Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Power of Adaptivity for Stochastic Submodular Cover

Rohan Ghuge, Anupam Gupta, Viswanath Nagarajan

To cite this article:
Rohan Ghuge, Anupam Gupta, Viswanath Nagarajan (2024) The Power of Adaptivity for Stochastic Submodular Cover. Operations Research 72(3):1156-1176. https://doi.org/10.1287/opre.2022.2388

https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Vol. 72, No. 3, May–June 2024, pp. 1156–1176
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

The Power of Adaptivity for Stochastic Submodular Cover

Rohan Ghuge,[a] Anupam Gupta,[b] Viswanath Nagarajan[a,*]

[a] Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Michigan 48109; [b] Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

*Corresponding author

Contact: rghuge@umich.edu (RG); anupamg@cs.cmu.edu (AG); viswa@umich.edu, https://orcid.org/0000-0002-9514-5581 (VN)

Received: September 10, 2021
Revised: June 29, 2022
Accepted: September 5, 2022
Published Online in Articles in Advance: October 19, 2022

Area of Review: Optimization

https://doi.org/10.1287/opre.2022.2388

Abstract. In the stochastic submodular cover problem, the goal is to select a subset of stochastic items of minimum expected cost to cover a submodular function. Solutions in this setting correspond to sequential decision processes that select items one by one adaptively (depending on prior observations). Whereas such adaptive solutions achieve the best objective, the inherently sequential nature makes them undesirable in many applications. We show how to obtain solutions that approximate fully adaptive solutions using only a few "rounds" of adaptivity. We study both independent and correlated settings, proving smooth trade-offs between the number of adaptive rounds and the solution quality. Experiments on synthetic and real data sets show qualitative improvements in the solutions as we allow more rounds of adaptivity; in practice, solutions with a few rounds of adaptivity are nearly as good as fully adaptive solutions.

Funding: R. Ghuge and V. Nagarajan were supported in part by the National Science Foundation (NSF) Division of Civil, Mechanical and Manufacturing Innovation [Grant CMMI-1940766] and Division of Computing and Communication Foundations [Grant CCF-2006778]. A. Gupta was supported in part by the NSF [Grants CCF-1907820, CCF1955785, and CCF-2006953].

Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2022.2388.

Keywords: submodularity • stochastic optimization • rounds of adaptivity • covering problems

1. Introduction

Submodularity is a fundamental notion that arises in applications such as image segmentation, data summarization, hypothesis identification, information gathering, and social networks (see, for example, Simon et al. 2007, Lin and Bilmes 2011, Barinova et al. 2012, Sipos et al. 2012, Chen et al. 2014, Kempe et al. 2015, Radanovic et al. 2018). Given a nonnegative, monotone, and integer-valued submodular function $f : 2^U \to \mathbb{Z}_{\geq 0}$, the *submodular cover* optimization problem requires us to pick a minimum-cost subset S of items to "cover" function f. In other words, we want that $f(S) = Q$, where $Q = f(U)$ is the total coverage value.

Submodular cover arises in many applications in computer science, machine learning, and operations research (see Wolsey 1982, Golovin and Krause 2011, Mirzasoleiman et al. 2015, Bateni et al. 2018). For instance, consider a sensor deployment setting, in which we need to place a collection of sensors to monitor some phenomenon, such as air quality or traffic behavior (González-Banos 2001, Sun et al. 2019). Each sensor covers a limited area depending on its sensing range and also obstructions and the local geography. The goal is to deploy the fewest sensors to entirely cover some target region. The covered area is a submodular function of the sensors deployed, so this is a special case of submodular cover. Alternatively, one may want to place sensors so as to achieve a target level of "information gain." In many settings (see, e.g., Krause and Guestrin 2005), the information gain can be quantified as a submodular function, so this is again a special case of submodular cover. See Section 4.2 for more details.

A different application arises in medical diagnosis, in which we know s possible conditions from which a patient may suffer along with the priors on their occurrence (see, e.g., Garey and Graham 1974, Kosaraju et al. 1999). Our goal is to perform tests to identify the correct condition as quickly as possible. This can be cast as submodular cover by viewing each test as eliminating all inconsistent conditions (or hypotheses): the number of eliminated hypotheses is a coverage function that is submodular. This is an instance of submodular cover with a coverage value of $s - 1$ because once the $s - 1$ inconsistent hypotheses are eliminated, the remaining one must be correct. See Section 4.3 for more details.

Observe that both these applications involve uncertain data: the precise area covered by a sensor may not be known before the sensor is deployed, and the precise outcome (positive/negative) of a test is not known

until the test is performed. This uncertainty can be modeled using the framework of *stochastic submodular optimization*, turning the desired solution into a *sequential decision process*. In the simplest case of such a problem, we are given an underlying set of *stochastic items*, each of which is either active or inactive (with known probabilities). Only the active items contribute to the submodular function coverage. At each step of the sequential decision process, an item is *probed*, and its realization (i.e., active or inactive) is observed. The goal is to probe items in way that minimizes the expected total cost incurred until the submodular function is covered.

The process is typically *adaptive* so that information from previously probed items is used to identify the next item to probe. Such adaptive solutions are inherently fully sequential, which is undesirable if probing an item is time-consuming. For example, probing/performing a test in medical diagnosis may involve a long wait for test results. Or, in sensor deployment, if probing a sensor corresponds to physically deploying it, the action may take hours or days. Therefore, we prefer solutions with only few *rounds* of adaptivity, in which several items are probed simultaneously in each round, and the solution can only adapt at the end of each round.

Motivated by this, we ask, can solutions with a few adaptive rounds approximate fully adaptive solutions for the stochastic submodular cover problem? We consider cases in which realizations of different items are both independent and allowed to be correlated. For both these situations we give nearly tight answers with smooth trade-offs between the number r of adaptive rounds and the solution quality relative to fully adaptive solutions.

We make the following main contributions:

- When items have independent realizations, for any integer $r \geq 1$, we provide an algorithm with r rounds of adaptivity that costs at most $O(Q^{1/r}\log Q)$ times the optimal fully adaptive solution (that may adapt an unbounded number of times). Here, Q is the maximal value of the submodular function to be covered; note that the function is integer-valued. Our performance guarantee nearly matches a previously known lower bound of $\Omega\left(\frac{1}{r^3}Q^{1/r}\right)$.
- When items have correlated realizations (with a joint distribution of support size s), for any integer $r \geq 1$, we provide an algorithm with r rounds of adaptivity that costs at most $O(rs^{1/r}\log(sQ))$ times the optimal fully adaptive solution.
- We also prove that the dependence on the support size s is necessary for correlated distributions. We show an $\Omega\left(\frac{s^{1/r}}{r\log s}\right)$ multiplicative gap between r-round adaptive solutions and fully adaptive solutions (even when $Q = 1$).
- Finally, we demonstrate the practical efficacy of our algorithms by testing them on both real-world and synthetic data sets. Even on instances with more than 1,000 items, we observe that about six rounds of adaptivity suffice to obtain solutions that are nearly as good as fully adaptive solutions.

Our algorithms are based on a greedy-style approach and, hence, are very easy to implement. Moreover, unlike prior work, they provide approximation guarantees that do not depend on item costs or probability values (that may even be exponentially worse than Q and s).

1.1. Problem Definition

In the stochastic submodular cover problem, the input is a collection of m random variables (or items) $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$. Each item $\mathcal{X}_i$ has a cost $c_i \in \mathbb{R}_+$ and realizes to a random element of ground set U. Our results extend easily to the more general setting in which each item realizes to a subset of U rather than a single element; see Online Appendix E. Let the joint distribution of $\mathcal{X}$ be denoted by $\mathcal{D}$. The random variables $\mathcal{X}_i$ may or may not be independent; we discuss this issue in more detail at the end of this section. The realization of item $\mathcal{X}_i$ is denoted by $X_i \in U$; this realization is only known when $\mathcal{X}_i$ is probed at a cost of c_i. Extending this notation, given a subset of items $\mathcal{S} \subseteq \mathcal{X}$, its realization is denoted $S = \{X_i : \mathcal{X}_i \in \mathcal{S}\} \subseteq U$.

In addition, we are given an integer-valued monotone submodular function $f : 2^U \to \mathbb{Z}_+$ with $f(U) = Q$. A realization S of items $\mathcal{S} \subseteq \mathcal{X}$ is *feasible* if and only if $f(S) = Q$ the maximal value; in this case, we also say that $\mathcal{S}$ covers f. The goal is to probe (possibly adaptively) a subset $\mathcal{S} \subseteq \mathcal{X}$ of items that gets realized to a feasible set. We use the shorthand $c(\mathcal{S}) := \sum_{i:\mathcal{X}_i \in \mathcal{S}} c_i$ to denote the total cost of items in $\mathcal{S} \subseteq \mathcal{X}$ and use $c_{\max} := \max_i c_i$. The objective is to minimize the expected cost of probed items, for which the expectation is taken over the randomness in $\mathcal{X}$. We assume that $f(X) = Q$ for every realization X of the full set of items $\mathcal{X}$; this ensures that the function can be covered with probability one. We consider the following types of solutions.

Definition 1. For an integer $r \geq 1$, an r-round adaptive solution in the set-based model proceeds as follows. For each round $k = 1, \ldots, r$, it specifies a subset $\mathcal{S}_k$ of items that is probed in parallel. The cost incurred in round k is $c(\mathcal{S}_k)$, the total cost of all probed items in that round. The choice of subset $\mathcal{S}_k$ in round k can depend on realizations seen in all previous rounds $1, \ldots, k-1$.

In the set-based model, we allow solutions to be infeasible (i.e., to fail to cover f) with some small probability $\eta > 0$. As shown in Online Appendix D, such a relaxed notion is necessary. In designing algorithms,

we find it more convenient to work with a slightly different "permutation" model, which we define next.

Definition 2. For an integer $r \geq 1$, an r-round adaptive solution in the permutation model proceeds in r rounds of adaptivity. In each round $k \in \{1, \ldots, r\}$, the solution specifies an ordering of all remaining items and probes them in this order until some stopping rule. The decisions in round k, that is, the ordering and the stopping rule, can depend on the realizations seen in all previous rounds $1, \ldots, k-1$.

The permutation model is also used in prior literature (Goemans and Vondrák 2006, Agarwal et al. 2019). In the permutation model, solutions must be feasible with probability one. In Online Appendix D, we show that our algorithms in the permutation model can be converted into algorithms in the set-based model with similar approximation ratios. Henceforth, an r-round adaptive algorithm refers to one in the permutation model unless specified otherwise.

Setting $r = 1$ in Definition 2 gives us a *nonadaptive* solution as studied in Goemans and Vondrák (2006) and Agarwal et al. (2019). Setting $r = m$ gives us a *fully adaptive* solution that may adapt after every probe. Having more rounds leads to a smaller objective value, so fully adaptive solutions have the least objective value. Our performance guarantees are relative to an optimal fully adaptive solution; let OPT denote this solution and its cost. The r-round adaptivity gap is defined as follows:

$$\sup_{\text{instance } I} \frac{\mathbb{E}[\text{cost of best } r\text{-round adaptive solution on } I]}{\mathbb{E}[\text{cost of best adaptive solution on } I]}. \tag{1}$$

Setting $r = 1$ gives the *adaptivity gap*.

1.1.1. Independent and Correlated Distributions.

We first study the case in which the random variables $\mathcal{X}$ are *independent*. In keeping with existing literature (Deshpande et al. 2016, Im et al. 2016, Agarwal et al. 2019), we refer to this case simply as stochastic submodular cover. We then consider the case when the random variables $\mathcal{X}$ are correlated with a joint distribution $\mathcal{D}$ of support size s: the realizations in the support of $\mathcal{D}$ are also called scenarios. We refer to the correlated setting as *scenario* submodular cover as in Grammel et al. (2016).

1.1.2. Comparison with "Adaptive Submodularity."

We note that some previous papers, such as Golovin and Krause (2017) and Esfandiari et al. (2021b), model correlations in stochastic submodular cover in a different way. This involves assuming that the function f and the item distribution $\mathcal{D}$ satisfy the following condition (called adaptive submodularity): the conditional expected increase in f resulting from any item never increases when we condition on more realizations. Adaptive submodularity is more general than (independent) stochastic submodular cover, but it does not generalize scenario submodular cover. See Section 1.3 for more details.

1.2. Results and Techniques

Our first result is when the items have *independent distributions*.

Theorem 1 (Independent Items: Permutation Model). *For any integer $r \geq 1$, there is an r-round adaptive algorithm for the stochastic submodular cover problem with expected cost $O(Q^{1/r} \cdot \log Q)$ times the cost of an optimal adaptive solution.*

This improves over the $O(r\, Q^{1/r} \log Q \log(mc_{max}))$ bound of Agarwal et al. (2019) by eliminating the dependence on the number of items m and the item costs (which could be much larger than Q). Moreover, our result nearly matches the lower bound of $\Omega\left(\frac{1}{r^3} Q^{1/r}\right)$ given by Agarwal et al. (2019). Setting $r = \log Q$ shows that $O(\log Q)$ adaptive rounds give an $O(\log Q)$-approximation. By transforming this algorithm into a set-based solution (using Theorem 10 in Online Appendix D), we get the following.

Corollary 1 (Independent Items: Set-Based Model). *There is an $O(\log Q)$-round algorithm for stochastic submodular cover in the set-based model that (i) has expected cost $O(\log Q)$ times the optimal adaptive cost and (ii) covers the function with probability at least $1 - \frac{1}{Q}$.*

This approximation ratio of $O(\log Q)$ is the best possible (unless $P = NP$) even with an arbitrary number ($r = m$) of adaptive rounds. Corollary 1 improves upon an $O(\log^2 Q \log(mc_{\max}))$-approximation in a logarithmic number of rounds (Agarwal et al. 2019) and an $O(\log(mQc_{\max}))$-approximation in $O(\log m \log(Qmc_{\max}))$ set-based rounds (Esfandiari et al. 2021a).

Moreover, Theorem 1 (with $r = 1$) implies an $O(Q \log Q)$ gap between nonadaptive and fully adaptive solutions for stochastic set cover, a special case of submodular cover. This resolves an open question of Goemans and Vondrák (2006) up to an $O(\log Q)$ factor, in which Q is the number of objects in the set cover instance.

Our second set of results is for the case when items have correlated distributions. Recall that s denotes the support size of the joint distribution $\mathcal{D}$, that is, the number of scenarios.

Theorem 2 (Correlated Items: Permutation Model). *For any integer $r \geq 1$, there is an r-round adaptive algorithm for scenario submodular cover with cost $O(s^{1/r}(\log s + r \log Q))$ times the optimal adaptive cost.*

We also obtain a $2r$-round adaptive algorithm with a better cost guarantee of $O(s^{1/r}\log(sQ))$ times the optimal adaptive cost (see Corollary 3). Setting $r = \log s$ and using the conversion to a set-based solution (Theorem 10), we infer the following.

Corollary 2 (Correlated Items: Set-Based Model). *There is an $O(\log s)$-round algorithm for scenario submodular cover in the set-based model that (i) has expected cost $O(\log(sQ))$ times the optimal adaptive cost and (ii) covers the function with probability at least $1-\frac{1}{s}$.*

This approximation guarantee is nearly the best possible even with an arbitrary number of adaptive rounds: there is an $\Omega(\log Q)$-factor hardness of approximation even for the deterministic case (in which $s = 1$). We note that scenario submodular cover generalizes the classic optimal decision tree problem, which is studied extensively; see, for instance, Garey and Graham (1974), Loveland (1985), Kosaraju et al. (1999), Dasgupta (2004), and Gupta et al. (2017a). Corollary 2 improves upon the fully adaptive $O(\log(sQ))$-approximation for scenario submodular cover (Grammel et al. 2016) by achieving the same approximation guarantee in just $O(\log s)$ rounds. It is also an improvement over the $O\left(\log\left(mQ\frac{c_{\max}}{p_{\min}}\right)\right)$-approximation in $O\left(\log m\log\left(Qm\frac{c_{\max}}{p_{\min}}\right)\right)$ set-based rounds, which follows from Grammel et al. (2016) and Esfandiari et al. (2021a); here, $p_{\min} \le \frac{1}{s}$ is the minimum probability of any scenario.

We note that, when the number of rounds is less than logarithmic, our result provides the first approximation guarantee even in the well-studied special case of optimal decision trees.

The results in Theorems 1 and 2 are incomparable: whereas the independent case has more structure in the distribution $\mathcal{D}$, its support size is exponential. Finally, the dependence on the support size s is necessary in the correlated setting as our next result shows.

Theorem 3 (Correlated Items Lower Bound). *For any integer $r \ge 1$, there is an instance of scenario submodular cover with $Q = 1$ for which the cost of any r-round adaptive solution is $\Omega\left(\frac{s^{1/r}}{r\log s}\right)$ times the optimal adaptive cost.*

This lower bound is information-theoretic and does not depend on computational assumptions, whereas the upper bound of Theorem 2 is given by a polynomial algorithm.

Finally, our algorithms are also easy to implement. We test our algorithms on both synthetic and real data sets; these tests validate the practical performance of our algorithms. Specifically, we test our algorithm for the independent case (Theorem 1) on instances of stochastic set cover and our algorithm for the correlated case (Theorem 2) on instances of optimal decision tree. For stochastic set cover, we use real-world data sets to generate instances with $\approx 1{,}200$ items. We observe a sharp improvement in performance within a few rounds of adaptivity and that six to seven rounds of adaptivity are nearly as good as fully adaptive solutions. For optimal decision tree, we use both real-world and synthetic data. The real-world data has ≈ 400 scenarios, and the synthetic data has 10,000 scenarios. Again, we find that about six rounds of adaptivity suffice to obtain solutions as good as fully adaptive ones. We also compared our algorithms' cost to information-theoretic lower bounds for both applications: our costs are typically within 50% of these lower bounds.

1.2.1. Techniques. The algorithms for the independent and correlated cases are similar in spirit but have some crucial technical differences. In each round of both algorithms, we iteratively compute a "score" for each item and greedily select the item of maximum score. This results in a nonadaptive list of all remaining items, and the items are probed in this order until a stopping rule is satisfied. The stopping rule in the independent case corresponds to reducing the remaining target (on the function value) by a factor of $Q^{1/r}$, whereas the stopping rule in the correlated case involves reducing the number of "compatible scenarios" by an $s^{1/r}$ factor.

The analysis for Theorems 1 and 2 follows parallel lines at the beginning. For each integer $i \ge 0$, we relate the following two quantities: (i) the probability that the algorithm does not complete within cost $\alpha \cdot 2^i$ and (ii) the probability that the optimal adaptive solution does not complete within cost 2^i. The "stretch factor" α corresponds to the approximation ratio and is chosen differently for the independent and correlated cases. In order to relate these noncompletion probabilities, we consider the total score G of items selected by the algorithm between costs $\alpha 2^{i-1}$ and $\alpha 2^i$. The crux of the analysis lies in giving lower and upper bounds on the total score G; this is when the arguments for the independent and correlated settings begin to differ.

In the independent case, the score of any item $\mathcal{X}_e$ is an estimate of its relative marginal gain, for which we take an expectation over all previous items as well as $\mathcal{X}_e$. We also normalize this gain by the item's cost. See Equation (2) for the definition. In lower bounding the total score G, we use a variant of a sampling lemma from Agarwal et al. (2019) as well as the constant-factor adaptivity gap for submodular maximization from Bradac et al. (2019). We also need to partition the outcome space (of all previous realizations) into "good" and "bad" outcomes: conditional on a good outcome, OPT has a high probability of completing before cost 2^i. Good outcomes are necessary in our proof of the sampling lemma, but luckily, the total probability of bad

outcomes is small (and they can be effectively ignored). In upper bounding G, we consider the total score as a sum over decision/sample paths and use the fact that the sum of relative gains corresponds to a harmonic series.

In the correlated case, the score of any item $\mathcal{X}_e$ is the sum of two terms: (i) its expected relative marginal gain as in the independent case and (ii) an estimate of the probability on "eliminated" scenarios. Both terms are needed because the algorithm needs to balance (i) covering the function and (ii) identifying the realized scenario (after which it is trivial to cover f). Again, we normalize by the item's cost; see Equation (17). In lower bounding the total score G, we partition the outcome space into good/okay/bad outcomes that correspond to a high conditional probability of OPT (i) covering function f by cost 2^i, (ii) eliminating a constant fraction of scenarios by cost 2^i, or (iii) neither of the two cases. Further, by restricting to outcomes that have a large number of compatible scenarios (else, the algorithm's stopping rule would apply), we can bound the number of relevant outcomes by $s^{1/r}$. Then, we consider OPT (up to cost 2^i) conditional on all good/okay outcomes and show that one of these items has a high score. To upper bound G, we again consider the total score as a sum over decision paths.

1.3. Related Work

A $(1+\ln Q)$-approximation algorithm for the basic submodular cover problem is obtained in Wolsey (1982). Dinur and Steurer (2014) show that this is also the best possible (unless $P = NP$) because submodular cover generalizes the set cover problem. In the past several years, there have been many papers on stochastic variants of submodular cover as this framework captures many different applications; see Golovin and Krause (2011), Im et al. (2016), Deshpande et al. (2016), Grammel et al. (2016), and Navidi et al. (2020).

The stochastic set cover problem was first studied by Goemans and Vondrák (2006), who show that the adaptivity gap lies between $\Omega(d)$ and $O(d^2)$, where d is the number of objects to be covered. Recently, Agarwal et al. (2019) improved the upper bound to $O(d \log d \log(mc_{max}))$. As a corollary of Theorem 1, we obtain a tighter $O(d \log d)$ adaptivity gap. Importantly, we eliminate the dependence of the items m and maximum cost, which may even be exponential in d.

An $O(\log Q)$ adaptive approximation algorithm for stochastic submodular cover (independent items) follows from the work of Im et al. (2016). Liu et al. (2008), Golovin and Krause (2011, 2017), and Deshpande et al. (2016) obtain related results for special cases or with weaker bounds. As mentioned earlier, Agarwal et al. (2019) obtain r-round adaptivity gaps for independent items. Theorem 1 improves their bound by an $O(r \cdot \log(mc_{max}))$ factor. We bypass computationally expensive steps in prior work, such as solving several instances of stochastic submodular maximization: so our algorithm is more efficient. Our analysis (outlined earlier) is also very different. Whereas we use a sampling lemma similar to Agarwal et al. (2019), this result is applied in different manner, and it only affects the analysis (see Lemma 4).

The scenario submodular cover problem was introduced by Grammel et al. (2016) as a common generalization of several problems, including optimal decision tree (Garey and Graham 1974, Gupta et al. 2017a), equivalence class determination (Cicalese et al. 2014), and decision region determination (Javdani et al. 2014). Grammel et al. (2016) obtain an $O(\log(sQ))$ fully adaptive algorithm for it. The same approximation ratio (in a more general setting) is also obtained by Navidi et al. (2020). In the correlated setting, we are not aware of any prior work on limited rounds of adaptivity (when the number of rounds $r < \log s$). Some aspects of our analysis (e.g., good/okay/bad outcomes) are similar to that of Navidi et al. (2020), but additional work is needed as we have to bound the r-round adaptivity gap.

As noted earlier, the framework of adaptive submodularity from Golovin and Krause (2017) models correlations in stochastic submodular cover in a different way. Whereas stochastic submodular cover with independent items satisfies adaptive submodularity, scenario submodular cover does not. Esfandiari et al. (2021a) give an $O(\log(mQ))$-approximation in $O(\log m \log(Qm))$ rounds of adaptivity for adaptive submodular cover (AdSubCov), *assuming unit costs*. So their result implies the same bounds for (independent) stochastic submodular cover. Although scenario submodular cover is *not* a special case of AdSubCov, Grammel et al. (2016) reformulate scenario submodular cover as AdSubCov but with a different goal function that is indeed adaptive submodular. However, this new goal function has a larger Q-value of $\frac{Q}{p_{min}}$ because of which the algorithm of Esfandiari et al. (2021a) only implies an $O\left(\log\left(mQ\frac{c_{\max}}{p_{\min}}\right)\right)$-approximation in $O\left(\log m \log\left(Qm\frac{c_{\max}}{p_{\min}}\right)\right)$ rounds (see Table 2). To the best of our knowledge, there are no algorithms for AdSubCov using fewer than squared-logarithmic rounds of adaptivity.

Tables 1 and 2 provide a comparison of results for stochastic and scenario submodular cover. These results are in the set-based model with failure probability $o(1)$.

The role of adaptivity is extensively studied for stochastic submodular *maximization*. Asadpour and Nazerzadeh (2016) give a constant adaptivity gap under matroid constraints (on the probed items). Gupta et al. (2017b) obtain a constant adaptivity gap for a very large class of *prefix-closed* constraints; the constant factor was subsequently

Table 1. Comparison of Results for Stochastic Submodular Cover

Stochastic submodular cover	Approximation guarantee	Set-based rounds
Im et al. (2016)	$O(\log Q)$	Fully adaptive
Agarwal et al. (2019)	$O(\log^2 Q \log(mc_{\max}))$	$O(\log Q)$
Esfandiari et al. (2021b)	$O(\log(mQc_{\max}))$	$O(\log m \log(Qmc_{\max}))$
Our result	$O(\log Q)$	$O(\log Q)$

improved to two, which is also the best possible as shown by Bradac et al. (2019). We make use of this result in our analysis. More generally, the role of adaptivity is extensively studied for various stochastic maximization problems. Dean et al. (2008) and Bhalgat et al. (2011) study the stochastic knapsack problem; Bansal et al. (2012) and Behnezhad et al. (2020) examine the stochastic matching problem; Gupta and Nagarajan (2013) obtain results for stochastic probing; and Guha and Munagala (2009), Gupta et al. (2015), and Bansal and Nagarajan (2015) study stochastic orienteering.

Karbasi et al. (2021), Esfandiari et al. (2021b), and Gao et al. (2019) study the role of adaptivity and batch processing in online learning. Their work was motivated by noting that many real-world data observations are processed in batches. Recently, there have been several results examining the role of adaptivity in *deterministic* submodular optimization (see, for example, Balkanski and Singer 2018, Balkanski et al. 2019, Chekuri and Quanrud 2019). The motivation here was to parallelize function queries, which are often expensive. In many settings, there are algorithms using a (poly)logarithmic number of rounds that nearly match the best sequential (or fully adaptive) approximation algorithms. Whereas our motivation for parallelizing the expensive probing steps in stochastic submodular cover is similar to that in these two lines of work (online learning and deterministic submodular optimization), the techniques we use are very different.

A different (and extensively studied) model for stochastic online optimization is the setting of *prophet inequalities*, in which the algorithm observes several random items sequentially and needs to make immediate selection decisions. The classic setting, introduced by Krengel and Sucheston (1977), involves selecting a single item so as to maximize its expected value. See, for example, Correa et al. (2018) for a recent survey. Kleinberg and Weinberg (2019) extend the classic prophet inequality setting to one in which multiple items may be selected subject to a matroid constraint (the objective is a linear function of the selected items). See also the extension to multiple matroid constraints by Feldman et al. (2021). Rubinstein and Singla (2017) generalize this setting even further to the case of submodular objectives. A key difference from our setting is that the benchmark in prophet inequalities is *omniscient* and allowed to be aware of all the random realizations before making its decisions, so the focus here is on *competitive analysis*, and constant competitive algorithms are known for all these settings. In contrast, for stochastic submodular cover, we cannot compare with such a strong omniscient benchmark: one cannot obtain competitive ratios better than $O(Q)$. Given this, we compare with the best policy restricted in the same manner as the algorithm, that is, which is not aware of item realizations up front.

2. Stochastic Submodular Cover

We now consider the (independent) stochastic submodular cover problem and prove Theorem 1. For simplicity, we assume that costs c_i are integers. Our results also hold for arbitrary costs (by replacing certain summations in the analysis by integrals). We work with the permutation model of rounds throughout this section. We use the notation $S \sim \mathcal{S}$ to denote realization S drawn from the distribution induced by $\mathcal{S}$; for example, $\mathbb{E}_{S\sim\mathcal{S}}[f(S)]$ is the expectation of $f(S)$ over the randomness of $\mathcal{S}$.

We find it convenient to solve a *partial cover* version of the stochastic submodular cover problem. In this partial cover version, we are given a parameter $\delta \in [0,1]$, and the goal is to probe some items $\mathcal{R} \subseteq \mathcal{X}$ that realize to a set R achieving value $f(R) > Q(1-\delta)$. We are interested in a nonadaptive algorithm for this problem. Clearly, if

Table 2. Comparison of Results for Scenario Submodular Cover

Scenario submodular cover	Approximation guarantee	Set-based rounds
Grammel et al. (2016)	$O(\log(sQ))$	Fully-adaptive
Esfandiari et al. (2021b), Grammel et al. (2016)	$O(\log(mQ\frac{c_{\max}}{p_{\min}}))$	$O(\log m \log(Qm\frac{c_{\max}}{p_{\min}}))$
Our result	$O(\log sQ)$	$O(\log s)$

Note. $s \le 1/p_{\min}$.

$\delta = 1/Q$, the integrality of the function f means that $f(R) = Q$, and we solve the original (full-coverage) problem. Moreover, we can use this algorithm with different thresholds to also get the r-round algorithm. The main result of this section is the following.

Theorem 4. *There is a nonadaptive algorithm for the partial cover version of stochastic submodular cover with cost $O\left(\frac{\ln 1/\delta}{\delta}\right)$ times the optimal adaptive cost for the (full) submodular cover.*

The algorithm first creates an ordering/list L of the items nonadaptively, that is, without knowing the realizations of the items. To do so, at each step, we pick a new item that maximizes a carefully defined score function (Equation (2)). The score of an item cannot depend on the realizations of previous items on the list (because we are nonadaptive). The summation in the score (2) corresponds to the increase in the function value for the new item $\mathcal{X}_e$ (scaled by the residual target) for a random realization of the previous items $\mathcal{S}$; we also refer to this term as the *incremental value* of the item. The actual score is the ratio of the incremental value and the cost of item $\mathcal{X}_e$. Once this ordering L is specified, the algorithm starts probing and realizing the items in this order and does so until the realized value exceeds $(1-\delta)Q$.

Algorithm 1 (Partial Covering Algorithm PARCA($\mathcal{X}, f, Q, \delta$))

1: $\mathcal{S} \leftarrow \emptyset$, list $L \leftarrow \langle\rangle$, $\tau \leftarrow Q(1-\delta)$
2: **while** $\mathcal{S} \neq \mathcal{X}$ **do** ▷ Building the list nonadaptively
3: select an item $\mathcal{X}_e \in \mathcal{X} \setminus \mathcal{S}$ that maximizes:

$$\text{score}(\mathcal{X}_e) := \frac{1}{c_e} \cdot \sum_{S \sim \mathcal{S}: f(S) \le \tau} \mathbf{P}(\mathcal{S} = S) \cdot \mathbb{E}_{X_e \sim \mathcal{X}_e}\left[\frac{f(S \cup X_e) - f(S)}{Q - f(S)}\right] \quad (2)$$

4: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{X}_e\}$ and list $L \leftarrow L \circ \mathcal{X}_e$
5: $\mathcal{R} \leftarrow \emptyset$, $R \leftarrow \emptyset$
6: **while** $f(R) \le \tau$ **do** ▷ Probing items on the list
7: $\mathcal{X}_e \leftarrow$ first r.v. in list L not in $\mathcal{R}$, and let $X_e \in U$ be its realization.
8: $R \leftarrow R \cup \{X_e\}, \mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{X}_e\}$
9: return probed items $\mathcal{R}$ and their realizations R.

Given this partial covering algorithm, we immediately get an algorithm for the r-round version of the problem, in which we are allowed to make r rounds of adaptive decisions. Indeed, we can first set $\delta = Q^{-1/r}$ and solve the partial covering problem with this value of δ. Suppose we probe variables $\mathcal{R}$, and their realizations are given by the set $R \subseteq U$. Then, we can condition on these values to get the marginal value function f_R, where $f_R(S) = f(R \cup S) - f(R)$. We note that f_R is submodular, and so we can inductively get an $(r-1)$-round solution for this problem. The following algorithm formalizes this.

Algorithm 2 (r-Round Adaptive Algorithm For Stochastic Submodular Cover SSC($r, \mathcal{X}, f$))

1: run PARCA $(\mathcal{X}, f, Q, Q^{-1/r})$ for round #1.
2: let $\mathcal{R}$ (respectively, R) denote the probed items (respectively, their realizations) in PARCA.
3: define residual submodular function $\widehat{f} := f_R$.
4: recursively solve SSC$(r-1, \mathcal{X} \setminus \mathcal{R}, \widehat{f})$.

Theorem 5. *Algorithm 2 is an r-round adaptive algorithm for stochastic submodular cover with cost $O(Q^{1/r}\log Q)$ times the optimal adaptive cost.*

Proof. We proceed by induction on the number of rounds r. Let OPT denote the cost of an optimal adaptive solution. The base case is $r = 1$, in which case $\delta = Q^{-1/r} = \frac{1}{Q}$. By Theorem 4, the partial cover algorithm PARCA $(\mathcal{X}, f, Q, Q^{-1/r})$ obtains a realization R with $f(R) > (1-\delta)Q = Q - 1$. As f is integer-valued, we must have $f(R) = Q$, which means the function is fully covered. So the algorithm's expected cost is $O(Q \log Q) \cdot \texttt{OPT}$.

We now consider $r > 1$ and assume (inductively) that Algorithm 2 finds an $r-1$-round $O\left(Q^{\frac{1}{r-1}}\log Q\right)$-approximation algorithm for any instance of stochastic submodular cover. Let $\delta = Q^{-1/r}$. By Theorem 4, the expected cost in round 1 (step 1 in Algorithm 2) is $O\left(\frac{1}{r}Q^{1/r}\log Q\right) \cdot \texttt{OPT}$. Let $\widehat{Q} := Q - f(R) = \widehat{f}(U)$ denote the maximal value of the residual submodular function $\widehat{f} = f_R$. Note that $\widehat{Q} < \delta Q = Q^{(r-1)/r}$ by the definition of the partial covering algorithm. The optimal solution OPT conditioned on the variables in $\mathcal{R}$ realizing to R gives a feasible adaptive solution to the residual problem of covering $\widehat{f}$; we denote this conditional solution by $\widehat{\texttt{OPT}}$. We inductively get that the cost of our $r-1$-round solution on $\widehat{f}$ is at most

$$O\left(\widehat{Q}^{\frac{1}{r-1}}\log \widehat{Q}\right) \cdot \widehat{\texttt{OPT}} \le O\left(\frac{r-1}{r}\, Q^{1/r}\log Q\right) \cdot \widehat{\texttt{OPT}},$$

where we used $\widehat{Q} < Q^{(r-1)/r}$. As this holds for every realization R, we can take expectations over R to get that the (unconditional) expected cost of the last $r-1$ rounds is $O\left(\frac{r-1}{r}\, Q^{1/r}\log Q\right) \cdot \texttt{OPT}$. Adding to this the cost of the first round, which is $O\left(\frac{1}{r}Q^{1/r}\log Q\right) \cdot \texttt{OPT}$, completes the proof. □

Remark 1. Assuming that the scores (2) can be computed in polynomial time, it is clear that our entire algorithm can be implemented in polynomial time. In particular, if T denotes the time taken to calculate the score of one item, then the overall algorithm runs in time *poly*(m, T),

where m is the number of items. We are not aware of a closed-form expression for the scores (for arbitrary submodular functions f). However, as discussed in Section A.2, we can use sampling to estimate these scores to within a constant factor. Moreover, our analysis works even if we only choose an approximate maximizer for (2) at each step. It turns out that $T = poly(m, c_{max})$ many samples suffice to estimate these scores. So the final runtime is $poly(m, c_{max})$; note that this does not depend on the number $|U|$ of elements. We note that even the previous algorithms (Agarwal et al. 2019, Esfandiari et al. 2021a) have a polynomial dependence on c_{max} in their runtime. In the following analysis, we assume that the scores (2) are computed exactly; see Online Appendix A.2 for the sampling details.

2.1. Analysis for a Call to ParCA

We now prove Theorem 4. We denote by OPT an optimal adaptive solution for the (full) covering problem on f. We denote by NA the nonadaptive strategy given by algorithm ParCA. Note that NA probes variables according to the order given by the list L and stops when the realized coverage exceeds $\tau := Q(1-\delta)$ (see Algorithm 1 for details).

Now, we relate the expected costs of NA and OPT. We refer to the cumulative cost incurred (by either OPT or NA) until any point in the solution as *time* elapsing. We say that OPT is in phase i when it is in the time interval $[2^i, 2^{i+1})$ for $i \geq 0$. Let $\alpha := O\left(\frac{\ln 1/\delta}{\delta}\right)$, where the constant factor is fixed later. We say that NA is in *phase* i when it is in time interval $[\alpha \cdot 2^{i-1}, \alpha \cdot 2^i)$ for $i \geq 1$; phase 0 refers to the time interval $[1, \alpha)$. Define

- u_i: probability that NA goes beyond phase i, that is, has cost at least $\alpha \cdot 2^i$.
- u_i^*: probability that OPT goes beyond phase $i-1$, that is, costs at least 2^i.

As all costs are integers, $u_0^* = 1$. For ease of notation, we also use OPT and NA to denote the *random* cost incurred by OPT and NA, respectively. The following lemma, which bounds the noncompletion probability u_i in terms of u_i^*, forms the crux of the analysis.

Lemma 1. *For any phase $i \geq 1$, we have $u_i \leq \frac{u_{i-1}}{4} + \frac{6}{5}u_i^*$.*

Lemma 1 implies Theorem 4 using standard techniques (see Online Appendix A.1).

2.2. Proof of the Key Lemma (Lemma 1)

Recall the setting of Lemma 1 and fix the phase $i \geq 1$. Consider the list L generated by ParCA$(\mathcal{X}, f, Q, \delta)$. Let NA denote both the nonadaptive strategy of Algorithm 1 as well as its cost. Note that NA probes items in the order given by L until a coverage value greater than $\tau := Q(1-\delta)$ is achieved. We assume (without loss of generality) that δ is a power of two, that is, $\delta = 2^{-z}$ for some integer $z \geq 0$. Indeed, if this is not the case, we can use a power-of-two value δ', where $\frac{\delta}{2} \leq \delta' \leq \delta$: this only increases the approximation ratio $O\left(\frac{1}{\delta}\ln\frac{1}{\delta}\right)$ in Theorem 4 by a constant factor.

For each time $t \geq 0$, let $\mathcal{X}_{e(t)}$ denote the item that is selected at time t. In other words, this is the item that causes the cumulative cost of L to exceed t for the first time. We define the *total gain* as the following quantity:

$$G := \sum_{t=\alpha 2^{i-1}}^{\alpha 2^i} \text{score}(\mathcal{X}_{e(t)}), \tag{3}$$

which corresponds to the sum of scores over the time interval $[\alpha \cdot 2^{i-1}, \alpha \cdot 2^i)$. The proof of Lemma 1 is completed by giving upper and lower bounds on G, which we do next. The lower bound views G as a sum over time steps, whereas the upper bound views G as a sum over decision paths.

2.2.1. A Lower Bound for G.

We show that $G = \Omega(\alpha\delta) \cdot \left(u_i - \frac{6}{5}u_i^*\right)$. To this end, we prove the contribution to G at each time $t \in [\alpha 2^{i-1}, \alpha 2^i), \text{score}(\mathcal{X}_{e(t)}) \geq \Omega\left(\frac{\delta}{2^i}\right) \cdot \left(u_i - \frac{6}{5}u_i^*\right)$. To prove the lower bound on $\text{score}(\mathcal{X}_{e(t)})$, we show that there exists a set $\mathcal{T}$ of items with total cost $O(2^i/\delta)$ and incremental value $\Omega\left(u_i - \frac{6}{5}u_i^*\right)$; see Lemma 5.

Henceforth, fix some time $t \in [\alpha 2^{i-1}, \alpha 2^i)$ in phase i of our algorithm. Let $\mathcal{S}$ be the set of chosen items (added to list L) until time t and let S denote its realization. Note that $\mathcal{S}$ is used crucially in the definition of score (2), and the total cost of the items $\mathcal{S}$ is at most t. We also need the following key definitions:

1. For any power-of-two $\theta \in \left\{\frac{1}{2^j} : 0 \leq j \leq \log Q\right\}$, we say that a realization S of $\mathcal{S}$ belongs to *scale* θ if and only if $\frac{\theta}{2} < \frac{Q - f(S)}{Q} \leq \theta$. Note that δ corresponds to some scale because it is a power of two. We use $\mathcal{E}_\theta$ to denote all outcomes of $\mathcal{S}$ that belong to scale θ. We also use $S \sim \mathcal{E}_\theta$ to denote the conditional distribution of $\mathcal{S}$ corresponding to $\mathcal{E}_\theta$.
2. For any scale θ, let $r_{i\theta}^* := \mathbf{P}(\text{OPT covers } f \text{ with cost at most } 2^i \textbf{ and } S \in \mathcal{E}_\theta)$.
3. Scale θ is called good if $\frac{r_{i\theta}^*}{\mathbf{P}(\mathcal{E}_\theta)} \geq \frac{1}{6}$, where $\mathbf{P}(\mathcal{E}_\theta) := \mathbf{P}_{S\sim\mathcal{S}}(S \in \mathcal{E}_\theta)$.

Here is an outline of the rest of the analysis. Lemma 2 characterizes the stopping rule in NA in terms of the scale θ of realization S. Lemma 3 lower bounds the total $r_{i\theta}^*$ value over good scales in terms of the noncompletion probabilities u_i and u_i^*. Then, Lemma 4 shows that, for each good scale θ, there is a subset $\mathcal{T}_\theta$ of items

in which (i) the total cost is $O(2^i/\theta)$ and (ii) the total incremental value is large *conditional on* realization $S \in \mathcal{E}_\theta$. Finally, Lemma 5 shows that taking all items in $\mathcal{T}_\theta$ (for all good scales $\theta \geq \delta$) provides a set $\mathcal{T}$ with total cost $O(2^i/\delta)$ and large incremental value (unconditionally). This can then be used to lower bound G as noted.

Lemma 2. *The realization S is in a scale $\theta \leq \delta$ if and only if $f(S) \geq \tau$. Hence, NA terminates by time t if and only if the realization of $\mathcal{S}$ is in a scale $\theta \leq \delta$.*

Proof. Note that, if the realization of $\mathcal{S}$ is in some scale $\theta \geq 2\delta$, then

$$f(S) < Q - \frac{\theta Q}{2} \leq Q - \delta Q = \tau,$$

and NA would not terminate before time t. On the other hand, if the realization of $\mathcal{S}$ is in a scale $\theta \leq \delta$ (note that all scales are powers of two), then $f(S) \geq Q - \theta Q \geq \tau$, and NA would terminate before time t. □

Lemma 3. *We have $\sum_{\theta > \delta,\ \text{good}} r^*_{i\theta} \geq u_i - \frac{6u^*_i}{5}$.*

Proof. First, we upper bound $\sum_{\theta \text{ not good}} r^*_{i\theta}$. Consider any scale θ that is not good. Then, $\frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)} < \frac{1}{6}$, that is, $\mathbf{P}(\texttt{OPT} < 2^i \mid S \in \mathcal{E}_\theta) < 1/6$, which implies $\mathbf{P}(\texttt{OPT} \geq 2^i \mid S \in \mathcal{E}_\theta) > \frac{5}{6} > 5\frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}$. So

$$\sum_{\theta \text{ not good}} r^*_{i\theta} < \frac{1}{5} \sum_{\theta \text{ not good}} \mathbf{P}(\mathcal{E}_\theta) \cdot \mathbf{P}(\texttt{OPT} \geq 2^i \mid S \in \mathcal{E}_\theta) \leq \frac{1}{5} \sum_{\theta} \mathbf{P}(\texttt{OPT} \geq 2^i \text{ and } S \in \mathcal{E}_\theta) \leq \frac{u^*_i}{5}, \quad (4)$$

where the last inequality uses the fact that $\sum_\theta \mathbf{P}(\texttt{OPT} \geq 2^i \text{ and } S \in \mathcal{E}_\theta) = u^*_i$.

We now upper bound $\sum_{\theta \leq \delta} r^*_{i\theta}$. By Lemma 2, if the realization S is in scale $\theta \leq \delta$, then NA ends before time $t \leq \alpha 2^i$; that is, it does not go beyond phase i. Hence,

$$\sum_{\theta \leq \delta} r^*_{i\theta} \leq \sum_{\theta \leq \delta} \mathbf{P}(S \in \mathcal{E}_\theta) \leq 1 - u_i. \quad (5)$$

We now use the fact that $1 - u^*_i = \sum_\theta r^*_{i\theta}$, where we sum over all scales. So

$$\sum_{\theta > \delta,\ \text{good}} r^*_{i\theta} \geq \sum_{\theta} r^*_{i\theta} - \sum_{\theta \text{ not good}} r^*_{i\theta} - \sum_{\theta \leq \delta} r^*_{i\theta} \geq (1 - u^*_i) - \frac{u^*_i}{5} - (1 - u_i) = u_i - \frac{6u^*_i}{5},$$

where we use (4) and (5). □

We now define a function $g : 2^U \to \mathbb{R}_{\geq 0}$ as follows:

$$g(T) := \sum_{\text{scale } \theta > \delta} \mathbf{P}(\mathcal{E}_\theta) \cdot \mathbb{E}_{S \sim \mathcal{E}_\theta}\left[\frac{f(S \cup T) - f(S)}{Q - f(S)}\right] = \sum_{S \sim \mathcal{S}: f(S) < \tau} \mathbf{P}(\mathcal{S} = S) \cdot \frac{f_S(T)}{Q - f(S)}, \quad \forall T \subseteq U. \quad (6)$$

We use $f_S(T) = f(S \cup T) - f(S)$. The second equality is by Lemma 2. The function g is monotone and submodular because f_S is monotone and submodular for each $S \subseteq U$, and g is a nonnegative linear combination of such functions. Moreover, for any item $\mathcal{X}_e$, its incremental value (the summation in (2)) is $\mathbb{E}_{\mathcal{X}_e}[g(X_e)]$, and we have

$$\text{score}(\mathcal{X}_e) = \frac{1}{c_e} \cdot \mathbb{E}_{\mathcal{X}_e}[g(X_e)].$$

Constrained Stochastic Submodular Maximization. Our analysis makes use of some known results for stochastic submodular maximization. Here, we are given as input a nonnegative monotone submodular function $h : 2^U \to \mathbb{R}_{\geq 0}$ and independent stochastic items $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$ such that each $\mathcal{X}_i$ realizes to some element of the ground set U. There is a cost c_i associated with each item $\mathcal{X}_i$ and a budget B on the total cost. The goal is to select $\mathcal{S} \subseteq \mathcal{X}$ (possibly adaptively) such that the total cost of $\mathcal{S}$ is at most B and it maximizes the expected value $\mathbb{E}_{S \sim \mathcal{S}}[h(S)]$. Note that an adaptive solution selects items $\mathcal{S}$ sequentially (based on prior realizations), so the subset $\mathcal{S}$ is itself random. On the other hand, a nonadaptive solution involves selecting a deterministic subset $\mathcal{S}$ up front. The adaptivity gap for stochastic submodular maximization is at most two; see theorem 1 in Bradac et al. (2019). In other words, for any instance, the optimal adaptive value is at most two times the optimal nonadaptive value.

Lemma 4. *For any good scale θ, there exists a subset $\mathcal{T}_\theta \subseteq \mathcal{X} \setminus \mathcal{S}$ with $c(\mathcal{T}_\theta) \leq \frac{144}{\theta} \cdot 2^i$ such that*

$$\mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_\theta \sim \mathcal{T}_\theta}[f(S \cup T_\theta) - f(S)] \geq \frac{\theta Q}{6} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}. \quad (7)$$

Proof. We construct set $\mathcal{T}_\theta$ as follows. Initially, $\mathcal{T}_\theta \leftarrow \emptyset$. For each $k = 1, 2, \ldots, \frac{144}{\theta}$,

1. Sample realization S of $\mathcal{S}$ from $\mathcal{E}_\theta$.
2. Let $\mathcal{T}_k \subseteq \mathcal{X} \setminus \mathcal{S}$ be an optimal nonadaptive solution to the stochastic submodular maximization instance with items $\mathcal{X} \setminus \mathcal{S}$, submodular function f_S, and cost budget $B = 2^i$. Note that the distribution of items $\mathcal{X} \setminus \mathcal{S}$ is independent of the realization S of $\mathcal{S}$.
3. Set $\mathcal{T}_\theta \leftarrow \mathcal{T}_\theta \cup \mathcal{T}_k$.

By construction, $c(\mathcal{T}_\theta) \leq \frac{144}{\theta} \cdot 2^i$. So we focus on the expected function value. Consider any $S \in \mathcal{E}_\theta$ as the realization of $\mathcal{S}$. Let $\mathcal{T}_S$ denote the nonadaptive solution obtained in step 2 for realization S. Define $w^*_{i,S}$ as the probability that OPT covers f with cost at most 2^i given that $\mathcal{S}$ realizes to S; that is,

$$w^*_{i,S} := \mathbf{P}(\texttt{OPT} \text{ covers } f \text{ with cost at most } 2^i \mid \mathcal{S} = S).$$

Note that $\sum_{S \in \mathcal{E}_\theta} w^*_{i,S} \cdot \mathbf{P}(\mathcal{S} = S) = r^*_{i\theta}$. Let $\texttt{AD}_S$ denote OPT conditional on $\mathcal{S} = S$, restricted to items $\mathcal{X} \setminus \mathcal{S}$ and

until time 2^i. Note that $\mathtt{AD}_S$ is a feasible adaptive solution to the stochastic submodular maximization instance in step 2: every decision path has total cost at most 2^i. The expected value of $\mathtt{AD}_S$ (under function f_S) is at least $(Q - f(S)) \cdot w^*_{i,S}$ by definition of $w^*_{i,S}$. As $S \in \mathcal{E}_\theta$, we have $Q - f(S) \geq \frac{\theta Q}{2}$, which implies $\mathtt{AD}_S$ has value at least $\frac{\theta Q}{2} \cdot w^*_{i,S}$. Now, using the factor two adaptivity gap for stochastic submodular maximization (discussed earlier), it follows that the nonadaptive solution $\mathcal{T}_S$ has expected value

$$\mathbb{E}_{T_S \sim \mathcal{T}_S}[f_S(T_S)] = \mathbb{E}_{T_S \sim \mathcal{T}_S}[f(S \cup T_S) - f(S)] \geq \frac{1}{2} \cdot (\text{value of } \mathtt{AD}_S) \geq \frac{\theta Q}{4} \cdot w^*_{i,S}, \quad \forall S \in \mathcal{E}_\theta.$$

Taking expectations over S (conditional on $S \in \mathcal{E}_\theta$),

$$\begin{aligned}
&\mathbb{E}_S \mathbb{E}_{T_S \sim \mathcal{T}_S}[f(S \cup T_S) - f(S) \mid S \in \mathcal{E}_\theta] \\
&= \sum_{S \in \mathcal{E}_\theta} \mathbb{E}_{T_S \sim \mathcal{T}_S}[f(S \cup T_S) - f(S)] \cdot \mathbf{P}(\mathcal{S} = S \mid S \in \mathcal{E}_\theta) \\
&\geq \sum_{S \in \mathcal{E}_\theta} \frac{\theta Q}{4} \cdot w^*_{i,S} \cdot \mathbf{P}(\mathcal{S} = S \mid S \in \mathcal{E}_\theta) \\
&= \frac{\theta Q}{4} \cdot \sum_{S \in \mathcal{E}_\theta} w^*_{i,S} \cdot \frac{\mathbf{P}(\mathcal{S} = S)}{\mathbf{P}(\mathcal{E}_\theta)} = \frac{\theta Q}{4} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}.
\end{aligned}$$

The first equality uses the fact that the item realizations in $\mathcal{T}_S$ are independent of the realization S of $\mathcal{S}$. The left-hand side of the relation can be rewritten to give

$$\mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S \sim \mathcal{T}_S}[f(S \cup T_S) - f(S)] \geq \frac{\theta Q}{4} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}. \tag{8}$$

For a contradiction to (7), suppose that

$$\mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_\theta \sim \mathcal{T}_\theta}[f(S \cup T_\theta) - f(S)] < \frac{\theta Q}{6} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}. \tag{9}$$

Subtracting Equation (9) from Equation (8) gives the following:

$$\begin{aligned}
\frac{\theta Q}{12} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)} &< \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta}[f(S \cup T_S) - f(S \cup T_\theta)] \\
&\leq \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta}[f(S \cup T_\theta \cup T_S) - f(S \cup T_\theta)] \\
&\leq \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta}[f(T_\theta \cup T_S) - f(T_\theta)],
\end{aligned} \tag{10}$$

where the second inequality uses monotonicity of f and the last one its submodularity.

Let $\mathcal{T}^{(k)} = \mathcal{T}_1 \cup \mathcal{T}_2 \ldots \cup \mathcal{T}_k$ denote the items selected until iteration k. Then,

$$\mathbb{E}_{T_\theta}[f(T_\theta)] = \sum_{k=1}^{144/\theta} \mathbb{E}_{T^{(k)}}[f(T^{(k)}) - f(T^{(k-1)})],$$

where we define $f(T^{(0)}) := 0$. Note that, in each iteration k, the sample S is drawn independently and identically from $\mathcal{E}_\theta$, and items $\mathcal{T}_k = \mathcal{T}_S$ are added.

$$\begin{aligned}
&\mathbb{E}_{T^{(k)}}[f(T^{(k)}) - f(T^{(k-1)})] \\
&= \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T^{(k-1)}}[f(T^{(k-1)} \cup T_S) - f(T^{(k-1)})] \\
&\geq \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta}[f(T_\theta \cup T_S) - f(T_\theta)] > \frac{\theta Q}{12} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}.
\end{aligned}$$

The first inequality uses $\mathcal{T}^{(k-1)} \subseteq \mathcal{T}_\theta$ and submodularity, and the second inequality uses (10). Adding over all iterations k,

$$\mathbb{E}_{T_\theta}[f(T_\theta)] > \sum_k \frac{\theta Q}{12} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)} = \frac{144}{\theta} \cdot \frac{\theta Q}{12} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)} \geq 2Q,$$

where the last inequality uses the fact that θ is a good scale. This is a contradiction because the maximum function value is Q. This completes the proof of (7). □

Lemma 5. *For any $\mathcal{S} \subseteq \mathcal{X}$, there exists a subset $\mathcal{T} \subseteq \mathcal{X} \setminus \mathcal{S}$ of total cost at most $\frac{144}{\delta} \cdot 2^i$ with*

$$\sum_{\mathcal{X}_e \in \mathcal{T}} \mathbb{E}[g(X_e)] \geq \frac{1}{6} \cdot \sum_{\theta > \delta, \text{ good}} r^*_{i\theta}.$$

Proof. Let $\mathcal{B}$ denote the set of good scales θ with $\theta > \delta$. Note that $\sum_{\theta \in \mathcal{B}} \frac{1}{\theta} \leq \frac{1}{\delta}$ as the scales are powers of two. From Lemma 4, let $\mathcal{T}_\theta$ denote the items satisfying (7) for each scale $\theta \in \mathcal{B}$. Define $\mathcal{T} = \cup_{\theta \in \mathcal{B}} \mathcal{T}_\theta$. As claimed, the total cost is

$$c(\mathcal{T}) \leq \sum_{\theta \in \mathcal{B}} c(\mathcal{T}_\theta) \leq 2^i \sum_{\theta \in \mathcal{B}} \frac{144}{\theta} \leq \frac{144}{\delta} \cdot 2^i.$$

Next, we bound the total incremental value as

$$\sum_{\mathcal{X}_e \in \mathcal{T}} \mathbb{E}[g(X_e)] = \mathbb{E}_{T \sim \mathcal{T}}\left[\sum_{u \in T} g(u)\right] \geq \mathbb{E}_{T \sim \mathcal{T}}[g(T)], \tag{11}$$

where the inequality is by submodularity of g.

By definition of function g (see (6)),

$$\begin{aligned}
\mathbb{E}_{T \sim \mathcal{T}}[g(T)] &= \sum_{\theta > \delta} \mathbf{P}(\mathcal{E}_\theta) \cdot \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim \mathcal{T}}\left[\frac{f(S \cup T) - f(S)}{Q - f(S)}\right] \\
&\geq \sum_{\theta \in \mathcal{B}} \mathbf{P}(\mathcal{E}_\theta) \cdot \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim \mathcal{T}}\left[\frac{f(S \cup T) - f(S)}{Q - f(S)}\right].
\end{aligned} \tag{12}$$

In order to bound the last term, consider any $\theta \in \mathcal{B}$. By (7) and $\mathcal{T} \supseteq \mathcal{T}_\theta$, it follows that

$$\mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim \mathcal{T}}[f(S \cup T) - f(S)] \geq \frac{\theta Q}{6} \cdot \frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}, \quad \forall \theta \in \mathcal{B}.$$

Using the fact that $\frac{\theta Q}{2} < Q - f(S) \le \theta Q$ for all $S \in \mathcal{E}_\theta$, we get

$$\mathbb{E}_{S\sim\mathcal{E}_\theta}\mathbb{E}_{T\sim\mathcal{T}}\left[\frac{f(S\cup T)-f(S)}{Q-f(S)}\right] \ge \frac{1}{6}\cdot\frac{r^*_{i\theta}}{\mathbf{P}(\mathcal{E}_\theta)}, \qquad \forall \theta \in \mathcal{B}.$$

Combined with (11) and (12), we get $\sum_{\mathcal{X}_e\in\mathcal{T}}\mathbb{E}[g(X_e)] \ge \frac{1}{6}\cdot\sum_{\theta\in\mathcal{B}} r^*_{i\theta}$, which completes the proof of the lemma. □

Using Lemma 5 and averaging,

$$\max_{\mathcal{X}_e\in\mathcal{X}\setminus\mathcal{S}} \text{score}(\mathcal{X}_e) \ge \max_{\mathcal{X}_e\in\mathcal{T}} \text{score}(\mathcal{X}_e) \ge \frac{\sum_{\mathcal{X}_e\in\mathcal{T}}\mathbb{E}[g(X_e)]}{\sum_{\mathcal{X}_e\in\mathcal{T}} c_e} \ge \frac{\delta}{\beta\cdot 2^i}\cdot\sum_{\theta>\delta,\ \text{good}} r^*_{i\theta}, \tag{13}$$

where $\beta = O(1)$.

Combining (13) and Lemma 3, and using the greedy choice in step 3 (Algorithm 1),

$$\text{score}(\mathcal{X}_{e(t)}) \ge \frac{\delta}{\beta 2^i}\left(u_i - \frac{6}{5}u^*_i\right).$$

We note that this inequality continues to hold (with a larger constant β) even if we choose an item that only maximizes the score (2) within a constant factor.

Using this inequality for each time t during phase i, we have

$$G \ge \alpha 2^{i-1}\cdot\frac{\delta}{\beta 2^i}\left(u_i - \frac{6}{5}u^*_i\right) = \frac{\alpha\delta}{2\beta}\cdot\left(u_i - \frac{6}{5}u^*_i\right). \tag{14}$$

We use this lower bound for G in conjunction with an upper bound, which we prove next.

2.2.2. An Upper Bound for *G*. Here, we show that $G = O(\ln(1/\delta))\cdot u_{i-1}$. We now consider the implementation of the nonadaptive list L and calculate G as a sum of contributions over the observed decision path. Let Π denote the (random) decision path followed by the nonadaptive strategy NA: this consists of a prefix of L along with their realizations. Denote by $\langle X_1, X_2, \dots, \rangle$ the sequence of realizations (each in U) observed on Π. So item $\mathcal{X}_j$ is selected between time $\sum_{\ell=1}^{j-1} c_\ell$ and $\sum_{\ell=1}^{j} c_\ell$. Let $h(p)$ index the first (last) item in Π (if any) that is selected (even partially) during phase i, that is, between time $\alpha 2^{i-1}$ and $\alpha 2^i$. For each index $h \le j \le p$, let t_j denote the duration of time that item $\mathcal{X}_j$ is selected during in phase i, so t_j is the width of interval $[\sum_{\ell=1}^{j-1} c_\ell, \sum_{\ell=1}^{j} c_\ell] \cap [\alpha 2^{i-1}, \alpha 2^i]$. It follows that $t_j \le c_j$. Define $G(\Pi) := 0$ if index h is undefined (i.e., Π terminates before phase i), and otherwise,

$$G(\Pi) := \sum_{j=h}^{p}\frac{t_j}{c_j}\cdot\frac{f(\{X_1,\dots,X_j\})-f(\{X_1,\dots,X_{j-1}\})}{Q-f(\{X_1,\dots,X_{j-1}\})} \le \sum_{j=h}^{p}\frac{f(\{X_1,\dots,X_j\})-f(\{X_1,\dots,X_{j-1}\})}{Q-f(\{X_1,\dots,X_{j-1}\})}. \tag{15}$$

By the stopping criterion for L, the f value *before* the end of Π remains at most $\tau = Q(1-\delta)$. So the denominator, that is, $Q - f(\{X_1,\dots,X_{j-1}\})$, is at least δQ for all j.

Lemma 6. *For any decision path Π, we have $G(\Pi) \le 1 + \ln(1/\delta)$.*

Proof. For each $h \le j \le p$, let $V_j := f(\{X_1,\dots,X_j\})$; also let $V_0 = 0$. For $j \le p-1$, as noted, $V_j \le \tau$; as f is integer-valued, $V_j \in \{0,1,\dots,\lfloor\tau\rfloor\}$. We have

$$\sum_{j=h}^{p-1}\frac{V_j - V_{j-1}}{Q-V_{j-1}} \le \sum_{j=h}^{p-1}\sum_{y=0}^{V_j - V_{j-1}-1}\frac{1}{Q-V_{j-1}-y} \le \sum_{\ell=\delta Q+1}^{Q}\frac{1}{\ell} \le \ln\left(\frac{Q}{\delta Q}\right) = \ln(1/\delta).$$

The second inequality uses $Q - V_{j-1} - y \ge Q - V_j + 1 \ge Q - \tau + 1 = \delta Q + 1$. The right-hand side of (15) is then

$$\sum_{j=h}^{p-1}\frac{V_j - V_{j-1}}{Q-V_{j-1}} + \frac{V_p - V_{p-1}}{Q-V_{p-1}} \le 1 + \ln(1/\delta),$$

where we used $V_p \le Q$. This completes the proof. □

Taking expectations over the various decision paths means $G = \mathbb{E}_\Pi[G(\Pi)]$, so Lemma 6 gives

$$G \le (1+\ln(1/\delta))\cdot\mathbf{P}(\Pi \text{ doesn't terminate before phase } i) = (1+\ln(1/\delta))\cdot u_{i-1}. \tag{16}$$

2.2.3. Wrapping Up. To complete the proof of Lemma 1, we set $\alpha := \frac{8\beta}{\delta}(1+\ln(1/\delta)) = O\left(\frac{\ln(1/\delta)}{\delta}\right)$ and combine (14) and (16) to get

$$u_{i-1} \ge \frac{G}{1+\ln(1/\delta)} \ge \frac{\alpha\delta}{2\beta(1+\ln(1/\delta))}\cdot\left(u_i - \frac{6}{5}u^*_i\right) = 4\left(u_i - \frac{6}{5}u^*_i\right).$$

This completes the proof of Lemma 1 and, hence, Theorem 1.

3. Scenario Submodular Cover

In this section, we describe an r-round adaptive algorithm for the scenario submodular cover problem.

Again, we work with the permutation model of rounds (unless stated otherwise). As before, we have a collection of m stochastic items $\mathcal{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$ with costs $\{c_i\}_{i=1}^m$. In contrast to the independent case, the stochastic items here are correlated, and their joint distribution $\mathcal{D}$ is given as input. The goal is to minimize the expected cost of a solution $\mathcal{S} \subseteq \mathcal{X}$ that realizes to a feasible set (i.e., $f(S) = Q$). The joint distribution $\mathcal{D}$ specifies the (joint) probability that $\mathcal{X}$ realizes to any outcome $X \in U^m$. We refer to the realizations $X \in U^m$ that have a nonzero probability of occurrence as scenarios. Let s denote the number of scenarios in $\mathcal{D}$, that is, the support size of distribution $\mathcal{D}$. The set of scenarios is denoted $M = \{1, \ldots, s\}$, and p_ω denotes the probability of each scenario $\omega \in M$. Note that $\sum_{\omega=1}^{s} p_\omega = 1$. For each scenario $\omega \in M$ and item $\mathcal{X}_e$, we denote by $X_e(\omega) \in U$ the realization of $\mathcal{X}_e$ in scenario ω. The distribution $\mathcal{D}$ can be viewed as selecting a random *realized* scenario $\omega^* \in M$ according to the probabilities $\{p_\omega\}$, after which the item realizations are deterministically set to $\langle X_1(\omega^*), \ldots, X_m(\omega^*)\rangle$. However, an algorithm does not know the realized scenario ω^*: it only knows the realizations of the probed items (using which it can infer a posterior distribution for ω^*). As stated in Section 1.1, our performance guarantee in this case depends on the support size s. We also show that such a dependence is necessary (even when Q is small).

For any subset $\mathcal{S} \subseteq \mathcal{X}$ of items, we denote by $S(\omega)$ the realizations for items in $\mathcal{S}$ under scenario ω. We say that scenario ω is *compatible* with some realization $\{u_e : \mathcal{X}_e \in \mathcal{S}\}$ if and only if, $\mathcal{X}_e(\omega) = u_e$ for all items $\mathcal{X}_e \in \mathcal{S}$.

3.1. The Algorithm

Similar to the algorithm for the independent case, it is convenient to solve a partial cover version of the scenario submodular cover problem. However, the notion of partial progress is different: we use the number of compatible scenarios instead of function value. Formally, in the partial version, we are given a parameter $\delta \in [0,1]$, and the goal is to probe some items $\mathcal{R}$ that realize to a set R such that either (i) the number of compatible scenarios is less than δs or (ii) the function f is fully covered. Clearly, if $\delta = 1/s$, then case (i) cannot happen (it corresponds to zero compatible scenarios), so the function f must be fully covered. We use this algorithm with different parameters δ to solve the r-round version of the problem. We show the following.

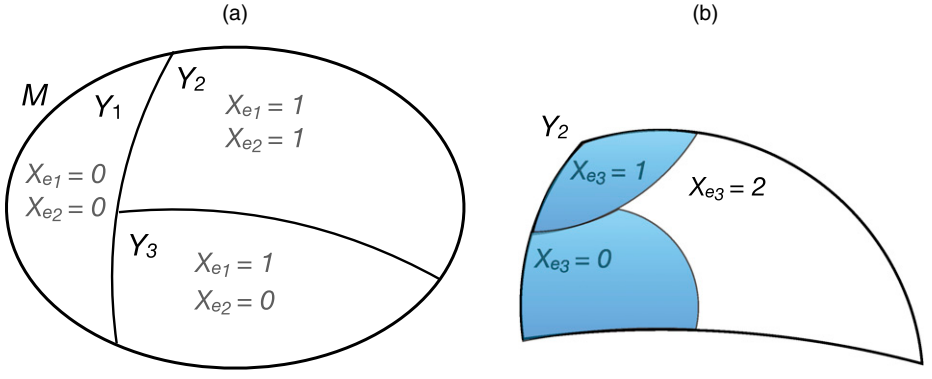
Theorem 6. *There is a nonadaptive algorithm for the partial version of scenario submodular cover with cost $O\left(\frac{1}{\delta}\left(\ln \frac{1}{\delta} + \log Q\right)\right)$ times the optimal adaptive cost of the (full) submodular cover.*

The algorithm first creates an ordering/list L of the items nonadaptively, that is, without knowing the realizations of the items. To do so, at each step we pick a new item that maximizes a carefully defined score function (Equation (17)). The score of an item depends on an estimate of progress toward (i) eliminating scenarios and (ii) covering function f. Before we state this score formally, we need some definitions.

Definition 3. For any $\mathcal{S} \subseteq \mathcal{X}$, let $\mathcal{H}(\mathcal{S})$ denote the partition $\{Y_1, \ldots, Y_\ell\}$ of the scenarios M in which all scenarios in a part have the same realization for items in $\mathcal{S}$. Let $\mathcal{Z} := \{Y \in \mathcal{H}(\mathcal{S}) : |Y| \geq \delta s\}$ be the set of "large" parts having size at least δs.

In other words, scenarios ω and σ lie in the same part of $\mathcal{H}(\mathcal{S})$ if and only if $S(\omega) = S(\sigma)$. Note that partition $\mathcal{H}(\mathcal{S})$ does not depend on the realization of $\mathcal{S}$. Moreover, after probing and realizing items $\mathcal{S}$, the set of compatible scenarios must be one of the parts in $\mathcal{H}(\mathcal{S})$. Also, the number of large parts $|\mathcal{Z}| \leq \frac{s}{\delta s} = \frac{1}{\delta}$ as the number of scenarios $|M| = s$. See Figure 1(a) for an example.

Figure 1. (Color online) Illustrations of Key Definitions



Notes. (a) $\mathcal{S} = \{\mathcal{X}_{e_1}, \mathcal{X}_{e_2}\}$ and we partition the scenarios M based on outcomes of $\mathcal{S}$ to get $\mathcal{H}(S) = \{Y_1, Y_2, Y_3\}$. (b) We further partition scenarios Y_2 based on realizations of $\mathcal{X}_{e_3}$.The part of Y_2 compatible with outcome $X_{e_3} = 2$ is the largest cardinality part, that is, $B_{e_3}(Y_2)$. The shaded region is $L_{e_3}(Y_2)$.

Definition 4. For any $\mathcal{X}_e \in \mathcal{X}$ and subset $Z \subseteq M$ of scenarios, consider the partition of Z based on the realization of $\mathcal{X}_e$. Let $B_e(Z) \subseteq Z$ be the largest cardinality part and define $L_e(Z) := Z \setminus B_e(Z)$.

Note that $L_e(Z)$ comprises several parts of the preceding partition of Z. If the realized scenario $\omega^* \in L_e(Z)$ and $\mathcal{X}_e$ is selected, then at least half the scenarios in Z are eliminated (as being incompatible with X_e). Figure 1(b) illustrates these definitions.

For any $Z \in \mathcal{H}(\mathcal{S})$, note that the realizations $S(\omega)$ are identical for all $\omega \in Z$: we use $S(Z) \subseteq U$ to denote the realization of $\mathcal{S}$ under each scenario in Z.

If $\mathcal{S}$ denotes the previously added items in list L, the score (17) of any item $\mathcal{X}_e$ involves a term for each part $Z \in \mathcal{Z}$, which itself comes from two sources:

- Information gain $\sum_{\omega \in L_e(Z)} p_\omega$, the total probability of scenarios in $L_e(Z)$.
- Relative function gain $\sum_{\omega \in Z} p_\omega \cdot \frac{f(S(Z) \cup X_e(\omega)) - f(S(Z))}{Q - f(S(Z))}$, the expected relative gain obtained by including element X_e, where the expectation is over the scenarios in part Z.

The overall score of item $\mathcal{X}_e$ is the sum of these terms (over all parts in $\mathcal{Z}$) normalized by the cost c_e. In defining the score, we only focus on the large parts $\mathcal{Z}$. If the realization of $\mathcal{S}$ corresponds to any other part, then the number of compatible scenarios is less than δs (and the partial-cover algorithm would terminate).

Once the list L is specified, the algorithm starts probing and realizing the items in this order and does so until either (i) the number of compatible scenarios drops below δs or (ii) the realized function value equals Q. Note that, in case (ii), the function is fully covered. See Algorithm 3 for a formal description of the nonadaptive partial-cover algorithm.

Algorithm 3 (Scenario Partial Covering Algorithm SPARCA$(\mathcal{X}, M, f, Q, \delta)$)

1: $\mathcal{S} \leftarrow \emptyset$ and list $L \leftarrow \langle\rangle$.
2: **while** $\mathcal{S} \neq \mathcal{X}$ **do** ▷ Building the list nonadaptively
3: define $\mathcal{Z}$ and $L_e(Z)$ for each $Z \in \mathcal{Z}$ as in Definitions 3 and 4.
4: select an item $\mathcal{X}_e \in \mathcal{X} \setminus \mathcal{S}$ that maximizes:

$$\text{score}(\mathcal{X}_e) = \frac{1}{c_e} \cdot \sum_{Z \in \mathcal{Z}} \left(\sum_{\omega \in L_e(Z)} p_\omega + \sum_{\omega \in Z} p_\omega \cdot \frac{f(S(Z) \cup X_e(\omega)) - f(S(Z))}{Q - f(S(Z))} \right) \tag{17}$$

5: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{X}_e\}$ and list $L \leftarrow L \circ \mathcal{X}_e$
6: $\mathcal{R} \leftarrow \emptyset$, $R \leftarrow \emptyset$, $H \leftarrow M$.
7: **while** $|H| \geq \delta |M|$ and $f(R) < Q$ **do** ▷ Probing items on the list
8: $\mathcal{X}_e \leftarrow$ first r.v. in list L not in $\mathcal{R}$
9: $X_e = v \in U$ be the realization of $\mathcal{X}_e$.
10: $R \leftarrow R \cup \{v\}, \mathcal{R} \leftarrow \mathcal{R} \cup \{\mathcal{X}_e\}$
11: $H \leftarrow \{\omega \in H : X_e(\omega) = v\}$
12: return probed items $\mathcal{R}$, realizations R and compatible scenarios H.

Given this partial covering algorithm, we immediately get an algorithm for the r-round version of the problem, in which we are allowed to make r rounds of adaptive decisions. Indeed, we can first set $\delta = s^{-1/r}$ and solve the partial covering problem. Suppose we probe the items $\mathcal{R}$ (with realizations $R \subseteq U$) and are left with compatible scenarios $H \subseteq M$. Then we can condition on scenarios H and the marginal value function f_R (which is submodular) and inductively get an $r-1$-round solution for this problem. The following algorithm and result formalizes this.

Algorithm 4 (r-Round Adaptive Algorithm For Scenario Submodular Cover NSC$(r, \mathcal{X}, M, f)$)

1: run SPARCA$(\mathcal{X}, M, f, Q, |M|^{-1/r})$ for round one. Let $\mathcal{R}$ denote the probed items, R their realizations, and $H \subseteq M$ the compatible scenarios returned by SPARCA.
2: define residual submodular function $\widehat{f} := f_R$.
3: recursively solve NSC$(r-1, \mathcal{X} \setminus \mathcal{R}, H, \widehat{f})$.

Theorem 7. *Algorithm 4 is an r-round adaptive algorithm for scenario submodular cover with cost $O(s^{1/r}(\log s + r \log Q))$ times the optimal adaptive cost, where s is the number of scenarios.*

3.2. Analysis for the Partial Covering Algorithm

We now prove Theorem 6. Consider any call to SPARCA. Let $s = |M|$ denote the number of scenarios in the instance. Recall that the goal is to probe items $\mathcal{R} \subseteq \mathcal{X}$ with some realization R and compatible scenarios $H \subseteq M$ such that (i) $|H| < \delta s$ or (ii) $f(R) = Q$. We denote by OPT an optimal adaptive solution for the covering problem on f. Let NA denote the nonadaptive strategy given by algorithm SPARCA. Note that NA probes items in the order given by the list L and stops when either condition (i) or (ii) occurs. We consider the expected cost of this strategy and relate it to the cost of OPT. The high-level approach is similar to that for the independent case, but the details are quite different.

We refer to the cumulative cost incurred until any point in a solution as time. We say that OPT is in phase i in the time interval $[2^i, 2^{i+1})$ for $i \geq 0$. We say that NA is in phase i in the time interval $[\beta \cdot 2^{i-1}, \beta \cdot 2^i)$ for $i \geq 1$. We use phase 0 to refer to the interval $[1, \beta)$. We set $\beta := \frac{16}{\delta} \log(Q/\delta)$; this choice becomes clear in the proof. For any phase $i \geq 0$, we use

- u_i: probability that NA goes beyond phase i, that is, costs at least $\beta \cdot 2^i$.
- u_i^*: probability that OPT goes beyond phase $i-1$, that is, costs at least 2^i.

Because all costs are integers, $u_0^* = 1$. For ease of notation, we also use OPT and NA to denote the random cost incurred by OPT and NA, respectively. The main result here is that $\mathbb{E}[\text{NA}] \le O(\beta) \cdot \mathbb{E}[\text{OPT}]$. As in the independent case, it suffices to show the following key lemma.

Lemma 7. *For any phase $i \ge 1$, we have $u_i \le \frac{u_{i-1}}{4} + 2u_i^*$.*

Using this lemma, we can immediately prove Theorem 6: this part of the analysis is identical to the one for Theorem 4 (using Lemma 1). We provide the complete proof of Lemma 7 in Online Appendix B.

3.3. Tight Approximation Using More Rounds

We now show that a better (and tight) approximation is achievable if we use $2r$ (instead of r) adaptive rounds. The main idea is to use the following variant of the partial covering problem, called scenario submodular partial cover (SSPC). The input is the same as scenario submodular cover: items $\mathcal{X}$, scenarios M with $|M| = s$, and submodular function f with maximal value Q. Given parameters $\delta, \varepsilon \in [0,1]$, the goal in SSPC is to probe some items $\mathcal{R}$ that realize to set $R \subseteq U$ such that either (i) the number of compatible scenarios is less than δs or (ii) the function value $f(R) > Q(1-\varepsilon)$. Unlike the partial version studied previously, we do not require f to be fully covered in case (ii). Note that setting $\varepsilon = \frac{1}{Q}$, we recover the previous partial covering problem, so SSPC is more general.

Corollary 3. *There is a nonadaptive algorithm for* SSPC *with cost $O\left(\frac{1}{\delta}\left(\ln\frac{1}{\delta} + \ln\frac{1}{\varepsilon}\right)\right)$ times the optimal adaptive cost for the (full) submodular cover.*

The algorithm and proof are nearly identical to SPARCA. See Online Appendix B for the details.

The following result shows how SSPC can be used iteratively to obtain a $2r$-round algorithm.

Theorem 8. *There is a $2r$-round adaptive algorithm for scenario submodular cover with cost $O(s^{1/r}\log(sQ))$ times the optimal adaptive cost, where s is the number of scenarios.*

Setting $r = \log s$, we achieve a tight $O(\log(sQ))$ approximation in $O(\log s)$ rounds. Combined with the conversion to set-based rounds (Theorem 10), this proves Corollary 2.

4. Applications

4.1. Stochastic Set Cover

The stochastic set cover problem is a special case of stochastic submodular cover. The input is a universe E of d objects and a collection $\{\mathcal{X}_1, \ldots, \mathcal{X}_m\}$ of m items. Each item $\mathcal{X}_i$ has a cost $c_i \in \mathbb{R}_+$ and corresponds to a random subset of objects (with a known explicit distribution). Different items are independent of each other. The goal is to select a set of items such that the realized subsets cover E and the expected cost is minimized. This problem was first studied by Goemans and Vondrák (2006), in which it is shown that the adaptivity gap lies between $\Omega(d)$ and $O(d^2)$. The correct adaptivity gap for stochastic set cover was posed as an open question by Goemans and Vondrák (2006). Subsequently, Agarwal et al. (2019) made significant progress by showing that the adaptivity gap is $O(d \log d \cdot \log(mc_{max}))$. However, as a function of the natural parameter d (number of objects), the best adaptivity gap remained $O(d^2)$ because the number of stochastic sets m and maximum cost c_{max} may be arbitrarily larger than d.

As a corollary of Theorem 1, we obtain an $O(d \log d)$ adaptivity gap that nearly matches the $\Omega(d)$ lower bound. In fact, for any $r \ge 1$, we obtain an r-round adaptive algorithm that costs at most $O(d^{1/r}\log d)$ times the fully adaptive cost. This nearly matches the $\Omega\left(\frac{1}{r^3}d^{1/r}\right)$ r-round adaptivity gap shown in Agarwal et al. (2019). We note that, when $r = \log d$, we obtain an $O(\log d)$-approximation algorithm with a logarithmic number of adaptive rounds; this approximation ratio is the best possible even for deterministic set cover.

4.2. Sensor Placement with Unreliable Sensors

In the sensor placement problem, we are concerned with deploying a collection of sensors for visual surveillance or to acquire information on air quality, temperature, humidity, etc. One approach to model this problem (suitable for visual surveillance) is to assume that each sensor has a particular sensing region and to minimize the number of sensors required to cover some target region. This corresponds to the art gallery problem (see González-Banos 2001), which, in turn, is a special case of set cover. In the setting with unreliable sensors that we consider, each sensor may fail independently with some probability (in which case it does not cover its region), and the goal is to minimize the expected number of sensors so as to cover the target region. This can be modeled as an instance of stochastic set cover discussed in Section 4.1.

An alternative approach (suitable for information acquisition) is to model sensor readings as Gaussian processes (Deshpande et al. 2004), in which the goal is to place the minimum number of sensors so as to achieve a target level of information gain. Formally, let $[m]$ denote the set of locations and $\mathcal{Z}_i$ be a random variable representing the information at location $i \in [m]$. Let $\mathcal{Z} = \langle \mathcal{Z}_i : i \in [m] \rangle$ denote the information at all locations. A sensor at location i makes observation $\mathcal{Y}_i = \mathcal{Z}_i + \epsilon_i$, where ϵ_i is an independent (random) noise term. For any subset $A \subseteq [m]$, let $\mathcal{Y}_A = \{\mathcal{Y}_i : i \in A\}$ denote the random observations at the locations in A.

Given $\mathcal{Y}_A = Y_A$, we can use $\mathbb{E}[\mathcal{Z}_i|\mathcal{Y}_A = Y_A]$ to make predictions on the information at *any* location i. The information gain of the system if we place sensors at locations $A \subseteq [m]$ is defined as

$$g(A) = \mathbb{H}(\mathcal{Z}) - \mathbb{H}(\mathcal{Z}|\mathcal{Y}_A),$$

where $\mathbb{H}(\mathcal{Z}|\mathcal{Y}_A) = \mathbb{E}_{Y_A}[\mathbb{H}(\mathcal{Z}|\mathcal{Y}_A = Y_A)]$ denotes the conditional entropy of $\mathcal{Z}$ given $\mathcal{Y}_A$.

Krause and Guestrin (2005) show that the function $g(A)$ is monotone and submodular in A assuming that the observations $\mathcal{Y}_A$ are conditionally independent given $\mathcal{Z}$; see, corollary 4 in their paper. The conditional independence assumption is satisfied in our setting because the noise terms ϵ_i are independent. Given a target $\widehat{Q}$ on the information gain, the goal in the deterministic problem is to find sensor locations $A \subseteq [m]$ so that $g(A) \geq \widehat{Q}$.

In the stochastic setting (with unreliable sensors), each sensor $i \in [m]$ is active independently with some probability p_i (and fails otherwise). The goal is to place sensors (possibly adapting based on the active/failure outcomes) so that the information gain from the active sensors is at least $\widehat{Q}$. This can be modeled as an instance of stochastic submodular cover as follows. The items correspond to the m locations. For each sensor at location $i \in [m]$, we define two elements T_i and F_i corresponding to active and failure outcomes, respectively. The ground set $U = \{T_i : i \in [m]\} \cup \{F_i : i \in [m]\}$. For each $i \in [m]$, random variable $\mathcal{X}_i = T_i$ with probability p_i and $\mathcal{X}_i = F_i$ otherwise. Define function $f : 2^U \to \mathbb{R}_+$ as follows

$$f(S) = g(W(S)), \text{ where } W(S) = \{i \in [m] : T_i \in S\}, \quad \forall S \subseteq U.$$

Observe that f is monotone and submodular because g is monotone and submodular (over $[m]$). Furthermore, let M be a large enough integer so that scaling f by a factor of M makes it integer-valued. Theorem 1 can be applied to the scaled function f with target $Q = M \cdot \widehat{Q}$. Then, we obtain an r-round adaptive algorithm of cost at most $O(Q^{1/r} \log Q)$ times an optimal fully adaptive algorithm for the unreliable sensor placement problem.

4.3. Optimal Decision Tree

The optimal decision tree problem captures problems from a variety of fields, such as learning theory, medical diagnosis, pattern recognition, and Boolean logic; see the surveys Moret (1982) and Murthy (1997). In an instance of optimal decision tree (ODT) we are given s hypotheses with probabilities $\{p_i\}_{i=1}^s$. An unknown random hypothesis $y^* \in [s]$ is drawn from this distribution. There is a collection of m tests, and test e costs $c_e \in \mathbb{R}_+$ and returns a positive result if y^* lies in some subset $T_e \subseteq [s]$ and a negative result if $y^* \notin T_e$. The goal is to identify y^* using tests of minimum expected cost.

We can transform any ODT instance into an instance of scenario submodular cover. We associate scenarios with hypotheses, and the distribution $\mathcal{D}$ is given by probabilities $\{p_i\}_{i=1}^s$. For each test e, we have an item $\mathcal{X}_e$ of cost c_e. The ground set $U = \{e^+, e^- : e \text{ test}\}$, which corresponds to all possible (individual) test results. Item $\mathcal{X}_e$ realizes to e^+ (e^-) if test e is positive (,negative). If y^* is the realized scenario, then the item realizations correspond to the test outcomes for hypothesis y^*. For each test e, define subsets $T(e^-) = T_e$ and $T(e^+) = [s] \setminus T_e$. The submodular function is $f(S) = \min\{|\cup_{v \in S} T(v)|, s - 1\}$ for $S \subseteq U$. Note that $f(S)$ is the number of incompatible hypotheses given the tests results S. Moreover, $Q = s - 1$. Applying Theorem 2, for any $r \geq 1$, we obtain an r-round adaptive algorithm that costs at most $O(rs^{1/r} \log s)$ times the fully adaptive optimum. Our lower bound for scenario submodual cover (Theorem 3) also implies an $\Omega\left(\frac{1}{r \log s} s^{1/r}\right)$ bound on the r-round-adaptivity gap for ODT. So, for any constant $r \geq 1$, our r-round-adaptivity gap is tight up to an $O(\log^2 s)$ factor. Moreover, using Corollary 2, we obtain an $O(\log s)$ round algorithm of cost $O(\log s)$ times the adaptive optimum. As shown in Chakaravarthy et al. (2011), $O(\log s)$ is the best possible approximation ratio even for fully adaptive algorithms.

4.3.1. Application to Medical Diagnosis. Recall our motivating application in medical diagnosis in which we know s possible conditions from which a patient may suffer (along with the priors of their occurrence), and our goal is to adaptively perform tests to identify the correct condition as quickly (and cheaply) as possible. This problem, known as automated diagnosis, can be directly cast as the optimal decision tree problem. See, for example, Azar and El-Metwally (2013) for applications of decision tree methods to diagnostic problems. See also Bellala et al. (2012) for an application of ODT in emergency response. Here, a first responder observes the symptoms of a victim of chemical exposure in order to identify the toxic chemical.

4.3.2. Application to Active Learning. In a typical (binary) classification problem, there is an X set of data points, each of which is associated with a + or – label. There is also a set $\mathcal{H}$ of hypotheses, in which each hypothesis provides a +/− labeling of the data points. It is assumed that the true classifier/hypothesis h^* belongs to $\mathcal{H}$. In the average-case setting that we consider, there is also a Bayesian prior $p_{\mathcal{H}}(\cdot)$ for the true hypothesis, that is, $\Pr[h^* = h] = p_{\mathcal{H}}(h)$ for all $h \in \mathcal{H}$. The learner needs to identify the true hypothesis h^*. In *active learning*, the learner can query the label of any point $e \in X$ by incurring some cost c_e (which corresponds to some expert labeling e). The goal is to identify h^* by

querying points (possibly adaptively) at the minimum expected cost. See, for example, Dasgupta (2004), Guillory and Bilmes (2009), and Cicalese et al. (2014) for more details on this approach. This is exactly an instance of an optimal decision tree, in which the data points correspond to tests. Applying Theorem 2, for any $r \geq 1$, we obtain an r-round adaptive algorithm that costs at most $O(r|\mathcal{H}|^{1/r}\log(|\mathcal{H}|))$ times the fully adaptive optimum.

4.4. Shared Filter Evaluation

This problem is introduced by Munagala et al. (2007) in the context of executing multiple queries on a data set with shared (Boolean) filters. There are n independent "filters" $\mathcal{X}_1, \ldots, \mathcal{X}_n$, and each filter i has cost c_i and evaluates to *true* with probability p_i (and *false* otherwise). There are also m conjunctive queries, in which each query $j \in [m]$ is the "and" of some subset $Q_i \subseteq [n]$ of the filters. So query j is true iff X_i = true for all $i \in Q_j$. The goal is to evaluate all queries at the minimum expected cost. This can be modeled as an instance of stochastic submodular cover. The items correspond to filters. The ground set $U = \{T_i, F_i\}_{i=1}^n$, where T_i (F_i) corresponds to filter i evaluating to true (false). The submodular function is

$$f(S) := \sum_{j=1}^{m} \min\{|Q_j|, |Q_j| \cdot |S \cap \{F_i : i \in Q_j\}| + |S \cap \{T_i : i \in Q_j\}|\}.$$

Note that the term for query Q_j is $|Q_j|$ iff the query's value is determined: it is false when a single false filter is seen and it's true when all its filters are true. The maximal value of the function is $Q = \sum_{j=1}^{m} |Q_j| \leq mn$. Using Theorem 1, for any $r \geq 1$, we obtain an r-round adaptive algorithm with cost at most $O((mn)^{1/r} \cdot \log(mn))$ times the optimal adaptive cost.

4.5. Correlated Knapsack Cover

There are n items, each of cost c_i and random (integer) reward $\mathcal{X}_i$. The rewards may be correlated and are given by a joint distribution $\mathcal{D}$ with s scenarios. The exact reward of an item is only known when it is probed. Given a target Q, the goal is to probe some items so that the total realized reward is at least Q, and we minimize the expected cost. This is a special case of scenario submodular cover. The ground set $U = \{(i, v) : i \in [n], 0 \leq v \leq Q\}$, where element (i, v) represents item $\mathcal{X}_i$ realizing to v. Any realization of value at least Q is treated as equal to Q: this is because the target is itself Q. For each element $e = (i, v) \in U$, let $a_e = v$ denote its value. Then, the submodular function is $f(S) = \min\{\sum_{e \in S} a_e, Q\}$ for $S \subseteq U$. By Theorem 1, we obtain an r-round adaptive algorithm with cost at most $O(s^{1/r}(\log s + r \log Q))$ times the optimal adaptive cost.

4.6. Stochastic Score Classification

The stochastic score classification (`SSClass`) problem introduced by Gkenosis et al. (2018) models applications such as assessing disease risk levels of patients and giving quality ratings to products. Formally, there are n binary random variables $\mathcal{X}_1, \ldots, \mathcal{X}_n$, which are used to compute a score $r(X) = \sum_{i=1}^{n} a_i X_i$, where $a_i \in \mathbb{Z}_+$ for all i. The realization $X_i \in \{0, 1\}$ of variable $\mathcal{X}_i$ can be determined by performing a query of cost $c_i \in \mathbb{R}_+$. Additionally, there are $B + 1$ integers $\alpha_1 < \cdots < \alpha_{B+1}$ that partition the possible scores into classes. Realization X of $\mathcal{X}$ is in *class* $j \in [B]$ iff $\alpha_j \leq r(X) \leq \alpha_{j+1} - 1$. We can view the α_j values as "cutoffs" for the classes. The goal is to determine the correct class by querying variables at minimum expected cost. Note that it is not necessary to query all variables to determine the class.

Gkenosis et al. (2018) show that any instance of `SSClass` can be converted into an instance of stochastic submodular cover as follows. The ground set $U = \{(i, 0), (i, 1) : i \in [n]\}$, corresponding to the possible realizations of the variables. Consider any index $j \in \{2, \ldots B\}$. Recall the cutoff α_j and let $\beta_j := \sum_{i=1}^{n} a_i - \alpha_j + 1$. Define two submodular functions:

$$R_j^1(S) := \min\left\{\alpha_j, \sum_{(i,1) \in S} a_i\right\} \quad \text{and}$$

$$R_j^0(S) := \min\left\{\beta_j, \sum_{(i,0) \in S} a_i\right\}, \qquad \forall S \subseteq U.$$

Observe that $R_j^1(S) = \alpha_j$ (i.e., R_j^1 is covered) iff the realizations in S imply that the score is *at least* α_j. Similarly, $R_j^0(S) = \beta_j$ iff the realizations in S imply that the score is *at most* $\alpha_{j+1} - 1$. We combine these two functions using the "OR" construction of submodular functions to get

$$g_j(S) = \alpha_j \beta_j - (\alpha_j - R_j^1(S)) \cdot (\beta_j - R_j^0(S)).$$

Note that g_j is covered (i.e., has value $\alpha_j \beta_j$) if and only if either R_j^1 or R_j^0 is covered. Moreover, g_j is monotone and submodular. Finally, we combine all these B functions using the "AND" construction to obtain $f(S) := \sum_{j=2}^{B} g_j(S)$. Note that f is monotone and submodular, and it is covered if and only if each of the g_j's is covered, which implies that the class is correctly identified. Moreover, the maximal value of f is $Q = \sum_{j=2}^{B} \alpha_j \beta_j \leq BW^2 \leq W^3$, where $W := \sum_{i=1}^{n} a_i$.

Using Theorems 1 and 2, we obtain r-round adaptive algorithms for independent and correlated distributions with approximation ratios $O(W^{3/r} \log W)$ and $O(rs^{1/r} \log(sW))$, respectively (relative to the fully adaptive optimum). Recall that s is the number of scenarios for correlated distributions. Very recently, Ghuge et al. (2022) obtained a nonadaptive (i.e., one round)

algorithm that achieves an $O(1)$ approximation for stochastic score classification with independent distributions; however, their result is not applicable for correlated distributions.

We can also extend our result to the d-dimensional score classification problem, in which one needs to determine the classes of d different score functions $r_1, \cdots r_d$. For each score r_k, we define a submodular function f_k as earlier and define a combined function $f(S) := \sum_{k=1}^{d} f_k(S)$. Now, the maximal value $Q \le dW^3$. So we obtain r-round adaptivity gaps of $O(d^{1/r}W^{3/r}\log(dW))$ and $O(rs^{1/r}\log(dsW))$ for the independent and correlated cases.

5. Computational Results

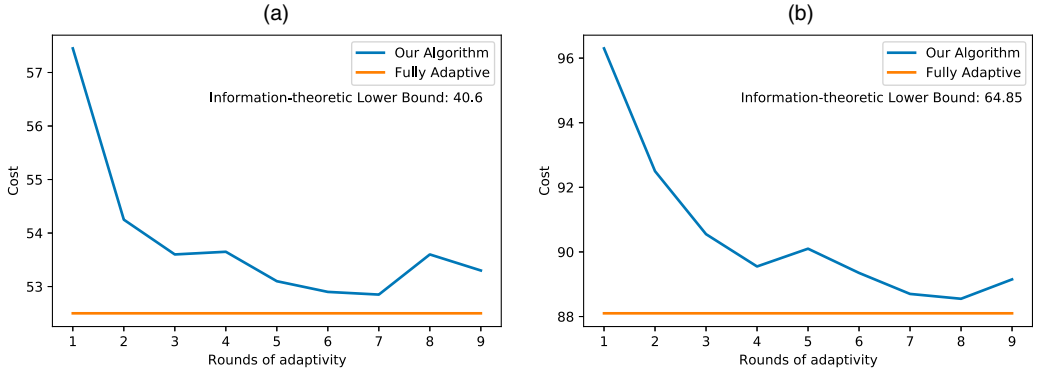
We provide a summary of computational results of our r-round adaptive algorithms for the stochastic set cover and optimal decision tree problems. We conducted all experiments using Python 3.8 and Gurobi 8.1 with a 2.3 Ghz Intel Core *i*5 processor and 16 GB 2133 MHz LPDDR3 memory.

5.1. Stochastic Set Cover

5.1.1. Instances. We use the Epinions network (http://snap.stanford.edu/) and a Facebook messages data set described in Rossi and Ahmed (2015) to generate instances of the stochastic set cover problem. The Epinions network consists of 75,879 nodes and 508,837 directed edges. We compute the subgraph induced by the top 1,000 nodes with the highest out-degree (this subgraph has 57,092 directed edges) and use this to generate the stochastic set cover instance. The Facebook messages data set consists of 1,266 nodes and 6,451 directed edges. Given an underlying directed graph, we generate an instance of the stochastic set cover problem as follows. We associate the ground set U with the set of nodes of the underlying graph. We associate an item $\mathcal{X}_u$ with each node u. Let $\Gamma(u)$ denote the out-neighbors of u. We sample a subset of $\Gamma(u)$ using the binomial distribution with $p = 0.1$ for 500 times. Let $S \subseteq \Gamma(u)$ be sampled α_S times: then, $\mathcal{X}_u$ realizes to $S \cup \{u\}$ with probability $\alpha_S/500$. This ensures that $\mathcal{X}_u$ always covers u. We set f to be the coverage function. We interpret the realization of a node as the set of neighbors it influences, and so f computes the total number of people that are influenced. We set $Q = \delta n$, where n represents the number of nodes in the underlying network. We use $\delta = 0.5$ for the Epinions network. However, because the Facebook messages network is sparse, we set $\delta = 0.25$ in the second instance. All costs are one.

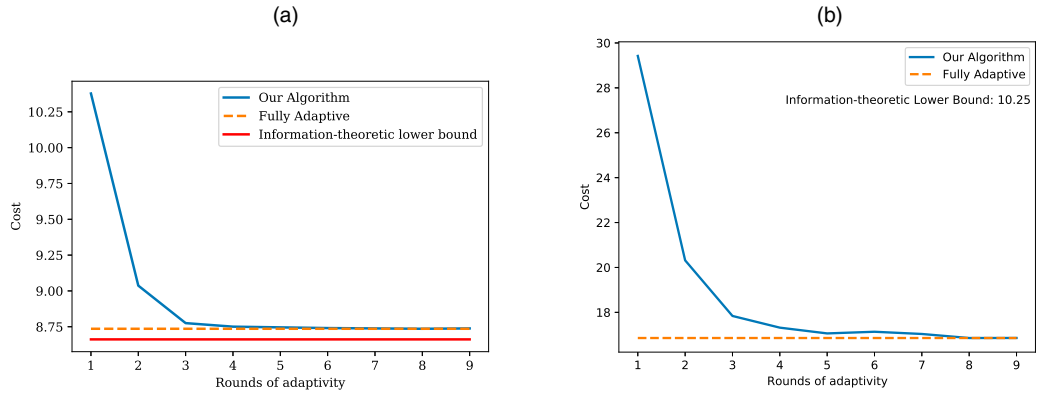
5.1.2. Results. We test our r-round adaptive algorithm on the two kinds of instances described. We vary the number r of rounds over integers in the interval $[1, \log n]$, where $n \approx 1,000$. We compare with our fully adaptive algorithm, which adapts after every probe: in each step, this algorithm probes the item that maximizes the score (2) with $\mathcal{S} = \emptyset$. We note that this also corresponds to the adaptive algorithm from Im et al. (2016). To compute an estimate of the expected cost of any algorithm, we sample item realizations 20 times independently and take the average cost incurred. We also find an estimate of an information-theoretic lower bound by sampling item realizations 20 times and taking the average cost of an integer linear program (solved using the Gurobi solver) that computes the optimal *off-line* cost to cover Q elements for a given realization. Note that no adaptive policy can do better than this lower bound. In fact, the gap between this information-theoretic lower bound and an optimal adaptive solution may be as large as $\Omega(Q)$ on worst case instances. We observe that, in both cases, the expected cost of our solution after only a few rounds of adaptivity is within 50% of the information-theoretic lower bound. Moreover, in six to seven rounds of adaptivity, we notice a decrease of $\approx 8\%$ in the expected cost, and these solutions are nearly as good as fully adaptive solutions. We plot this trend in Figure 2.

Figure 2. (Color online) Computational Results for Stochastic Set Cover



Notes. (a) Epinions network. (b) Facebook messages network.

Figure 3. (Color online) Computational Results for ODT on WISER Data Set



Notes. (a) WISER − U. (b) WISER − R.

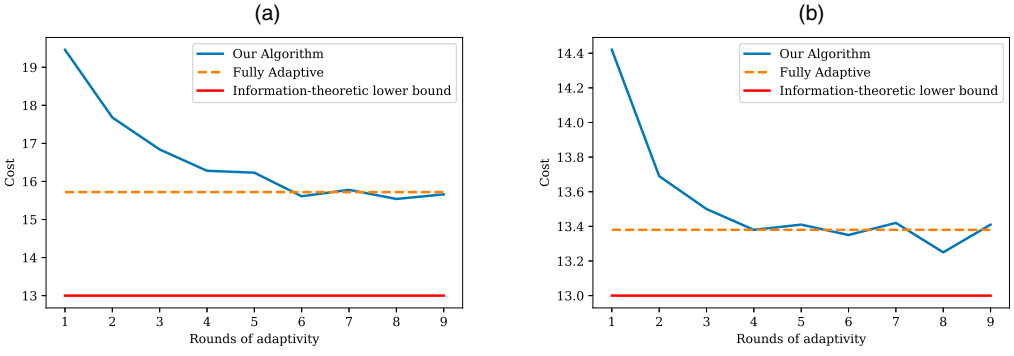
Finally, note that the increase in expected cost with rounds of adaptivity (see Figure 2(a)) can be attributed to the probabilistic nature of our algorithm (and the experimental setup). We also notice this in the next batch of experiments.

5.2. Optimal Decision Tree

5.2.1. Instances. We use a real-world data set called WISER (http://wiser.nlm.nih.gov/) for our experiments. The WISER data set describes symptoms that one may suffer from after being exposed to certain chemicals. It contains data corresponding to 415 chemicals (scenarios for ODT) and 79 symptoms (elements with binary outcomes). Given a patient exhibiting certain symptoms, the goal is to identify the chemical to which the patient has been exposed (by testing as few symptoms as possible). This data set is used for evaluating algorithms for similar problems in other papers (Bhavnani et al. 2007; Bellala et al. 2011, 2012; Navidi et al. 2020). For each symptom–chemical pair, the data specifies whether someone exposed to the chemical exhibits the given symptom. However, the WISER data has "unknown" entries for some pairs. In order to obtain instances for ODT from this, we generate 10 different data sets by assigning random binary values to the unknown entries. Then, we remove all identical scenarios to ensure that the ODT instance is feasible. We use the uniform probability distribution for the scenarios. Given these 10 data sets, we consider two cases. The first case (called WISER − U) has all tests with unit costs. In the second case, we generate costs randomly for each test from $\{1,4,7,10\}$ according to the probabilities $[0.1,0.2,0.4,0.3]$; for example, with probability 0.4, a test is assigned cost 7. Varying cost captures the setting in which tests may have different costs, and we may not want to schedule an expensive test without sufficient information. We refer to this case as WISER − R.

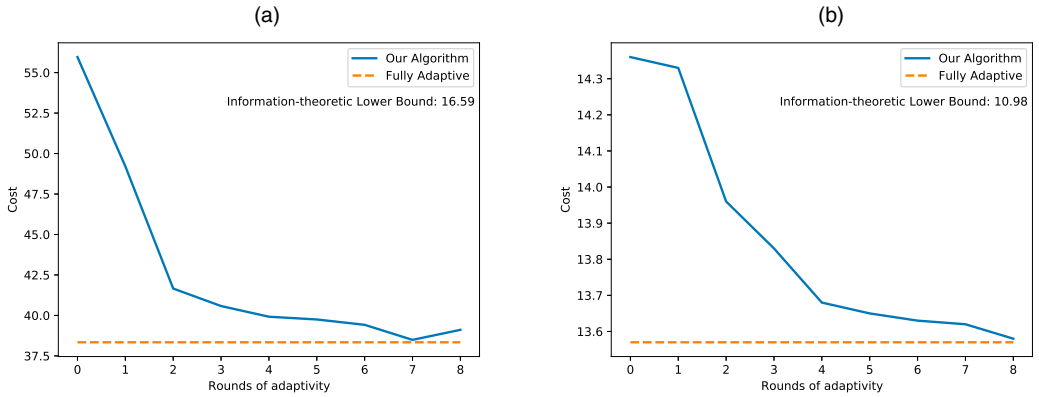
We also test our algorithm on synthetic data that we generate as follows. We set $s = 10{,}000$ and $m = 100$. For each $y \in [s]$, we randomly generate a binary sequence of outcomes that correspond to how y reacts to the tests. We do this in two ways: for test e, we set $y \in T_e$ with probability $p \in \{0.2,0.5\}$. If a sequence of outcomes is repeated, we discard the scenario to ensure feasibility of the ODT instance. We assign equal probability to each

Figure 4. (Color online) Computational Results for ODT on Synthetic Data with Unit Costs



Notes. (a) SYN − U − 0.2. (b) SYN − U − 0.5.

Figure 5. (Color online) Computational Results for ODT on Synthetic Data with Nonuniform Costs



Notes. (a) SYN − R − 0.2. (b) SYN − R − 0.5.

scenario. We generate instances using (i) unit costs or (ii) random costs from $\{1,4,7,10\}$ with respective probabilities $[0.1,0.2,0.4,0.3]$. Thus, we generate four types of instances with synthetic data. We refer to the instance generated with $p = 0.2$ and unit costs as SYN − U − 0.2. Other instances are named similarly.

5.2.2. Results. We test our r-round adaptive algorithm on all of the aforementioned data sets. We vary r over integers in the interval $[1, \log s]$. Again, we compare with our fully adaptive algorithm, which adapts after every probe: in each step, this algorithm probes the item that maximizes the score (17) in which $\mathcal{S} = \emptyset$ and $\mathcal{Z}$ consists of a single part with all scenarios. We also implemented the fully adaptive "generalized binary search" algorithm from Dasgupta (2004) as an algorithmic benchmark: on the instances considered, this algorithm performed nearly identically to our fully adaptive algorithm, so we exclude it from the plots.

We also compare with information-theoretic lower bounds obtained as follows. For instances with unit costs (WISER − U, SYN − U − 0.2 and SYN − U − 0.5) this lower bound corresponds to the entropy that is $\log_2(s)$, where s is the number of scenarios. Recall that all instances have uniform probabilities across scenarios. For instances with nonuniform costs (WISER − R, SYN − R − 0.2 and SYN − R − 0.5), the entropy is no longer a valid lower bound. Instead, we use an integer programming formulation to compute the optimal cost to identify a given scenario (see Online Appendix F for details) and then average over all scenarios. We note that there are instances in which this information-theoretic lower bound is smaller than the optimal adaptive cost by an $O(s)$ factor.

For the WISER data sets, we compute averages using all scenarios. Figure 3(a) plots our algorithm's expected cost as a function of r for WISER − U. We observe that our algorithm gets very close to the information-theoretic lower bound in only three rounds of adaptivity. Figure 3(b) plots our algorithm's costs for WISER − R. Here, we observe a sharp decrease in costs within four rounds of adaptivity, after which our algorithm's cost is within $\approx$ 50% of the information-theoretic lower bound. Note that we only plot results on our first data set for WISER − U and WISER − R. We include plots for all 10 WISER data sets in Online Appendix G.

For the synthetic data, we compute averages by sampling scenarios over 100 trials (because $s = 10{,}000$, computing expectation over all s would be very slow). We plot the results in Figures 4 and 5. We observe that, with six rounds of adaptivity, our algorithm's cost nearly matches that of the fully adaptive algorithm.

Acknowledgments

A preliminary version of this paper appeared as Ghuge et al. (2021) in the proceedings of the 38th International Conference on Machine Learning (ICML), 2021.

References

Agarwal A, Assadi S, Khanna S (2019) Stochastic submodular cover with limited adaptivity. *Proc. 30th Annual ACM-SIAM Sympos. Discrete Algorithms* (ACM-SIAM), 323–342.

Asadpour A, Nazerzadeh H (2016) Maximizing stochastic monotone submodular functions. *Management Sci.* 62(8):2374–2391.

Azar AT, El-Metwally SM (2013) Decision tree classifiers for automated medical diagnosis. *Neural Comput. Appl.* 23:2387–2403.

Balkanski E, Singer Y (2018) The adaptive complexity of maximizing a submodular function. *Proc. 50th Annual ACM SIGACT Sympos. Theory Comput.* (ACM, New York), 1138–1151.

Balkanski E, Rubinstein A, Singer Y (2019) An exponential speedup in parallel running time for submodular maximization without loss in approximation. *Proc. 30th Annual ACM-SIAM Sympos. Discrete Algorithms* (ACM-SIAM), 283–302.

Bansal N, Nagarajan V (2015) On the adaptivity gap of stochastic orienteering. *Math. Programming* 154(1–2):145–172.

Bansal N, Gupta A, Li J, Mestre J, Nagarajan V, Rudra A (2012) When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica* 63(4):733–762.

Barinova O, Lempitsky V, Kholi P (2012) On detection of multiple object instances using hough transforms. *IEEE Trans. Pattern Anal. Machine Intelligence* 34(9):1773–1784.

Bateni M, Esfandiari H, Mirrokni VS (2018) Optimal distributed submodular optimization via sketching. *Proc. 24th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 1138–1147.

Behnezhad S, Derakhshan M, Hajiaghayi M (2020) Stochastic matching with few queries: (1-ϵ) approximation. *Proc. 52nd Annual ACM SIGACT Sympos. Theory Comput.*, 1111–1124.

Bellala G, Bhavnani S, Scott C (2011) Active diagnosis under persistent noise with unknown noise distribution: A rank-based approach. *Proc. 14th Internat. Conf. Artificial Intelligence Statist.*, vol. 15, 155–163.

Bellala G, Bhavnani SK, Scott C (2012) Group-based active query selection for rapid diagnosis in time-critical situations. *IEEE Trans. Inform. Theory* 58(1):459–478.

Bhalgat A, Goel A, Khanna S (2011) Improved approximation results for stochastic knapsack problems. *Proc. 22nd Annual ACM-SIAM Sympo. Discrete Algorithms* (ACM-SIAM), 1647–1665.

Bhavnani SK, Abraham A, Demeniuk C, Gebrekristos M, Gong A, Nainwal S, Vallabha GK, Richardson RJ (2007) Network analysis of toxic chemicals and symptoms: Implications for designing first-responder systems. *AMIA Annual Sympos. Proc.*, 51–55.

Bradac D, Singla S, Zuzic G (2019) (Near) optimal adaptivity gaps for stochastic multi-value probing. *Approx. Randomization Combin. Optim.*, 145:49:1–49:21.

Chakaravarthy VT, Pandit V, Roy S, Awasthi P, Mohania MK (2011) Decision trees for entity identification: Approximation algorithms and hardness results. *ACM Trans. Algorithms* 7(2):1–22.

Chekuri C, Quanrud K (2019) Parallelizing greedy for submodular set function maximization in matroids and beyond. *Proc. 51st Annual ACM SIGACT Sympos. Theory Comput.* (ACM, New York), 78–89.

Chen Y, Shio H, Montesinos CAF, Koh LP, Wich S, Krause A (2014) Active detection via adaptive submodularity. *Proc. 31st Internat. Conf. Machine Learn.*, 55–63.

Cicalese F, Laber ES, Saettler AM (2014) Diagnosis determination: Decision trees optimizing simultaneously worst and expected testing cost. *Proc. 31th Internat. Conf. Machine Learn.*, 414–422.

Correa JR, Foncea P, Hoeksma R, Oosterwijk T, Vredeveld T (2018) Recent developments in prophet inequalities. *ACM SIGecom Exchanges* 17(1):61–70.

Dasgupta S (2004) Analysis of a greedy active learning strategy. *Adv. Neural Inform. Processing Systems* 17:337–344.

Dean BC, Goemans MX, Vondrák J (2008) Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.* 33(4):945–964.

Deshpande A, Hellerstein L, Kletenik D (2016) Approximation algorithms for stochastic submodular set cover with applications to Boolean function evaluation and min-knapsack. *ACM Trans. Algorithms* 12(3):1–28.

Deshpande A, Guestrin C, Madden SR, Hellerstein JM, Hong W (2004) Model-driven data acquisition in sensor networks. *Proc. 30th Internat. Conf. Very Large Data Bases*, vol. 30, 588–599.

Dinur I, Steurer D (2014) Analytical approach to parallel repetition. *Proc. 46th Annual ACM Sympos. Theory Comput.*, 624–633.

Esfandiari H, Karbasi A, Mirrokni V (2021a) Adaptivity in adaptive submodularity. *Proc. 34th Conf. Learn. Theory*, vol. 134 (PMLR), 1823–1846.

Esfandiari H, Karbasi A, Mehrabian A, Mirrokni V (2021b) Regret bounds for batched bandits. *Proc. Conf. AAAI Artificial Intelligence* 35(8):7340–7348.

Feldman M, Svensson O, Zenklusen R (2021) Online contention resolution schemes with applications to Bayesian selection problems. *SIAM J. Comput.* 50(2):255–300.

Gao Z, Han Y, Ren Z, Zhou Z (2019) Batched multi-armed bandits problem. *Adv. Neural Inform. Processing Systems*, vol. 32 (Curran Associates, Inc., Red Hook, NY), 501–511.

Garey M, Graham R (1974) Performance bounds on the splitting algorithm for binary testing. *Acta Informatica* 3:347–355.

Ghuge R, Gupta A, Nagarajan V (2021) The power of adaptivity for stochastic submodular cover. *Proc. 38th Internat. Conf. Machine Learn.*, 3702–3712.

Ghuge R, Gupta A, Nagarajan V (2022) Non-adaptive stochastic score classification and explainable halfspace evaluation. Aardal K, Sanità L, eds. *Proc. Integer Programming Combin. Optim. 23rd Internat. Conf.* (Springer), 277–290.

Gkenosis D, Grammel N, Hellerstein L, Kletenik D (2018) The stochastic score classification problem. *Proc. 26th Annual Eur. Sympos. Algorithms*, vol. 112 (Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik), 1–14.

Goemans M, Vondrák J (2006) Stochastic covering and adaptivity. LATIN 2006: Theoretical Informatics (Springer, Berlin/Heidelberg), 532–543.

Golovin D, Krause A (2011) Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artificial Intelligence Res.* 42(1):427–486.

Golovin D, Krause A (2017) Adaptive submodularity: A new approach to active learning and stochastic optimization. Preprint, submitted December 6, https://arxiv.org/abs/1003.3967.

González-Banos H (2001) A randomized art-gallery algorithm for sensor placement. *Proc. 17th Annual Sympos. Comput. Geometry*, 232–240.

Grammel N, Hellerstein L, Kletenik D, Lin P (2016) Scenario submodular cover. *Internat. Workshop Approx. Online Algorithms* (Springer), 116–128.

Guha S, Munagala K (2009) Multi-armed bandits with metric switching costs. *Proc. Automata Languages Programming 36th Internat. Colloquium*, 496–507.

Guillory A, Bilmes JA (2009) Average-case active learning with costs. *Algorithmic Learn. Theory 20th Internat. Conf. Proc.* (Springer), 141–155.

Gupta A, Nagarajan V (2013) A stochastic probing problem with applications. *Integer Programming Combin. Optim. 16th Internat. Conf.*, 205–216.

Gupta A, Nagarajan V, Ravi R (2017a) Approximation algorithms for optimal decision trees and adaptive TSP problems. *Math. Oper. Res.* 42(3):876–896.

Gupta A, Nagarajan V, Singla S (2017b) Adaptivity gaps for stochastic probing: Submodular and XOS functions. *Proc. 28th Annual ACM-SIAM Sympos. Discrete Algorithms* (ACM-SIAM), 1688–1702.

Gupta A, Krishnaswamy R, Nagarajan V, Ravi R (2015) Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.* 40(1):56–79.

Im S, Nagarajan V, van der Zwaan R (2016) Minimum latency submodular cover. *ACM Trans. Algorithms* 13(1):1–28.

Javdani S, Chen Y, Karbasi A, Krause A, Bagnell D, Srinivasa SS (2014) Near optimal Bayesian active learning for decision making. *AISTATS*, 430–438.

Karbasi A, Mirrokni VS, Shadravan M (2021) Parallelizing Thompson sampling. *Adv. Neural Inform. Processing Systems* 34:10535–10548.

Kempe D, Kleinberg JM, Tardos É (2015) Maximizing the spread of influence through a social network. *Theory Comput.* 11:105–147.

Kleinberg R, Weinberg SM (2019) Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econom. Behav.* 113:97–115.

Kosaraju SR, Przytycka TM, Borgstrom RS (1999) On an optimal split tree problem. *Proc. Sixth Internat. Workshop Algorithms Data Structures*, 157–168.

Krause A, Guestrin C (2005) Near-optimal nonmyopic value of information in graphical models. *Proc. 21st Conf. Uncertainty Artificial Intelligence*, 324–331.

Krengel U, Sucheston L (1977) Semiamarts and finite values. *Bull. Amer. Math. Soc.* 83(4):745–747.

Lin H, Bilmes J (2011) A class of submodular functions for document summarization. *Proc. 49th Annual Meeting Assoc. Comput. Linguistics: Human Language Tech.*, vol. 1, 510–520.

Liu Z, Parthasarathy S, Ranganathan A, Yang H (2008) Near-optimal algorithms for shared filter evaluation in data stream systems. *Proc. ACM SIGMOD Internat. Conf. Management Data* (ACM, New York), 133–146.

Loveland DW (1985) Performance bounds for binary testing with arbitrary weights. *Acta Informatica* 22(1):101–114.

Mirzasoleiman B, Karbasi A, Badanidiyuru A, Krause A (2015) Distributed submodular cover: Succinctly summarizing massive data. *Adv. Neural Inform. Processing Systems* 28:2881–2889.

Moret BME (1982) Decision trees and diagrams. *ACM Comput. Surveys* 14(4):593–623.

Munagala K, Srivastava U, Widom J (2007) Optimization of continuous queries with shared expensive filters. *27th ACM SIGMOD-SIGACT-SIGART Sympos. Principles Database Systems*, 215–224.

Murthy SK (1997) Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining Knowledge Discovery* 2:345–389.

Navidi F, Kambadur P, Nagarajan V (2020) Adaptive submodular ranking and routing. *Oper. Res.* 68(3):856–877.

Radanovic G, Singla A, Krause A, Faltings B (2018) Information gathering with peers: Submodular optimization with peer-prediction constraints. *Proc. Conf. Artificial Intelligence*, 1603–1618.

Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. *Proc. 29th AAAI Conf. Artificial Intelligence*, 4292–4293.

Rubinstein A, Singla S (2017) Combinatorial prophet inequalities. Klein PN, ed. *Proc. Annual ACM-SIAM Sympos. Discrete Algorithms* (ACM-SIAM), 1671–1687.

Simon I, Snavely N, Seitz SM (2007) Scene summarization for online image collections. *Proc. 11th Internat. Conf. Comput. Vision*, 1–8.

Sipos R, Swaminathan A, Shivaswamy P, Joachims T (2012) Temporal corpus summarization using submodular word coverage. *Proc. 21st ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 754–763.

Sun C, Li VOK, Lam JCK, Leslie I (2019) Optimal citizen-centric sensor placement for air quality monitoring: A case study of city of Cambridge, the United Kingdom. *IEEE Access* 7:47390–47400.

Wolsey L (1982) An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* 2(4):385–393.

Rohan Ghuge is a PhD candidate in industrial and operations engineering at the University of Michigan. Rohan's research interests include combinatorial and stochastic optimization, approximation algorithms, and problems in assortment optimization and network design.

Anupam Gupta is a professor in the computer science department at Carnegie Mellon University. His research interests are broadly in the design and analysis of algorithms, primarily in optimization under uncertainty, in developing approximation algorithms for NP-hard optimization problems, and in understanding the algorithmic properties of metric spaces. He is an ACM Fellow and a recipient of the Alfred P. Sloan Research Fellowship and the NSF Career award.

Viswanath Nagarajan is an associate professor of industrial and operations engineering at the University of Michigan. Viswanath's research interests are in combinatorial optimization, approximation algorithms, and stochastic optimization.