The Power of Adaptivity for Stochastic Submodular Cover

Rohan Ghuge∗ Anupam Gupta† Viswanath Nagarajan∗

July 1, 2021

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). While such adaptive solutions achieve the best objective, the inherently sequential nature makes them undesirable in many applications. We ask: how well can solutions with only a few adaptive rounds approximate fully-adaptive solutions? We give nearly tight answers for both independent and correlated settings, proving smooth tradeoffs between the number of adaptive rounds and the solution quality, relative to fully adaptive solutions. Experiments on synthetic and real datasets 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.

1 Introduction

Submodularity is a fundamental notion that arises in applications such as image segmentation, data summarization [SSS07, LB11, SSSJ12], hypothesis identification [BLK12, CSM+14], information gathering [RSKF18], and social networks [KKT15]. The submodular cover optimization problem requires us to pick a minimum-cost subset \( S \) of items to cover a monotone submodular function \( f \). Submodular cover has been extensively studied in machine learning, computer science and operations research. Here are two examples from sensor deployment and medical diagnosis.

In the sensor deployment setting, we consider the problem of deploying a collection of sensors to monitor some phenomenon such as air quality or traffic situations. The area each sensor can cover depends on its sensing range. The goal of the problem is to deploy as few sensors as possible to cover a desired region entirely. The area covered as a function of the sensors deployed is a submodular function, and we can cast the sensor deployment problem as a special case of submodular cover.

In the medical diagnosis example, we have \( s \) possible conditions (hypotheses) the patient may suffer from along with the priors on their occurrence, and our goal is to perform tests to identify the correct condition as quickly as possible [GG74, KP99, Das04, CLS14]. This can be cast as submodular cover by viewing each test as eliminating all inconsistent hypotheses. Hence we want a

∗Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, USA. Research supported in part by NSF grants CMMI-1940766 and CCF-2006778.

†Department of Computer Science, Carnegie Mellon University, Pittsburgh, USA. Supported in part by NSF awards CCF-1907820, CCF-1955785, and CCF-2006953.
coverage value of \( s - 1 \): once \( s - 1 \) inconsistent hypotheses are eliminated, the remaining one must be correct.

Observe that both these applications involve uncertain data: the precise area covered by a sensor is not known before the sensor is actually setup and the precise outcome (positive/negative) of a test is not known until the action has been taken. This uncertainty can be modeled using stochastic submodular optimization, where the items are stochastic. As a simple example of the stochastic nature, each item may be active or inactive (with known probabilities), and only active items contribute to the submodular function.

A solution for stochastic submodular cover is now a sequential decision process. At each step, an item is probed and its realization (e.g., active or inactive) is observed. The process is typically adaptive, where all the information from previously probed items is used to identify the next item to probe. This process continues until the submodular function is covered, and the goal is to minimize the expected cost of probed items. Such adaptive solutions are inherently fully sequential, which is undesirable if probing an item is time-consuming. E.g., in sensor deployment, probing a sensor corresponds to physically deploying a sensor and observing whether it functions as expected, which may take hours or days. Or, probing/performing a test in medical diagnosis may involve a long wait for test results. Therefore, we prefer solutions with only few rounds of adaptivity.

Motivated by this, we ask: how well do solutions with only a few adaptive rounds approximate fully-adaptive solutions for the stochastic submodular cover problem? We consider both cases where realizations of different items are independent, and where they are allowed to be correlated. For both these situations we give nearly tight answers, with smooth tradeoffs between the number \( r \) of adaptive rounds and the solution quality (relative to fully adaptive solutions).

The main contribution of our work is an \( r \)-round adaptive solution for stochastic submodular cover in the “set-based” model for adaptive rounds. In this model, a fixed subset of items is probed in parallel every round (and their total cost is incurred). The decisions in the current round can depend on the realizations seen in all previous rounds. However, as noted in [AAK19], if we require function \( f \) to be covered with probability one then the \( r \)-round-adaptivity gap turns out to be very large. (See §A for an example.) Therefore, we focus on set-based solutions that are only required to cover the function with high probability.

In designing algorithms, it turns out to be more convenient to work with the “permutation” model for adaptive rounds, where the function is covered with probability one. This model was also used in prior literature [GV06,AAK19]. Here, every round of an \( r \)-round-adaptive solution specifies an ordering of all remaining items and probes them in this order until some stopping rule. See Definition 2.1 for a formal definition. Moreover, our \( r \)-round adaptive algorithm in the permutation model can be transformed into an \( r \)-round adaptive algorithm in the set-based model. We obtain algorithms in the set-based model that:

- for any \( \eta \in (0, 1) \), finds an \( r \)-round adaptive solution that has expected cost at most \( \frac{r\alpha}{\eta} \cdot \OPT \) and covers the function with probability at least \( 1 - \eta \).

- finds an \( O(r) \)-round adaptive solution that has expected cost at most \( O(\alpha) \cdot \OPT \) and covers the function with probability at least \( 1 - e^{-\Omega(r)} \).

Here \( \OPT \) is the cost of an optimal fully-adaptive solution and \( \alpha \) is the approximation ratio of our algorithm in the permutation model.

The first algorithm above is for the case where \( r \), the number of rounds of adaptivity, is small (say, a constant). In this, we keep the number of rounds the same, but we lose a factor \( r \) in the
expected cost. The second algorithm is for the case that \( r \) is large, e.g., more than a constant. Here, the number of set-based rounds increases by a factor 2, but we only lose a constant factor in expected cost. We formalize and prove these results in \( \S A \). For the rest of the paper, an \( r \)-round adaptive algorithm refers to an an \( r \)-round adaptive algorithm in the permutation model (unless specified otherwise).

### 1.1 Results and Techniques

Consider a monotone submodular function \( f : 2^U \rightarrow \mathbb{Z}_{\geq 0} \) with \( Q := f(U) \). There are \( m \) items, where each item \( i \) is a random variable \( X_i \) having cost \( c_i \) and corresponding to a random element of \( U \). (Our results extend to the more general setting where each item realizes to a subset of \( U \).) The goal is to select a set of items such that the union \( S \) of their corresponding realizations satisfies \( f(S) = Q \), and the expected cost is minimized. Our first result is when the items have independent distributions.

**Theorem 1.1 (Independent Items).** For any integer \( r \geq 1 \), there is an \( r \)-round adaptive algorithm for the stochastic submodular cover problem with cost \( O(Q^{1/r} \cdot \log Q) \) times the cost of an optimal adaptive algorithm.

This improves over an \( O(r Q^{1/r} \log Q \log(mc_{\text{max}})) \) bound from \([AAK19]\) 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 \( \Omega(Q^{1/r}) \) from \([AAK19]\). 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 A.2, we get:

**Corollary 1.2 (Independent Items: Set-Based Model).** There is an \( O(\log Q) \) round algorithm for stochastic submodular cover in the set-based model with cost \( O(\log Q) \) times the optimal (fully) adaptive cost.

This approximation ratio of \( O(\log Q) \) is the best possible (unless \( P=NP \)) even with an arbitrary number \( (r = m) \) of adaptive rounds. Previously, one could only obtain an \( O(\log^2 Q \log(mc_{\text{max}})) \)-approximation in a logarithmic number of rounds \([AAK19]\), or an \( O(\log(mQc_{\text{max}})) \)-approximation in \( O(\log m \log(Qmc_{\text{max}})) \) set-based rounds \([EKM19]\).

Moreover, Theorem 1.1 (with \( r = 1 \)) implies an \( O(Q \log Q) \) adaptivity gap for stochastic set cover (a special case of submodular cover), resolving an open question from \([GV06]\) up to an \( O(\log Q) \) factor, where \( Q \) is the number of objects in set cover.

Our second set of results is for the case where items have correlated distributions. Let \( s \) denote the support size of the joint distribution \( \mathcal{D} \), i.e., the number of scenarios.

**Theorem 1.3 (Correlated Items).** For any integer \( r \geq 1 \), there is an \( r \)-round adaptive algorithm for scenario submodular cover with cost \( O\left(s^{1/r}(\log s + r \log Q)\right) \) times the cost of an optimal adaptive algorithm.

We also obtain a \( 2r \)-round adaptive algorithm with an better cost guarantee of \( O\left(s^{1/r} \log(sQ)\right) \) times the cost of an optimal adaptive algorithm (see Corollary 6.6). Combined with the conversion to a set-based solution (Theorem A.2) and setting \( r = \log s \), we can then infer:

**Corollary 1.4 (Correlated Items: Set-Based Model).** There is an \( O(\log s) \) round algorithm for scenario submodular cover in the set-based model with cost \( O(\log(sQ)) \) times the optimal (fully) adaptive cost.
The above approximation guarantee is nearly the best possible, even with an arbitrary number of adaptive rounds: there is an $\Omega(\log s)$-factor hardness of approximation \cite{CPR}. Scenario submodular cover generalizes the classic optimal decision tree problem \cite{GG74,HR77,Lov85,KPB99,Das04,AH12,GNR17}. A fully-adaptive $O(\log(sQ))$-approximation for scenario submodular cover was obtained in \cite{GHKL}; see also \cite{NKN20} for a more general result. In terms of rounds-of-adaptivity, an $O(\log(mQ^{\frac{\text{cmax}}{p_{\text{min}}}}))$-approximation in $O(\log m \log(Qm^{\frac{\text{cmax}}{p_{\text{min}}}}))$ set-based rounds follows from \cite{GHKL,EKM}. Here $p_{\text{min}} \leq \frac{1}{s}$ 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 tree.

The results in Theorem 1.1 and Theorem 1.3 are incomparable: while 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 1.5** (Lower Bound). For any integer $r \geq 1$, there is an instance of scenario submodular cover with $Q = 1$ where the cost of any $r$-round adaptive solution is $\Omega(\frac{1}{r^2 \log s} \cdot s^{1/r})$ times the optimal adaptive cost.

This lower bound is information-theoretic and does not depend on computational assumptions, whereas the upper bound of Theorem 1.3 is given by a polynomial algorithm.

Finally, we note that our algorithms are also easy to implement. We tested both algorithms on synthetic and real datasets that validate the practical performance of our algorithms. Specifically, we test our algorithm for the independent case (Theorem 1.1) on instances of stochastic set cover, and our algorithm for the correlated case (Theorem 1.3) on instances of optimal decision tree. For stochastic set cover, we use real-world datasets to generate instances with $\approx 1200$ items. We observe a sharp improvement in performance within a few rounds of adaptivity, and that 6-7 rounds of adaptivity are nearly as good as fully adaptive solutions. For optimal decision tree, we use both real-world data and synthetic data. The real-world data has $\approx 400$ scenarios and the synthetic data has 10,000 scenarios. Again, we find that about 6 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.

### 1.1.1 Techniques

The algorithms for the independent and correlated cases are similar, with some crucial 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 non-adaptive list of all remaining items, and the items are probed in this order until a stopping rule. The PARCA 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 SPARCA rule involves reducing the number of “compatible scenarios” by an $s^{1/r}$ factor in the correlated case.

The analysis for both Theorems 1.1 and 1.3 follows parallel lines at the beginning. For each $i \geq 0$, we relate the “non-completion” probabilities of the algorithm after cost $\alpha \cdot 2^i$ to the optimal adaptive solution after cost $2^i$. The “stretch” factor $\alpha$ corresponds to the approximation ratio, which is different for the independent and correlated cases. In order to relate these non-completion probabilities, we consider the total score $G$ of items selected by the algorithm between cost $\alpha 2^{i-1}$ and $\alpha 2^i$. The crux of the analysis lies in giving lower and upper bounds on the total score $G$; the arguments here are different for the independent and correlated settings.
In the independent case, the score of any item \( X_e \) is an estimate of its relative marginal gain, where we take an expectation over all previous items as well as \( X_e \). We also normalize this gain by the item’s cost. See Equation (1) for the definition. In lower bounding the total score \( G \), we use a variant of a sampling lemma from \[AAK19\] as well as the constant-factor adaptivity gap for submodular maximization \[BSZ19\]. We also need to partition the outcome space (of all previous realizations) into “good” and bad outcomes: conditional on a good outcome, \( \text{OPT} \) 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 \( X_e \) 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 (16). 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 \( \text{OPT} \) (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 \( \text{OPT} \) (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.

1.2 Related Work

A \((1 + \ln Q)\)-approximation algorithm for the basic submodular cover problem was obtained in \[Wol82\]. This is also the best possible (unless P=NP) as the set cover problem is a special case \[Fei98\]. In the past several years, there have been many papers on stochastic variants of submodular cover \[GK11, INvdZ16, DHK16, GHKL16, NKN20\] as this framework captures many different applications.

The stochastic set cover problem was first studied in \[GV06\], which showed that the adaptivity gap lies between \( \Omega(d) \) and \( O(d^2) \) where \( d \) is the number of objects to be covered. Recently, \[AAK19\] improved the upper bound to \( O(d \log d \log(mc_{\text{max}})) \). As a corollary of Theorem 1.1, 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 \[INvdZ16\]. Related results for special cases or with weaker bounds were obtained in \[LPRY08, GK11, GK17, DHK16\]. As mentioned earlier, upper/lower bounds for algorithms with limited rounds-of-adaptivity were obtained in \[AAK19\]. Theorem 1.1 improves on these upper bounds by an \( O(r \cdot \log(mc_{\text{max}})) \) factor. Moreover, our algorithm is greedy-style and much simpler to implement. We bypass computationally expensive steps in prior work such as solving several instances of stochastic submodular maximization. Our analysis (outlined above) is also very different. We use a variant of a key sampling lemma from \[AAK19\], but it is applied in different manner and only affects the analysis. Our high-level approach of lower/upper bounding the total score is similar to the analysis in \[INvdZ16\] for the fully adaptive problem. However, the details are very different because we need to handle issues of \( r \)-round-adaptivity gaps.
The scenario submodular cover problem was introduced in [GHKL16] as a common generalization of several problems including optimal decision tree [GG74, GNR17], equivalence class determination [CLS14] and decision region determination [JCK14]. An $O(\log(sQ))$ fully adaptive algorithm was obtained in [GHKL16]. The same approximation ratio (in a more general setting) was also obtained in [NKN20]. 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 [NKN20], but more work is needed as we have to bound the $r$-round-adaptivity gap.

The framework of “adaptive submodularity”, introduced by [GK17], models correlations in stochastic submodular cover in a different way. Adaptive submodularity is a combined condition on the goal function $f$ (that needs to be covered) and the distribution $\mathcal{D}$ on items. While stochastic submodular cover with independent items satisfies adaptive-submodularity, scenario submodular cover does not. [GK17] gave a fully-adaptive $O(\log^2(mQ))$-approximation algorithm for adaptive-submodular cover (AdSubCov) when all costs are unit. Recently, [EKM19] improved this to an $O(\log(mQ))$-approximation in $O(\log m \log(Qm))$ rounds of adaptivity, still assuming unit costs. When costs are arbitrary, the result in [EKM19] implies an $O(\log(mQc_{\text{max}}))$-approximation in $O(\log m \log(Qmc_{\text{max}}))$ rounds: this result also applies to stochastic submodular cover and can be compared to Corollary 1.2 that we obtain. Although scenario submodular cover is not a special case of AdSubCov, [GHKL16] showed that scenario submodular cover can be re-formulated as AdSubCov with a different goal function that is adaptive-submodular. However, this new goal function has a larger “$Q$ value” of $Q_{\text{pmin}}$. So, when the algorithm from [EKM19] is applied to this new goal function, it only implies an $O(\log(mQ_{\text{pmax}}^{\text{cmax}}))$-approximation in $O(\log m \log(Qmc_{\text{max}}^{\text{pmax}}))$ rounds (this can be compared to Corollary 1.4). To the best of our knowledge, there are no algorithms for AdSubCov using fewer than squared-logarithmic rounds of adaptivity.

The role of adaptivity has been extensively studied for stochastic submodular maximization. A constant adaptivity gap under matroid constraints (on the probed items) was obtained in [AN16]. Later, [GNS17] obtained a constant adaptivity gap for a very large class of “prefix closed” constraints; the constant factor was subsequently improved to 2 which is also the best possible [BSZ19]. We make use of this result in our analysis. More generally, the role of adaptivity has been extensively studied for various stochastic “maximization” problems such as knapsack [DGV08, BGK11], matching [BGL12, BDH20], probing [GN13] and orienteering [GM09, GKNR15, BN15].

Recently, there have been several results examining the role of adaptivity in (deterministic) submodular optimization [BS18a, BBS18, BS18b, BRS19, CQ19]. The motivation here was in parallelizing function queries (that are often expensive). In many settings, there are algorithms using (poly)logarithmic number of rounds that nearly match the best sequential (or fully adaptive) approximation algorithms. While our motivation is similar (in parallelizing the expensive probing steps), the techniques used are completely different.

## 2 Definitions

In the stochastic submodular cover problem, the input is a collection of $m$ random variables (or items) $\mathcal{X} = \{X_1, \ldots, X_m\}$. Each item $X_i$ has a cost $c_i \in \mathbb{R}_+$, and realizes to a random element of groundset $U$. Let the joint distribution of $\mathcal{X}$ be denoted by $\mathcal{D}$. The random variables $X_i$ may or may not be independent; we discuss this issue in more detail below. The realization of item $X_i$ is denoted by $X_i \in U$; this realization is only known when $X_i$ is probed at a cost of $c_i$. Extending this
notation, given a subset of items \( S \subseteq X \), its realization is denoted \( S = \{ X_i : X_i \in S \} \subseteq U \).

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

**Definition 2.1.** For an integer \( r \geq 1 \), an \( r \)-round-adaptive solution 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 \( k \) can depend on the realizations seen in all previous rounds \( 1, \ldots, k-1 \).

Setting \( r = 1 \) gives us a non-adaptive solution, and setting \( r = m \) gives us a (fully) adaptive solution. Having more rounds leads to a smaller objective value, so adaptive solutions have the least objective value. Our performance guarantees are relative to an optimal fully adaptive solution; let \( \text{OPT} \) denote this solution and its cost. The \( r \)-round-adaptivity gap is defined as follows:

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

Setting \( r = 1 \) gives the adaptivity gap.

**Independent and Correlated Distributions** We first study the case where the random variables \( X \) are independent. In keeping with existing literature [INvdZ16, DHK16, AAK19], we refer to this problem simply as the stochastic submodular cover problem. We then consider the case when the random variables \( X \) are correlated with a joint distribution of polynomial size, and refer to it as the scenario submodular cover problem [GHKL16].

## 3 Stochastic Submodular Cover

We now consider the (independent) stochastic submodular cover problem and prove Theorem 1.1.

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 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 \( R \) that realizes to a set \( R \) achieving value \( f(R) > Q(1 - \delta) \). We are interested in a non adaptive algorithm for this problem. Clearly, if \( \delta = 1/Q \), 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 version of the problem. The main result of this section is:

**Theorem 3.1.** There is a non-adaptive 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 non-adaptively; 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 (1)). The score of an item cannot depend on the realizations of previous items on the list (since we are non-adaptive). However, it depends on the
Proof. We proceed by induction on the number of rounds \( r \). Let \( \text{OPT} \) denote the cost of an optimal adaptive solution. The base case is \( r = 1 \), in which case \( \delta = Q^{-1/r} = 1/Q \). By Theorem 3.1, the partial cover algorithm \( \text{ParCA}(\mathcal{X}, f, Q, Q^{-1/r}) \) 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 \( O(Q \log Q) \cdot \text{OPT} \).

We now consider \( r > 1 \) and assume (inductively) that Algorithm 2 finds an \( r - 1 \)-round \( O(Q^{\frac{1}{r-1}} \log Q) \)-approximation algorithm for any instance of stochastic submodular cover. Let \( \delta = Q^{-1/r} \). By Theorem 3.1, the expected cost in round 1 (step 1 in Algorithm 2) is \( O(\frac{1}{r} Q^{1/r} \log Q) \). Let \( Q := Q - f(R) = \hat{f}(U) \) denote the maximal value of the residual submodular function \( \hat{f} = f_R \). Note that \( Q < \delta Q = Q^{(r-1)/r} \), by the definition of the partial covering algorithm. The optimal
solution $\tilde{\text{OPT}}$ conditioned on the variables in $\mathcal{R}$ realizing to $R$ gives a feasible adaptive solution to the residual problem of covering $\hat{f}$; we denote this conditional solution by $\tilde{\text{OPT}}$. We inductively get that the cost of our $r-1$-round solution on $\hat{f}$ is at most

$$O(\hat{Q}^{1/r} \log \hat{Q}) \cdot \tilde{\text{OPT}} \leq O \left( \frac{r-1}{r} Q^{1/r} \log Q \right) \cdot \tilde{\text{OPT}},$$

where we used $Q < Q^{(r-1)/r}$. 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 $O(\frac{r-1}{r} Q^{1/r} \log Q) \cdot \text{OPT}$. Adding to this the cost of the first round, which is $O(\frac{1}{r} Q^{1/r} \log Q)$, completes the proof. 

**Remark:** Assuming that the scores (1) 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 $\text{poly}(m, T)$ 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 §C, 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 (1) at each step. It turns out that $T = \text{poly}(m, c_{\max})$ many samples suffice to estimate these scores. So, the final runtime is $\text{poly}(m, c_{\max})$; note that this does not depend on the number $|U|$ of elements. We note that even the previous algorithms [AAK19, EKM19] have a polynomial dependence on $c_{\max}$ in their runtime. In the following analysis we will assume that the scores (1) are computed exactly; see §C for the sampling details.

### 3.1 Analysis for a Call to ParCA

We now prove Theorem 3.1.

We denote by $\text{OPT}$ an optimal adaptive solution for the covering problem on $f$. Now we analyze the cost incurred by following the non-adaptive strategy (which we call $\text{NA}$): probe variables according to the order given by the list $L$ generated by ParCA, and stop when the realized coverage exceeds $\tau := Q(1 - \delta)$ (see Algorithm 1 for details). We consider the expected cost of this strategy, and relate it to the cost of $\text{OPT}$.

We refer to the cumulative cost incurred (either by $\text{OPT}$ or by $\text{NA}$) until any point in a solution as time elapsing. We say that $\text{OPT}$ is in phase $i$ when it is in the time interval $[2^i, 2^{i+1})$ for $i \geq 0$. Let $\alpha := O \left( \frac{\ln 1/\delta}{\delta} \right)$. We say that $\text{NA}$ 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 interval $[1, \alpha)$. Define

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

Since all costs are integers, $u_0^\ast = 1$. For ease of notation, we also use $\text{OPT}$ and $\text{NA}$ to denote the random cost incurred by $\text{OPT}$ and $\text{NA}$ respectively. The following lemma forms the crux of the analysis.

**Lemma 3.3.** For any phase $i \geq 1$, we have $u_i \leq \frac{u_{i-1}}{4} + \frac{6}{5} u_i^\ast$.

Before we prove this lemma, we use it to prove Theorem 3.1.
Proof of Theorem 3.1

With probability \((u_{i-1} - u_i)\), \(NA\) ends in phase \(i\) with cost at most \(\alpha \cdot 2^i\). As a consequence of this, we have

\[
\mathbb{E}[NA] \leq \alpha \cdot (1 - u_0) + \sum_{i \geq 1} \alpha \cdot 2^i \cdot (u_{i-1} - u_i) = \alpha + \alpha \cdot \sum_{i \geq 0} 2^i u_i. \tag{2}
\]

Similarly, we can bound the cost of the optimal adaptive algorithm as

\[
\mathbb{E}[\text{OPT}] \geq \sum_{i \geq 0} 2^i (u^*_i - u^*_{i+1}) = u^*_0 + \frac{1}{2} \sum_{i \geq 1} 2^i u^*_i = 1 + \frac{1}{2} \sum_{i \geq 1} 2^i u^*_i, \tag{3}
\]

where the final equality uses the fact that \(u^*_0 = 1\). Define \(\Gamma := \sum_{i \geq 0} 2^i u_i\). We have

\[
\Gamma := \sum_{i \geq 0} 2^i u_i \leq u_0 + \frac{1}{4} \sum_{i \geq 1} 2^i \cdot u_{i-1} + \frac{6}{5} \sum_{i \geq 1} 2^i \cdot u^*_i \leq u_0 + \frac{1}{4} \sum_{i \geq 1} 2^i u_{i-1} + \frac{12}{5} \cdot (\mathbb{E}[\text{OPT}] - 1) = u_0 + \frac{1}{2} \left( \sum_{i \geq 0} 2^i u_i \right) + \frac{12}{5} \cdot (\mathbb{E}[\text{OPT}] - 1) \leq \frac{1}{2} \Gamma + \frac{12}{5} \mathbb{E}[\text{OPT}] - 1,
\]

where the first inequality follows from Lemma 3.3, the second inequality from (3), and the last inequality from the fact that \(u_0 \leq 1\). Thus, \(\Gamma \leq \frac{1}{2} \cdot \mathbb{E}[\text{OPT}] - 2\). From (2), we conclude \(\mathbb{E}[NA] \leq \frac{24}{5} \alpha \cdot \mathbb{E}[\text{OPT}]\). Setting \(\alpha = O\left(\frac{\ln(1/\delta)}{\delta}\right)\) completes the proof. \(\square\)

4 Proof of the Key Lemma (Lemma 3.3)

Recall the setting of Lemma 3.3 and fix the phase \(i \geq 1\). Consider the list \(L\) generated by \(\text{PARCA}(\mathcal{X}, f, Q, \delta)\). Let \(NA\) denote both the non-adaptive 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 will assume (without loss of generality) that \(\delta\) is a power-of-two, i.e., \(\delta = 2^{-z}\) for some integer \(z \geq 0\). Indeed, if this is not the case, we can use a power-of-two value \(\delta'\) where \(\frac{z}{2} \leq \delta' \leq \delta\): this only increases the approximation ratio \(O\left(\frac{1}{\delta} \ln \frac{1}{\delta}\right)\) in Theorem 3.1 by a constant factor.

For each time \(t \geq 0\), let \(X_{e(t)}\) denote the item that would be selected at time \(t\). In other words, this is the item which causes the cumulative cost of \(L\) to exceed \(t\) for the first time. We define the total gain as the random variable

\[
G := \sum_{t = \alpha 2^{i-1}}^{\alpha 2^i} \text{score}(X_{e(t)}), \tag{4}
\]

which corresponds to the sum of scores over the time interval \([\alpha \cdot 2^{i-1}, \alpha \cdot 2^i]\). The proof of Lemma 3.3 will be 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.
4.1 A Lower Bound for $G$

Fix some time $t \in [\alpha 2^{i-1}, \alpha 2^i]$ in phase $i$ of our algorithm. Let $S$ be the random variable denoting the set of chosen items (added to list $L$) until some time $t$, and let $S$ denote its realization. We also need the following definitions:

1. For any power-of-two $\theta \in \{\frac{1}{2^j} : 0 \leq j \leq \log Q\}$, we say that a realization $S$ of $S$ belongs to scale $\theta$ iff $0 < \frac{\theta}{2} < \frac{Q-f(S)}{Q} \leq \theta$. We use $\mathcal{E}_\theta$ to denote all outcomes of $S$ that belong to scale $\theta$. We also use $S \sim \mathcal{E}_\theta$ to denote the conditional distribution of $S$ corresponding to $\mathcal{E}_\theta$.

2. For any scale $\theta$, let $r_{i\theta}^* := \mathbb{P}(\text{OPT covers } f \text{ with cost at most } 2^i \text{ and } S \in \mathcal{E}_\theta)$.

3. Scale $\theta$ is called good if $\mathbb{P}(\mathcal{E}_\theta) \geq \frac{1}{6}$ where $\mathbb{P}(\mathcal{E}_\theta) := \mathbb{P}_{S \sim S}(S \in \mathcal{E}_\theta)$.

Lemma 4.1. $\mathcal{NA}$ terminates by time $t$ if and only if the realization of $S$ is in a scale $\theta \leq \delta$.

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

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

and $\mathcal{NA}$ would not terminate before time $t$. On the other hand, if the realization of $S$ 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 $\mathcal{NA}$ would terminate before time $t$.

We now define a function

$$g(T) := \sum_{\theta > \delta} \mathbb{P}(\mathcal{E}_\theta) \cdot \mathbb{E}_{S \sim \mathcal{E}_\theta} \left[ \frac{f(S \cup T) - f(S)}{Q - f(S)} \right] = \sum_{S \sim S: f(S) < \tau} \mathbb{P}(S = S) \cdot \frac{f_S(T)}{Q - f(S)}, \quad \forall T \subseteq U. \quad (5)$$

The second equality is by Lemma 4.1. The function $g$ is monotone and submodular, since $f_S$ is monotone and submodular for each $S \subseteq U$, and $g$ is a non-negative linear combination of such functions. Moreover, for any item $X_e$, we have $\text{score}(X_e) = \frac{1}{c_{X_e}} \cdot \mathbb{E}_{X_e}[g(X_e)]$.

Constrained stochastic submodular maximization. Our analysis makes use of some known results for stochastic submodular maximization. Here, we are given as input a non-negative monotone submodular function $g : 2^U \rightarrow \mathbb{R}_{\geq 0}$, independent stochastic items $\mathcal{X} = \{X_1, \ldots, X_m\}$ such that each $X_i$ realizes to some element of the groundset $U$. There is a cost $c_i$ associated with each item $X_i$, and a budget $B$ on the total cost. The goal is to select $S \subseteq \mathcal{X}$ such that the total cost of $S$ is at most $B$ and it maximizes the expected value $\mathbb{E}_{S \sim \mathcal{S}}[g(S)]$. The adaptivity gap for stochastic submodular maximization is at most 2; see Theorem 1 in [BSZ19].

Lemma 4.2. For any good scale $\theta$, there exists a subset $T_\theta \subseteq \mathcal{X} \setminus S$ with $c(T_\theta) \leq \frac{144}{\theta} \cdot 2^i$ such that:

$$\mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_\theta \sim T} \left[ f(S \cup T_\theta) - f(S) \right] \geq \frac{\theta Q}{6} \cdot \frac{r_{i\theta}^*}{\mathbb{P}(\mathcal{E}_\theta)} \cdot \frac{1}{6}. \quad (6)$$

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

1. Sample realization $S$ of $S$ from $\mathcal{E}_\theta$. 

11
2. Let \( T_k \subseteq \mathcal{X} \setminus \mathcal{S} \) be an optimal non-adaptive solution to the stochastic submodular maximization instance with function \( f_S \) and cost budget \( 2^i \).

3. Set \( T_\theta \leftarrow T_\theta \cup T_k \).

By construction, \( c(T_\theta) \leq \frac{144}{9} \cdot 2^i \). 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 non-adaptive solution obtained in step 2 above for realization \( S \). Define \( w_{i,S}^* \) as the probability that \( \text{OPT} \) covers \( f \) with cost at most \( 2^i \) given that \( S \) realizes to \( S \), i.e.,

\[
 w_{i,S}^* := \mathbb{P}(\text{OPT covers } f \text{ with cost at most } 2^i \mid S = S).
\]

Note that \( \sum_{S \in \mathcal{E}_\theta} w_{i,S}^* \cdot \mathbb{P}(S = S) = r_{i\theta}^* \). Let \( \text{AD} \) denote \( \text{OPT} \) until time \( 2^i \) and restricted to items \( \mathcal{X} \setminus \mathcal{S} \). Note that \( \text{AD} \) is a feasible adaptive solution to the stochastic submodular maximization instance with function \( f_S \); every decision path has total cost at most \( 2^i \). The expected value of \( \text{AD} \) 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 \( \text{AD} \) has value at least \( \frac{\theta Q}{2} \cdot w_{i,S}^* \). Now, using the factor 2 adaptivity gap for stochastic submodular maximization (discussed above), it follows that the non-adaptive solution \( \mathcal{T}_S \) has expected value:

\[
 \mathbb{E}_{T_S \sim \mathcal{T}_S} [f(S \cup T_S) - f(S)] \geq \frac{\theta Q}{4} \cdot w_{i,S}^*, \quad \forall S \in \mathcal{E}_\theta.
\]

Taking expectations,

\[
 \mathbb{E}_S \mathbb{E}_{T_S} [f(S \cup T_S) - f(S) \mid S \in \mathcal{E}_\theta] \geq \sum_{S \in \mathcal{E}_\theta} \frac{\theta Q}{4} \cdot w_{i,S}^* \cdot \mathbb{P}(S = S \mid S \in \mathcal{E}_\theta)
 = \frac{\theta Q}{4} \cdot \sum_{S \in \mathcal{E}_\theta} w_{i,S}^* \cdot \frac{\mathbb{P}(S = S)}{\mathbb{P}(\mathcal{E}_\theta)} = \frac{\theta Q}{4} \cdot \frac{r_{i\theta}^*}{\mathbb{P}(\mathcal{E}_\theta)}.
\]

The left-hand-side of the above relation can be rewritten to give

\[
 \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} [f(S \cup T_S) - f(S)] \geq \frac{\theta Q}{4} \cdot \frac{r_{i\theta}^*}{\mathbb{P}(\mathcal{E}_\theta)}. \tag{7}
\]

For a contradiction to (6), suppose that

\[
 \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_\theta} [f(S \cup T_\theta) - f(S)] < \frac{\theta Q}{6} \cdot \frac{r_{i\theta}^*}{\mathbb{P}(\mathcal{E}_\theta)}. \tag{8}
\]

Subtracting Equation (8) from Equation (7) gives the following:

\[
 \frac{\theta Q}{12} \cdot \frac{r_{i\theta}^*}{\mathbb{P}(\mathcal{E}_\theta)} < \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta} [f(S \cup T_S) - f(S \cup T_\theta)]
 \leq \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta} [f(S \cup T_\theta \cup T_S) - f(S \cup T_\theta)]
 \leq \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T_S} \mathbb{E}_{T_\theta} [f(T_\theta \cup T_S) - f(T_\theta)] \tag{9}
\]

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 \( k \). We write a telescoping
sum as follows
\[ \mathbb{E}_{T_\theta}[f(T_\theta)] = \sum_k \mathbb{E}_{T^{(k)}}[f(T^{(k)}) - f(T^{(k-1)})] \]

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 \( T_k = T_S \) are added.

From Lemma 4.2, let \( T \). By definition of function \( g \), we have
\[ \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim T_S} \mathbb{E}_{T^{(k-1)}} f(T^{(k-1)}) \]
\[ \geq \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim T_S} \mathbb{E}_{T_\theta} f(T_\theta) f(T_S) \]
\[ > \frac{\theta Q}{12} \cdot \frac{r^*_i}{P(E_\theta)}. \]

The first inequality above uses \( T^{(k-1)} \subseteq T \) and submodularity, and the second inequality uses \( Q \). Adding over all iterations \( k \),
\[ \mathbb{E}_{T_\theta}[f(T_\theta)] > \sum_k \frac{\theta Q}{12} \cdot \frac{r^*_i}{P(E_\theta)} = \frac{144}{\theta} \cdot \frac{\theta Q}{12} \cdot \frac{r^*_i}{P(E_\theta)} \geq 2Q, \]

where the last inequality uses the fact that \( \theta \) is a good scale. This is a contradiction since the maximum function value is \( Q \). This completes the proof of (6).

**Lemma 4.3.** For any \( S \subseteq \mathcal{X} \), there exists a subset \( T \subseteq \mathcal{X} \setminus S \) of total cost at most \( \frac{144}{\delta} \cdot 2^i \) with
\[ \mathbb{E}_{T \sim T}[g(T)] \geq \frac{1}{6} \cdot \sum_{\theta > \delta, \text{good}} r^*_i. \]

**Proof.** Let \( B \) denote the set of **good** scales \( \theta \) with \( \theta > \delta \). Note that \( \sum_{\theta \in B} \frac{1}{\theta^2} \leq \frac{1}{\delta^2} \) as the scales are powers of two. From Lemma 4.2, let \( T_\theta \) denote the items satisfying (6) for each scale \( \theta \in B \). Define \( T = \cup_{\theta \in B} T_\theta \). As claimed, the total cost is
\[ c(T) \leq \sum_{\theta \in B} c(T_\theta) \leq 2^i \sum_{\theta \in B} \frac{144}{\theta} \leq \frac{144}{\delta} \cdot 2^i. \]

Next, we bound the expected increase in the value of \( f \). By (6) and \( T \supseteq T_\theta \), it follows that
\[ \mathbb{E}_{S \sim E_\theta} \mathbb{E}_{T \sim T}[f(S \cup T) - f(S)] \geq \frac{\theta Q}{6} \cdot \frac{r^*_i}{P(E_\theta)} \]
for each \( \theta \in B \). Using the fact that \( \frac{\theta Q}{2} < Q - f(S) \leq \theta Q \) for all \( S \in E_\theta \), we get:
\[ \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim T}[f(S \cup T) - f(S)] \geq \frac{1}{6} \cdot \frac{r^*_i}{P(E_\theta)}, \quad \forall \theta \in B. \]

By definition of function \( g \) (see (5)),
\[ \mathbb{E}_{T \sim T}[g(T)] = \sum_{\theta > \delta} P(E_\theta) \cdot \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim T}[f(S \cup T) - f(S)] \]
\[ \geq \sum_{\theta \in B} P(E_\theta) \cdot \mathbb{E}_{S \sim \mathcal{E}_\theta} \mathbb{E}_{T \sim T}[f(S \cup T) - f(S)] \geq \frac{1}{6} \cdot \sum_{\theta \in B} r^*_i, \]

13
which completes the proof of the lemma.

Using Lemma 4.3 and averaging,

$$\max_{X_e \in X \setminus S} \text{score}(X_e) \geq \max_{X_e \in T} \text{score}(X_e) \geq \frac{\sum_{X_e \in \mathcal{T}} \mathbb{E}[g(X_e)]}{c(T)} \geq \frac{\mathbb{E}_{T \sim \mathcal{T}}[\sum_{X_e \in T} g(X_e)]}{c(T)}$$

$$\geq \frac{\delta}{\beta} \cdot 2^i \cdot \sum_{\theta \geq \delta, \text{good}} r^*_\theta,$$

(10)

where $\beta = O(1)$.

**Lemma 4.4.** We have $\sum_{\theta \geq \delta, \text{good}} r^*_\theta \geq u_i - \frac{6u^*_i}{5}$.

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

$$\sum_{\theta \not\in \text{good}} r^*_\theta < \frac{1}{5} \sum_{\theta \not\in \text{good}} \mathbb{P}(E_\theta) \cdot \mathbb{P}(\text{OPT} \geq 2^i | S \sim \mathcal{E}_\theta) \leq \frac{1}{5} \sum_{\theta} \mathbb{P}(\text{OPT} \geq 2^i \text{ and } S \sim \mathcal{E}_\theta) \leq \frac{u^*_i}{5},$$

(11)

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

We now upper bound $\sum_{\theta \leq \delta} r^*_\theta$. By Lemma 4.1, if the realization $S$ is in scale $\theta \leq \delta$ then NA ends before time $t \leq \alpha 2^i$, i.e., it does not go beyond phase $i$. Hence,

$$\sum_{\theta \leq \delta} r^*_\theta \leq \sum_{\theta \leq \delta} \mathbb{P}(S \sim \mathcal{E}_\theta) \leq 1 - u_i.$$

(12)

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

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

where we used (11) and (12).

Combining (10) and Lemma 4.4 and using the greedy choice in step 3 (Algorithm 1), the score at time $t$,

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

We note that this inequality continues to hold (with a larger constant $\beta$) even if we choose an item that only maximizes the score (1) within a constant factor.

Using the above inequality for each time $t$ during phase $i$, we have

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

(13)

We will use this lower bound for $G$ in conjunction with an upper bound, which we prove next.
4.2 An Upper Bound for \( G \)

We now consider the implementation of the non-adaptive list \( L \) and calculate \( G \) as a sum of contributions over the observed decision path. Let \( \Pi \) denote the (random) decision path followed by the non-adaptive strategy \( \mathcal{NA} \): this consists of a prefix of \( L \) along with their realizations. Denote by \( \langle X_1, X_2, \ldots, \rangle \) the sequence of realizations (each in \( U \)) observed on \( \Pi \). So item \( X_j \) is selected between time \( \sum_{\ell=1}^{j-1} c_\ell \) and \( \sum_{j=1}^{j} c_\ell \). Let \( h \) (resp. \( p \)) index the first (resp. last) item in \( \Pi \) (if any) that is selected (even partially) during phase \( i \), i.e., between time \( \alpha 2^{i-1} \) and \( \alpha 2^i \). For each index \( h \leq j \leq p \), let \( t_j \) denote the duration of time that item \( X_j \) 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 \( t_j \leq c_j \).

Define \( G(\Pi) := 0 \) if index \( h \) is undefined (i.e., \( \Pi \) terminates before phase \( i \)), and otherwise:

\[
G(\Pi) := \sum_{j=h}^{p} \frac{t_j \cdot f(\{X_1, \ldots, X_j\}) - f(\{X_1, \ldots, X_{j-1}\})}{Q - f(\{X_1, \ldots, X_{j-1}\})} \leq \sum_{j=h}^{p} \frac{f(\{X_1, \ldots, X_{j}\}) - f(\{X_1, \ldots, X_{j-1}\})}{Q - f(\{X_1, \ldots, X_{j-1}\})} \tag{14}
\]

By the stopping criterion for \( L \), the \( f \) value before the end of \( \Pi \) remains at most \( \tau = Q(1 - \delta) \). So the denominator above, i.e., \( Q - f(\{X_1, \ldots, X_{j-1}\}) \) is at least \( \delta Q \) for all \( j \).

**Lemma 4.5.** For any decision path \( \Pi \), we have \( G(\Pi) \leq 1 + \ln(1/\delta) \).

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

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

The second inequality above uses \( Q - V_{j-1} - y \geq Q - V_{j-1} \geq Q - \tau + 1 = \delta Q + 1 \). The right-hand-side of (14) is then:

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

where we used \( V_p \leq Q \). This completes the proof. \( \square \)

Taking expectations over the various decision paths means \( G = \mathbb{E}_\Pi[G(\Pi)] \), so Lemma 4.5 gives

\[
G \leq (1 + \ln(1/\delta)) \cdot P(\Pi \text{ doesn’t terminate before phase } i) = (1 + \ln(1/\delta)) \cdot u_{i-1}. \tag{15}
\]

4.3 Wrapping Up

To complete the proof of Lemma 3.3, we set \( \alpha := \frac{8\beta}{\delta} (1 + \ln(1/\delta)) = \mathcal{O} \left( \frac{\ln(1/\delta)}{\delta} \right) \), and combine (13) and (15) to get

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

This completes the proof of Lemma 3.3 and hence Theorem 1.1.
5 Scenario Submodular Cover

In this section, we describe an $r$-round adaptive algorithm for the scenario submodular cover problem. As before, we have a collection of $m$ stochastic items $X = \{X_1, \ldots, X_m\}$ with costs $c_i$. In contrast to the independent case, the stochastic items here are correlated, and their joint distribution $D$ is given as input. The goal is to minimize the expected cost of a solution $S \subseteq X$ that realizes to a feasible set (i.e., $f(S) = Q$).

The joint distribution $D$ specifies the (joint) probability that $X$ realizes to any outcome $X \in U^m$. We refer to the realizations $X \in U^m$ that have a non-zero probability of occurrence as scenarios. Let $s = |D|$ denote the number of scenarios in $D$ and $p_\omega$ denotes the probability of each scenario $\omega \in M$. For each scenario $\omega \in M$ and item $X_e$, we denote by $X_e(\omega) \in U$ the realization of $X_e$ in scenario $\omega$. The distribution $D$ 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 $\omega^*$: it only knows the realizations of the probed items (using which it can infer a posterior distribution for $\omega^*$). As stated in §2, our performance guarantee in this case depends on the support-size $s$. We will also show that such a dependence is necessary (even when $Q$ is small).

For any subset $S \subseteq X$ of items, we denote by $S(\omega)$, the realizations for items in $S$ under scenario $\omega$. We say that scenario $\omega$ is compatible with a realization of $S \subseteq X$ if and only if, $X_e$ realizes to $X_e(\omega)$ for all items $X_e \in S$.

5.1 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 will 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 $R$ that realize to a set $R$ such that either (i) the number of compatible scenarios is less than $\delta s$ 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 will use this algorithm with different parameters $\delta$ to solve the $r$-round version of the problem. The main result of this section is:

**Theorem 5.1.** There is a non-adaptive algorithm for the partial cover version of scenario submodular cover with cost $O\left(\frac{1}{\delta}(\ln \frac{1}{\delta} + \log Q)\right)$ times the optimal adaptive cost for the (full) submodular cover.

The algorithm first creates an ordering/list $L$ of the items non-adaptively; 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 (16)). The score of an item depends on an estimate of progress towards (i) eliminating scenarios and (ii) covering function $f$. Before we state this score formally, we need some definitions.

**Definition 5.1.** For any $S \subseteq X$ let $\mathcal{H}(S)$ denote the partition $\{Y_1, \ldots, Y_\ell\}$ of the scenarios $M$ where all scenarios in a part have the same realization for items in $S$. Let $Z := \{Y \in \mathcal{H}(S) : |Y| \geq \delta s\}$ be the set of “large” parts having size at least $\delta s$. 

16
We have \( S = \{X_{e_1}, X_{e_2}\} \), and we partition the set of scenarios \( M \) based on outcomes of \( S \) to get \( \mathcal{H}(S) = \{Y_1, Y_2, Y_3\} \).

(b) We further partition scenarios \( Y_2 \) based on realizations of \( X_{e_3} \). 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 represents \( L_{e_3}(Y_2) \).

Figure 1: Illustrations of Key Definitions

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

**Definition 5.2.** For any \( X_e \in \mathcal{X} \) and subset \( Z \subseteq M \) of scenarios, consider the partition of \( Z \) based on the realization of \( X_e \). Let \( B_e(Z) \subseteq Z \) be the largest cardinality part, and define \( L_e(Z) := Z \setminus B_e(Z) \).

Note that \( L_e(Z) \) is comprised of several parts of the above partition of \( Z \). If the realized scenario \( \omega^* \in L_e(Z) \) and \( X_e \) is selected then, at least half the scenarios in \( Z \) will be eliminated (as being incompatible with \( X_e \)). Figure 1b illustrates these definitions.

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

If \( S \) denotes the previously added items in list \( L \), the score \( (16) \) involves a term for each part drawn 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 \( X_e \) is the sum of these terms (over all parts in \( Z \)) normalized by the cost \( c_e \) of item \( X_e \). In defining the score, we only focus on the “large” parts \( Z \). If the realization of \( S \) corresponds to any other part then the number of compatible scenarios would be less than \( \delta s \) (and the partial-cover algorithm would have terminated).

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 \( \delta s \), 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 non-adaptive partial-cover algorithm.
Algorithm 3 Scenario PARtial Covering Algorithm SPARCA(\(\mathcal{X}, M, f, Q, \delta\))

1: \(S \leftarrow \emptyset\) and list \(L \leftarrow \{\}\).
2: \(\textbf{while} S \neq \mathcal{X} \textbf{ do} \) \text{/* Building the list non-adaptively */}
3: define \(Z\) and \(L_e(Z)\) as in Definitions 5.1 and 5.2
4: select an item \(\mathcal{X}_e \in \mathcal{X} \setminus S\) that maximizes:

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

(16)
5: \(S \leftarrow S \cup \{\mathcal{X}_e\}\) and list \(L \leftarrow L \circ \mathcal{X}_e\)
6: \(R \leftarrow \emptyset, R \leftarrow \emptyset, H \leftarrow M\).
7: \(\textbf{while} |H| \geq \delta|\mathcal{M}|\text{ and } f(R) < Q \textbf{ do} \) \text{/* Probing items on the list */}
8: \(\mathcal{X}_e \leftarrow \text{first r.v. in list } L \text{ not in } R\)
9: \(X_e = v \in U\) be the realization of \(\mathcal{X}_e\).
10: \(R \leftarrow R \cup \{v\}, R \leftarrow R \cup \{\mathcal{X}_e\}\)
11: \(H \leftarrow \{\omega \in H : X_e(\omega) = v\}\)
12: return probed items \(\mathcal{R}, \text{realizations } R\text{ and compatible scenarios } H\).

Given this partial covering algorithm we immediately get an algorithm for the \(r\)-round version of the problem, where 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 \textit{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 \(\hat{f} := f_R\).
3: recursively solve NSC\((r - 1, \mathcal{X} \setminus \mathcal{R}, H, \hat{f})\).

Theorem 5.2. Algorithm 3 is an \(r\)-round adaptive algorithm for scenario submodular cover with cost \(O\left(s^{1/r}(\log s + r \log Q)\right)\) times the optimal adaptive cost, where \(s\) is the number of scenarios.

Proof. We proceed by induction on the number of rounds \(r\). Let \(\text{OPT}\) denote the cost of an optimal fully adaptive solution. The base case is \(r = 1\), in which case \(\delta = s^{-1/r} = \frac{1}{s}\). By Theorem 5.1 the partial cover algorithm SPARCA\((\mathcal{X}, M, f, Q, s^{-1/r})\) obtains a realization \(R\) and compatible scenarios \(H \subseteq M\) where either (i) \(|H| < \delta s = 1\) or (ii) \(f(R) = Q\). Clearly, we cannot have case (i) as there is always at least one compatible scenario. So \(f\) is fully covered. Moreover, the algorithm's expected cost is \(O(s(\log s + \log Q)) \cdot \text{OPT}\), as claimed.

We now consider \(r > 1\) and assume (inductively) that Algorithm 3 finds an \(r - 1\)-round \(O\left(s^{1/r}(\log s + r \log Q)\right)-\)approximation algorithm for any instance of scenario submodular cover. Let \(\delta = s^{-1/r}\). By Theorem 5.1 the expected cost in round 1 (step 1 in Algorithm 3) is \(O(s^{1/r}(\frac{1}{s} \log s + \log Q)) \cdot \text{OPT}\). Let \(\hat{s} = |H|\) denote the number of scenarios in the residual instance. Let \(\hat{Q} = Q - f(R) = \hat{f}(U)\) denote the maximal value of the residual submodular function \(\hat{f} = f_R\). By
definition of the partial covering problem, we have either $\hat{s} < \delta s$ or $\hat{Q} = 0$. If $\hat{Q} = 0$ then our algorithm incurs no further cost and the inductive statement follows. We now assume $\hat{s} < \delta s = s^{r-1}$. The optimal solution $OPT$ conditioned on the scenarios $H$ gives a feasible adaptive solution to the residual instance of covering $\hat{f}$; we denote this conditional solution by $\hat{OPT}$. We inductively get that the cost of our $r - 1$-round solution on $\hat{f}$ is at most

$$O(s^{r-1}(\log \hat{s} + (r - 1) \log \hat{Q})) \cdot \hat{OPT} \leq O(s^{1/r} \left( \frac{r-1}{r} \log s + (r - 1) \log Q \right)) \cdot OPT,$$

where we used $\hat{s} < s^{(r-1)/r}$ and $\hat{Q} \leq Q$. As this holds for every subset of scenarios $H$, we can take expectations over $H$ to get that the (unconditional) expected cost of the last $r - 1$ rounds is $O(s^{1/r} (\frac{r-1}{r} \log s + (r - 1) \log Q)) \cdot OPT$. Adding to this the cost of the first round, which is $O(s^{1/r}(\frac{1}{r} \log s + \log Q)) \cdot OPT$, completes the proof. \hfill \square

### 5.2 Analysis for the partial covering algorithm

We now prove Theorem 5.1. Consider any call to SPARCA. Let $s = |M|$ denote the number of scenarios in the instance. Recall that the goal is to probe items $R \subseteq X$ 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 fully adaptive solution for the covering problem on $f$. Now we analyze the cost incurred by our algorithm’s non-adaptive strategy (which we call $NA$). Note that $NA$ probes items in the order given by the list $L$ (generated by SPARCA) and stops when either condition (i) or (ii) above 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 will become clear in the proof of Lemma 5.3. The following notation is associated with any phase $i \geq 0$:

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

Since 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 $E[NA] \leq O(\beta) \cdot E[OPT]$. As in the independent case, it suffices to show:

**Lemma 5.3.** For any phase $i \geq 1$, we have $u_i \leq \frac{u_{i-1}}{4} + 2u_i^*$.\hfill

Using this lemma, we can immediately prove Theorem 5.1. This part of the analysis is identical to the one presented in §3 for Theorem 3.1.

### 6 Proof of the key lemma for Scenario Submodular Cover

We now prove Lemma 5.3. Let $L$ be the list returned by SPARCA. Recall that $NA$ is the cost incurred by non-adaptively realizing items in the order of $L$ until (i) the number of compatible scenarios $|H| < \delta s$ or (ii) $f$ gets fully covered. For each time $t \geq 0$, let $X_{e(t)}$ denote the item that
would be selected at time $t$. In other words, this is the item which causes the cumulative cost of $L$ to exceed $t$ for the first time. We define the total gain as the random variable

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

which corresponds to the sum of scores over the time interval $[\beta \cdot 2^{i-1}, \beta \cdot 2^i)$. The proof will be completed by upper and lower bounding $G$, 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.

### 6.1 Lower bounding $G$

For the analysis, it will be convenient to view $\text{OPT}$ as a decision tree where nodes correspond to probed items and branches correspond to item realizations. Fix some time $t \in [\beta \cdot 2^{i-1}, \beta \cdot 2^i)$ in phase $i$. Let $\mathcal{S}$ denote the items that have already been added to the list $L$: so $\mathcal{X} \setminus \mathcal{S}$ are the available items. The partition $\mathcal{H}(\mathcal{S})$ of scenarios $M$ and the large parts $Z$ are as in Definition 5.1. For any $Z \in \mathcal{Z}$, we define:

- $S(Z) \subseteq U$ is the realization from items $\mathcal{S}$ compatible with all scenarios in $Z$.
- $Q_Z := Q - f(S(Z))$ is the residual target (after selecting $\mathcal{S}$) under scenarios $Z$.
- residual submodular function $f_Z := f_{S(Z)}$ under scenarios $Z$.
- for any item $\mathcal{X}_e \in \mathcal{X} \setminus \mathcal{S}$, scenarios $L_e(Z) \subseteq Z$ are as in Definition 5.2. This is the complement of the largest part of $Z$ (based on the realization of $\mathcal{X}_e$).
- $\text{OPT}_Z$ is the subtree of $\text{OPT}$ until time $2^i$, restricted to paths traced by scenarios in $Z$; this only includes items that are completely selected by time $2^i$.
- $\text{Stem}_Z$ is the path in $\text{OPT}_Z$ that at each node $\mathcal{X}_e$ follows the branch corresponding to the realization compatible with scenarios $B_e(Z) = Z \setminus L_e(Z)$. We use $\text{Stem}_Z$ to also denote the set of items on this path.

See Figure 2 for an illustration of $\text{OPT}_Z$ and $\text{Stem}_Z$.

For each $Z \in \mathcal{Z}$, we partition scenarios $Z$ based on the realizations on $\text{Stem}_Z$:

- $Z_{\text{good}} = \{\omega \in Z : \omega \text{ compatible at the end of } \text{Stem}_Z \text{ and } f \text{ covered}\}$,
- $Z_{\text{bad}} = \{\omega \in Z : \omega \text{ compatible at the end of } \text{Stem}_Z \text{ and } f \text{ uncovered}\}$,
- $Z_{\text{ok}} = \{\omega \in Z : \omega \in L_e(Z) \text{ for some } \mathcal{X}_e \in \text{Stem}_Z\}$; these are the scenarios that diverge from $\text{Stem}_Z$ at some item $\mathcal{X}_e$.

See Figure 2 for an illustration.

**Definition 6.1.** We define part $Z \in \mathcal{Z}$ as **good** if $p(Z_{\text{good}}) \geq p(Z)/2$, **bad** if $p(Z_{\text{bad}}) \geq p(Z)/2$, or **okay** if $p(Z_{\text{okay}}) \geq p(Z)/2$.

If there are ties, they can be broken arbitrarily.
Figure 2: The tree on the left represents the decision tree $\text{OPT}$ for scenarios $\{s_1, ..., s_8\}$. Each node represents the element chosen at the node, and the set of scenarios compatible until the node (before element at node is chosen). We redraw the decision tree for $Z = \{s_1, s_2, s_3, s_5, s_7\}$ and restrict attention to scenarios in $Z$. The decision tree on the right represents $\text{OPT}\_Z$. Note that both trees are restricted to time $2^i$. Furthermore, we add $\text{Stem}_Z$ (using dotted lines) in $\text{OPT}\_Z$. Note that scenario $s_5$ is a good scenario since it is covered by the end of $\text{Stem}_Z$, and scenarios $s_1, s_2, s_3, s_7$ are all okay scenarios. Thus $Z_{\text{good}} = \{s_5\}$ and $Z_{\text{okay}} = \{s_1, s_2, s_3, s_7\}$. If $s_5$ was not covered by time $2^i$, it would be a bad scenario.

**Lemma 6.1.** Each $Z \in Z$ is either good, bad or okay.

**Proof.** Observe that one of $Z_{\text{good}}$ or $Z_{\text{bad}}$ is always empty and the other equals all scenarios compatible at the end of $\text{Stem}_Z$. Moreover, $Z_{\text{okay}}$ consists of all scenarios that diverge from $\text{Stem}_Z$. So $Z_{\text{good}} \cup Z_{\text{bad}} \cup Z_{\text{okay}} = Z$. It follows that $\max\{p(Z_{\text{good}}), p(Z_{\text{bad}}), p(Z_{\text{okay}})\} \geq p(Z)/2$, which proves the lemma.

For each $Z \in Z$, let $T_Z = \text{Stem}_Z \setminus S \subseteq X \setminus S$ denote the available items on $\text{Stem}_Z$. In the next two lemmas, we lower bound the total score from items in $T_Z$. We consider separately the terms corresponding to function gain and information gain. For any scenario $\omega \in Z$, we use $\text{Stem}_Z(\omega) \subseteq U$ denotes the realization of items $\text{Stem}_Z$ under scenario $\omega$.

**Lemma 6.2.** If $Z$ is good, then $\sum_{\omega \in Z} p_\omega \cdot f_Z(T_Z(\omega)) > p(Z)/2$.

**Proof.** Consider any scenario $\omega \in Z_{\text{good}}$. If $\omega$ is realized then $f$ is covered by the realization $\text{Stem}_Z(\omega)$ of $\text{Stem}_Z$. Thus, $f(\text{Stem}_Z(\omega)) = Q$. We have

$$f_Z(T_Z(\omega)) = f(T_Z(\omega) \cup S(Z)) - f(S(Z)) = f(\text{Stem}_Z(\omega) \cup S(Z)) - f(S(Z)) = Q - f(S(Z)) = Q_Z,$$

where the second equality uses that $T_Z(\omega) = \text{Stem}_Z(\omega) \setminus S(Z)$. As $Z$ is good, $p(Z_{\text{good}}) \geq p(Z)/2$, and the lemma follows by summing over $\omega \in Z_{\text{good}}$.

**Lemma 6.3.** If $Z$ is okay, then $p\left(\bigcup_{\chi \in T_Z} L_\chi(Z)\right) \geq p(Z)/2$.
Finally, we have $\cup_{\mathcal{X}_e \in \mathcal{T}_Z} L_e(Z) \supseteq \left( \cup_{\mathcal{X}_e \in \mathcal{T}_Z} L_e(Z) \right) \setminus \left( \cup_{\mathcal{X}_e \in \mathcal{S}} L_e(Z) \right) = \cup_{\mathcal{X}_e \in \mathcal{T}_Z} L_e(Z) = Z_{\text{okay}}$. 

Hence, $p\left( \cup_{\mathcal{X}_e \in \mathcal{T}_Z} L_e(Z) \right) \geq \frac{p(Z)}{2}$, which completes the proof. \hfill \square

We now prove a bound on the overall probability of parts that are either good or okay.

**Lemma 6.4.** We have $\sum_{Z: \text{okay or good}} p(Z) \geq (u_i - 2u_i^*)$.

**Proof.** We first show that $\sum_{Z: \text{bad}} p(Z) \leq 2 \cdot u_i^*$. For any $Z \in \mathcal{Z}$, if scenario $\omega \in Z_{\text{bad}}$ is realized then we know that $\text{OPT}$ does not cover $f$ at the end of $\text{stem}_Z$, which implies that $\text{OPT}$ costs at least $2^t$. Thus $u_i^* \geq \sum_{Z \in \mathcal{Z}} \omega \in Z_{\text{bad}} p_\omega$. Moreover, $p(Z_{\text{bad}}) \geq p(Z)/2$ if $Z$ is bad. So,

$$u_i^* \geq \sum_{Z \in \mathcal{Z}} \sum_{\omega \in Z_{\text{bad}}} p_\omega \geq \sum_{Z: \text{bad}} \sum_{\omega \in Z_{\text{bad}}} p_\omega \geq \sum_{Z: \text{bad}} p(Z)/2. \quad (17)$$

Thus, $\sum_{Z: \text{bad}} p(Z) \leq 2u_i^*$ as claimed.

Consider any part $Y \in \mathcal{H}(\mathcal{S}) \setminus \mathcal{Z}$; recall Definition 5.1. Note that $|Y| < \delta s$. So, if $\mathcal{S}$ realizes to the set $S(Y)$ then $\text{NA}$ would terminate by time $t \leq \beta 2^t$, which implies that $\text{NA}$ does not go beyond phase $i$. Thus,

$$\sum_{Y \in \mathcal{H}(\mathcal{S}) \setminus \mathcal{Z}} p(Y) \leq \Pr[\text{NA terminates by phase } i] = 1 - u_i. \quad (18)$$

Finally, we have

$$\sum_{Z: \text{okay or good}} p(Z) = 1 - \sum_{Z: \text{bad}} p(Z) - \sum_{Y \in \mathcal{H}(\mathcal{S}) \setminus \mathcal{Z}} p(Y) \geq 1 - 2u_i^* - (1 - u_i) = u_i - 2u_i^*,$$ 

where we used (17) and (18). This completes the proof. \hfill \square

We are now ready to prove the lower bound on $\text{score}(\mathcal{X}_{e(t)})$.

**Lemma 6.5.** For any time $t$ in phase $i$, we have $\text{score}(\mathcal{X}_{e(t)}) \geq \delta \cdot \frac{1}{2^t} \cdot (u_i - 2u_i^*)$.

**Proof.** Recall the subsets $\mathcal{T}_Z \subseteq \mathcal{X} \setminus \mathcal{S}$ for $Z \in \mathcal{Z}$. Let $\mathcal{T} = \bigsqcup_{Z \in \mathcal{Z}} \mathcal{T}_Z$ denote the multiset that contains each item as many times as it occurs. By definition of $\text{stem}_Z$, the cost $c(\mathcal{T}_Z) \leq 2^t$ for each $Z$. So, $c(\mathcal{T}) = \sum_{Z \in \mathcal{Z}} c(\mathcal{T}_Z) \leq |\mathcal{Z}| \cdot 2^t \leq \frac{s}{\delta} \cdot 2^t$. Moreover, $\mathcal{T} \subseteq \mathcal{X} \setminus \mathcal{S}$: so each item in $\mathcal{T}$ is available to be added to $L$ at time $t$. We can lower bound $\text{score}(\mathcal{X}_{e(t)})$ by averaging over the multiset $\mathcal{T}$:

$$\text{score}(\mathcal{X}_{e(t)}) \geq \max_{\mathcal{X}_e \in \mathcal{T}} \text{score}(\mathcal{X}_e) \geq \frac{1}{c(\mathcal{T})} \cdot \sum_{Z \in \mathcal{Z}} \sum_{X_e \in \mathcal{T}_Z} \sum_{Y \in \mathcal{Z}} \left( p(L_e(Y)) + \sum_{\omega \in Y} p_\omega \cdot \frac{f_Y(X_e(\omega))}{Q_Y} \right)$$

$$\geq \frac{1}{c(\mathcal{T})} \cdot \sum_{Z \in \mathcal{Z}} \sum_{X_e \in \mathcal{T}_Z} \left( p(L_e(Z)) + \sum_{\omega \in Z} p_\omega \cdot \frac{f_Z(X_e(\omega))}{Q_Z} \right). \quad (19)$$

We now consider the two terms above (information and function gain) separately.
Bounding the information gain. We only consider \textit{okay} sets \( Z \).

\[
\sum_{Z \in \mathcal{Z}} \sum_{\mathcal{X}_c \in \mathcal{T}_Z} p(L_e(Z)) \geq \sum_{Z \in \mathcal{Z}} p \left( \bigcup_{\mathcal{X}_c \in \mathcal{T}_Z} L_e(Z) \right) \geq \sum_{Z : \text{okay}} \frac{p(Z)}{2},
\]

where the first inequality is by submodularity of the \( p \)-weighted coverage function and the second inequality is by Lemma \ref{lem:submodularity_coverage}.

Bounding the function gain. Consider any \textit{good} set \( Z \).

\[
\sum_{\mathcal{X}_c \in \mathcal{T}_Z} \sum_{\omega \in \mathcal{Z}} p_\omega \cdot \frac{f_Z(X_\omega(\omega))}{Q_\omega} = \sum_{\omega \in \mathcal{Z}} \sum_{\mathcal{X}_c \in \mathcal{T}_Z} p_\omega \cdot \frac{f_Z(X_\omega(\omega))}{Q_\omega} \geq \sum_{\omega \in \mathcal{Z}} p_\omega \cdot \frac{f_Z(T_\omega(\omega))}{Q_\omega} \geq \frac{p(Z)}{2},
\]

where the first inequality is by submodularity of \( f_Z \) and the second inequality is by Lemma \ref{lem:submodularity_fZ}.

Adding over all good sets \( Z \),

\[
\sum_{Z \in \mathcal{Z}} \sum_{\mathcal{X}_c \in \mathcal{T}_Z} \sum_{\omega \in \mathcal{Z}} p_\omega \cdot \frac{f_Z(X_\omega(\omega))}{Q_\omega} \geq \sum_{Z : \text{good}} \frac{p(Z)}{2}.
\]

Combining the two bounds above,

\[
\sum_{Z \in \mathcal{Z}} \sum_{\mathcal{X}_c \in \mathcal{T}_Z} \left( p(L_e(Z)) + \sum_{\omega \in \mathcal{Z}} p_\omega \cdot \frac{f_Z(X_\omega(\omega))}{Q_\omega} \right) \geq \sum_{Z : \text{okay or good}} \frac{p(Z)}{2} \geq \frac{u_i - 2u^*_i}{2},
\]

where the last inequality is by Lemma \ref{lem:upper_bound}. Combined with (19) and \( c(T) \leq \frac{1}{8} \cdot 2^i \), this completes the proof.

As Lemma \ref{lem:upper_bound} holds for each time \( t \) in phase \( i \),

\[
G = \sum_{t=\beta^{2i-1}}^{\beta^{2i}} \text{score}(\mathcal{X}_c(t)) \geq \beta^{2i-1} \cdot \delta \cdot \frac{1}{2^i+1} \cdot (u_i - 2u^*_i) \geq \frac{\beta \delta}{4} \cdot (u_i - 2u^*_i) \quad (20)
\]

6.2 Upper bounding \( G \)

We now consider the implementation of the non-adaptive list \( L \) and calculate \( G \) as a sum of contributions over the decision path in the non-adaptive solution \( \mathcal{NA} \). Let \( \omega \in \mathcal{M} \) denote the realized scenario, and consider the decision path under scenario \( \omega \). Let \( \langle X_1, X_2, \ldots \rangle \) be the sequence of realizations (each in \( U \)) of items in \( L \) under scenario \( \omega \). So item \( X_j \) is selected between time \( \sum_{t=1}^{j-1} c_t \) and \( \sum_{t=1}^{j} c_t \). Let \( h \) (resp. \( p \)) index the first (resp. last) item (if any) that is selected (even partially) during phase \( i \), i.e., between time \( \beta^{2i-1} \) and \( \beta^{2i} \). For each index \( h \leq j \leq p \), let \( t_j \) denote the duration of time that item \( X_j \) is selected during in phase \( i \); so \( t_j \) is the width of interval \( [\sum_{t=1}^{j-1} c_t, \sum_{t=1}^{j} c_t] \cap [\beta^{2i-1}, \beta^{2i}] \). Furthermore, for each index \( j \), let \( Z_j \subseteq \mathcal{M} \) be the set of scenarios compatible with outcomes \( \langle X_1, \cdots, X_{j-1} \rangle \). Note that \( M = Z_1 \supseteq Z_2 \supseteq \cdots \supseteq Z_p \supseteq Z_{p+1} \). Define
Using the fact that
\[ G(\omega) := 0 \] if index \( h \) is undefined (i.e., NA ends before phase \( i \) under scenario \( \omega \)) and otherwise:

\[ G(\omega) := \sum_{j=h}^{p} \frac{t_j}{e_j} \left( 1[\omega \in L_j(Z_j)] + \frac{f(\{X_1,\ldots,X_j\}) - f(\{X_1,\ldots,X_{j-1}\})}{Q - f(\{X_1,\ldots,X_{j-1}\})} \right) \]

\[ \leq \sum_{j=h}^{p} \left( 1[\omega \in L_j(Z_j)] + \frac{f(\{X_1,\ldots,X_j\}) - f(\{X_1,\ldots,X_{j-1}\})}{Q - f(\{X_1,\ldots,X_{j-1}\})} \right) \] \hspace{1cm} (21)

By the stopping criterion for \( L \), before the end of NA we must have (i) the number of compatible scenarios remains at least \( \delta s \) and (ii) the \( f \) value remains less than \( Q \). In particular, we must have \( |Z_p| \geq \delta s \).

We analyze the two quantities in the above expression separately. We begin by analyzing \( \sum_{j=h}^{p} 1[\omega \in L_j(Z_j)] \). Fix any \( h \leq j \leq p \). If the realized scenario \( \omega \in L_j(Z_j) \), then the number of compatible scenarios drops by a factor of at least 2 upon observing \( X_j \); that is, \( |Z_{j+1}| \leq \frac{1}{2}|Z_j| \).

Using the fact that \( |Z_1| = |M| = s \) and \( |Z_p| \geq \delta s \), it follows that

\[ \sum_{j=h}^{p} 1[\omega \in L_j(Z_j)] \leq \log_2 \frac{s}{\delta s} = \log_2(1/\delta). \]

Next, we analyze the second term in (21). Using the fact that \( f \) is integral, monotone and takes values in the range \([0, Q]\), we have:

\[ \sum_{j=h}^{p} \frac{f(\{X_1,\ldots,X_j\}) - f(\{X_1,\ldots,X_{j-1}\})}{Q - f(\{X_1,\ldots,X_{j-1}\})} \leq \sum_{\ell=1}^{Q} \frac{1}{\ell} \leq \ln Q. \]

Thus, we have \( G(\omega) \leq \log \frac{1}{\delta} + \ln Q \leq \log(Q/\delta) \).

Note that \( G = \mathbb{E}_{\omega \sim M} [G(\omega)] \). It now follows that

\[ G \leq \log(Q/\delta) \cdot \mathbb{P}(\text{NA doesn’t terminate before phase } i) = \log(Q/\delta) \cdot u_{i-1}. \] \hspace{1cm} (22)

### 6.3 Completing proof of Lemma 5.3

Using (20), (22) and setting \( \beta = \frac{16}{\delta} \log(Q/\delta) \), we get

\[ \frac{16 \log(Q/\delta)}{4} \cdot (u_i - 2u^*_i) \leq \log(Q/\delta) \cdot u_{i-1} \]

which on rearranging gives \( u_i \leq \frac{u_{i-1}}{4} + 2u^*_i \), as desired.

This completes the proof of Theorem 1.3

### 6.4 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 \( X \), 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 \( R \) that realize to set \( R \subseteq U \).
such that either (i) the number of compatible scenarios is less than $\delta s$ 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 6.6.** There is a non-adaptive algorithm for SSPC with cost $O\left(\frac{1}{s}(\ln \frac{1}{s} + \ln \frac{1}{\varepsilon})\right)$ times the optimal adaptive cost for the (full) submodular cover.

**Proof.** The algorithm is nearly identical to SPARCA (Algorithm 3). In Definition 5.1 we change

$$Z := \{Y \in \mathcal{H}(S) : |Y| \geq \delta s \text{ and } f(S(Y)) \leq Q(1 - \varepsilon)\}.$$  

In other words $Z$ contains parts having size at least $\delta s$ and for which the realized function value is at most the target $Q(1 - \varepsilon)$. Note that scenarios in $Z$ correspond to those under which the SSPC stopping rule does not already apply. After this, we use the same steps to produce list $L$ in Algorithm 3. The only difference is in the stopping rule (when probing items from $L$) which reflects the definition of SSPC. In particular, the condition in the while-loop (step 7 of Algorithm 3) is now replaced by:

$$\text{While } |H| \geq \delta |M| \text{ and } f(R) \leq Q(1 - \varepsilon).$$

The analysis is also nearly identical: we prove Lemma 5.3 by lower/upper bounding the total gain $G$. The lower bound (20) remains the same. For the upper bound, the analysis for the first term in (21) remains the same, but its second term is now bounded as:

$$\sum_{j=h}^{p} \frac{V_j - V_{j-1}}{Q - V_{j-1}} \leq 1 + \sum_{j=h}^{p-1} \frac{V_j - V_{j-1}}{Q - V_{j-1}} \leq 1 + \sum_{\ell=Q}^{Q} \frac{1}{\ell} \leq 1 + \ln \frac{1}{\varepsilon},$$

where $V_j := f(\{X_1, ..., X_j\})$. The second inequality uses $V_1 \leq V_2 \leq \cdots V_{p-1} \leq Q(1 - \varepsilon)$. This implies (just as before) that

$$G \leq \left(1 + \log \frac{1}{\delta} + \log \frac{1}{\varepsilon}\right) \cdot u_{i-1}.$$

Finally, choosing $\beta = \frac{16}{s} (1 + \log \frac{1}{\delta} + \log \frac{1}{\varepsilon})$ and simplifying, we get $u_i \leq \frac{u_{i-1}}{4} + 2u_i^*$ (Lemma 5.3). This completes the proof.

Next, we show how SSPC can be used iteratively to obtain a $2r$-round algorithm.

**Theorem 6.7.** There is a $2r$-round adaptive algorithm for scenario submodular cover with cost $O\left(s^{1/r} \log(sQ)\right)$ times the optimal adaptive cost, where $s$ is the number of scenarios.

**Proof.** We use SSPC with $\delta = s^{-1/r}$ and $\varepsilon = Q^{-1/r}$ repeatedly for $2r$ adaptive rounds. Let $\rho = O\left(s^{1/r} \log(sQ)\right)$ denote the approximation ratio from Corollary 6.6 for these $\delta, \varepsilon$ values.

The state at any round consists of previously probed items and their realizations. For the analysis, we view the iterative algorithm as a $2r$ depth tree, where the nodes at depth $i$ are the states in round $i$ and the branches out of each node represent the observed realizations in that round. For any $i \in [2r]$, let $\Omega_i$ denote all the states in round $i$: note that these form a partition of the outcome space.

For any state $\omega \in \Omega_i$, let $\text{OPT}(\omega)$ denote the expected cost of the optimal adaptive policy conditioned on the realizations in $\omega$. By Corollary 6.6 conditioned on $\omega$, the expected cost in
round $i$ is at most $\rho \cdot \OPT(\omega)$. Hence, the (unconditional) expected cost in round $i$ is at most $\rho \sum_{\omega \in \Omega_i} \Pr[\omega] \cdot \OPT(\omega) = \rho \cdot \OPT$ where we used that $\Omega_i$ is a partition of all outcomes. So the expected cost of the $2r$-round algorithm is at most $2r \cdot \rho \cdot \OPT$ as claimed.

It remains to show that $f$ is fully covered after $2r$ rounds. Note that after each round, we have one of the following (i) the number of compatible scenarios drops by factor $s^{1/r}$, or (ii) the residual target drops by factor $Q^{1/r}$. Clearly, case (i) can happen at most $r$ times (as there is always some compatible scenario). So case (ii) occurs at least $r$ times, which implies that the residual target is less than 1, i.e., $f$ is fully covered.

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 \[A.2\]), this proves Corollary \[1.4\].

7 Lower Bound for Scenario Submodular Cover

Recall that we obtained an $r$-round $O(s^{1/r} \cdot (\log s + r \log Q))$-approximation algorithm for scenario submodular cover. It is natural to ask if the dependence on the support-size $s$ is necessary, given that the bound for the independent case only depends on $Q$. The main result of this section is Theorem \[1.5\] which is a nearly matching lower bound on the $r$-round-adaptivity gap. In particular, it shows that the $s^{1/r}$ dependence is necessary even when $Q = 1$.

7.1 Hard Instances for Scenario Submodular Cover

As a warm-up we first consider the case when $r = 1$. We design an instance to prove a 1-round adaptivity gap. The instances that we use to prove the adaptivity gap for $r > 1$ will build on this. The instance has groundset $U = \{0,1,\star,\bot\}$ and function $f(S) = |S \cap \{\star\}|$. Clearly, $f$ is a $0-1$ valued monotone submodular function; and $Q = 1$. Let $\ell$ be an integer parameter and $N = 2^\ell$. We have two types of items: $\mathcal{Y}_0, ..., \mathcal{Y}_{\ell-1}$ each of which realizes to either 0 or 1, and $N$ items $\mathcal{Z}_0, ..., \mathcal{Z}_{N-1}$ each of which realizes a symbol $\star$ or $\bot$. All items have cost 1. The joint distribution $\mathcal{D}$ of these items has $s = N$ scenarios, each of uniform probability. For each $B \in \{0,1,\cdots,N-1\}$, scenario $B$ has the following realizations (below, $b_0b_1\cdots b_{\ell-1}$ is the binary representation of $B$).

\[
Y_i(B) = b_i, \forall i = 0, \cdots, \ell - 1 \quad \text{and} \quad Z_j(B) = \bot, \forall j \in \{0, \cdots, N-1\} \setminus \{B\} \text{ and } Z_B(B) = \star.
\]

We say that item $Z_k$ is active if, and only if, $Z_k$ realizes to $\star$. Note that there is a unique active item in each scenario, and this item needs to be probed to cover $f$.

An adaptive solution is as follows. First, probe items $\mathcal{Y}_0, ..., \mathcal{Y}_{\ell-1}$, and based on their realizations calculate the number $B$ represented by $Y_0, \cdots, Y_{\ell-1}$. Then, probe item $Z_B$ (the active one) to cover $f$. This policy has cost $\ell + 1$. On the other hand, since each $Z_k$ is equally likely to be active, any non-adaptive policy needs to pick $\Omega(2^\ell)$ items, in expectation, to cover $f$. In conclusion, this proves an $\Omega(2^\ell / \ell) = \Omega(s / \log s)$ lower bound on the 1-round adaptivity gap for scenario submodular cover.

The $r$-round instance. This is based on recursively applying the above instance structure. Recall that $\ell$ is some parameter and $N = 2^\ell$. Moreover, the realizations of $\mathcal{Y}$-items above point to the unique $\mathcal{Z}$-item that is active. To extend this idea, we consider an $N$-ary tree $T$ of depth $r$. For each internal node, we index its $N$ children by $\{0, \cdots, N-1\}$ in an arbitrary order. We define a collection of items, each of unit cost, using $T$ as follows.
• For each depth \(i = 0, \ldots, r-1\) and node \(v \in T\) at depth \(i\), there are \(\ell\) items \(Y_0(v), \ldots, Y_{\ell-1}(v)\), each of which realizes 0 or 1.

• For each leaf node \(w\) of \(T\) (i.e., at depth \(r\)), there is an item \(Z_w\) which realizes to one of the symbols \(\star\) or \(\perp\).

For any non-leaf node \(v\), we use the shorthand \(Y(v) = \langle Y_0(v), \ldots, Y_{\ell-1}(v) \rangle\) to denote all its \(Y\)-items. The function \(f\) is as defined earlier: \(f(S) = |S \cap \{\star\}|\). So \(f(S) = 1\) if \(\star \in S\) (else \(f(S) = 0\)). The distribution \(D\) involves \(s = 2^{r \ell}\) scenarios with uniform probability. Each scenario is associated with a unique leaf node (i.e. \(Z\)-item); note that the number of leaves is exactly \(s\).

For each leaf \(w\), we define its corresponding scenario as follows. Let \(v_0, \ldots, v_{r-1}, v_r = w\) denote the path from the root of \(T\) to \(w\). For each depth \(i = 0, \ldots, r-1\), let node \(v_{i+1}\) be the \(B_i^r\)th child of node \(v_i\). Then, scenario \(w\)'s realizations are:

• The realization \(Y_0(v_i), \ldots, Y_{\ell-1}(v_i)\) of \(Y(v_i)\) equals the binary representation of \(B_i\), for each depth \(0 \leq i \leq r-1\). The realization of all other \(Y\)-items is 0.

• The realization \(Z_w = \star\) and \(Z_u = \perp\) for all leaf \(u \neq w\).

In this way, the realizations at the depth \(i\) node \(v_i\) point to the “correct” depth \(i + 1\) node \(v_{i+1}\) that should be probed next. This completes the description of the instance.

**Adaptive solution.** We now describe a good \((r+1)\)-round adaptive solution \(OPT\) of cost \(r \ell + 1\). It is a natural top-down strategy. It starts by probing the \(\ell\) items \(Y(v_0)\) at the root \(v_0\), and based on their realization calculates the number \(B_0\) represented by \(Y_0(v_0), \ldots, Y_{\ell-1}(v_0)\); then it probes the \(Y\)-items at the \(B_0^r\)th child of \(v_0\); and so on. After \(r\) rounds, it is able to discern the unique active item \(Z_w\) and probes it (in the final round) to cover \(f\).

### 7.2 Lower bound proof

Consider any \(r\)-round solution \(A\) for the above instance. By Yao’s minimax principle, we can assume that \(A\) is deterministic. We denote by random variables \(L_0, \ldots, L_{r-1}\), the ordered lists used by \(A\) in each of its \(r\) rounds. Moreover, \(S_0, \ldots, S_{r-1}\) denote the sets of items probed by \(A\) in each round; note that each \(S_i\) is a prefix of \(L_i\). Also, \(L_i\) is deterministic conditioned on the realizations of \(S_0, \ldots, S_i\). The following proof strategy is similar to that in \([AAK19]\) for the independent setting; however, note that the instance itself is very different.

We use \(w^*\) to denote the realized scenario (leaf node) and \(\cap_0, \ldots, \cap_r = w^*\) to denote the (random) nodes of \(T\) on the path from the root to \(w^*\). For any depth \(0 \leq i \leq r\), random variable \(T_i\) denotes the set of all items at all descendant nodes of \(\cap_i\) (including itself). Note that conditioned on \(w^*\), all other random variables \((L_i, S_i, \cap_i, T_i\) etc.) are deterministic. For this reason, each random event can be seen as a subset of the scenarios.

Roughly speaking, we show that one of the following events occurs: (i) either the size of probed set \(S_i\) is large for some round \(i\), or (ii) if sets \(S_0, \ldots, S_{i-1}\) are small, then with high probability \(S_i\) does not contain any item from \(T_{i+1}\). In the first case, we are immediately done. If the second case keeps occurring then the problem eventually reduces to an instance with \(r = 1\) rounds (for which we already know a lower bound). For a contradiction, assume:

**Assumption 7.1.** The expected cost of \(A\) is at most \(\frac{N}{(4r)^r}\).
For each $i = 0, \ldots, r$, define the following events.

- $\mathcal{F}_i$ is the event that
  \[(S_0 \cup \cdots S_{i-1}) \cap T_i = \emptyset,\]
  i.e., no item from $T_i$ is probed before round $i$ (i.e. in rounds $0, \cdots, i-1$).

- $\mathcal{E}_i$ is the event that
  \[|S_j| \leq \frac{N}{4r}, \text{ for all } j = 0, \cdots, i-1,\]
  i.e., at most $\frac{N}{4r}$ items are probed in each round $0, \cdots, i-1$.

Note that $|\mathcal{F}_0| = |\mathcal{E}_0| = N^r$ as they include all scenarios.

**Lemma 7.1.** For any $i \in \{0, \cdots, r-1\}$, we have $|\mathcal{E}_i \cap \mathcal{F}_{i+1}| \geq |\mathcal{E}_i \cap \mathcal{F}_i| - \frac{N^r}{2r}$.

**Proof.** Note that $\mathbb{E}[|S_i|] \leq N/(4r)^2$ by Assumption 7.1. So,
\[
\Pr[\mathcal{E}_i \setminus \mathcal{E}_{i+1}] \leq \Pr\left[|S_i| > \frac{N}{4r}\right] \leq \frac{1}{4r},
\]
where the first inequality is by definition of events $\mathcal{E}_i$s and the second is Markov’s inequality. As all scenarios have uniform probability, $|\mathcal{E}_i \setminus \mathcal{E}_{i+1}| = N^r \cdot \Pr[\mathcal{E}_i \setminus \mathcal{E}_{i+1}] \leq N^r/(4r)$.

Let $\mathcal{G} := \mathcal{E}_{i+1} \cap \mathcal{F}_i$. Note that
\[
|\mathcal{G}| \geq |\mathcal{E}_i \cap \mathcal{F}_i| - |\mathcal{E}_i \setminus \mathcal{E}_{i+1}| \geq |\mathcal{E}_i \cap \mathcal{F}_i| - N^r/(4r). \tag{23}
\]

We now partition scenarios in $\mathcal{G}$ according to the depth $i$ nodes in tree $T$. For any node $u$ at depth $i$, let $\mathcal{G}(u) \subseteq \mathcal{G}$ denote the scenarios that appear in the subtree below $u$. Note $|\mathcal{G}| = \sum_u |\mathcal{G}(u)|$.

Fix any depth $i$ node $u$. Note that under any scenario of $\mathcal{G}(u)$, set $T_i$ consists of all items $T(u)$ in the subtree below $u$. We first claim that

The list $\mathcal{L}_i$ in round $i$ is the same for all scenarios $w \in \mathcal{G}(u)$.

As $w \in \mathcal{F}_i$, we know that no item in $T_i = T(u)$ is probed in rounds $0, \cdots, i-1$ (under scenario $w$). Moreover, by construction of our instance, each item $I \in \mathcal{X} \setminus T_i = \mathcal{X} \setminus T(u)$ has the same realization $I$ under all scenarios in $\mathcal{G}(u)$. So the realizations seen before round $i$ are identical under all scenarios in $\mathcal{G}(u)$, which implies that the list $\mathcal{L}_i$ in round $i$ is also the same (for all scenarios in $\mathcal{G}(u)$).

Let $\mathcal{P}_u$ denote the first $N/(4r)$ items in list $\mathcal{L}_i$ under scenarios $\mathcal{G}(u)$. Note that $\mathcal{S}_i \subseteq \mathcal{P}_i$ under all scenarios in $\mathcal{G}(u) \subseteq \mathcal{E}_{i+1}$: this follows from the definition of $\mathcal{E}_{i+1}$.

Let $\mathcal{G}'(u) = \mathcal{G}(u) \cap \mathcal{F}_{i+1}$. We now claim that
\[
|\mathcal{G}(u) \setminus \mathcal{G}'(u)| \leq N^{r-i}/(4r). \tag{24}
\]

To see this, let $N_u$ denote all children nodes of $u$ and $C_u \subseteq N_u$ denote those children that do not contain any $\mathcal{P}_u$-item in in their subtree. As $u$ has $N$ children, $|C_u| \geq N - |\mathcal{P}_u|$. Now consider any depth $i+1$ node $v \in C_u$ and scenario $w \in \mathcal{G}(u)$ that lies in the subtree below $v$. Note that, under scenario $w$, $T_{i+1} = T(v)$ the items in subtree below $v$. We now claim that $w \in \mathcal{F}_{i+1}$ (and hence $w \in \mathcal{G}'(u)$) because:
• no item probed in rounds before $i$ is in $\mathcal{T}_i \supseteq \mathcal{T}_{i+1}$ as $w \in \mathcal{G}(u) \subseteq \mathcal{F}_i$.

• no item probed in round $i$ is in $\mathcal{T}_{i+1} = \mathcal{T}(v)$; recall that $\mathcal{S}_i \subseteq \mathcal{P}(u)$ (as $w \in \mathcal{G}(u) \subseteq \mathcal{E}_{i+1}$) and $\mathcal{P}(u) \cap \mathcal{T}(v) = \emptyset$.

It follows that every scenario in $\mathcal{G}(u) \setminus \mathcal{G}'(u)$ appears in a subtree below a node of $N_u \setminus C_u$. Hence, $|\mathcal{G}(u) \setminus \mathcal{G}'(u)| \leq |N_u \setminus C_u| \cdot N^{r-i-1} \leq |\mathcal{P}(u)| \cdot N^{r-i-1}$ as the number of scenarios under each such subtree is $N^{r-i-1}$. Equation (24) follows by using $|\mathcal{P}(u)| \leq N/(4r)$.

Finally, let $\mathcal{G}' = \cup_{u} \mathcal{G}'(u)$ over all depth-$i$ nodes $u$. Note that $\mathcal{G}' = \mathcal{G} \cap \mathcal{F}_{i+1} \subseteq \mathcal{E}_{i+1} \cap \mathcal{F}_{i+1}$. We have

$$|\mathcal{G}'| = \sum_u |\mathcal{G}'(u)| \geq \sum_u |\mathcal{G}(u)| - N^{i} \frac{N^{r-i}}{4r} = |\mathcal{G}| - \frac{N^{r}}{4r} \geq |\mathcal{E}_i \cap \mathcal{F}_i| - \frac{N^{r}}{2r}.$$ 

Above, the first inequality uses (24) and the fact that the number of depth-$i$ nodes is $N^i$. The last inequality uses (23). This completes the proof.

Applying this lemma repeatedly, we obtain $|\mathcal{E}_r \cap \mathcal{F}_r| \geq |\mathcal{E}_0 \cap \mathcal{F}_0| - \frac{N^{r}}{4r} \geq N^{r}/2$. Consider any scenario $w \in \mathcal{E}_r \cap \mathcal{F}_r$. Note that $w \in \mathcal{F}_r$ means that, under scenario $w$, no $\mathcal{T}_i$ item is probed in rounds 0, · · · , $r$ - 1, i.e., all $r$ rounds. However, we need to probe item $Z_w$ in $\mathcal{T}_r$ in order to cover $f$ (under scenario $w$). This is a contradiction to solution $A$ being feasible and hence Assumption 7.1.

Thus, the cost of any $r$-round-adaptive solution must be at least $\frac{N}{(4r)^r} = \Omega(2^{r}/r^2)$. Using the fact that there is an adaptive solution of cost at most $r^\ell + 1$, this implies an adaptivity gap of $\Omega(\frac{1}{r^\ell} 2^s) = \Omega(\frac{s^{1/r}}{r^{\ell+1} \log s})$ as $s = 2^r \cdot r$. This completes the proof of Theorem 1.5. Note that the above instance can also be viewed as one for optimal decision tree: by just ignoring the $Z$-items, and with the goal of identifying the realized scenario.

8 Applications

8.1 Stochastic Set Cover

The stochastic set cover problem is a special case of stochastic submodular cover. It abstracts the unreliable sensor deployment problem mentioned in §1. The input is a universe $E$ of $d$ objects and a collection $\{X_1, \ldots, X_m\}$ of $m$ items. Each item $X_i$ 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 [GV06], where it was shown that the adaptivity gap (relative gap between the best non-adaptive and adaptive solutions) is between $\Omega(d)$ and $O(d^2)$. The correct adaptivity gap for stochastic set cover was posed as an open question by [GV06]. Subsequently, [AAK19] made significant progress, by showing that the adaptivity gap is $O(d \log d \cdot \log (mc_{\text{max}}))$. 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 $c_{\text{max}}$ may be arbitrary larger than $d$.

As a corollary of our result for (independent) stochastic submodular cover, we obtain a nearly-tight $O(d \log d)$ adaptivity gap for this problem. In fact, for any $r \geq 1$, we obtain an $r$-round adaptive algorithm that costs at most $O(d^{1/r} \log d)$ times the optimal (fully) adaptive cost. This nearly matches the $\Omega(\frac{1}{r^{1/r}} d^{1/r})$ gap for $r$-round adaptive algorithms shown in [AAK19]. 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.
8.2 Optimal Decision Tree

The optimal decision tree problem captures problems from a variety of fields including, but not limited to, learning theory, medical diagnosis, pattern recognition, and boolean logic (see [Mor82] and [Mur97] for a comprehensive survey on the use of decision trees). In an instance of optimal decision tree (ODT) we are given \( s \) hypotheses from which an unknown random hypothesis \( y^* \) is drawn. There is a collection of \( m \) tests, where 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.

Note that ODT is a special case of the scenario submodular cover problem. We can transform an instance of ODT into an instance for scenario submodular cover. We associate scenarios with hypotheses. For each test \( e \), we have an item \( \mathcal{X}_e \) of cost \( c_e \). Let \( \mathcal{X}_e(y) \) denote the realization of \( \mathcal{X}_e \) under scenario \( y \). We set

\[
\mathcal{X}_e(y) = \begin{cases} [s] \setminus T_e, & \text{if } y \in T_e \\ T_e, & \text{otherwise.} \end{cases}
\]

The joint distribution \( \mathcal{D} \) realizes to \( (\mathcal{X}_1(y), \ldots, \mathcal{X}_m(y)) \) with probability \( p_y \). We define the submodular function \( g(S) = |\bigcup_{\mathcal{X}_e \in S} \mathcal{X}_e| \) where \( \mathcal{X}_e \) corresponds to the realization of item \( \mathcal{X}_e \). Observe that \( g(S) \) computes the number of eliminated scenarios. Finally, we define \( f(S) = \min\{g(S), s - 1\} \); that is, we truncate full coverage at \( s - 1 \). This is because we can identify the realized scenario once \( s - 1 \) scenarios have been eliminated. Noting the \( f \) is an integer-valued monotone submodular function completes the description of the instance for scenario submodular cover. For any \( r \geq 1 \), we obtain an \( r \)-round adaptive algorithm that costs at most \( O(rs^{1/r} \log s) \) times the optimal (fully) adaptive cost. We also obtain a nearly matching lower bound: any \( r \)-round adaptive algorithm must incur approximation ratio \( \Omega(\frac{1}{r}s^{1/r}) \).

8.3 Shared Filter Evaluation

This problem was introduced by [MSW07] in the context of executing multiple queries on a dataset with shared (boolean) filters. There are \( n \) independent “filters” \( \mathcal{X}_1, \ldots, \mathcal{X}_n \), where each filter \( i \) has cost \( c_i \) and evaluates to \textit{true} with probability \( p_i \) (and \textit{false} otherwise). There are also \( m \) conjunctive queries, where each query \( j \in [m] \) is the “and” of some subset \( Q_j \subseteq [n] \) of the filters. So query \( j \) is \textit{true} iff \( \mathcal{X}_i = \text{true} \) 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 groundset \( U = \{T_i, F_i\}_{i=1}^m \) where \( T_i \) (resp. \( F_i \)) corresponds to filter \( i \) evaluating to \textit{true} (resp. \textit{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 \(|Q_j| \) if the query’s value is determined: it is \textit{false} when a single false filter is seen and it’s \textit{true} when all its filters are true. The maximal value of the function is \( Q = \sum_{j=1}^m |Q_j| \leq mn \). So, 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.
8.4 Correlated Knapsack Cover

There are \( n \) items, each of cost \( c_i \) and random (integer) reward \( X_i \). The rewards may be correlated and are given by a joint distribution \( D \) 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 groundset \( U = \{(i, v) : i \in [n], 0 \leq v \leq Q\} \) where element \((i, v)\) represents item \( X_i \) 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 \( S \subseteq U \). For any \( r \geq 1 \), 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.

9 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 of our computational experiments using Python 3.8 and Gurobi 8.1 with a 2.3 Ghz Intel Core i5 processor and 16 GB 2133 MHz LPDDR3 memory.

9.1 Stochastic Set Cover

**Instances.** We use the Epinions network\(^1\) and a Facebook messages dataset described in [RA15] 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 1000 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 dataset consists 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 with each node. 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 \( \alpha_S \) times: then \( X_u \) realizes to \( S \cup \{u\} \) with probability \( \alpha_S/500 \). This ensures that \( X_u \) always covers \( u \). We set \( f \) to be the coverage function and 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, since the Facebook messages network is sparse, we set \( \delta = 0.25 \) in the second instance.

**Results.** We test our \( r \)-round adaptive algorithm on the two kinds of instances described above. We vary \( r \) over integers in the interval \([1, \log n]\), where \( n \approx 1000 \). To compute an estimate of the expected cost, we sample new realizations 20 times and take the average cost incurred by the algorithm over these trials. In each trial, we also solve an integer linear program (using the Gurobi solver) to determine the optimal offline cost to cover \( Q \) elements for the given realization: we use the average cost over the trials as an estimate on an information-theoretic lower bound: 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 6 − 7 rounds of adaptivity

\(^1\)http://snap.stanford.edu/
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 3. Finally, note that the increase in expected cost with rounds of adaptivity (see Figure 3a) can be attributed to the probabilistic nature of our algorithm (and the experimental setup). We also notice this in the next batch of experiments.

9.2 Optimal Decision Tree

Instances. We use a real-world dataset called WISER\footnote{http://wiser.nlm.nih.gov/} for our experiments. The WISER dataset 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 that the patient has been exposed to (by testing as few symptoms as possible). This dataset has been used for evaluating algorithms for similar problems in other papers \cite{BAD+07, BBS11, BBS12, NKN20}. For each symptom-chemical pair, the data specifies whether or not 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 datasets 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 datasets, we consider two cases. The first case considered is one where all tests have unit cost. We refer to this as the WISER – U case. For the second case, we generate costs randomly for each test from \{1, 4, 7, 10\} according to the weight vector \{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 where tests may have different costs, and we may not want to schedule an expensive test without sufficient information. We refer to this as WISER – R case.

We also test our algorithm on synthetic data which we generate as follows. We set $s = 10000$ and $m = 100$. For each $y \in [s]$, we randomly generate a binary sequence of outcomes which corresponds 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 scenario. We generate instances using unit costs or by assigning a random cost from \{1, 4, 7, 10\} to each test according to the weight vector.
Results. We test our $r$-round adaptive algorithm on all of the above mentioned datasets. We vary $r$ over integers in the interval $[1, \log s]$. For the WISER $- U$ datasets, we compute the expected cost of our algorithm over all scenarios. We plot the expected costs over all rounds, and compare it to $\log(s)$ which is an information-theoretic lower bound for the ODT problem with unit costs and uniform probabilities over the scenarios. We observe that our algorithm gets very close to this lower bound in only 3 rounds of adaptivity. See Figure 4a. In the case of WISER $- R$, we compute the expected cost incurred by our $r$-round adaptive algorithm for $r$ varying over integers in the interval $[1, \log s]$ (expectation taken over all scenarios). We observe a sharp decrease in costs within 4 rounds of adaptivity. We plot the results in Figure 4b.

For the synthetic data, we sample scenarios over 100 trials (since $s = 10000$, computing expectation over all $s$ would be very slow). As in the previous case, for each trial, we compute the cost incurred by our $r$-round adaptive algorithm for $r$ varying over integers in the interval $[1, \log s]$. Then, we average all these costs and use it as an estimate for the expected cost. For SYN $- U - 0.2$ and SYN $- U - 0.5$, we compare the results to $\log(s)$ which is an information-theoretic lower bound for the instances (since scenarios are sampled uniformly at random). We observe a improvement in the costs within 6 rounds of adaptivity. Beyond this however, the costs do not improve (and we exclude it from our plot). We observe a similar trend for the case of SYN $- R - 0.2$ and SYN $- R - 0.5$. We plot the results in Figures 5 and 6.

Note that we only plot results related to the first dataset for the WISER $- U$ and WISER $- R$ cases. We include plots for all 10 datasets in Appendix D.

References

[AAK19] Arpit Agarwal, Sepehr Assadi, and Sanjeev Khanna. Stochastic submodular cover with limited adaptivity. In Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, page 323–342, 2019.
Figure 5: Computational results for ODT on synthetic data with unit costs

(a) SYN − U − 0.2

(b) SYN − U − 0.5

Figure 6: Computational results for ODT on synthetic data with non-uniform costs

(a) SYN − R − 0.2

(b) SYN − R − 0.5

[AH12] M. Adler and B. Heeringa. Approximating optimal binary decision trees. *Algorithmica*, 62(3-4):1112–1121, 2012.

[AN16] Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. *Manag. Sci.*, 62(8):2374–2391, 2016.

[BAD+07] Suresh K Bhavnani, Annie Abraham, Christopher Demeniuk, Messeret Gebrekristos, Abe Gong, Satyendra Nainwal, Gautam K Vallabha, and Rudy J Richardson. Network analysis of toxic chemicals and symptoms: implications for designing first-responder systems. *AMIA Annual Symposium Proceedings*, pages 51–55, 2007.

[BBS11] Gowtham Bellala, Suresh Bhavnani, and Clayton Scott. Active diagnosis under persistent noise with unknown noise distribution: A rank-based approach. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15, pages 155–163, 2011.

[BBS12] G. Bellala, S. K. Bhavnani, and C. Scott. Group-based active query selection for rapid diagnosis in time-critical situations. *IEEE Transactions on Information Theory*, 58(1):459–478, 2012.

[BBS18] Eric Balkanski, Adam Breuer, and Yaron Singer. Non-monotone submodular maximization in exponentially fewer iterations. In *Advances in Neural Information Processing Systems*, pages 2359–2370, 2018.
Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. Stochastic matching with few queries: (1-\(\epsilon\)) approximation. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1111–1124, 2020.

MohammadHossein Bateni, Hossein Esfandiari, and Vahab S. Mirrokni. Optimal distributed submodular optimization via sketching. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1138–1147, 2018.

Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved approximation results for stochastic knapsack problems. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1647–1665, 2011.

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

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

Nikhil Bansal and Viswanath Nagarajan. On the adaptivity gap of stochastic orienteering. *Math. Program.*, 154(1-2):145–172, 2015.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 283–302, 2019.

Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1138–1151, 2018.

Eric Balkanski and Yaron Singer. Approximation guarantees for adaptive sampling. In *Proceedings of the 35th International Conference on Machine Learning*, pages 393–402, 2018.

Domagoj Bradac, Sahil Singla, and Goran Zuzic. (Near) Optimal Adaptivity Gaps for Stochastic Multi-Value Probing. In *Approximation, Randomization, and Combinatorial Optimization*, volume 145, pages 49:1–49:21, 2019.

F. Cicalese, E. S. Laber, and A. M. Saettler. Diagnosis determination: decision trees optimizing simultaneously worst and expected testing cost. In *Proceedings of the 31th International Conference on Machine Learning*, pages 414–422, 2014.

V. T. Chakaravarthy, V. Pandit, S. Roy, P. Awasthi, and M. K. Mohania. Decision trees for entity identification: Approximation algorithms and hardness results. *ACM Transactions on Algorithms*, 7(2):15, 2011.

Chandra Chekuri and Kent Quanrud. Parallelizing greedy for submodular set function maximization in matroids and beyond. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 78–89, 2019.

Yuxin Chen, Hiroaki Shio, César Antonio Fuentes Montesinos, Lian Pin Koh, Serge Wich, and Andreas Krause. Active detection via adaptive submodularity. In *Proceedings of the 31st International Conference on Machine Learning*, page 1–55–I–63, 2014.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems*, pages 337–344, 2004.

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

35
[DHK16] Amol Deshpande, Lisa Hellerstein, and Devorah Kletenik. Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack. *ACM Trans. Algorithms*, 12(3), April 2016.

[EKM19] Hossein Esfandiari, Amin Karbasi, and Vahab S. Mirrokni. Adaptivity in adaptive submodularity. *CoRR*, abs/1911.03620, 2019.

[Fei98] U. Feige. A threshold of \( \ln n \) for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[GG74] M.R. Garey and R.L. Graham. Performance bounds on the splitting algorithm for binary testing. *Acta Informatica*, 3:347–355, 1974.

[GHKL16] Nathaniel Grammel, Lisa Hellerstein, Devorah Kletenik, and Patrick Lin. Scenario submodular cover. In *International Workshop on Approximation and Online Algorithms*, pages 116–128. Springer, 2016.

[GK11] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res. (JAIR)*, 42:427–486, 2011.

[GK17] Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. *CoRR*, abs/1003.3967, 2017.

[GK15] Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Math. Oper. Res.*, 40(1):56–79, 2015.

[GM09] Sudipto Guha and Kamesh Munagala. Multi-armed bandits with metric switching costs. In *Automata, Languages and Programming, 36th International Colloquium (ICALP)*, pages 496–507, 2009.

[GN13] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Integer Programming and Combinatorial Optimization - 16th International Conference*, pages 205–216, 2013.

[GNR17] Anupam Gupta, Viswanath Nagarajan, and R. Ravi. Approximation algorithms for optimal decision trees and adaptive TSP problems. *Math. Oper. Res.*, 42(3):876–896, 2017.

[GNS17] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and XOS functions. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1688–1702, 2017.

[GV06] Michel Goemans and Jan Vondrák. Stochastic covering and adaptivity. In *LATIN 2006: Theoretical Informatics*, pages 532–543. Springer Berlin Heidelberg, 2006.

[HR77] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is \( NP \)-complete. *Information Processing Lett.*, 5(1):15–17, 1976/77.

[INvdZ16] Sungjin Im, Viswanath Nagarajan, and Ruben van der Zwaan. Minimum latency submodular cover. *ACM Trans. Algorithms*, 13(1):13:1–13:28, 2016.

[JCK+14] Sh. Javdani, Y. Chen, A. Karbasi, A. Krause, D. Bagnell, and S. S. Srinivasa. Near optimal bayesian active learning for decision making. In *AISTATS*, pages 430–438, 2014.

[KG07] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, page 1650–1654. AAAI Press, 2007.

[KKT15] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.
S. R. Kosaraju, T. M. Przytycka, and R. S. Borgstrom. On an Optimal Split Tree Problem. In Proceedings of the 6th International Workshop on Algorithms and Data Structures, pages 157–168, 1999.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT ’11, page 510–520, 2011.

D. W. Loveland. Performance bounds for binary testing with arbitrary weights. Acta Inform., 22(1):101–114, 1985.

Zhen Liu, Srinivasan Parthasarathy, Anand Ranganathan, and Hao Yang. Near-optimal algorithms for shared filter evaluation in data stream systems. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD), pages 133–146, 2008.

Baharan Mirzasoleiman, Amin Karbasi, Ashwinkumar Badanidiyuru, and Andreas Krause. Distributed submodular cover: Succinctly summarizing massive data. In Advances in Neural Information Processing Systems, pages 2881–2889, 2015.

Bernard M. E. Moret. Decision trees and diagrams. ACM Comput. Surv., 14(4):593–623, December 1982.

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

Sreerama K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery, 2:345–389, 1997.

S. Mini, S. K. Udgata, and S. L. Sabat. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. IEEE Sensors Journal, 14(3):636–644, 2014.

Fatemeh Navidi, Prabhanjan Kambadur, and Viswanath Nagarajan. Adaptive submodular ranking and routing. Oper. Res., 68(3):856–877, 2020.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015.

Goran Radanovic, Adish Singla, Andreas Krause, and Boi Faltings. Information gathering with peers: Submodular optimization with peer-prediction constraints. In Proc. Conference on Artificial Intelligence (AAAI), February 2018.

C. Sun, V. O. K. Li, J. C. K. Lam, and I. Leslie. Optimal citizen-centric sensor placement for air quality monitoring: A case study of city of cambridge, the united kingdom. IEEE Access, 7:47390–47400, 2019.

I. Simon, N. Snively, and S. M. Seitz. Scene summarization for online image collections. In 11th International Conference on Computer Vision, pages 1–8, 2007.

Ruben Sipos, Adith Swaminathan, Pannaga Shivashwamy, and Thorsten Joachims. Temporal corpus summarization using submodular word coverage. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM ’12, page 754–763, 2012.

L.A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica, 2(4):385–393, 1982.
A Set-based Model for Rounds

Here we discuss the “set based” model for adaptive rounds, where each round probes a fixed subset of items (and incurs their total cost). In this model, as noted in [AAK19], we can no longer require the function $f$ to be covered with probability one. We also provide an example below where the $r$-round-adaptivity gap is very large if we require coverage with probability one. Therefore, we consider solutions that are only required to cover the function with high probability. We will still compare to the (fully) adaptive optimum $\text{OPT}$ that always covers the function.

Formally, an $r$-round adaptive solution in the set-based model proceeds as follows. For each round $i = 1, \ldots, r$, it specifies a subset $S_i$ of items (that depends on all realizations in rounds $1, \ldots, i − 1$) that are probed in parallel. The cost incurred in round $i$ is $c(S_i)$ the total cost of all probed items in that round.

**Example:** Consider an instance with $|U| = 1$, and function $f$ with $f(U) = 1$ and $f(\emptyset) = 0$. So parameter $Q = 1$. There are $m$ items, where each item $i \in \{1, \ldots, m − 1\}$ has cost $2^i$ and instantiates to $U$ with probability $\frac{1}{2}$. The last item $m$ costs $2^m$ and instantiates to $U$ with probability one. In the permutation model, there is a non-adaptive solution of cost $O(m)$, which just probes items in the order $1, 2, \ldots, m$. On the other hand, in the set-based model with $r < m$ rounds we claim that the optimal cost is $\Omega(2^{m/r})$ if we require $f$ to be covered with probability one. Note that any solution $B$ in the set-based model with $r$ rounds is given by indices $0 = i(0) \leq i(1) \leq \ldots, i(r − 1) \leq i(r) = m$ where the $j$th round probes items $i(j − 1) + 1, \ldots, i(j)$ (if $f$ is not already covered). The probability that round $j$ is needed in $B$ is $2^{−i(j − 1)}$. So the expected cost of $B$ is at least $\sum_{j=1}^{r} 2^{−i(j − 1)} \cdot 2^{i(j)} \geq 2^{m/r}$ where we used $\sum_{j=1}^{r} (i(j) − i(j − 1)) = i(r) − i(0) = m$. It follows that the $r$-round adaptivity gap in the set-based model is exponential in $m$ (while the adaptivity gap is constant for the permutation model). This is the reason our set-based algorithms only cover $f$ with high probability (rather than probability one).

A.1 Conversion Theorems

While we give our main technical results in the permutation model to make the analysis easier, we can translate our $r$-round algorithm in the permutation model into one in the set-based model, as we show next. Below, $\text{OPT}$ is the cost of an optimal fully-adaptive solution.

**Theorem A.1.** For any $r \geq 1$ and $\eta > 0$, there is a set-based $r$-round adaptive solution for stochastic submodular cover (resp. scenario submodular cover) with expected cost at most $O(\frac{\eta}{\eta} Q^{1/r} \log Q) \cdot \text{OPT}$ (resp. $O(\frac{\eta}{\eta} (s^{1/r} \log s + r \log Q) \cdot \text{OPT})$) and covers the function with probability at least $1 − \eta$.

**Proof.** We will show the following “black box” reduction. Suppose there is an $r$-round adaptive algorithm in the permutation model with approximation ratio $\alpha$. Then, there is an algorithm in the set-based model that for any $\eta \in (0, 1)$, finds an $r$-round adaptive solution that has expected cost at most $\frac{r\alpha}{\eta} \cdot \text{OPT}$ and covers the function with probability at least $1 − \eta$. The theorem would then follow from Theorems 1.1 and 1.3.

Consider any solution $A$ in the permutation-model that always covers the function. For each round $i = 1, \ldots, r$ in $A$, let $L_i$ be the permutation to probe items and let $\mu_i$ be the expected cost of probed items (after applying the stopping rule). Note that $L_i$ and $\mu_i$ depend on realizations in all previous rounds. Let $S_i$ denote the maximal prefix of $L_i$ that has total cost at most $\frac{\eta}{\eta} \mu_i$. We define the set-based solution $B$ as follows. In round $i$, we probe all items $S_i$ in parallel. Note that
the cost of solution $B$ in round $i$ is at most $\frac{r}{\eta} \mu_i$. Taking expectations and adding over all rounds, the expected cost of $B$ is at most $\frac{r}{\eta}$ times that of $A$. So the cost is at most $\frac{r}{\eta} \cdot \text{OPT}$.

We refer to round-$i$ as a failure if the round-$i$ stopping rule (in $A$) does not apply by the end of $S$. By Markov’s inequality, the probability that round-$i$ is a failure is at most $\frac{r}{\eta}$. Using a union bound over all $r$ rounds, the probability of a failure in any round is at most $\eta$. Note that, if there is no failure then solution $B$ indeed covers the function fully: as the items probed by $B$ is a superset of those probed by $A$. Hence, solution $B$ covers the function with probability at least $1 - \eta$.

Note that the approximation ratios for the set-based model are only a factor $r$ more than in the permutation model (assuming that the “failure probability” $\eta$ is any constant). So this is a good result for small (constant) $r$. When the number of rounds $r$ is larger, we have a different approach that does not lose this factor $r$ in the approximation ratio, but uses $O(r)$ set-based rounds.

**Theorem A.2.** For any $r \geq 1$, in the set-based model:

- there is a $2r$-round adaptive algorithm for stochastic submodular cover with expected cost at most $O(Q^{1/r} \log Q) \cdot \text{OPT}$
- there is a $4r$-round adaptive algorithm for scenario submodular cover with expected cost at most $O(s^{1/r} \log(sQ)) \cdot \text{OPT}$

that covers the function with probability at least $1 - e^{-\Omega(r)}$.

**Proof.** We first consider stochastic submodular cover. We make use of the partial cover algorithm PARCA (Theorem 3.1) in an iterative manner (similar to the $r$-round algorithm in the permutation model). The difference is that we will run PARCA for $2r$ set-based rounds (instead of $r$). The items probed in each set-based round is a prefix of the permutation-based round. Algorithm 5 gives a formal description. The state at any round is the tuple $(T, \tilde{T})$ consisting of previously probed items and their realizations. For the analysis, we view the iterative algorithm as a $2r$ depth tree, where the nodes at depth $i$ are the states in round $i$ and the branches out of each node represent the observed realizations in that round. For any $i \in [2r]$, let $\Omega_i$ denote all the states in round $i$; note that these form a partition of the outcome space.

**Algorithm 5** 2r-round set-based algorithm for stochastic submodular cover

1: let $T \leftarrow \emptyset$ denote the probed items and $T \leftarrow \emptyset$ their realization.
2: for $i = 1, 2, \ldots, 2r$ do
3: run PARCA $(X \setminus T, f_T, Q - f(T), Q^{-1/r})$.
4: let $L_i$ denote the permutation on items $X \setminus T$ from PARCA.
5: let $\mu_i$ be the expected cost of probing $L_i$ (in the permutation model).
6: let $R_i$ denote the maximal prefix of $L_i$ of cost at most $4\mu_i$.
7: probe items $R_i$ in the set-based solution and let $R_i$ denote their realizations.
8: update $T \leftarrow T \cup R_i$ and $T \leftarrow T \cup R_i$.

**Expected cost.** For any state $\omega \in \Omega_i$, let $\text{OPT}(\omega)$ denote the expected cost of the optimal adaptive policy conditioned on the realizations in $\omega$. By Theorem 3.1 conditioned on $\omega$, the expected cost $\mu_i(\omega)$ of the permutation $L_i(\omega)$ is at most $\alpha \cdot \text{OPT}(\omega)$ where $\alpha = O(\frac{1}{r}Q^{1/r} \log Q)$. Note that the cost
of the set-based round is at most $4 \cdot \mu_i(\omega)$. Hence, the (unconditional) expected cost in round $i$ is at most

$$4 \sum_{\omega \in \Omega_i} \Pr[\omega] \cdot \mu_i(\omega) \leq 4\alpha \sum_{\omega \in \Omega_i} \Pr[\omega] \cdot \OPT(\omega) = 4\alpha \cdot \OPT,$$

where we used that $\Omega_i$ is a partition of all outcomes. It follows that the expected cost of the set-based algorithm is at most $2r \cdot 4\alpha \cdot \OPT = 8r\alpha \cdot \OPT$ as claimed.

**Completion probability.** Consider any round $i$ and state $\omega \in \Omega_i$. Recall that $T$ denotes the realizations from prior rounds. The stopping rule for the permutation $L_i(\omega)$ in ParCA is that the residual target drops by a factor $\delta = Q^{-1/r}$. In other words, if $S$ denotes the realizations observed in permutation $L_i(\omega)$ then the stopping rule is $Q - f(T \cup S) < \delta \cdot (Q - f(T))$. Recall that the set-based round-$i$ involves probing the prefix $R_i(\omega)$ of $L_i(\omega)$ having cost $4 \cdot \mu_i(\omega)$. This round is said to be a success if the stopping rule applies by the end of $R_i(\omega)$. By the choice of $R_i$ and Markov’s inequality, the probability that this round is a success is at least $\frac{3}{4}$. So the expected number of successful rounds is at least $2r \cdot \frac{3}{4} = \frac{3}{2}r$. Moreover, the success events in different rounds $i = 1, \ldots, 2r$ are independent. Hence, by a Chernoff bound, the probability of seeing less than $r$ successes out of $2r$ rounds is at most $e^{-\Omega(r)}$. Finally, note that if there are at least $r$ successes then the function $f$ gets covered completely as each success reduces the residual target by a factor $Q^{-1/r}$. It follows that our algorithm covers $f$ with probability $1 - e^{-\Omega(r)}$ as claimed.

The proof for scenario submodular cover is identical, except for the use of SParCA (Corollary 6.6) instead of ParCA.

**B Items realizing to subsets**

Here, we consider a seemingly more general model for stochastic submodular cover, where each item realizes to a subset of elements. As before, there is a monotone submodular function $f : 2^U \to \mathbb{Z}_{\geq 0}$ with $Q := f(U)$. There are $m$ items, where each item $X_i$ has cost $c_i$ and realizes to a random subset of $U$. (In the basic model, each item realizes to a single element.) The goal is to select a set of items such that the union $S$ of their corresponding realizations satisfies $f(S) = Q$, and the expected cost is minimized. We can reduce this model to the usual one (considered in the paper) as follows. Let $W := 2^U$ denote an expanded groundset consisting of all subsets of $U$. Let function $g : 2^W \to \mathbb{Z}_{\geq 0}$ be defined as $g(R) = f(\cup_{r \in R} r)$ for any $R \subseteq W$. Note that $g$ is monotone submodular as $f$ is. Note that each item $X_i$ realizes to a single element of the expanded set $W$. Moreover, covering function $g$ is equivalent to covering $f$, and both have maximal value $Q$. So it suffices to solve the usual model with single-element realizations. However, the difficulty with this reduction is that the number of elements $|W|$ is exponential. Using the fact that our algorithm’s runtime is independent of the size of the groundset (see remark after Algorithm 2), it follows that we obtain a polynomial (in $m, Q, c_{\text{max}}$) time algorithm for this more general model as well.

**C Estimating Scores in Algorithm ParCA**

The partial covering algorithm ParCA (Algorithm 1) relies on computing the maximum score according to 1. We are not aware of a closed-form expression to calculate the score for general functions $f$. Instead, we show that sampling can be used to estimate these scores, which suffices for the overall algorithm. At any point in algorithm ParCA, let $S \subseteq X$ denote the items already
adapted to the non-adaptive list $L$. Then, we need to compute the maximum score $\max_{e \in \mathcal{X} \setminus S}$, where

$$g_e := \sum_{S \sim \mathcal{X} \setminus S: f(S) \leq \tau} P(S = S) \cdot \mathbb{E}_{X_e \sim X_e} \left[ \frac{f(S \cup X_e) - f(S)}{Q - f(S)} \right] = \mathbb{E}_{S, X_e} \left[ 1_{f(S) \leq \tau} \cdot \frac{f(S \cup X_e) - f(S)}{Q - f(S)} \right].$$

To this end, we use $K = O(m^2c_{\text{max}} \log(mc_{\text{max}}))$ independent samples to obtain an estimate $\tilde{g}_e$ for each item $e$. Then, we choose the item that maximizes $\tilde{g}_e$. Note that the time taken in each step is $O(m \cdot K)$, so the overall time taken by algorithm PARCA is $O(m^2K) = \text{poly}(m, c_{\text{max}})$.

Let $L$ denote the list produced by the (sampling based) PARCA algorithm, and let $\mathcal{N}A$ be the resulting non-adaptive solution. We now show that the expected cost of $\mathcal{N}A$ is $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon}) \cdot \text{OPT}$. As before, $\text{OPT}$ is the cost of an optimal fully-adaptive solution. This would prove Theorem 3.1 even for the sampling-based PARCA algorithm.

**Lemma C.1.** At any step, with already added items $S$, we have $\Pr[f(S) \leq \tau] \leq \sum_{x_e \in \mathcal{X} \setminus S} g_e$.

**Proof.** We can write:

$$\sum_{x_e \in \mathcal{X} \setminus S} g_e = \sum_{x_e \in \mathcal{X} \setminus S} \mathbb{E}_{S, X_e} \left[ 1_{f(S) \leq \tau} \cdot \frac{f(S \cup X_e) - f(S)}{Q - f(S)} \right] = \mathbb{E}_{\mathcal{X}} \left[ 1_{f(S) \leq \tau} \cdot \sum_{x_e \in \mathcal{X} \setminus S} \frac{f(S \cup X_e) - f(S)}{Q - f(S)} \right] \geq \mathbb{E}_{\mathcal{X}} \left[ 1_{f(S) \leq \tau} \cdot \frac{f(X) - f(S)}{Q - f(S)} \right] = \mathbb{E}_{\mathcal{X}} \left[ 1_{f(S) \leq \tau} \right] = \Pr[f(S) \leq \tau].$$

Above, the inequality is by submodularity and the second-last equality uses the fact that $f(X) = Q$ with probability one.

Let $L'$ denote the maximal prefix of list $L$ where $\max_e g_e \geq \varepsilon := \frac{1}{m^2c_{\text{max}}}$ at each step. Let $\mathcal{N}A'$ denote the cost incurred by the non-adaptive solution on items in $L'$ and $\mathcal{N}A''$ be the cost incurred on items $L \setminus L'$. Clearly, $\mathbb{E}[\mathcal{N}A] = \mathbb{E}[\mathcal{N}A'] + \mathbb{E}[\mathcal{N}A'']$.

By Lemma C.1, it follows that after $L'$ ends, we have $\Pr[f(S) \leq \tau] \leq m \cdot \max_e g_e \leq m \varepsilon \leq \frac{1}{mc_{\text{max}}}$. In other words, $\Pr[\mathcal{N}A \text{ continues after } L'] \leq \frac{1}{mc_{\text{max}}}$ and hence

$$\mathbb{E}[\mathcal{N}A''] \leq \Pr[\mathcal{N}A \text{ continues after } L'] \cdot mc_{\text{max}} = O(1).$$

It now remains to bound $\mathbb{E}[\mathcal{N}A']$.

Let $\bar{\mathcal{N}}A$ denote any non-adaptive solution obtained by algorithm PARCA assuming that it always selects an item $X_e$ with score($X_e$) $\geq \frac{1}{4}$ max$_{x \in \mathcal{X} \setminus S}$ score($x_i$). In words, $\bar{\mathcal{N}}A$ is built using items having $\frac{1}{4}$-approximately maximum score. It can be easily verified that the analysis in §3.1 can be extended to prove $\mathbb{E}[\bar{\mathcal{N}}A] \leq O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon}) \cdot \text{OPT}$. (The only change is an additional factor of $\frac{1}{4}$ in the right-hand-side of (13).) We now bound $\mathbb{E}[\mathcal{N}A']$ using $\mathbb{E}[\mathcal{N}A]$.

**Lemma C.2.** Consider any item $X_e$ in $L'$ (with $S$ denoting all previous items). Then, we have score($X_e$) $\geq \frac{1}{4}$ max$_{x \in \mathcal{X} \setminus S}$ score($x_i$) with probability at least $1 - \frac{1}{m^2c_{\text{max}}}$. 

**Proof.** We partition the remaining items $\mathcal{X} \setminus S$ into $I^+ = \{i: g_i \geq \varepsilon/4\}$ and $I^- = \{j: g_j < \varepsilon/4\}$.

Consider any item $i \in I^+$: so $g_i \geq \varepsilon/4$. As $\bar{g}_i$ is the average of $K$ independent samples each of mean $g_i$, using a Chernoff bound we obtain $\Pr[\frac{1}{2}g_i \leq \bar{g}_i \leq 2g_i] \geq 1 - e^{-\Omega(\varepsilon K)} \geq 1 - \frac{1}{m^2c_{\text{max}}}$. Now
consider an item $j \in I^-$: so $g_j < \varepsilon/4$. By a Chernoff bound again, $\Pr[\bar{g}_j > \varepsilon/2] \leq e^{-\Omega(\varepsilon K)} \leq \frac{1}{m^2 c_{\text{max}}}$. So, with probability at least $1 - \frac{1}{m^2 c_{\text{max}}}$,

$$\frac{1}{2} g_i \leq \bar{g}_i \leq 2g_i \text{ for all } i \in I^+, \text{ and } \bar{g}_j \leq \frac{\varepsilon}{2} \text{ for all } j \in I^-.$$ 

Below, we condition on this event.

As item $e$ is in $L'$, we know that $\max_i g_i \geq \varepsilon$. So $I^+ \neq \emptyset$. Recall that $\bar{g}_e = \max_i \bar{g}_i$. Combined with the above, it follows that $e \in I^+$. Moreover,

$$g_e \geq \frac{1}{2} \bar{g}_e = \frac{1}{2} \max_i \bar{g}_i = \frac{1}{2} \max_{i \in I^+} \bar{g}_i \geq \frac{1}{4} \max_{i \in I^+} g_i = \frac{1}{4} \max_i g_i.$$ 

This completes the proof. \qed

We say that the sampling was successful if the event in Lemma C.2 occurs for every item in $L'$. As there are at most $m$ items, sampling is successful with probability at least $1 - \frac{1}{m c_{\text{max}}}$. Conditioned on the sampling being successful, $L'$ is a prefix of some non-adaptive solution $\overline{NA}$ that always selects an item having $\frac{1}{4}$-approximately maximum score. Hence,

$$\mathbb{E}[\overline{NA}' | \text{sampling successful}] \leq \mathbb{E}[\overline{NA}] \leq O\left(\frac{1}{\delta} \log \frac{1}{\delta}\right) \cdot \mathbb{O}T.$$ 

If sampling is not successful then $\overline{NA}'$ costs at most $mc_{\text{max}}$. So,

$$\mathbb{E}[\overline{NA}] \leq \mathbb{E}[\overline{NA}' | \text{sampling successful}] + \Pr(\text{sampling not successful}) \cdot mc_{\text{max}} \leq O\left(\frac{1}{\delta} \log \frac{1}{\delta}\right) \cdot \mathbb{O}T.$$ 

It now follows that $\mathbb{E}[\overline{NA}] = \mathbb{E}[\overline{NA}'] + \mathbb{E}[\overline{NA}'' \leq O\left(\frac{1}{\delta} \log \frac{1}{\delta}\right) \cdot \mathbb{O}T$ as desired.

**D Additional Plots**

In the main body of the paper, we only include plots related to the first dataset for the WISER – U and WISER – R cases. The trends observed in the other datasets are similar. We include plots for all 10 datasets here for completeness.
Figure 7: Computational results for ODT on WISER – U

43
Figure 8: Computational results for ODT on WISER – R