Set Covering with Our Eyes Closed*

Fabrizio Grandoni† Anupam Gupta‡ Stefano Leonardi§ Pauli Miettinen¶
Piotr Sankowski§ ‖ 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, so that $X \subseteq U$ is covered by $\mathbf{S}(X) = \cup_{u \in X} \mathbf{S}(u)$. The aim is finding 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* stochastic online algorithms *with the same competitive factors.*

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 $c(\mathscr{S}') := \sum_{S \in \mathscr{S}'} c(S)$. Each feasible solution can also be interpreted 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 minimizing the number $|\mathbf{S}(X)|$ of subsets used to cover X.

In their seminal work, Jia et al. [37] 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 on-

*Part of this work was done when the non-Roman authors were visiting the "Sapienza" Università di Roma.

†Dipartimento di Informatica, Sistemi e Produzione, Università di Roma "Tor Vergata", via del Politecnico 1, 00133, Roma, Italy. Partially supported by MIUR under project MAINSTREAM.

‡Carnegie Mellon University. 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. Partially supported by the EU within the 6th Framework Programme under contract no. 001907 "Dynamically Evolving, Large Scale Information Systems" (DELIS)

¶Helsinki Institute for Information Technology, University of Helsinki, P.O. box 68 (Gustaf Hällströmin katu 2b), 00014 Helsinki, Finland.

‖Institute of Informatics, University of Warsaw, ul. Banacha 2, 02097 Warsaw, Poland

**Microsoft Research, New England, Cambridge, USA. Part of the work was done when the author was at Carnegie Mellon University.

line) algorithms assumes that the input is chosen adversarially, and often this setting is too pessimistic: indeed, for universal set cover, Jia et al. [37] gave $\tilde{\Theta}(\sqrt{n})$ bounds. However, in many situations it is often more 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 $\pi : U \to [0,1]$. Let $\omega \in U^k$ be the random sequence of elements obtained (possibly with repetitions), and let us interpret ω as a set of elements when the ordering (and multiplicity) of elements in the sequence is not relevant. 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.

An algorithm for the universal stochastic set cover problem is *length-aware* if it is given the length k of the sequence in input, and *length-oblivious* otherwise.

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 the 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* $\mathbf{E}[c(\mathbf{S})] = O(\log mn)\mathbf{E}[c(\mathsf{opt})]$.

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 good ratio sets. Here we use a novel 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.

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, or 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 6.3. 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 [39, 35] as well. The details will be given in the full version of this paper.

Our results naturally extend to the stochastic version of the *online set cover* problem. Here the random sequence ω is presented to the algorithm element by element, 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, 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 Section 6 we describe universal algorithms for the stochastic versions of (non-metric) *facility location*, *multi-cut*, and *disc covering* in the plane.

In this paper, $\log x$ denotes the logarithm at base 2 of x. In the remaining of this paper, we assume that π is a uniform distribution: This assumption is without loss of generality using the standard reduction described in Appendix A.

1.2. Related Work

Universal, Oblivious and A-Priori Algorithms. These are algorithms where a single solution is constructed which is evaluated given multiple inputs—and either the worst-case or the average-case performance is considered. E.g., 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 [47, 8] and general metrics [37, 26, 29], as well as in the average-case [36, 9, 52, 24, 54]. (For the related problem of universal Steiner tree, see [39, 37, 26, 24].) For *universal set cover* and *facility location*, the previous results are in the worst-case: Jia et al. [37] 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. For *oblivious routing* [48, 32, 10] (see, e.g., [56, 57] for special cases), a tight logarithmic competitive result as well as a polynomial-time algorithm to compute the best routing is known for the worst case for undirected graphs [6, 49]. For *oblivious routing on directed graphs* the situation is similar to our problem: in the worst case the lower bound of $\Omega(\sqrt{n})$ [6] nearly matches upper bounds [30] but for the average case, [27] 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 Algorithms. Online algorithms have a long history (see, e.g., [11, 21]), and there have been many attempts to relax the strict worst-case notion of competitive analysis: see, e.g., [17, 1, 24] and the references therein. Online algorithms 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 [45, 46, 24], secretary problems [23], mechanism design [28, 41, 7], and matching problems in Ad-auctions [44, 13, 25].

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, 14, 4]). This ratio is the best possible under complexity-theoretic assumptions [19]; 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 [45, 22], the average case [24], as well as in the stronger *random permutation model* [45], 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 poster children for approximation algorithms, for which a $\Theta(\ln n)$-approximation has been long known [38, 16, 42], and this is the best possible [43, 18, 51, 5]. For the special case of set systems with small VC-dimension, a better algorithm is known [12]. Other objective functions have also been used, e.g., the min-latency [20] and min-entropy [31, 15] set cover problems. The $O(\log n)$ approximation for non-metric facility location is due to Hochbaum [33].

Stochastic Optimization. Research in (offline) stochastic optimization gives results for k-stage stochastic set cover; however, the approximation in most papers [35, 50, 53] is dependent on the number of stages k. Srinivasan [55] 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 $\mathrm{poly}(mn)$ time. In contrast to this, our results get within $O(\log nm)$ of the *best expected offline* cost.

2. 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.

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*

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 $|S \cap U|$;
  $\mathbf{S}(v) \leftarrow S$ for each $v \in S \cap U$;
  $U \leftarrow U \setminus S$;

exists 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 $\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_p$, and let $\cup \mathscr{C}_i$ be the union of the sets in $\mathscr{C}_i$. We now show that $|\cup \mathscr{C}_i| \geq n(1-\delta)$ for some i.

Suppose for 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,

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

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

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

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

We can now use the fact that for the *partial coverage problem* (pick the fewest sets to cover some $(1-\delta)$ fraction of the elements), the greedy algorithm is a $O(\log n)$-approximation [40, Thm 5.15] to get:

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 of 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 at most δn elements, we expect to see $\frac{k}{n}\delta n = 3\mu \ln 2m$ elements in a random sequence of length k. Whenever such an element 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 $O(\mu(\log n + \log m))$ as claimed. ■

An Exponential-Time Variant. Surprisingly, we can trade off the $\ln n$ factor in the approximation for a worse running time; this is quite unusual for competitive analysis where the lack of information rather lack of computational resources is the deciding factor. Instead of running the greedy algorithm to find the first $4\mu \ln n$ 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 cost is at most $O(\mu \log m)$. In Section 6.3 we give a polynomial-time algorithm achieving an $O(\log m)$-competitiveness when the set system has constant VC-dimension, and also give 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}), the adversary chooses $k = n$ and 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. In each iteration of Algorithm 2, 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 (*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.

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 $c(S)$;
  $\mathsf{S}(u)\leftarrow S$ 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 ;

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

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

which is at most $64\mathbf{E}[c(\mathsf{opt})]\ln n$. ■

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 $1\leq i\leq\ell-1$. Before bounding the mentioned cost (Lemma 3.6), we need a few intermediate results.

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. Denote by ω_i the subsequence 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 cheapest set in $\mathsf{opt}=\mathsf{opt}_\omega$ containing u. (Note that this is *not* the optimal set cover for ω_i.) As usual, $c(\mathsf{opt}|_{\omega_i})$ and $|\mathsf{opt}|_{\omega_i}|$ denote the cost and number of the sets in $\mathsf{opt}|_{\omega_i}$. Let Ω_i^q be the set of scenarios ω's such that $|\omega_i|=q$. The proofs of the following two technical lemmas are given in Appendix B.

Lemma 3.3 *For every $i\in\{1,\ldots,\ell\}$, if $k_i\geq 8\log 2n$ then there exists $q\geq k_i/2$ such that $\Pr_{\omega\in\Omega_i^q}\big[\,c(\mathsf{opt}|_{\omega_i})\leq 8\mathbf{E}[c(\mathsf{opt})]$ and $|\mathsf{opt}|_{\omega_i}|\leq 8\mathbf{E}[|\mathsf{opt}|]\,\big]\geq\frac{1}{2}$.*

Lemma 3.4 *For all $1\leq i\leq\ell$, $c(S_i)\mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big]\leq\mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big]$ and $c(S_i)\big(\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]-\mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big]\big)\leq\mathbf{E}[c(\mathsf{opt}|_{\omega_i})]-\mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big]$.*

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 $\mathsf{opt}|_{\omega_i}$.

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

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.

Let $\mathscr{S}_i$ be the family of sets $\{S\cap U_i\mid S\in\mathscr{S}\}$, and denote by $\mathscr{C}_1,\mathscr{C}_2,\ldots,\mathscr{C}_p$ 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 $\mathscr{C}_j$. Lemma 3.3 says that with probability at least $1/2$, the solution $\mathsf{opt}|_{\omega_i}$ uses at most $8\mathbf{E}[|\mathsf{opt}|]$ sets and costs at most $8\mathbf{E}[c(\mathsf{opt})]$, hence

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

Analogously to the proof of Lemma 2.1, we can infer that there is a collection $\mathscr{C}_j$ with

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

due to the assumption $q\geq k_i/2>8\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]\log m$. 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

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

However, the Type II set S_i was picked by the algorithm because there were no sets 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 Lemma 3.5, the expected cost incurred by using the remaining sets $S_1,\ldots,S_j$ is at most

$$\begin{aligned}
&\textstyle\sum_{i=1}^{j} c(S_i)\mathbf{Pr}[\omega\cap(S_i\cap U_i)\neq\emptyset]\\
&\le \textstyle\sum_{i=1}^{j} c(S_i)\mathbf{E}[|\omega\cap(S_i\cap U_i)|]\\
&\le \textstyle\sum_{i=1}^{j} c(S_i)\mathbf{E}[|\omega\cap(U_i\setminus U_{i+1})|]\\
&\le \textstyle\sum_{i=1}^{j} c(S_i)\,(k_i-k_{i+1}) \le \sum_{i=1}^{j} k_i\,(c(S_i)-c(S_{i-1}))\\
&\le \textstyle\sum_{i=1}^{j} 16\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]\log m\cdot(c(S_i)-c(S_{i-1}))\\
&= 16\log m\cdot\Big(c(S_j)\mathbf{E}\Big[|\mathsf{opt}|_{\omega_{j+1}}|\Big]\\
&\quad+\textstyle\sum_{i=1}^{j} c(S_i)\big(\mathbf{E}[|\mathsf{opt}|_{\omega_i}|]-\mathbf{E}\big[|\mathsf{opt}|_{\omega_{i+1}}|\big]\big)\Big).
\end{aligned}$$

It follows by Lemma 3.4 that the expected cost due to the sets $S_1,\ldots,S_j$ is at most

$$\begin{aligned}
&16\log m\cdot\Big(\mathbf{E}\Big[c(\mathsf{opt}|_{\omega_{j+1}})\Big]\\
&\quad+\textstyle\sum_{i=1}^{j}\big(\mathbf{E}[c(\mathsf{opt}|_{\omega_i})]-\mathbf{E}\big[c(\mathsf{opt}|_{\omega_{i+1}})\big]\big)\Big)\\
&= 16\mathbf{E}[c(\mathsf{opt}|_{\omega_1})]\log m \le 16\mathbf{E}[c(\mathsf{opt})]\log m,
\end{aligned}$$

concluding the proof of the lemma. ■

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

Proof of 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

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

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 particular value 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 $m=\mathrm{poly}(n)$, 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.

Let us slightly modify the universal algorithm for weighted set cover as follows: fixing a value $0<x\le\log m$, the set S minimizing $c(S)/|S\cap U|$ is selected only if $c(S)/|S\cap U| > 64\cdot 2^x\mathbf{E}[c(\mathsf{opt})]/|U|$. By adapting the analysis, the cost of Type I sets is bounded by $O(2^x\log n)\mathbf{E}[c(\mathsf{opt})]$, and the expected cost of Type II sets is $O(\log n+\frac{\log m}{x})\mathbf{E}[c(\mathsf{opt})]$. A similar result can be shown for Algorithm 1, in the unweighted (length-oblivious) case. Setting x suitably (details appear in the full version), we get:

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.*

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 $\log\log m-\log\log n=\Theta(\log n)$. Let the sequence length be $k=\sqrt{n}/2$. Consider any mapping. The sets included in the solution covering the first i elements cover at most $i\sqrt{n}\le\frac{n}{2}$ of the total elements. Hence, with probability at least half, the mapping must pick a new set to cover the $(i+1)$-th element. Hence, in expectation the mapping picks $\frac{\sqrt{n}}{4}$ sets while it is enough to select one set, proving the lemma. ■

5. Online Stochastic Set Cover

The universal algorithm for (weighted) stochastic set cover can be turned into an online algorithm 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 i. 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 update k' to the smallest value $k'' > k'$ which satisfies $\mathbf{E}[c(\mathbf{S}_{k''}(\omega^{k''}))] > 2\mathbf{E}[c(\mathbf{S}_{k'}(\omega^{k'}))]$ and modify the mapping consequently (we set $k' = \infty$ if such value k'' does not exist). We remark that the algorithm above takes polynomial time per sample, and does not assume any knowledge of the final number of samplings.

Proof of Theorem 1.3: Let $k \geq 1$ be the final number of samplings performed, and $\mathbf{S}$ be the actual 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 k_1, so assume $h \geq 2$ and hence $k_h \geq 2$. By the choice of the k_i's,

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

By definition, $\mathbf{E}\big[c(\mathbf{S}_{k_h-1}(\omega^{k_h-1}))\big] \leq 2\mathbf{E}\big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\big]$: this is trivially true for $k_h = \infty$ and holds by the minimality of k_h otherwise.

We need the following technical Lemma.

Lemma 5.1 *We have* $\mathbf{E}\big[c(\mathbf{S}_i(\omega^i))\big] \leq \mathbf{E}\big[c(\mathbf{S}_{i+1}(\omega^{i+1}))\big]$ *and* $\mathbf{E}\big[c(\mathbf{S}_i(\omega^i))\big] \leq 2\mathbf{E}\big[c(\mathbf{S}_{\lceil i/2\rceil}(\omega^{\lceil i/2\rceil}))\big]$, *for all* $i \geq 1$.

Proof: We observe that the pool of possible universal mappings from which each $\mathbf{S}_i$ is chosen is the same for every value of i (i.e. one for every possible breaking point). Moreover, the expected cost of each such mapping is an increasing function of the length of the sequence. As a consequence, $\mathbf{E}\big[c(\mathbf{S}_i(\omega^i))\big] \leq \mathbf{E}\big[c(\mathbf{S}_{i+1}(\omega^i))\big] \leq \mathbf{E}\big[c(\mathbf{S}_{i+1}(\omega^{i+1}))\big]$. The second claim follows along the same line. ■

It follows from Lemma 5.1 that

$$\begin{aligned}\mathbf{E}\Big[c(\mathbf{S}_{k_h}(\omega^{k_h}))\Big] &\leq 2\mathbf{E}\Big[c(\mathbf{S}_{\lceil k_h/2\rceil}(\omega^{\lceil k_h/2\rceil}))\Big] \\ &\leq 2\mathbf{E}\Big[c(\mathbf{S}_{k_h-1}(\omega^{k_h-1}))\Big] \leq 4\mathbf{E}\Big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\Big].\end{aligned}$$

We can conclude by the properties of the universal stochastic set cover algorithm that

$$\begin{aligned}\mathbf{E}\Big[c(\mathbf{S}(\omega^k))\Big] &\leq 6\mathbf{E}\Big[c(\mathbf{S}_{k_{h-1}}(\omega^{k_{h-1}}))\Big] \\ &= O(\log mn)\mathbf{E}\Big[c(\mathsf{opt}(\omega^{k_{h-1}}))\Big] \\ &= O(\log mn)\mathbf{E}\Big[c(\mathsf{opt}(\omega^k))\Big].\end{aligned}$$

■

6. Extensions and Applications

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

6.1. Universal Stochastic Facility Location

In this section we consider the universal stochastic version of (non-metric) facility location, a generalization of 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 the closest facility in $\mathbf{S}(X)$. We also denote by $|\mathbf{S}(X)|$ the number of facilities in $\mathbf{S}(X)$. With the usual notation, the aim is 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.

Our algorithm is an extension of the algorithm from Section 3, where the new challenge is to handle the connection costs for clients. As for weighted set cover, we first assume that the algorithm is given as input $\mathbf{E}[c(\mathsf{opt})]$; we later show how to remove this assumption.

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

Data: $C, F, d\colon C \times F \to \mathbb{R}_{\geq 0}$, $c\colon F \to \mathbb{R}_{\geq 0}$, k, $\mathbf{E}[c(\mathsf{opt})]$.
while $C \neq \emptyset$ **do**
  let $f \in F$ and $S \subseteq C$ minimize $\mathsf{avg} := \frac{c(f)+\min\{1,\frac{k}{n}\}\sum_{v\in S} d(v,f)}{|S\cap C|}$;
  if $\mathsf{avg} > \frac{192\mathbf{E}[c(\mathsf{opt})]}{|C|}$ **then** let $f \in F$ and $S = \{v\} \subseteq C$ minimize $c(f) + d(v,f)$;
  $\mathbf{S}(u) \leftarrow S$ for each $u \in S \cap C$;
  $C \leftarrow C \setminus S$;

The first step in the while loop can be implemented in polynomial time even if the number of candidate sets S is exponential, since it suffices to consider, for each facility f, the closest i clients still in C, for every $i = 1, \ldots, |C|$. The proof of the following lemma follows on similar lines to proof of Theorem 1.1 given in Section 3. Due to space limitations, the proof is omitted and will appear in the full version of the paper.

Lemma 6.1 *Algorithm 3 returns a universal mapping* $\mathbf{S}$ *to the universal stochastic facility location problem with* $\mathbf{E}[c(\mathbf{S})] = O(\log n)\mathbf{E}[c(\mathsf{opt})]$. *The same claim holds if a*

constant approximation to $\mathbf{E}[c(\mathsf{opt})]$ *is given as input instead of* $\mathbf{E}[c(\mathsf{opt})]$.

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

Theorem 6.2 *There exists a polynomial-time length-aware algorithm that returns a universal mapping* $\mathbf{S}$ *to the universal stochastic facility location problem with* $\mathbf{E}[c(\mathbf{S})] = O(\log n)\mathbf{E}[c(\mathsf{opt})]$.

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, $\mathbf{E}_{\omega\in C^1}[c(\mathsf{opt}(\omega))] \leq \mathbf{E}_{\omega\in C^k}[c(\mathsf{opt}(\omega))] \leq \mathbf{E}_{\omega\in C^n}[c(\mathsf{opt}(\omega))]$ for $1 \leq k \leq n$. 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, we run Algorithm 3 for all $\log n$ values x_i to obtain $\log n$ 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 is 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 6.3 *There is an* $O(\log n)$*-competitive algorithm for the online stochastic facility location problem.*

6.2. 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 $\omega \in D^k$ is defined as usual to be $c(\mathbf{S}(\omega))$—the total cost of edges in $\mathbf{S}(\omega)$. 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 multi-cut in trees (i.e., 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 e is in the unique path from s_i to t_i. Thus we can use the algorithm from Section 3 to obtain a $O(\log n)$-competitive algorithm for stochastic universal multi-cut in trees. Using results from Räcke [49], we can generalize this result to general graphs obtaining a $O(\log^2 n)$-competitive algorithm. The proof of the following theorem is omitted due to space constraints.

Theorem 6.4 *There exists an* $O(\log^2 n)$*-competitive polynomial-time algorithm for the online 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 multi-cut problem.*

6.3. Disc Covering in the Plane

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, such 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 [34], and by Bronnimann and Goodrich [12]: among other results, they gave a constant-factor approximation for the problem based on set cover for set systems with small VC-dimension.

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 map*. For ease of exposition, let us discretize the plane into n points by placing a fine-enough mesh on the plane. Using arguments in Section 2 and in Bronimamm and Goodrich [12] we can show that for the case of points chosen randomly from some known distribution from the plane (or more precisely, from this mesh), there *exists* a universal map whose expected set-cover cost is at most $O(\log m)$ times the expected optimum. Moreover, using the k-coverage algorithm for set systems of finite VC-dimension from the same section, we can also find such a universal map in randomized polynomial-time. The details are omitted and will appear in the full version of the paper.

References

[1] S. Albers and S. Leonardi. On-line algorithms. *ACM Comput. Surv.*, 31(3es):4, 1999.

[2] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. The Online Set Cover Problem. In *STOC'03*, pages 100–105, 2003.

[3] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. S. Naor. A general approach to online network optimization problems. In *SODA'04*, pages 577–586, 2004.

[4] N. Alon, Y. Azar, and S. Gutner. Admission control to minimize rejections and online set cover with repetitions. In *SPAA'05*, pages 238–244, 2005.

[5] N. Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2(2):153–177, 2006.

[6] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal oblivious routing in polynomial time. In *STOC'03*, pages 383–388, 2003.

[7] M. Babaioff, N. Immorlica, and R. Kleinberg. Matroids, secretary problems, and online mechanisms. In *SODA'07*, pages 434–443, 2007.

[8] D. Bertsimas and M. Grigni. Worst-case examples for the spacefilling curve heuristic for the Euclidean traveling salesman problem. *Oper. Res. Lett.*, 8(5):241–244, 1989.

[9] D. J. Bertsimas, P. Jaillet, and A. R. Odoni. A priori optimization. *Oper. Res.*, 38(6):1019–1033, 1990.

[10] M. Bienkowski, M. Korzeniowski, and H. Räcke. A practical algorithm for constructing oblivious routing schemes. In *SPAA'03*, pages 24–33, 2003.

[11] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, New York, 1998.

[12] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. *Discrete Comput. Geom.*, 14(4):463–479, 1995. ACM Symposium on Computational Geometry (Stony Brook, NY, 1994).

[13] N. Buchbinder, K. Jain, and J. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA'07*, pages 253–264, 2007.

[14] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. In *ESA'05*, pages 689–701, 2005.

[15] J. Cardinal, S. Fiorini, and G. Joret. Tight results on minimum entropy set cover. In *APPROX'06*, pages 61–69. 2006.

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

[17] R. Dorrigiv and A. Lopez-Ortiz. A survey of performance measures for on-line algorithms. *SIGACT News*, 36(3):67–81, 2005.

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

[19] U. Feige and S. Korman. On the use of randomization in the online set cover problem. *Technical Report*.

[20] U. Feige, L. Lovász, and P. Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.

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

[22] D. Fotakis. On the competitive ratio for online facility location. In *ICALP'03*, pages 637–652. 2003.

[23] P. R. Freeman. The secretary problem and its extensions: a review. *Internat. Statist. Rev.*, 51(2):189–206, 1983.

[24] N. Garg, A. Gupta, S. Leonardi, and P. Sankowski. Stochastic analyses for online combinatorial optimization problems. In *SODA'08*, pages 942–951, 2008.

[25] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA'08*, pages 982–991, 2008.

[26] A. Gupta, M. T. Hajiaghayi, and H. Räcke. Oblivious network design. In *SODA'06*, pages 970–979, 2006.

[27] M. T. Hajiaghayi, J. H. Kim, T. Leighton, and H. Räcke. Oblivious routing in directed graphs with random demands. In *STOC'05*, pages 193–201, 2005.

[28] M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In *EC'04*, pages 71–80, 2004.

[29] M. T. Hajiaghayi, R. D. Kleinberg, and F. T. Leighton. Improved lower and upper bounds for universal tsp in planar metrics. In *SODA'06*, pages 649–658, 2006.

[30] M. T. Hajiaghayi, R. D. Kleinberg, T. Leighton, and H. Räcke. Oblivious routing on node-capacitated and directed graphs. In *SODA'05*, pages 782–790, 2005.

[31] E. Halperin and R. M. Karp. The minimum-entropy set cover problem. *Theoret. Comput. Sci.*, 348(2-3):240–250, 2005.

[32] C. Harrelson, K. Hildrum, and S. Rao. A polynomial-time tree decomposition to minimize congestion. In *SPAA'03*, pages 34–43, 2003.

[33] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, Dec 1982.

[34] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.

[35] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *SODA'04*, pages 684–693, 2004.

[36] P. Jaillet. A priori solution of a travelling salesman problem in which a random subset of the customers are visited. *Oper. Res.*, 36(6):929–936, 1988.

[37] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for tsp, steiner tree, and set cover. In *STOC'05*, pages 386–395, 2005.

[38] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.

[39] D. R. Karger and M. Minkoff. Building Steiner trees with incomplete global knowledge. In *FOCS'00*, pages 613–623, 2000.

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

[41] R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA'05*, pages 630–631, 2005.

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

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

[44] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. AdWords and generalized online matching. *J. ACM*, 54(5):Art. 22, 19 pp., 2007.

[45] A. Meyerson. Online facility location. In *FOCS'01*, pages 426–431. 2001.

[46] A. Meyerson, K. Munagala, and S. Plotkin. Designing networks incrementally. In *FOCS'01*, pages 406–415, 2001.

[47] L. K. Platzman and J. J. Bartholdi, III. Spacefilling curves and the planar travelling salesman problem. *J. ACM*, 36(4):719–737, 1989.

[48] H. Räcke. Minimizing congestion in general networks. In *FOCS'02*, pages 43–52, 2002.

[49] H. Räcke. Optimal Hierarchical Decompositions for Congestion Minimization in Networks. In *STOC'08*, 2008.

[50] R. Ravi and A. Sinha. Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In *IPCO'04*, pages 101–115, 2004.

[51] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *STOC'97*, pages 475–484, 1997.

[52] F. Schalekamp and D. B. Shmoys. Algorithms for the universal and a priori tsp. *Oper. Res. Lett.*, 36(1):1–3, Jan 2008.

[53] D. B. Shmoys and C. Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *J. ACM*, 53(6):978–1012, 2006.

[54] D. B. Shmoys and K. Talwar. A constant approximation algorithm for the a priori traveling salesman problem. In *IPCO'08*, 2008.

[55] A. Srinivasan. Approximation algorithms for stochastic and risk-averse optimization. In *SODA'07*, pages 1305–1313, 2007.

[56] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *STOC'81*, pages 263–277, 1981.

[57] B. Vöcking. Almost optimal permutation routing on hypercubes. In *STOC'01*, pages 530–539, 2001.

A Non-Uniform Probability Distributions

We show that given an arbitrary distribution we can convert it to the uniform distribution. The only assumption that we need is that the algorithms have polynomial approximation ratios in the worst case, which is the case for all algorithms presented here. Assume we are given an $\alpha(n,m)$-competitive algorithm that is n^β-worst-case competitive, for some function α and a constant β. We replace each element $u \in U$ with $\lceil n^{\beta+1}\pi(u)\rceil$ copies of u in all the sets containing u. Denote the set system obtained this way by $(U', \mathscr{S}')$. Note that a uniform distribution on U' simulates the given distribution π in such a way that elements with $\pi(u) \geq \frac{1}{n^{\beta+1}}$ are generated with probability changed only by a factor of at most 2. Hence, the algorithm gives an $2\alpha(n,m)$ approximation on the sequences when only such elements are generated. The cost of all other sequences can be bounded by $n^\beta \times n\frac{1}{n^{\beta+1}}\mathbf{E}[c(\mathsf{opt}_\omega)] = \mathbf{E}[c(\mathsf{opt}_\omega)]$. Hence, finally we get an $2\alpha(n^{\beta+1},m)+1$ competitive algorithm. In particular, for $\alpha(n,m) = O(\log(mn))$ the competitive factor is $O(\log(mn))$. The following lemmas show how the reduction above applies to our algorithms.

Lemma A.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 A.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. Let $cheap(x)$ be the minimum cost of a set covering x. Observe that $\mathsf{opt}_\omega \geq \frac{1}{n}\sum_{x\in\omega} cheap(x)$. For any element $x \in \omega$ covered by a Type I set, it holds $c(\mathbf{S}(x)) \leq n \cdot cheap(x)$. For the remaining elements $x \in \omega$, $c(\mathbf{S}(x)) = cheap(x)$. As a consequence the cost of the solution returned by the algorithm is at most $\sum_{x\in\omega} n \cdot cheap(x)$. The claim follows. ■

A similar argument holds for the facility location problem as well. This justifies our assumption that π is a uniform probability distribution.

Lemma A.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. For any $x \in \omega$, let $cheap(x) = \min_{f\in F}\{c(f) + d(x,f)\}$. Observe that $\mathsf{opt}_\omega \geq \frac{1}{n}\sum_{x\in\omega} cheap(x)$. The cost paid by $\mathbf{S}$ for any $x \in \omega$ covered by a facility of Type II is at most $cheap(x)$. Consider now any $x \in \omega$ covered by a facility $f = \mathbf{S}(x)$ of Type I. Let $S = \mathbf{S}^{-1}(f)$. $\mathbf{S}$ pays for x at most

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

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

B Proofs from Section 3

Proof of Lemma 3.3: We restrict our attention to scenarios in $\Omega_i^{\geq k_i/2} := \uplus_{p\geq k_i/2} \Omega_i^p$, i.e., scenarios where the sampled k elements contain at least $\frac{k_i}{2}$ elements from U_i. Let d_i be the upper quartile of $|\mathsf{opt}|_{\omega_i}|$, i.e., in at least three-quarters of the scenarios in Ω_i, the optimal solution $\mathsf{opt} = \mathsf{opt}(\omega)$ uses at most d_i sets to cover the elements in the scenario. A Chernoff's bound implies that $\Pr[|\omega_i| < k_i/2] \leq \exp\left(-\frac{(1/2)^2 8\log 2n}{2}\right) \leq \frac{1}{2n}$. Hence, conditioning on the event $\omega \in \Omega_i^{\geq k_i/2} = \{\omega : |\omega_i| \geq k_i/2\}$ and observing that $\frac{1}{1-(1/2n)} \leq 2$, we obtain By the above equations and the definition of $\mathsf{opt}|_{\omega_i}$, $d_i \leq 4\mathbf{E}\left[|\mathsf{opt}|_{\omega_i}| \mid \omega \in \Omega_i^{\geq k_i/2}\right] \leq 8\mathbf{E}[|\mathsf{opt}|_{\omega_i}|] \leq 8\mathbf{E}[|\mathsf{opt}_\omega|]$.

An analogous argument shows that the cost $c(\mathsf{opt}|_{\omega_i})$ is at most $8\mathbf{E}[c(\mathsf{opt}_\omega)]$ with probability at least $3/4$. Hence, a trivial union bound implies that $\Pr_{\omega\in\Omega_i^{\geq k_i/2}}\left[\, c(\mathsf{opt}|_{\omega_i}) \leq 8\mathbf{E}[c(\mathsf{opt})] \;\wedge\; |\mathsf{opt}|_{\omega_i}| \leq 8\mathbf{E}[|\mathsf{opt}|]\,\right] \geq \frac{1}{2}$. Since $\Omega_i^{\geq k_i/2} = \uplus_{p\geq k_i/2}\Omega_i^p$, an averaging argument implies that some $q \geq k_i/2$ satisfies the lemma. ■

Proof of Lemma 3.4: 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 definition $c(S_i) \leq c(S_{i+1})$,

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

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

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

Taking expectations on the inequalities gives the lemma. ■