f, S) is zero if no element in S is sampled, and otherwise is $c(f)$ plus the sum of the distances from the sampled elements in S to f. We next bound the cost of the pairs of Type I.

LEMMA 6.1. *The expected total cost of Type I pairs is* $O(\log n)\mathbf{E}[c(\mathsf{opt})]$.

Proof. Let (f_i, S_i) be the i-th pair of Type I selected by the algorithm, $i = 1, \ldots, \ell$. Moreover, let C_i denote the set C before f_i was selected. The expected cost paid by our solution for buying f_i and connecting the sampled clients in S_i to f_i is

$$c(f_i)\Pr[S_i \cap \omega \neq \emptyset] + \sum_{v\in S_i} d(v, f_i)\Pr[v \in \omega] \leq c(f_i) + \left(1 - \left(1 - \frac{1}{n}\right)^k\right) \cdot \sum_{v\in S_i} d(v, f_i).$$

Since, the pair (f_i, S_i) is selected in the considered step, the latter quantity is at most $1280\mathrm{e}\cdot\mathsf{apx}\frac{|S_i\cap C_i|}{|C_i|} \leq 2\cdot 1280\mathrm{e}\cdot\mathbf{E}[c(\mathsf{opt})]\frac{|S_i\cap C_i|}{|C_i|}$. The claim follows by the same argument as in Lemma 3.2. □

Consider now the pairs of Type II. We need some notation (analogous to the set cover case). We denote by $(f_i, S_i) = (f_i, \{v_i\})$ the i-th pair of Type II selected by

[1] We remark that the algorithm and analysis in the set cover case can be analogously modified in order to exploit a constant approximation of the optimum.

[2] The constant e in the algorithm can be replaced by a larger constant, so that the algorithm has to deal with polynomially bounded rational numbers only.

the algorithm, $i = 1, \ldots, \ell$, and let $c_i := c(f_i) + d(v_i, f_i)$. We let C_i denote the set C before f_i was selected, $n_i = |C_i|$, and $k_i = n_i \frac{k}{n}$. We also let ω_i be the subsequence (or multi-set) obtained from the random sequence ω by taking only elements belonging to C_i. By $\mathsf{opt}|_{\omega_i}$ we denote the set of facilities in opt which serve clients in ω_i. In particular, $|\mathsf{opt}|_{\omega_i}|$ is the number of such facilities. We let $c(\mathsf{opt}|_{\omega_i})$ be the cost of opening facilities in $\mathsf{opt}|_{\omega_i}$ plus connecting each client in ω_i to the closest facility in $\mathsf{opt}|_{\omega_i}$. (In the connection cost, elements of ω_i are considered without repetitions). Of course, $c(\mathsf{opt}|_{\omega_\ell}) \leq c(\mathsf{opt}|_{\omega_{\ell-1}}) \leq \ldots \leq c(\mathsf{opt}|_{\omega_1}) \leq c(\mathsf{opt}(\omega))$. Additionally, for each client $v \in C$, let D_v denote the expected connection cost paid by opt for v given that v appears in the random sequence ω, and let $D := \left(1 - \left(1 - \frac{1}{n}\right)^k\right) \cdot \sum_{v \in C} D_v$. Observe that D is the total expected connection cost payed by opt. In particular, $D \leq \mathbf{E}[c(\mathsf{opt})]$. We let Ω_i^q and $\Omega_i^{\geq q}$ be the set of ω's such that $|\omega_i| = q$ and $|\omega_i| \geq q$, respectively.

The following lemma is analogous to Lemma 3.3 and is proved in the same way.

LEMMA 6.2. *For every $i \in \{1, \ldots, \ell\}$, if $k_i \geq 8 \log 2n$ then there exists $q \geq \frac{k_i}{2}$ such that*

$$\Pr\left[c(\mathsf{opt}) \leq 8\mathbf{E}[c(\mathsf{opt})] \;\; \textit{and} \;\; |\mathsf{opt}| \leq 8\mathbf{E}[|\mathsf{opt}|] \mid \omega \in \Omega_i^q\right] \geq \frac{1}{2}.$$

Proof. The proof is the same as for Lemma 3.3, where elements are replaced by clients and U_i by C_i. □

Next lemma extends Lemma 3.5. Let us remark that we are considering iterations where there does not exist a pair (f, S) such that $\frac{c(f) + \left(1 - \left(1 - \frac{1}{n}\right)^k\right) \cdot \sum_{v \in S} d(v,f)}{|S|} \leq \frac{1280\mathrm{e} \cdot \mathsf{apx}}{|C|}$, where C is the set of clients at the beginning of the iteration.

LEMMA 6.3. *For every $i \in \{1, \ldots, \ell\}$, if $k_i \geq 8 \log 2n$ then $k_i \leq 64\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] \log n$.*

Proof. Suppose the lemma does not hold, i.e., $k_i > 64\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] \log n$. Apply Lemma 6.2 in order to define q. In particular, in at least one half of the scenarios Ω_i^q the cost and cardinality (i.e., number of facilities) of the optimal solution are at most 8 times the corresponding expected values over all scenarios. We show a contradiction by providing a pair (f, S) which violates the condition in the second step of the while loop of Algorithm 3.

Consider the n_i^q different outcomes for a sequence of q clients in C_i with repetitions. In the optimal solution, disconnect v from its closest facility in any scenario where $D_v \geq 4\mathbf{E}[D_v \mid \omega \in \Omega_i^q]$. We next show that in at least two thirds of the scenarios the reduced optimum solution still connects at least $\frac{q}{4}$ clients from C_i (counting repetitions). To see that, define a 0-1 matrix, which has a row for each pair $(v, i) \in C_i \times \{1, \ldots, q\}$ and a column for each scenario $\omega \in \Omega_i^q$. Entry $((v, i), \omega)$ is initially set to one if and only if v appears at least i times in ω. Hence, there are qn_i^q ones altogether. Now zero the entries $((v, i), \omega)$ where v is disconnected in scenario ω. By Markov's inequality, we zero at most one fourth of the ones of each row (v, i). Hence the number of ones becomes at least $\frac{3}{4} qn_i^q$. Let x denote the number of columns with less than $\frac{q}{4}$ ones, i.e., the number of scenarios where less that $\frac{q}{4}$ clients are connected (counting repetitions). One has

$$\frac{3}{4} qn_i^q \leq \frac{q}{4} \cdot x + q \cdot (n_i^q - x) \quad \Rightarrow \quad x \leq \frac{n_i^q}{3}.$$

It follows from the union bound and Lemma 6.2 that at least a fraction $1-\frac{1}{2}-\frac{1}{3}=\frac{1}{6}$ of the scenarios will satisfy the conditions of Lemma 6.2 and additionally connect at least $\frac{q}{4}$ elements in the reduced optimum solution.

For each facility f, let S_f denote the set of clients v within distance at most $4\mathbf{E}[D_v \mid \omega \in \Omega_i^q]$ from f. Consider the collections of at most $d = 8\mathbf{E}[|\mathsf{opt}|]$ facilities with total weight of opening all the d facilities less than or equal to $8\mathbf{E}[c(\mathsf{opt})]$. Let F_j, $1 \le j \le p \le (2n)^d$, be the j-th such collection, and let $\cup F_j := \cup_{f\in F_j} S_f$ be the union of the S_f over the facilities of the collection. We must have:

$$\sum_{j=1}^{p} \binom{q}{\frac{q}{4}} |\cup F_j|^{\frac{q}{4}} n_i^{\frac{3q}{4}} \ge \frac{1}{6} n_i^q.$$

Using $\binom{q}{\frac{q}{4}} \le \left(\frac{\mathrm{e}q}{\frac{q}{4}}\right)^{\frac{q}{4}} = (4\mathrm{e})^{\frac{q}{4}}$ and the assumption that $q \ge \frac{k_i}{2} > 32\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]\log n \ge 4d\log n$, there must be one F_j such that:

$$|\cup F_j| \ge \frac{n_i}{\left(\binom{q}{q/4} 6p\right)^{4/q}} \ge \frac{n_i}{4\mathrm{e}\cdot 6^{\frac{4}{q}} (2n)^{\frac{4d}{q}}} \ge \frac{n_i}{4\mathrm{e}\cdot 6^{\frac{1}{8\log n}} 2^{\frac{1}{\log n}} n^{\frac{1}{\log n}}} \ge \frac{n_i}{32\mathrm{e}}.$$

The cost of opening all the facilities F_j is at most $8\mathbf{E}[c(\mathsf{opt})]$. Furthermore, for the considered scenarios we have

$$\begin{aligned}\sum_{f\in F_j}\sum_{v\in S_f}\left(1-\left(1-\frac{1}{n}\right)^k\right)\cdot d(v,f) &\le \sum_{f\in F_j}\sum_{v\in S_f}\left(1-\left(1-\frac{1}{n}\right)^k\right)\cdot 4\mathbf{E}[D_v \mid \omega\in\Omega_i^q] \\ &\le 4\mathbf{E}[D \mid \omega\in\Omega_i^q] \\ &\le 4\mathbf{E}[c(\mathsf{opt}) \mid \omega\in\Omega_i^q] \le 32\mathbf{E}[c(\mathsf{opt})].\end{aligned}$$

Hence, by an averaging argument, there must exist a facility f such that

$$\begin{aligned}\frac{c(f)+\sum_{v\in S_f}\left(1-\left(1-\frac{1}{n}\right)^k\right)\cdot d(v,f)}{|S_f|} &\le \frac{8\mathbf{E}[c(\mathsf{opt})]+32\mathbf{E}[c(\mathsf{opt})]}{n_i/32\mathrm{e}} \\ &= \frac{1280\mathrm{e}\mathbf{E}[c(\mathsf{opt})]}{n_i} \le \frac{1280\mathrm{e}\cdot \mathsf{apx}}{n_i}.\end{aligned}$$

This contradicts the fact that a pair of Type II is selected in the iteration considered. □

The following lemma is analogous to Lemma 3.4

LEMMA 6.4. *For each* $1 \le i \le \ell$,

(a) $c_i\mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big] \le \mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big]$;

(b) $c_i(\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] - \mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big]) \le \mathbf{E}[c(\mathsf{opt}|_{\omega_i})] - \mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big]$.

Proof. By construction, $c_i \le c_{i+1}$. Furthermore, for any facility $f \in \mathsf{opt}|_{\omega_{i+1}}$ and any client $v \in \omega_{i+1}$ served by f (there must be at least one such client), one has $c_{i+1} \le c(f) + d(v,f)$. Consequently, $c_{i+1}\cdot|\mathsf{opt}|_{\omega_{i+1}}|$ is a lower bound on $c(\mathsf{opt}|_{\omega_{i+1}})$. Altogether:

$$c_i|\mathsf{opt}|_{\omega_{i+1}}| \le c_{i+1}|\mathsf{opt}|_{\omega_{i+1}}| \le c(\mathsf{opt}|_{\omega_{i+1}}).$$

By applying the same argument to the facilities in $\mathsf{opt}|_{\omega_i} \setminus \mathsf{opt}|_{\omega_{i+1}}$:

$$c_i(|\mathsf{opt}|_{\omega_i}| - |\mathsf{opt}|_{\omega_{i+1}}|) \le c(\mathsf{opt}|_{\omega_i}) - c(\mathsf{opt}|_{\omega_{i+1}}).$$

The claim follows by taking expectations. □

We now bound the cost of the Type II pairs in the solution constructed by the algorithm.

LEMMA 6.5. *The expected cost of Type II pairs selected by Algorithm 3 is* $O(\log n) \cdot \mathbf{E}[c(\mathsf{opt})]$.

Proof. For notational convenience, we set $k_{\ell+1} = 0$ and $c_0 = 0$. Moreover, let j be such that $k_j \geq 8 \log 2n$ and $k_{j+1} < 8 \log 2n$. Note that in expectation we see at most $8 \log 2n$ elements in C_{j+1}. Each of these elements are connected in opt to a facility f for which $c(f) + d(v_i, f)$ is not smaller than c_i. Hence, the cost of connecting v_i's to f_i's for $j < i \leq \ell$ is bounded by $8 \log 2n \mathbf{E}[c(\mathsf{opt})]$.

By Lemma 6.3 and Lemma 6.4, the cost of the pairs $(f_1, S_1), \ldots, (f_j, S_j)$ is upper bounded by

$$\begin{aligned}
\sum_{i=1}^{j} c_i \Pr[v_i \in \omega] &= \sum_{i=1}^{j} c_i (k_i - k_{i+1}) \leq \sum_{i=1}^{j} k_i (c_i - c_{i-1}) \\
&\overset{\text{Lem. 6.3}}{\leq} \sum_{i=1}^{j} 64 \mathbf{E}[|\mathsf{opt}|_{\omega_i}|] \log n (c_i - c_{i-1}) \\
&= 64 \log n \left(c_j \mathbf{E}\big[|\mathsf{opt}|_{\omega_{j+1}}|\big] + \sum_{i=1}^{j} c_i (\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] - \mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big]) \right) \\
&\overset{\text{Lem. 6.4}}{\leq} 64 \log n \left(\mathbf{E}\big[c(\mathsf{opt}|_{\omega_{j+1}})\big] + \sum_{i=1}^{j} \big(\mathbf{E}[c(\mathsf{opt}|_{\omega_i})] - \mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big]\big) \right) \\
&= 64 \log(n) \mathbf{E}[c(\mathsf{opt}|_{\omega_1})] \leq 64 \log(n) \mathbf{E}[c(\mathsf{opt})].
\end{aligned}$$

The claim follows. □

The following lemma follows immediately from Lemmas 6.1 and 6.5.

LEMMA 6.6. *Algorithm 3 returns a universal mapping* S *to the universal stochastic facility location problem with* $\mathbf{E}[c(\mathsf{S})] = O(\log n) \mathbf{E}[c(\mathsf{opt})]$.

Using the above lemma it is easy to prove the main theorem.

THEOREM 6.7. *There exists a polynomial-time length-aware algorithm that returns a universal mapping* S *to the universal stochastic facility location problem with* $\mathbf{E}[c(\mathsf{S})] = O(\log n) \mathbf{E}[c(\mathsf{opt})]$.

Proof. First, note that the value of $\mathbf{E}_{\omega \in C^1}[c(\mathsf{opt}(\omega))]$ can be easily computed, by finding for each $v \in C$ the facility f minimizing $c(f) + d(c, f)$. Trivially, for $1 \leq k \leq n$,

$$\mathbf{E}_{\omega \in C^1}[c(\mathsf{opt}(\omega))] \leq \mathbf{E}_{\omega \in C^k}[c(\mathsf{opt}(\omega))] \leq \mathbf{E}_{\omega \in C^n}[c(\mathsf{opt}(\omega))].$$

Moreover, by subadditivity, $\mathbf{E}_{\omega \in C^n}[c(\mathsf{opt}(\omega))] \leq n \mathbf{E}_{\omega \in C^1}[c(\mathsf{opt}(\omega))]$. Hence one of the values $x_i := 2^i \mathbf{E}_{\omega \in C^1}[c(\mathsf{opt}(\omega))]$ for $0 \leq i \leq \log n$ is a 2-approximation for $\mathbf{E}_{\omega \in C^k}[c(\mathsf{opt}(\omega))] = \mathbf{E}[c(\mathsf{opt})]$. Therefore, it is sufficient to run Algorithm 3 with $\mathsf{apx} = x_i$, for all the $\log n$ values x_i. This way one obtains $\log n$ different mappings. Afterwards, we choose the one with the smallest expected cost, which is guaranteed to be $O(\log n)$ approximate. The expected costs above can be computed analogously to the set cover case. □

The same reduction as in Section 5 leads to an $O(\log n)$-competitive algorithm for the online version of the problem.

THEOREM 6.8. *There is an* $O(\log n)$*-competitive algorithm for the online stochastic facility location problem.*

Proof. Consider the same algorithm as in the set cover case, where now each mapping $\mathsf{S}_{k'}$ is computed with the universal facility location algorithm of this section. The proof of the lemma follows along the same line as in Theorem 1.3, with the difference that now the competitive factor of each mapping is $O(\log n)$ rather than $O(\log mn)$. □

7. Non-Uniform Probability Distributions. In this section we show how to handle the case of non-uniform probability distributions, given our results for the uniform case. For the sake of simplicity, we focus on the set cover case: The same argument works for facility location.

Consider the following reduction to the uniform case. Let $(U, \mathscr{S}, \vec{\pi})$ be the input, where $(U, \mathscr{S})$ is the set system and $\vec{\pi} \in \mathbb{Q}_{\geq 0}^{|U|}$ is the probability distribution (i.e., π_u is the probability of sampling element u and $\sum_{u \in U} \pi_u = 1$). We multiply all the probabilities (which are rational numbers) by a proper integer N, so that $n_u := \pi_u \cdot N$ is integral. (Without loss of generality, $N \gg 1$). Now we replace u with n_i copies $u_1, u_2, \ldots, u_{n_i}$. Let $(U', \mathscr{S}')$ be the resulting set system, where sets $S' \in \mathscr{S}'$ correspond to sets $S \in \mathscr{S}$ and $|U'| = N$. We run our universal algorithm for the uniform case on $(U', \mathscr{S}')$. It is easy to see that all the copies u_i will be mapped to the same set S'_u (if we break ties consistently): we return the solution which maps each u into S_u.

Observe that $\Pr[u \text{ is sampled}] = \pi_u$ and $\Pr[\text{at least one } u_i \text{ is sampled}] = 1 - (1 - 1/N)^{n_u} \simeq 1 - \mathrm{e}^{-\pi_u}$. In particular, the two probabilities differ at most by a constant factor. It follows by subadditivity that the expected optimal cost of the modified instance is within a constant factor from the original expected optimal cost. Hence the solution produced is $O(\log mn)$ approximate.

Of course, the reduction above is not polynomial. However, we can consider an implicit representation of $(U', \mathscr{S}')$, where we associate a multiplicity n_u to each element u. It is not hard to adapt our algorithms to make them run in polynomial time with this implicit representation. In particular, in Algorithms 1 and 2 one can compute $|S \cap U|$ for a given set S in time polynomial in $|U|$ and in the number of bits needed to represent the probability distribution. Furthermore, all the comparisons in the mentioned algorithms involve rational numbers which can be represented with a polynomial number of bits in the input size.

8. The Independent Activation Model. In the *independent activation model* each element (client) u is independently sampled with a given probability π_u. In particular, the number K of sampled elements (clients) is a random variable in $\{0, 1, \ldots, n\}$. Due to concentration bounds, our algorithms can be adapted to work in this case as well.

Let $\tilde{k} := \mathbf{E}[K] = \sum_{u \in U} \pi_u$ be the expected number of sampled elements (clients). Simple greedy algorithms ensure a K-approximation, and hence an expected $\tilde{k}$-approximation. In more detail, it is sufficient to map each element u in the cheapest set S containing it, in the set cover case. For facility location, we map each client u into the facility f which minimizes $c(f) + d(u, f)$. It is easy to see that the resulting mapping costs at most $K \cdot \mathsf{opt}$.

Hence, without loss of generality, we can assume $\tilde{k} \geq c \log n$ for a sufficiently large constant $c > 0$. In this case we run our algorithms with $k = 2\tilde{k}$. For any given $\beta > 0$ and c large enough, Chernoff's bound guarantees that $\Pr[K > k] \leq n^{-\beta}$. Hence it is sufficient to show that our algorithms are n^{β} approximate in the worst case, for a

proper constant $\beta > 0$. We next show that this is the case.

LEMMA 8.1. *Any universal mapping for the unweighted set cover problem is n-approximate in the worst case.*

Proof. The optimal solution needs at least one set whereas the mapping returns at most n sets. The claim follows. □

LEMMA 8.2. *The universal mapping $\mathbf{S}$ generated by Algorithm 2 is n^2-approximate in the worst case for the set cover problem.*

Proof. Consider any sequence ω of k elements, and let $\overline{\omega}$ be the corresponding set. Let $cheap(v)$ be the minimum cost of a set covering v. Observe that $\mathsf{opt}(\omega) \geq \frac{1}{n}\sum_{v\in\overline{\omega}} cheap(v)$. For any element $v \in \overline{\omega}$ covered by a Type I set, it holds that $c(\mathbf{S}(v)) \leq n \cdot cheap(v)$. For the remaining elements $v \in \overline{\omega}$, $c(\mathbf{S}(v)) = cheap(v)$. As a consequence the cost of the solution returned by the algorithm is at most $\sum_{v\in\overline{\omega}} n \cdot cheap(v)$. The claim follows. □

LEMMA 8.3. *The universal mapping $\mathbf{S}$ generated by Algorithm 3 is n^3-approximate in the worst case for the facility location problem.*

Proof. Consider any sequence ω of k clients, and let $\overline{\omega}$ be the corresponding set. For any $v \in \overline{\omega}$, let $cheap(v) = \min_{f\in F}\{c(f) + d(v,f)\}$. Observe that $\mathsf{opt}(\omega) \geq \frac{1}{n}\sum_{v\in\overline{\omega}} cheap(v)$. The cost paid by $\mathbf{S}$ for any $v \in \overline{\omega}$ covered by a facility of Type II is at most $cheap(v)$. Consider now any $v \in \overline{\omega}$ covered by a facility $f = \mathbf{S}(v)$ of Type I. Let $S = \mathbf{S}^{-1}(f)$. $\mathbf{S}$ pays for v at most

$$c(f) + d(v,f) \leq c(f) + \sum_{v\in S} d(v,f) \leq n \cdot \left(c(f) + \left(1 - \left(1 - \frac{1}{n}\right)^k\right) \sum_{v\in S} d(v,f) \right)$$
$$\leq n^2 \cdot \left(\frac{c(f) + \left(1 - \left(1 - \frac{1}{n}\right)^k\right) \sum_{v\in S} d(v,f)}{|S \cap C|} \right) \leq n^2 \cdot cheap(v).$$

Altogether, the cost of the solution returned by the algorithm is at most $n^2 \sum_{v\in\overline{\omega}} cheap(v)$. The claim follows. □

9. Other Applications. Our techniques can be applied to other *covering-like* problems. In this section we sketch two such applications.

9.1. Universal Stochastic Multi-Cut. In an instance of the *universal multi-cut* problem we are given a graph $G = (V, E)$ with edge costs $c\colon E \to \mathbb{R}_{\geq 0}$, and a set of demand pairs $D = \{(s_i, t_i) : 1 \leq i \leq m\}$. The task is to return a mapping $\mathbf{S}\colon D \to 2^E$ so that $\mathbf{S}((s_i,t_i)) \subseteq E$ disconnects s_i from t_i. The cost of the solution for a sequence $\omega \in D^k$, i.e., the total cost of the edges in $\mathbf{S}(\omega) = \cup_{(s_i,t_i)\in\omega}\mathbf{S}((s_i,t_i))$, is denoted by $c(\mathbf{S}(\omega))$. The universal and online stochastic versions are defined analogously, and again the goal is to minimize the ratio $\mathbf{E}_\omega[c(\mathbf{S}(\omega))]/\mathbf{E}_\omega[c(\mathsf{opt}(\omega))]$.

Notice first that the multi-cut problem on trees (i.e., when the graph G is a tree) is a special case of weighted set cover: each demand pair (s_i, t_i) is an element in U, each edge e corresponds to a set S_e, and an element (s_i, t_i) is contained in a set S_e if the edge e lies on the unique path from s_i to t_i. Hence, we can use the algorithm from Section 3 to obtain a $O(\log n)$-competitive algorithm for stochastic universal multi-cut on trees.

Now, using the congestion-preserving hierarchical decompositions of Räcke [44], we can generalize this result to arbitrary graphs obtaining a $O(\log^2 n)$-competitive algorithm. The randomized reduction from multi-cut in general graphs to multi-cut in trees is detailed in [44, Section 3], and shows that an $O(\alpha)$-approximation

to the multi-cut problem on trees gives an $O(\alpha \log n)$-approximation to multi-cut on general graphs. Using our result for multi-cut on trees immediately gives the following theorem.

THEOREM 9.1. *There exists an $O(\log^2 n)$-competitive polynomial-time algorithm for the online stochastic multi-cut problem, and a polynomial-time algorithm that, given the length of the input sequence, is $O(\log^2 n)$-competitive for the universal stochastic multi-cut problem.*

9.2. Constant VC-Dimension. In this section we present better algorithms for universal stochastic (unweighted) set cover for the case when the set system has constant VC-dimension. (For formal definitions of VC-dimension and related concepts, see, e.g., [10]).

Before presenting our improved algorithm, let us describe an application where it is potentially useful. Consider a region $U \subseteq \mathbb{R}^2$ of the 2-dimensional plane, and a set of m "base-stations" $v_i \in \mathbb{R}^2$, each with a coverage radius r_i. In other words, v_i covers the disk $\mathbf{B}(v_i, r_i)$ of radius r_i centered at v_i. We assume that $U \subseteq \cup_i \mathbf{B}(v_i, r_i)$, i.e., the discs cover the entire region. Given a set $X \subseteq U$, the goal is to find a small set cover, i.e., to map each point $x \in X$ to a base-station covering it so that not too many base-stations are in use. This problem was studied by Hochbaum and Maas [30], and by Brönnimann and Goodrich [10], among others. However, one might want to hard-wire this mapping from locations in the plane to base-stations, so that we do not have to solve a set-cover problem each time a device wants to access a base-station; i.e., we want a *universal mapping*. For ease of exposition, let us discretize the plane into n points by placing a fine-enough mesh on the plane. Assume that the locations to be covered are chosen randomly from some known distribution from the plane (or more precisely, from this mesh). In this case, we can show that there *exists* a universal mapping whose expected set-cover cost is at most $O(\log m)$ times the expected optimum, and this mapping can be computed in randomized polynomial-time.

By the discussion in Section 2 (and with the same notation), the main challenge is to find the 2μ sets which cover all but δn elements. When the set system has VC-dimension d, we give a polynomial-time algorithm in Lemma 9.2 which finds $O(d2^d \mu \log \mu)$ sets which cover all but $2\delta n$ elements; for the special case of discs in the plane, our algorithm improves and returns just $O(\mu)$ sets. Combining this result with the argument in Theorem 1.2 implies a polynomial-time $O(\log m)$-competitive algorithm for the case of discs in the plane, and an $O(2^d \log \mu + \log m)$-competitive algorithm in general.

Let us state our result more generally: the (unweighted) *partial hitting set* problem takes as input a set system $(U, \mathscr{S})$ and an integer threshold $\tau \leq m = |\mathscr{S}|$. A feasible solution to this problem is subset $U' \subseteq U$ of elements such that all but τ sets from $\mathscr{S}$ contain at least one element from U'. The goal is to find a feasible solution with smallest cardinality. We give the following bi-criteria approximation algorithm for this problem. Our algorithm and analysis for partial hitting set builds on results for hitting set for bounded VC-dimension set systems [10].

To state our theorem formally, we introduce some notation: any weight function $w : U \to \mathbb{R}_{\geq 0}$ on the elements induces a weight function on the sets, where the weight of a set $S \subseteq U$ is $\sum_{e \in S} w(e)$—we denote the weight of S by $w(S)$. Given element weights and a parameter $\epsilon > 0$, an *ϵ-net* $A \subseteq U$ is a set of elements which hits all sets in $\mathscr{S}$ with weight at least $\epsilon\, w(U)$. We use $s(\epsilon)$ to denote an upper bound on the size of ϵ-nets for a set system. For any set system with VC-dimension d, it is known that

$s(\epsilon) = O((d/\epsilon)\log(d/\epsilon))$, and that such a net can be found in polynomial time (see, e.g., [10]. For the special case of disks in the plane, $s(\epsilon) = O(d/\epsilon)$ [41].

LEMMA 9.2. *There is a polynomial-time algorithm that, given an instance of partial hitting set where the optimal solution contains k elements and hits all but τ sets, outputs a subset with $s(4k)$ elements that hits all but 2τ elements, where d is the VC-dimension of the set system and $s(\epsilon)$ is an upper bound of the size of ϵ-nets in the set system.*

Proof. We assume we know k, the size of the optimal partial hitting set—this assumption can be discharged by running over all possible values of k. The algorithm proceeds in iterations. Initially, we give unit weight to each element, set $\epsilon = \frac{1}{4k}$, and find an ϵ-net A. If the net A hits all but 2τ sets, we stop and return A. Else, we pick a set R that is not hit uniformly at random, double the weight of all the elements in R, and go to the next iteration. By construction, the hitting set returned by the algorithm is of size $s(4k)$, so it suffices to show that the algorithm terminates in polynomial time.

We claim that the algorithm terminates in $O(k \log \frac{n}{k})$ iterations. Let A^t be the ϵ-net computed at iteration t. Let also $w^t(U)$ be the weight of U at the end of the same iteration ($w^0(U) = n$ is the initial weight). Any set R which is not hit by A_t must have weight at most $\epsilon\, w^{t-1}(U)$, since A^t is an ϵ-net. As a consequence, the total weight grows at most by a factor $(1+\epsilon)$ at each iteration. We can conclude by induction that

$$w^t(U) \leq n(1+\epsilon)^t \leq n\mathrm{e}^{\epsilon t}. \tag{9.1}$$

Let H^* denote the set of size k which hits all but τ sets. For each $h \in H^*$, let Z_h^t be the random variable denoting the number of iterations $i \leq t$ where the weight of the element h is doubled. For any iteration $i \leq t$, at least 2τ sets are not hit by A^i, and hence at least half of the sets not hit by A^i are hit by H^*. Thus, at any iteration, we double the weight of some element in H^* with probability at least $\frac{1}{2}$. We can conclude that $\sum_{h\in H^*} \mathbf{E}[Z_h^t] \geq \frac{t}{2}$. Observe that $w^t(H^*) = \sum_{h\in H^*} 2^{Z_h^t}$. Now using the fact that $\mathbf{E}\big[2^Y\big] \geq 2^{\mathbf{E}[Y]}$ for any random variable Y, we get that $\mathbf{E}[w(H^*)] \geq \sum_{h\in H^*} 2^{\mathbf{E}[Z_h]}$. Since $\sum_{h\in H^*} \mathbf{E}[Z_h] \geq \frac{t}{2}$, we can use the convexity of the exponential function to claim that $\sum_{h\in H^*} 2^{\mathbf{E}[Z_h]}$ is minimized when all terms $\mathbf{E}[Z_h]$ are equal—this gives us

$$\mathbf{E}\big[w^t(U)\big] \geq \mathbf{E}\big[w^t(H^*)\big] \geq k \cdot 2^{t/2k}. \tag{9.2}$$

Combining equations (9.1) and (9.2) and using the fact that $\epsilon = 1/4k$, we get that $k \cdot 2^{t/2k} \leq n\mathrm{e}^{t/4k}$, from which we obtain that $t \leq 8k\log(n/k)$. Hence, the algorithm terminates after $O(k\log(n/k))$ iterations, as claimed. This completes the proof of the lemma. □

COROLLARY 9.3. *Given a set system $(U, \mathscr{S})$ such that some 2μ sets from $\mathscr{S}$ cover all but δn elements from U, there is a polynomial time algorithm that outputs $O(2^{d+2}\mu\log(2^{d+2}\mu))$ sets that cover all but $2\delta n$ elements, where d is the VC-dimension of the set system. For the special case where sets are discs in the plane, the number of sets is $O(2^d\mu)$.*

Proof. Given the set system $(U, \mathscr{S})$, construct the dual set system $(U^*, \mathscr{S}^*)$ where $U^* = \mathscr{S}$ and $\mathscr{S}^* = U$, such that a set $x \in \mathscr{S}^*$ contains an element $S \in U^*$ if S contained x in the original set system. Consequently, the partial set cover problem on the set system $(U, \mathscr{S})$ ("pick the fewest sets from $\mathscr{S}$ that cover all elements from U but at most τ elements") is identical to the partial hitting set problem in the dual

set system $(U^*, \mathscr{S}^*)$ ("pick the fewest elements from U^* that hit all sets from $\mathscr{S}^*$ but at most τ sets"). Moreover, if the VC-dimension of the original set system is d, then the VC-dimension of the dual set system is at most 2^{d+1} [10].

Finally, since there are at most 2μ dual elements in U^* that hit all but δn dual sets in $\mathscr{S}^*$, and the VC-dimension d^* of the dual set system is at most 2^{d+1}, Lemma 9.2 and the fact that $s(\epsilon) = O(d^*/\epsilon \log d^*/\epsilon)$ immediately implies that we can find $O(2^{d+1} \cdot 2\mu \log(2^{d+1} \log 2\mu))$ elements in the dual set system that hit the same number of sets. Considering the sets in the original set system corresponding to these dual elements completes the proof. For the second part of the proof, we use the fact that $s(\epsilon) = O(d^*/\epsilon)$ for discs in the plane in this argument to get the bound of $O(2^d\mu)$. □

The following theorem summarizes the discussion above.

THEOREM 9.4. *There exists a randomized polynomial-time length-oblivious algorithm that returns a universal mapping* $\mathbf{S}$ *to the universal stochastic unweighted set cover problem on set systems of VC-dimension d, with* $\mathbf{E}[|\mathbf{S}|] = O(2^d \log \mathbf{E}[|\mathsf{opt}|] + \log m)\, \mathbf{E}[|\mathsf{opt}|]$. *For the special case of discs in the plane, we get an algorithm for universal stochastic unweighted set cover with* $\mathbf{E}[|\mathbf{S}|] = O(\log m)\, \mathbf{E}[|\mathsf{opt}|]$

REFERENCES

[1] S. Albers and S. Leonardi. On-line Algorithms. *ACM Computing Surveys*, 31(3es):4, 1999.

[2] 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.

[3] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A General Approach to Online Network Optimization Problems. *ACM Transactions on Algorithms*, 2(4): 640–660, 2006.

[4] A. Anagnostopoulos, F. Grandoni, S. Leonardi, and P. Sankowski. Online network design with outliers. In *Proceedings of the 37th Annual International Colloquium on Automata, Languages and Programming*, 2010. To appear.

[5] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal Oblivious Routing in Polynomial Time. *Journal of Computer and System Sciences*, 69(3): 383–394, 2004.

[6] D. Bertsimas and M. Grigni. Worst-case Examples for the Spacefilling Curve Heuristic for the Euclidean Traveling Salesman problem. *Operations Research Letters*, 8(5):241–244, 1989.

[7] D. J. Bertsimas, P. Jaillet, and A. R. Odoni. A Priori Optimization. *Operations Research*, 38(6):1019–1033, 1990.

[8] M. Bienkowski, M. Korzeniowski, and H. Räcke. A Practical Algorithm for Constructing Oblivious Routing Schemes. In *Proceedings of the 15th annual ACM symposium on Parallel Algorithms and Architectures*, pages 24–33, 2003.

[9] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, New York, 1998.

[10] H. Brönnimann and M. T. Goodrich. Almost Optimal Set covers in Finite VC-Dimension. *Discrete Computational Geometry*, 14(4):463–479, 1995.

[11] N. Buchbinder and J. Naor. Online Primal-Dual Algorithms for Covering and Packing. *Mathematics of Operations Research*, 34(2): 270–286, 2009.

[12] J. Cardinal, S. Fiorini, and G. Joret. Tight Results on Minimum Entropy Set Cover. *Algorithmica*, 51(1): 49–60, 2008.

[13] R. Dorrigiv and A. Lopez-Ortiz. A Survey of Performance Measures for On-line Algorithms. *SIGACT News*, 36(3):67–81, 2005.

[14] U. Feige. A Threshold of $\ln n$ for Approximating Set Cover. *Journal of the ACM*, 45(4):634–652, 1998.

[15] U. Feige and S. Korman. On the use of Randomization in the Online Set Cover Problem. *Technical Report*.

[16] U. Feige, L. Lovász, and P. Tetali. Approximating Min Sum Set Cover. *Algorithmica*, 40(4):219–234, 2004.

[17] A. Fiat and G. J. Woeginger, editors. *Online Algorithms*, volume 1442 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1998.

[18] D. Fotakis. On the Competitive Ratio for Online Facility Location. *Algorithmica*, 50(1): 1–57, 2008.

[19] P. R. Freeman. The Secretary Problem and its Extensions: a Review. *International Statistical Review*, 51(2):189–206, 1983.

[20] N. Garg, A. Gupta, S. Leonardi, and P. Sankowski. Stochastic Analyses for Online Combinatorial Optimization Problems. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 942–951, 2008.

[21] F. Grandoni, A. Gupta, S. Leonardi, P. Miettinen, P. Sankowski, and M. Singh. Set Covering with our Eyes Closed. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 347–356, 2008.

[22] A. Gupta, M. T. Hajiaghayi, and H. Räcke. Oblivious Network Design. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 970–979, 2006.

[23] M. T. Hajiaghayi, J. H. Kim, T. Leighton, and H. Räcke. Oblivious Routing in Directed Graphs with Random Demands. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 193–201, 2005.

[24] M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive Limited-Supply Online Auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 71–80, 2004.

[25] M. T. Hajiaghayi, R. D. Kleinberg, and F. T. Leighton. Improved Lower and Upper Bounds for Universal TSP in Planar Metrics. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 649–658, 2006.

[26] M. T. Hajiaghayi, R. D. Kleinberg, T. Leighton, and H. Räcke. Oblivious Routing on Node-Capacitated and Directed Graphs. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 782–790, 2005.

[27] E. Halperin and R. M. Karp. The Minimum-Entropy Set Cover Problem. *Theoretical Computer Science*, 348(2-3):240–250, 2005.

[28] C. Harrelson, K. Hildrum, and S. Rao. A Polynomial-time Tree Decomposition to minimize Congestion. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 34–43, 2003.

[29] D. S. Hochbaum. Heuristics for the Fixed Cost Median Problem. *Mathematical Programming*, 22(1):148–162, Dec 1982.

[30] D. S. Hochbaum and W. Maass. Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI. *Journal of the ACM*, 32(1):130–136, 1985.

[31] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the Costs and Benefits of Procrastination: Approximation Algorithms for Stochastic Combinatorial Optimization Problems. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 684–693, 2004.

[32] P. Jaillet. A Priori Solution of a Travelling Salesman Problem in which a Random Subset of the Customers are Visited. *Operations Research*, 36(6):929–936, 1988.

[33] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal Approximations for TSP, Steiner Tree, and Set Cover. In *Proceedings of 37th ACM Symposium on Theory of Computing*, pages 386–395, 2005.

[34] D. S. Johnson. Approximation Algorithms for Combinatorial Problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[35] D. R. Karger and M. Minkoff. Building Steiner Trees with Incomplete Global Knowledge. In *Proceedings of 41st Annual Symposium on Foundations of Computer Science*, pages 613–623, 2000.

[36] M. J. Kearns. *The Computational Complexity of Machine Learning*. MIT Press, 1990.

[37] L. Lovász. On the Ratio of Optimal Integral and Fractional Covers. *Discrete Mathematics*, 13(4):383–390, 1975.

[38] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. AdWords and Generalized Online Matching. *Journal of the ACM*, 54(5):Art. 22, 19 pp., 2007.

[39] A. Meyerson. Online Facility Location. In *Proceedings of 42nd Annual Symposium on Foundations of Computer Science*, pages 426–431. 2001.

[40] A. Meyerson, K. Munagala, and S. Plotkin. Designing Networks Incrementally. In *Proceedings of 42nd Annual Symposium on Foundations of Computer Science*, pages 406–415, 2001.

[41] J. Matoušek, R. Seidel, and E. Welzl. How to net a lot with little: small ϵ-nets for disks and halfspaces. In *Proceedings of the 6th Annual Symposium on Computational Geometry*, pages 16–22, 1990.

[42] L. K. Platzman and J. J. Bartholdi, III. Spacefilling Curves and the Planar Travelling Salesman Problem. *Journal of the ACM*, 36(4):719–737, 1989.

[43] H. Räcke. Minimizing Congestion in General Networks. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pages 43–52, 2002.

[44] H. Räcke. Optimal Hierarchical Decompositions for Congestion Minimization in Networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 2008.

[45] R. Raz and S. Safra. A Sub-constant Error-probability Low-degree Test, and a Sub-constant Error-probability PCP Characterization of NP. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pages 475–484, 1997.

[46] F. Schalekamp and D. B. Shmoys. Algorithms for the Universal and A Priori TSP. *Operations Research Letters*, 36(1):1–3, Jan 2008.

[47] D. B. Shmoys and C. Swamy. An Approximation Scheme for Stochastic Linear Programming and its Application to Stochastic Integer Programs. *Journal of the ACM*, 53(6):978–1012, 2006.

[48] D. B. Shmoys and K. Talwar. A Constant Approximation Algorithm for the A Priori Traveling Salesman Problem. In *Proceedings of the 13th Conference on Integer Programming and Combinatorial Optimization.*, 2008.

[49] A. Srinivasan. Approximation Algorithms for Stochastic and Risk-averse Optimization. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1305–1313, 2007.

[50] L. G. Valiant and G. J. Brebner. Universal Schemes for Parallel Communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, pages 263–277, 1981.

[51] B. Vöcking. Almost Optimal Permutation Routing on Hypercubes. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 530–539, 2001.