Semi-Bandit Learning for Monotone Stochastic Optimization*

Arpit Agarwal
Computer Science & Engineering
Indian Institute of Technology Bombay
Mumbai, India
aarpit@iitb.ac.in

Rohan Ghuge
H. Milton Stewart School of
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, USA
rghuge3@gatech.edu

Viswanath Nagarajan
Industrial and Operations Engineering
University of Michigan
Ann Arbor, USA
viswa@umich.edu

2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS) | 979-8-3315-1674-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/FOCS61266.2024.00083

***Abstract*—Stochastic optimization is a widely used approach for optimization under uncertainty, where uncertain input parameters are modeled by random variables. Exact or approximation algorithms have been obtained for several fundamental problems in this area. However, a significant limitation of this approach is that it requires full knowledge of the underlying probability distributions. *Can we still get good (approximation) algorithms if these distributions are unknown, and the algorithm needs to learn them through repeated interactions?* In this paper, we resolve this question for a large class of "monotone" stochastic problems, by providing a generic online learning algorithm with $\sqrt{T \log T}$ regret relative to the best approximation algorithm (under known distributions). Importantly, our online algorithm works in a semi-bandit setting, where in each period, the algorithm only observes samples from the random variables that were actually probed. Our framework applies to several fundamental problems in stochastic optimization such as prophet inequality, Pandora's box, stochastic knapsack, stochastic matchings and stochastic submodular optimization.**

***Index Terms*—online learning, online-to-offline, adaptivity gaps**

I. Introduction

Stochastic optimization problems have been a subject of intense investigation, as they offer a powerful lens through which we can handle uncertain inputs. In stochastic problems, uncertain input parameters are modeled by random variables, which are usually independent. Solutions (or policies) to a stochastic problem are sequential decision processes, where the next decision depends on all previously observed information. This approach has been applied to many domains: optimal stopping (Pandora's box [1], [2] and prophet inequality [3]–[5]), stochastic submodular optimization [6]–[8], stochastic probing [9]–[11] and various other stochastic combinatorial optimization problems (e.g., knapsack [12], [13] and matching [14], [15]). A fundamental assumption in all these results is that the underlying probability distributions are known to the algorithm. While the known-distributions assumption holds in some settings, it is not satisfied in many practical applications, e.g., in the absence of historical data. Our main goal in this paper is to relax this assumption and handle stochastic problems with *unknown distributions*. We start by providing some concrete examples.

Series testing. There is a system with n components, where each component i is "working" independently with some probability p_i. All n components must be working for the system to be functional. Moreover, it costs c_i to test component i and determine whether/not it is working. The goal is to test components sequentially to determine whether/not the system is functional, at the minimum expected cost. Clearly, testing will continue until some component is found to be not working, or we have tested all components and found them to be working. In the standard setting, where the probabilities $\{p_i\}_{i=1}^n$ are known upfront, the greedy policy that tests components in decreasing order of $\frac{1-p_i}{c_i}$ is optimal [16].

Pandora's box. There are n items, where each item i has an independent random value X_i with a known distribution $\mathcal{D}_i$. The realized value of X_i can be revealed by inspecting item i, which incurs cost c_i. The goal is to inspect a subset S of the items (possibly adaptively) to maximize the expected utility $\mathbb{E}\left[\max_{i\in S} \mathsf{X}_i - \sum_{i\in S} c_i\right]$. We emphasize that a solution to this problem can be quite complex: the choice of the next item to inspect may depend on the realizations of all previous items. Nevertheless, there is an elegant optimal solution to this problem when the distributions $\{\mathcal{D}_i\}_{i=1}^n$ are known [1].

Stochastic covering knapsack. There are n items: each item i has a (deterministic) cost c_i and independent random reward R_i with known distribution $\mathcal{D}_i$. The realized value of R_i is only known when item i is selected. Given a target Q, the goal is to select items sequentially (and adaptively) until the total reward is at least Q. The

V. Nagarajan's research was supported in part by NSF grant CCF-2418495.

2575-8454/24/$31.00 ©2024 IEEE
DOI 10.1109/FOCS61266.2024.00083

objective is to minimize the expected cost of selected items. Although the deterministic knapsack problem is already NP-hard, there is an elegant 3-approximation algorithm for stochastic covering knapsack [12] with known distributions.

In each of the above problems, what if the underlying probabilities/distributions are unknown? Is it still possible to obtain good performance guarantees? In order to address this question formally, we utilize the framework of *online learning*. Here, the algorithm interacts with an unknown-but-fixed input distribution $\mathcal{D}$ over multiple time periods.[1] In each period $t = 1, \cdots T$, the online algorithm comes up with a solution/policy σ^t to the stochastic problem and receives some *feedback* based on the performance of σ^t on the (unknown) distribution $\mathcal{D}$.

The type of feedback received is a crucial component in this learning-based framework. The simplest setting is *full feedback*, where the algorithm receives one sample from every random variable (r.v.). However, this is unrealistic for stochastic problems because the policy σ^t in period t only observes the r.v.s corresponding to some *subset* of the items. For example, in series testing, σ^t tests the n components in some order until the first non-working component is found: if the first $k-1$ components are working and the k^{th} component is not working then we would only observe the outcomes/samples of the first k r.v.s (and the remaining $n-k$ r.v.s are unobserved). In this paper we consider the more natural setting of *semi-bandit feedback*, where in each period t, the algorithm only receives samples from the r.v.s that the policy σ^t actually observed.

Our goal is to minimize the expected *regret* of the online algorithm, which is the difference between the total T-period objective of our algorithm and the optimum (which knows the distribution $\mathcal{D}$). Obtaining $o(T)$ regret dependence with respect to T means that our online algorithm, which doesn't know $\mathcal{D}$, asymptotically approaches the optimum. An additional difficulty arises from the fact that many problems that we consider are NP-hard (e.g., knapsack and submodular optimization). In such cases, we cannot hope to approach the optimum (via a polynomial algorithm). Here, we use the notion of α-regret, where the online algorithm approaches the α approximately optimal value. We ask: *can we transform any offline (approximation) algorithm for a stochastic problem into an online learning algorithm that has low-regret with respect to this offline algorithm?*

Our main result is an affirmative answer to this question. We give a general method for transforming any offline α-approximation algorithm (with known distributions) to an online learning algorithm with α-regret of $\widetilde{O}(\sqrt{T})$. It is well-known that the $\sqrt{T}$ regret bound cannot be improved, even in very special cases. Our method works for a wide range of stochastic problems that satisfy a natural *monotonicity* condition. This includes several fundamental problems in stochastic optimization such as prophet inequality, Pandora's box, stochastic knapsack, stochastic matching, stochastic submodular maximization and stochastic submodular cover.

A. Problem Setup

We first set-up notation for stochastic optimization problems in the known distribution setting.

a) Stochastic Optimization: Consider a stochastic problem $\mathcal{P}$ where the input consists of n items with an independent real-valued random variable (r.v.) X_i associated with each item $i \in [n] = \{1, 2, \cdots n\}$. There is a *known* probability distribution $\mathcal{D}_i$ for X_i; that is, $\mathsf{X}_i \sim \mathcal{D}_i$. We assume that each $\mathcal{D}_i$ has finite support, and denote the set of outcomes of all r.v.s by O. (We relax the discrete-distribution assumption in §II-B.) In order to determine the realization of any r.v. X_i, we need to *probe* item i.

A solution or *policy* for $\mathcal{P}$ is given by a decision tree σ, where each node is labeled by an item to probe next, and the branches out of a node correspond to the random realization of the probed item. Each node in decision tree σ also corresponds to the current "state" of the policy, which is given by the sequence of previously-probed items along with their realizations; we will refer to nodes and states of the policy interchangeably. The root node of σ is the starting state of the policy, at which point no item has been probed. Leaf nodes in σ are also called *terminal* nodes/states. The *policy execution* under any realization $\mathbf{x} = (x_1, \ldots, x_n)$ corresponds to a root-leaf path $\sigma_{\mathbf{x}}$ in the decision tree σ, where at any node labeled by item i, path $\sigma_{\mathbf{x}}$ follows the branch corresponding to outcome x_i. Note that running policy σ under $\mathbf{x}$ corresponds to traversing path $\sigma_{\mathbf{x}}$. We also define $S(\sigma, \mathbf{x})$ as the sequence of items probed by policy σ under realization $\mathbf{x}$.

The *cost* of policy σ depends on the realization $\mathbf{x}$, and we use $f(\sigma, \mathbf{x}) \geq 0$ to denote the cost of policy σ under realization $\mathbf{x}$. Specifically, $f(\sigma, \mathbf{x})$ depends only on the sequence of probed items $S(\sigma, \mathbf{x})$ and their realizations. We assume that cost is accrued only at terminal/leaf nodes: this is without loss of generality as every policy execution ends at a leaf node.[2] We make no assumptions about the cost function f beyond boundedness: it does not have to be linear, submodular etc. Our goal is

[1] In the single period setting, it is easy to construct examples that rule out any reasonable performance bound for these problems with unknown distributions.

[2] In some problems, it is natural to accrue cost at each state that the policy goes through. We can easily convert such a cost structure to our setting by placing all the cost accrued from a policy execution at its terminal state.

to find a policy σ that minimizes the expected cost $f(\sigma) := \mathbb{E}_{\mathbf{x}}[f(\sigma, \mathbf{x})]$. We also allow for constraints in problem $\mathcal{P}$, which must be satisfied under all realizations (i.e., with probability 1). Let $\mathcal{C}$ denote the set of feasible policies (that satisfy the constraints in $\mathcal{P}$). Sometimes, we work with *randomized* policies, where each node in the decision tree corresponds to a probability distribution over items (rather than a single item). Randomized policies are not any stronger than deterministic ones: there is always an optimal deterministic policy for this class of problems. Succinctly, the stochastic problem $\mathcal{P}$ is as follows (maximization problems are defined similarly).

$$\text{minimize} \quad f(\sigma) = \mathbb{E}_{\mathbf{x}}\left[f(\sigma, \mathbf{x})\right] \quad \text{subject to} \quad \sigma \in \mathcal{C}.$$

Observe that the number of nodes in policy σ may be exponential. So, we are interested in "efficient" policies that can be implemented in polynomial-time for any realization $\mathbf{x}$. For some stochastic problems like series systems and Pandora's box, efficient optimal policies are known. On the other hand, there are many problems like stochastic knapsack, matching and set-cover, where optimal policies may not be efficient: in these cases, we will focus on (efficient) *approximately* optimal policies. We use σ^* to denote an optimal policy, and denote its expected cost by $\texttt{OPT}$. An α-approximation algorithm for a stochastic problem $\mathcal{P}$ takes as input the probability distributions $\{\mathcal{D}_i\}_{i=1}^n$ (along with any objective/constraint parameters) and returns a policy that has expected cost at most $\alpha \cdot \texttt{OPT}$. We use the convention that for minimization problems, the approximation ratio $\alpha \geq 1$, whereas for maximization problems $\alpha \leq 1$.

b) The Online Semi-Bandit Setting: In this setting, the distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are *unknown* (other parameters such as the cost function and constraints are known). Moreover, we have to repeatedly solve the stochastic problem $\mathcal{P}$ over T time periods. We assume throughout the paper that $T \geq n$. The distributions $\{\mathcal{D}_i\}_{i=1}^n$ remain the same across all T periods. The goal is to simultaneously learn the distributions and converge to a good policy over time. At time $t \leq T$, we use all prior observations to present policy σ^t. The policies σ^t may be different for each time $t \in [T]$. We measure our learning algorithm in terms of *expected total regret*. For minimization problems, we define it as follows.

$$\begin{aligned} R(T) &= \mathbb{E}_{\mathbf{x}^1,\ldots,\mathbf{x}^T}\left[\sum_{t=1}^{T}\left(f(\sigma^t, \mathbf{x}^t) - f(\sigma^*, \mathbf{x}^t)\right)\right] \\ &= \mathbb{E}_{\mathbf{x}^1,\ldots,\mathbf{x}^T}\left[\sum_{t=1}^{T} f(\sigma^t, \mathbf{x}^t)\right] - T \cdot \texttt{OPT}. \end{aligned} \tag{1}$$

Here $\mathbf{x}^t$ represents the realization at time t. For conciseness, we use $\mathbf{h}^{t-1}$ to denote the *history* until time t; that is, $\mathbf{h}^{t-1} = (\mathbf{x}^1, \ldots, \mathbf{x}^{t-1})$. Note that policy σ^t is itself random because it depends on prior observations, i.e., on the history $\mathbf{h}^{t-1}$. The algorithm's cost at time t is $f(\sigma^t, \mathbf{x}^t)$, which depends additionally on the realizations $\mathbf{x}^t$ at time t. We can also re-write (1) as follows.

$$\begin{aligned} R(T) &= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\left[\mathbb{E}_{\mathbf{x}^t}\left[f(\sigma^t, \mathbf{x}^t) - f(\sigma^*, \mathbf{x}^t)\right]\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\mathbb{E}_{\mathbf{x}^t}\left[f(\sigma^t, \mathbf{x}^t)\right] - T \cdot \texttt{OPT}. \end{aligned} \tag{2}$$

Note that many of the stochastic optimization problems are known to be NP-hard, even when the distributions are known. Thus, we do not expect to learn policies that approach $\texttt{OPT}$. To get around this issue, we define the notion of α-regret as follows.

$$\alpha\text{-}R(T) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\left[\mathbb{E}_{\mathbf{x}^t}\left[f(\sigma^t, \mathbf{x}^t)\right]\right] - \alpha \cdot T \cdot \texttt{OPT}. \tag{3}$$

An important component in this learning setup is what type of *feedback* is received by the algorithm in each round. In the *full feedback* setting, the algorithm gets to know the entire sample $\mathbf{x}^t = (x_1^t, \ldots, x_n^t)$, which could be used to update beliefs regarding all the distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$. However, as mentioned earlier, this is unrealistic in our setting because in each period, the policy only observes a subset of r.v.s. So, we consider *semi-bandit* feedback, where the algorithm only observes the realizations of items that it probes. That is, at time t, the algorithm only observes $(X_i^t : i \in S(\sigma^t, \mathbf{x}^t))$ where $S(\sigma^t, \mathbf{x}^t)$ is the set of items probed by policy σ^t before it terminates.

B. Results and Techniques

For any stochastic problem instance $\mathcal{I}$ and distribution $\mathbf{D} = \{\mathcal{D}_i\}_{i=1}^n$, let $\texttt{OPT}_{\mathcal{I}}(\mathbf{D})$ denote the optimal cost of instance $\mathcal{I}$ with r.v.s having distribution $\mathbf{D}$. We first define a monotonicity condition for a stochastic problem $\mathcal{P}$ which will be used in our main result.

Definition I.1 (Monotonicity)**.** *A stochastic problem $\mathcal{P}$ is* up-monotone *if for any instance $\mathcal{I}$ and probability distributions $\mathbf{D}$ and $\mathbf{E}$ where $\mathbf{E}$ stochastically dominates $\mathbf{D}$, we have $\texttt{OPT}_{\mathcal{I}}(\mathbf{E}) \leq \texttt{OPT}_{\mathcal{I}}(\mathbf{D})$. Similarly, the problem is said to be* down-monotone *if for any instance $\mathcal{I}$ and probability distributions $\mathbf{D}$ and $\mathbf{E}$ where $\mathbf{E}$ stochastically dominates $\mathbf{D}$, we have $\texttt{OPT}_{\mathcal{I}}(\mathbf{E}) \geq \texttt{OPT}_{\mathcal{I}}(\mathbf{D})$.*

Theorem I.1. *Suppose that a stochastic problem $\mathcal{P}$ has an α-approximation algorithm, and it is either up-monotone or down-monotone. Then, there is a polynomial time semi-bandit learning algorithm for $\mathcal{P}$ (with unknown distributions) that has α-regret $O(nkf_{max}\sqrt{T\log(kT)})$. Here, n is the number of items,*

k is the maximum support size of any distribution, f_{max} is the maximal value of the objective function f (over all realizations) and T is the number of time periods.

We note that the dependence on the support-size k can be replaced by the maximum number of "thresholds" used at any node of the stochastic policy (which corresponds to the number of branches out of any node). For most of our applications, we have approximate policies with $O(1)$ thresholds: so the regret bound is just $O(nf_{max}\sqrt{T\log T})$. See Theorem II.4 for the formal statement, which also handles continuous r.v.s.

Consequently, we get polynomial time online learning algorithms for a number of stochastic optimization problems. We list a few of them next, highlighting the dependence on n and T (see §III for further details and the full version for more applications).

- Pandora's box: 1-regret $O\left(n\sqrt{T\log T}\right)$
- Prophet inequality: 1-regret $O\left(n\sqrt{T\log T}\right)$
- Matroid Prophet inequality [17]: $\frac{1}{2}$-regret $O\left(n\sqrt{T\log T}\right)$
- Stochastic (maximization) knapsack [13], [18]: $\frac{1}{2}$-regret $O\left(n\sqrt{T\log T}\right)$
- Stochastic covering knapsack [12]: 3-regret $O\left(n\sqrt{T\log T}\right)$

The first two results improve over the regret bounds of $O\left(n^3\sqrt{T}\log(T)\right)$ and $O\left(n^{4.5}\sqrt{T}\log(T)\right)$ for prophet inequality and Pandora's box from [19]. To the best of our knowledge, our results for all other applications are the first α-regret bounds of $\sqrt{T}$. We note that the result of [19] for Pandora's box and Prophet inequality holds under the more restrictive *bandit* feedback model (only the objective value is observed). However, it is limited to these two problems, whereas our approach for *semi-bandit* feedback provides $\sqrt{T}$ regret for a much broader class of problems. Prior work (based on sample complexity) only implies $T^{2/3}$ regret for these problems [20] (see §A for details).

The regret bound in Theorem I.1 is nearly optimal in the following sense. There is an $\Omega(\sqrt{T})$ lower bound on the regret even in the much simpler setting of multi-armed-bandits [21]. Furthermore, if we move beyond our setting of independent-identically-distributed (i.i.d.) distributions across periods, to the "adversarial" setting with different distributions for each period, then there is a linear $\Omega(T)$ lower-bound on regret for prophet inequality [19], which shows that sublinear regret is not possible for our class of problems in the adversarial case.

Technical Overview. At a high-level, our algorithm is based on the *principle of optimism in the face of uncertainty* which is well-studied in the multi-armed bandits literature: see e.g., the UCB algorithm [22]. Given observations from previous rounds, our algorithm first constructs a modified empirical distribution $\mathbf{E} = \{\mathcal{E}_i\}_{i=1}^n$ which *stochastically dominates* the true distribution $\mathbf{D}$. It then executes policy σ given by the offline (approximation) algorithm under distribution $\mathbf{E}$.

We now discuss how to bound the regret of the policy σ under the true distribution $\mathbf{D}$. To keep things simple here, we assume that $\alpha = 1$. We also assume that the problem is up-monotone and has a minimization objective (our analysis for down-monotonicity and/or maximization objective is identical). The difficulty in our setting is that the decision to probe/observe an item is *random*: it depends on the choice of policy σ and the underlying distribution $\mathbf{D}$ of items. In contrast, the classical multi-armed bandits setting allows direct control to the algorithm on which item to probe: so the UCB algorithm can control the rate of *exploration* of individual items and bound the regret of individual items in terms of this rate. How do we bound the rate of exploration of different items when we do not have direct control on which items are probed? Moreover, how do we bound the regret contribution of individual items in terms of their rate of exploration?

The key insight in our analysis is the following "stability" bound, which answers the above questions. Given product distributions $\mathbf{D}$ and $\mathbf{E}$, where $\mathbf{E}$ stochastically dominates $\mathbf{D}$, we show[3]

$$\begin{aligned} f(\sigma|\mathbf{D}) - f(\sigma^*|\mathbf{D}) &\leq f(\sigma|\mathbf{D}) - f(\sigma|\mathbf{E}) \\ &\leq f_{\max}\sum_{i=1}^{n} Q_i(\sigma)\cdot\epsilon_i, \end{aligned}$$

where σ^* (resp. σ) is an optimal policy under $\mathbf{D}$ (resp. $\mathbf{E}$), $Q_i(\sigma)$ is the probability that policy σ probes item i, and ϵ_i is the *total-variation* distance between $\mathcal{E}_i$ and $\mathcal{D}_i$. The first inequality above follows from the monotone property. The second inequality requires analyzing the decision tree of σ carefully and is a key technical contribution of this paper (see §II-A).

The above stability bound can be interpreted as follows: the contribution of item i in the total regret is the probability that it is probed times the total variation error in estimating $\mathcal{D}_i$. The dependence on the probability of probing is crucial here. This allows us to analyze the regret using a *charging argument* where we charge item i for regret in a *pay-per-use* manner. The paths in the execution tree of the online algorithm that probe i will pay for the error in estimation of $\mathcal{D}_i$, but then ϵ_i will also reduce along these paths. Moreover, we can bound the overall regret irrespective of the policy σ that is used at periods. The error $\epsilon_i \approx \sqrt{1/m}$ when i has been probed m times in the past, which gives a total regret contribution of $\sum_{m=1}^{T}\sqrt{1/m} = O(\sqrt{T})$ for any item. Finally, we lose an extra logarithmic factor to ensure that

[3]For policy σ and distribution $\mathbf{D}$, $f(\sigma|\mathbf{D})$ is the expected objective of σ when the r.v.s have distribution $\mathbf{D}$.

our empirical distribution $\mathbf{E}$ stochastically dominates $\mathbf{D}$ with high probability.

Let us contrast our algorithm (and stability bound) with another algorithm that balances the *explore-exploit* trade-off in a different manner: the *explore-then-commit* algorithm. This algorithm will explore each item $O(T^{2/3})$ times (say, by probing it first in policy σ) and then commit to a policy based on these $O(nT^{2/3})$ observations. Using the sample complexity bound of [20] it is easy to show that this algorithm achieves $O(T^{2/3})$ regret; see §A for details. However, each item is explored an equal number of times regardless of its cost and we might end up exploring some high-cost spurious items too many times. Our stability bound highlights a crucial difference between our algorithm and this algorithm: we only explore items that are needed in the policy. If there are items which are rarely used in the policy, they will only be explored infrequently and will not contribute to the regret as well. This also explains why our approach achieves a near-optimal $\sqrt{T \log T}$ regret bound compared to the $T^{2/3}$ bound of the sampling-based approach.

C. Related Work

As mentioned earlier, stochastic optimization problems (under known distributions) have been studied extensively. Several papers have extended the classic prophet inequality [3] and Pandora's box [1] results to more complex settings, e.g., [2], [4], [5], [23]–[25]. Moreover, good approximation bounds have been achieved for stochastic versions of various combinatorial optimization problems [7]–[9], [11]–[15], [18], [26], [27].

There has also been extensive work in online learning, where one considers unknown distributions (in the stochastic learning setting), see e.g. books [28]–[30]. All our stochastic problems can be modeled as Markov decision processes (MDPs) with unknown transition probabilities, and there have been some works on achieving sublinear regret in this setting [31]–[33]. However, the regret bounds from these works have a polynomial dependence on the "state space" of the MDP, which in our setting, is exponential in n (the number of r.v.s). It is also known that any online algorithm for arbitrarty MDPs incurs an $\Omega(\sqrt{S \cdot T})$ regret, where $S = \exp(n)$ is the size of the state space [34], [35]. In contrast, we obtain regret bounds that depend polynomially on n.

Although both stochastic optimization and online learning have been subjects of comprehensive research, there has been relatively limited work at the intersection of these two domains. For instance, [36] studied a special case of adaptive submodular maximization [7] in the semi-bandit setting. Some recent papers [37], [38] considered online learning for Pandora's box in a correlated setting, which is much harder than the independent case: one needs to resort to weaker "partially adaptive" benchmarks here. [39] also considered a different "multi armed" Pandora's box problem, and obtained a $\frac{1}{2}$-competitive ratio relative to a suitable benchmark. Drawing closer to our work is [19], which explores Pandora's box and prophet inequality within the online learning framework with bandit feedback, which is even more restrictive than our semi-bandit feedback. In *bandit* feedback, the algorithm only observes the realized objective value of its policy σ. [19] gave algorithms with 1-regret $O\left(n^3\sqrt{T}\log(T)\right)$ and $O\left(n^{4.5}\sqrt{T}\log(T)\right)$ for Pandora's box and prophet inequality respectively. While our feedback model is more relaxed than [19], we think that semi-bandit feedback already captures the issue of partial observations in policies. Moreover, under semi-bandit feedback we obtain a general framework that applies to several stochastic optimization problems.

There have also been several works studying sample complexity bounds for stochastic optimization problems. The goal in these works is to understand how much data is necessary and sufficient to guarantee near-optimal algorithms. Such results have been obtained for single-parameter revenue maximization [40], [41] and prophet inequality [42]. Recent work [20] gives optimal sample complexity bounds for stochastic optimization problems that exhibit *strong monotonicity*, which is a slightly stronger condition than our monotonicity definition. These results imply $\sqrt{T}$ regret under the full-feedback model, but only $T^{2/3}$ regret under the semi-bandit feedback model considered in this paper (see Appendix A).

Another relevant line of work is on combinatorial multi-armed bandits (CMAB), which also involves semi-bandit feedback [43]–[46]. Here, there are n base arms that produce stochastic outcomes drawn from an unknown fixed distribution. There is also a collection $\mathcal{F} \subseteq 2^{[n]}$ of allowed "super arms". In each period t, the algorithm selects a super-arm $S^t \in \mathcal{F}$ and observes the realizations of all arms $i \in S^t$. The result closest to ours is by [46], which considers a class of non-linear objectives satisfying a "monotone" condition and obtains a UCB-type algorithm achieving α-regret of $O(n\sqrt{T \log T})$ where α is the approximation ratio for the offline problem. Our setting is much more general because we need to select *policies* (not static subsets). When we select a policy, we do not know which arms will actually be observed. So, we only have an indirect control on what arms will be observed in each round. While our algorithm can be seen as a natural extension of [46], our analysis requires new ideas: in particular in proving the stability Lemma II.3 and its use in bounding overall regret.

There are also some online-to-offline reductions that work in the adversarial setting (which is harder than

our stochastic setting). In particular, [47] considered combinatorial optimization with *linear* objective functions, and obtained 1-regret of $O(n\sqrt{T})$ under full-feedback (assuming an exact offline algorithm). Then, [48] extended this result to linear problems with only an α-approximation algorithm, and obtained α-regret of $O(n\sqrt{T})$ under full-feedback and $O(nT^{2/3})$ under bandit-feedback. Recent work of [49], [50] considers certain combinatorial optimization problems with non-linear objectives, assuming an α-approximation via a greedy-type algorithm. For such problems, [49] obtained $\sqrt{T}$ regret under full-feedback and $T^{2/3}$ regret under bandit-feedback. While these results work in the harder adversarial online setting, our result handles a much wider class of problems: we learn policies (rather than just subsets) and handle arbitrary objectives. As noted before, there is an $\Omega(T)$ regret lower-bound for some of our applications in the adversarial setting.

D. Preliminaries

We present some preliminaries that are required in our algorithm and proofs.

Definition I.2 (Stochastic Dominance)**.** *A probability distribution $\mathcal{E}$ (over $\mathbb{R}$) stochastically dominates another distribution $\mathcal{D}$ if, for all $a \in \mathbb{R}$, we have: $\mathbf{P}_{X \leftarrow \mathcal{E}}(X \geq a) \geq \mathbf{P}_{Y \leftarrow \mathcal{D}}(Y \geq a)$.*

We use $\mathcal{D} \preceq_{\mathsf{SD}} \mathcal{E}$ to denote that distribution $\mathcal{E}$ stochastically dominates $\mathcal{D}$. For product distributions $\mathbf{D} = \{\mathcal{D}_i\}_{i=1}^n$ and $\mathbf{E} = \{\mathcal{E}_i\}_{i=1}^n$, we say that $\mathbf{E}$ stochastically dominates $\mathbf{D}$ if $\mathcal{E}_i$ stochastically dominates $\mathcal{D}_i$ for all $i \in [n]$; we also denote this by $\mathbf{D} \preceq_{\mathsf{SD}} \mathbf{E}$.

Let $\mathtt{TV}(\mathcal{D}, \mathcal{E})$ denote the *total variation distance* between discrete distributions $\mathcal{D}$ and $\mathcal{E}$. The total variation distance is half of the ℓ_1 distance between the two distributions, i.e., $\mathtt{TV}(\mathcal{D}, \mathcal{E}) = \frac{1}{2} \cdot ||\mathcal{D} - \mathcal{E}||_1$. The following standard result (see, for example, Lemma B.8 in [51]) bounds the total variation distance between product distributions.

Lemma I.2. *Given product distributions $\mathbf{D} = \{\mathcal{D}_i\}_{i=1}^n$ and $\mathbf{E} = \{\mathcal{E}_i\}_{i=1}^n$ over n r.v.s, we have*

$$\mathtt{TV}(\mathbf{D}, \mathbf{E}) \leq \sum_{i \in [n]} \mathtt{TV}(\mathcal{D}_i, \mathcal{E}_i).$$

Consider independent r.v.s $\mathsf{X}_1, \ldots, \mathsf{X}_n$ where T_i denotes the domain of X_i. Let h be a function from $\mathbf{T} = \mathsf{T}_1 \times \cdots \times \mathsf{T}_n$ to $[0, U]$; that is, h is a function on the *outcomes* of the random variables that is bounded by U. Thus, for any $\mathbf{x} \in \mathbf{T}$, $h(\mathbf{x})$ denotes the value of h on the outcome $\mathbf{x} = (x_1, \ldots x_n)$. Given a product distribution $\mathbf{P}$ over the r.v.s, define $h(\mathbf{P}) := \mathbb{E}_{\mathbf{x} \sim \mathbf{P}}[h(\mathbf{x})]$. The following useful fact bounds the difference in function value at two different distributions.

Lemma I.3. *Given discrete distributions $\mathbf{D}$ and $\mathbf{E}$ over n random variables $\mathsf{X}_1, \ldots, \mathsf{X}_n$, and a $[0, U]$ bounded function h on the outcomes of these r.v.s (as above), we have*

$$|h(\mathbf{D}) - h(\mathbf{E})| \leq U \cdot \mathtt{TV}(\mathbf{D}, \mathbf{E}).$$

II. The Online Framework

In this section we present our main algorithm. We will assume access to an α-approximation algorithm $\mathtt{ALG}$ for the stochastic problem $\mathcal{P}$. For concreteness, we assume that $\mathcal{P}$ is a minimization problem. Thus, given an instance of $\mathcal{P}$ (with a probability distribution for each item), $\mathtt{ALG}$ finds a policy of expected cost at most α times the optimum. We also assume that $\mathcal{P}$ is up-monotone; see Definition I.1. The online framework for down-monotone problems and maximization problems is almost identical: the changes are explained in the full version [52].

Our online algorithm is based on the principle of *optimism in the face of uncertainty*, and is very simple to describe. At each time step t, we construct an "optimistic" empirical distribution $\mathbf{E}^t$ that stochastically dominates the true (unknown) distribution $\mathbf{D}$. Then, we run the offline algorithm $\mathtt{ALG}$ on this empirical distribution $\mathbf{E}^t$ to obtain policy σ^t, which is the online policy at time t.

Given i.i.d. samples of any random variable, we show that it is possible to compute a stochastically dominating empirical distribution that has a small total-variation distance from the true distribution.

Theorem II.1. *There is an algorithm $\mathtt{EmpStocDom}$ that, given m i.i.d. samples from a distribution $\mathcal{D}$ with finite support-size k, and parameter $\delta > 0$, computes a distribution $\mathcal{E}$ that satisfies the following properties with probability at least $1 - \delta$:*

- *$\mathcal{E}$ stochastically dominates $\mathcal{D}$, and*
- *the total-variation distance $\mathtt{TV}(\mathcal{E}, \mathcal{D}) < k\sqrt{\frac{\log(2k/\delta)}{2m}}$.*

The algorithm $\mathtt{EmpStocDom}$ and proof of this result are deferred to the full version of the paper [52]. The main idea is to "shift" some probability mass in the usual empirical distribution from low to high values. Algorithm 1 describes our online framework, which uses algorithm $\mathtt{EmpStocDom}$.

Algorithm 1 Online-to-Offline framework

1: **for** $t = 1, \ldots T$ **do**
2: for each item $i \in [n]$, obtain distribution $\mathcal{E}_i^t$ by running $\mathtt{EmpStocDom}$ with $\delta = \frac{2}{(nT)^3}$ on all samples of r.v. X_i observed so far.
3: obtain policy σ^t by running the offline algorithm $\mathtt{ALG}$ on distribution $\mathbf{E}^t = \{\mathcal{E}_i^t\}_{i=1}^n$.
4: run policy σ^t on realization $\mathbf{x}^t$ and observe the probed items $S(\sigma^t, \mathbf{x}^t)$.

We now analyze Algorithm 1, and prove Theorem I.1. The basic idea in the analysis is that as we get more and more samples with increasing t, the total variation distance between $\mathbf{E}^t$ and $\mathbf{D}$ reduces, and the policy $\texttt{ALG}(\mathbf{E}^t)$ will become closer and closer to the optimal policy.

We will assume, without loss of generality, that we have observed at least one sample from each random variable. This can be ensured by adding n special time periods and choosing for each time $i \in [n]$, a policy σ^i that first probes item i; note that this contributes at most $n \cdot f_{max}$ to the total regret. Let N_j^t be the number of times r.v. X_j has been sampled before time t. By our assumption, $N_j^1 = 1$ for all $j \in [n]$. Note that N_j^t is a random variable and depends on the history $\mathbf{h}^{t-1} = \left(\mathbf{x}^1, \cdots \mathbf{x}^{t-1}\right)$. When needed, we will use $N_j^t(\mathbf{h}^{t-1})$ to make this dependence explicit. The following result follows directly from Theorem II.1 (proof deferred to the full version).

Lemma II.2. *With probability at least $1 - \frac{1}{nT}$, we have $\mathcal{E}_j^t$ stochastically dominates $\mathcal{D}_j$ and*

$$\texttt{TV}\left(\mathcal{E}_j^t, \mathcal{D}_j\right) < k \cdot \sqrt{\frac{3\log(knT)}{2N_j^t}},$$

for all $j \in [n]$ and $t \in [T]$.

Let G denote the *"good" event* corresponding to the condition in Lemma II.2 holding for all j and t. First, we complete the proof assuming that G holds (this assumption is removed later).

We now state a crucial "stability" property for the stochastic problem $\mathcal{P}$.

Lemma II.3 (Stability lemma)**.** *Consider a stochastic problem that is up-monotone. Suppose that $\mathbf{E} = \{\mathcal{E}_i\}_{i=1}^n$ and $\mathbf{D} = \{\mathcal{D}_i\}_{i=1}^n$ are product distributions such that $\mathbf{D} \preceq_{\mathsf{SD}} \mathbf{E}$ and $\texttt{TV}(\mathcal{E}_i, \mathcal{D}_i) \le \epsilon_i$ for each $i \in [n]$. If σ is the policy returned by $\texttt{ALG}(\mathbf{E})$ and σ^* is an optimal policy under $\mathbf{D}$, then:*

$$\begin{aligned} f(\sigma) - \alpha \cdot f(\sigma^*) &= \mathbb{E}_{\mathbf{x}\sim\mathbf{D}}\left[f(\sigma, \mathbf{x}) - \alpha \cdot f(\sigma^*, \mathbf{x})\right] \\ &\le f_{\max} \sum_{i=1}^{n} Q_i(\sigma) \cdot \epsilon_i \,, \end{aligned} \tag{4}$$

where $Q_i(\sigma)$ denotes the probability that item i is probed by policy σ under distribution $\mathbf{D}$.

This lemma relies on the monotonicity assumption, and is proved in §II-A. We now complete the proof of Theorem I.1 using Lemma II.3.

Bounding the regret as sum over time t. To bound the overall regret, it suffices to bound the regret at each time $t \in [T]$, defined as:

$$\begin{aligned} R^t(\mathbf{h}^{t-1}) &:= \mathbb{E}_{\mathbf{x}^t}\left[f(\sigma^t, \mathbf{x}^t) - \alpha \cdot f(\sigma^*, \mathbf{x}^t)\right] \\ &= f(\sigma^t) - \alpha \cdot f(\sigma^*), \end{aligned}$$

where $\mathbf{h}^{t-1}$ is the history at time t, policy $\sigma^t = \texttt{ALG}(\mathbf{E}^t)$ and policy σ^* is the optimal policy for the stochastic problem instance (under the true distribution $\mathbf{D}$), and policy σ^t is the policy used by our online algorithm at time t. By (3), the overall regret is $\alpha\text{-}R(T) = \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\left[R^t(\mathbf{h}^{t-1})\right]$. For each time $t \in [T]$ and history $\mathbf{h}^{t-1}$, we apply Lemma II.3 with distributions $\mathbf{E}^t(\mathbf{h}^{t-1})$ and $\mathbf{D}$, and parameters $\epsilon_i^t(\mathbf{h}^{t-1}) = k \cdot \sqrt{\frac{3\log(knT)}{2N_i^t(\mathbf{h}^{t-1})}}$ for all $i \in [n]$. Note that $\texttt{TV}(\mathcal{E}_i^t, \mathcal{D}_i) \le \epsilon_i^t(\mathbf{h}^{t-1})$ and $\mathcal{D}_i \preceq_{\mathsf{SD}} \mathcal{E}_i$ for each $i \in [n]$ because we assumed the good event G. Hence, Lemma II.3 implies that:

$$R^t(\mathbf{h}^{t-1}) \le f_{\max} \sum_{i=1}^{n} Q_i(\sigma^t) \cdot \epsilon_i^t(\mathbf{h}^{t-1}). \tag{5}$$

We note that R^t, σ^t, N^t and ϵ^t all depend on the history $\mathbf{h}^{t-1}$; to keep notation simple, we drop the explicit dependence on $\mathbf{h}^{t-1}$. Using the regret definition and (5),

$$\begin{aligned} \alpha\text{-}R(T) &\le f_{\max} \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\left[\sum_{i=1}^{n} Q_i(\sigma^t) \cdot \epsilon_i^t\right] \\ &= k f_{\max}\sqrt{1.5\log(knT)} \cdot \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\left[\sum_{i=1}^{n} \frac{Q_i(\sigma^t)}{\sqrt{N_i^t}}\right]. \end{aligned}$$

It now suffices to show

$$\sum_{t=1}^{T}\sum_{i=1}^{n} \mathbb{E}_{\mathbf{h}^{t-1}}\left[\frac{Q_i(\sigma^t)}{\sqrt{N_i^t}}\right] \le 2n\sqrt{T}. \tag{6}$$

Indeed, combining (6) with the above bound on regret, we get $\alpha\text{-}R(T) \le knf_{max}\sqrt{6T\log(knT)}$, which completes the proof of Theorem I.1 (assuming event G).

Proving (6) **as sum over decision paths.** We refer to the full history $\mathbf{h}^T = \left(\mathbf{x}^1, \cdots, \mathbf{x}^T\right)$ of the algorithm as its *decision path*. The main idea in this proof is to view the left-hand-side in (6) as a sum over decision paths rather than a sum over time. To this end, define, for all $i \in [n]$:

$$Z_i(\mathbf{h}^T) := \sum_{t=1}^{T} \frac{\mathbf{I}\left[\text{item } i \text{ probed by } \sigma^t(\mathbf{h}^{t-1})\right]}{\sqrt{N_i^t(\mathbf{h}^{t-1})}}. \tag{7}$$

Above, $\mathbf{I}$ is the indicator function. Also, let $Z(\mathbf{h}^T) := \sum_{i=1}^{n} Z_i(\mathbf{h}^T)$. By linearity of expectation,

$$\begin{aligned} \mathbb{E}_{\mathbf{h}^T}\left[Z_i(\mathbf{h}^T)\right] &= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^T}\left[\frac{\mathbf{I}\left[i \text{ probed by } \sigma^t(\mathbf{h}^{t-1})\right]}{\sqrt{N_i^t(\mathbf{h}^{t-1})}}\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1},\mathbf{x}^t}\left[\frac{\mathbf{I}\left[i \text{ probed by } \sigma^t(\mathbf{h}^{t-1})\right]}{\sqrt{N_i^t(\mathbf{h}^{t-1})}}\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}}\left[\frac{1}{\sqrt{N_i^t}} \cdot \mathbf{P}_{\mathbf{x}^t}[i \text{ probed by } \sigma^t]\right] \end{aligned}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{\mathbf{h}^{t-1}} \left[\frac{Q_i(\sigma^t)}{\sqrt{N_i^t}} \right], \tag{8}$$

where the second equality uses the fact that event $\{i$ probed by $\sigma^t\}$ only depends on $\mathbf{h}^t = (\mathbf{h}^{t-1}, \mathbf{x}^t)$, the third equality uses the fact that N_i^t only depends on $\mathbf{h}^{t-1}$ and that $\mathbf{x}^t$ is independent of $\mathbf{h}^{t-1}$, and the last equality is by the definition of $Q_i(\sigma^t)$ and the fact that $\mathbf{x}^t \sim \mathbf{D}$. Using (8) and adding over $i \in [n]$, we get

$$\sum_{t=1}^{T} \sum_{i=1}^{n} \mathbb{E}_{\mathbf{h}^{t-1}} \left[\frac{Q_i(\sigma^t)}{\sqrt{N_i^t}} \right] = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{h}^T} \left[Z_i(\mathbf{h}^T) \right] = \mathbb{E}_{\mathbf{h}^T}[Z(\mathbf{h}^T)].$$

Therefore, proving (6) is equivalent to:

$$\mathbb{E}_{\mathbf{h}^T}[Z(\mathbf{h}^T)] = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{h}^T}[Z_i(\mathbf{h}^T)] \leq 2n\sqrt{T} \tag{9}$$

We now prove (9) by showing that $\mathbb{E}_{\mathbf{h}^T}[Z_i(\mathbf{h}^T)] \leq 2\sqrt{T}$ for each $i \in [n]$. Indeed,

$$Z_i(\mathbf{h}^T) = \sum_{t=1}^{T} \frac{\mathbf{I}\left[i \text{ probed by } \sigma^t(\mathbf{h}^{t-1})\right]}{\sqrt{N_i^t(\mathbf{h}^{t-1})}} \leq \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T}.$$

The first inequality uses the fact that $N_i^t(\mathbf{h}^{t-1})$ equals the number of probes of item i in the first $t-1$ time steps: so $N_i^{t+1} = N_i^t + 1$ whenever item i is probed by $\sigma^t(\mathbf{h}^{t-1})$. This completes the proof of (9) and hence (6).

Removing the "good" event assumption. In the analysis above, we assumed that event G holds in (5). We now modify (5) as follows (which holds irrespective of G).

$$R^t(\mathbf{h}^{t-1}) \leq k f_{\max} \sum_{i=1}^{n} Q_i(\sigma^t) \cdot \epsilon_i^t(\mathbf{h}^{t-1}) + f_{max} \cdot \mathbf{I}[\overline{G}].$$

We used the fact that the maximum 1-step regret is f_{max}. Combined with the previous analysis (which handles the first term), we have

$$\begin{aligned} \alpha\text{-}R(T) &= \sum_{t=1}^{T} \mathbb{E}[R^t(\mathbf{h}^{t-1})] \\ &\leq kn f_{max} \sqrt{6T \log(knT)} + T f_{max} \cdot \mathbf{P}[\overline{G}] \\ &\leq kn f_{max} \sqrt{6T \log(knT)} + f_{max}. \end{aligned}$$

The last inequality uses Lemma II.2. This completes the proof of Theorem I.1.

A. Proving the Stability Lemma

We now prove Lemma II.3. Recall that there are two product distributions $\mathbf{E} = \{\mathcal{E}_i\}_{i=1}^n$ and $\mathbf{D} = \{\mathcal{D}_i\}_{i=1}^n$ with $\mathbf{D} \preceq_{\mathsf{SD}} \mathbf{E}$ and $\mathtt{TV}(\mathcal{E}_i, \mathcal{D}_i) \leq \epsilon_i$ for each $i \in [n]$. Let $\sigma = \mathtt{ALG}(\mathbf{E})$ be the policy returned by the (offline) α-approximation algorithm, and σ^* be an optimal policy under distribution $\mathbf{D}$. We want to upper bound $f(\sigma) - \alpha \cdot f(\sigma^*)$. Let $\mathcal{C}$ denote the set of all feasible policies to the given instance. Then,

$$\mathtt{OPT}(\mathbf{D}) = \min_{\tau \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim \mathbf{D}}[f(\tau, \mathbf{x})], \quad \text{and}$$
$$\mathtt{OPT}(\mathbf{E}) = \min_{\tau \in \mathcal{C}} \mathbb{E}_{\mathbf{x} \sim \mathbf{E}}[f(\tau, \mathbf{x})].$$

We use $f(\tau|\mathbf{U}) := \mathbb{E}_{\mathbf{x} \sim \mathbf{U}}[f(\tau, \mathbf{x})]$ to denote the expected cost of any policy τ when the r.v.s have product distribution $\mathbf{U}$. Note that $f(\tau) = f(\tau|\mathbf{D})$ where $\mathbf{D}$ is the true distribution. We now have:

$$\begin{aligned} & f(\sigma) - \alpha \cdot f(\sigma^*) = f(\sigma|\mathbf{D}) - \alpha \cdot f(\sigma^*|\mathbf{D}) \\ &= f(\sigma|\mathbf{D}) - f(\sigma|\mathbf{E}) + f(\sigma|\mathbf{E}) - \alpha \cdot f(\sigma^*|\mathbf{D}) \\ &\leq f(\sigma|\mathbf{D}) - f(\sigma|\mathbf{E}) + \alpha \cdot (\mathtt{OPT}(\mathbf{E}) - f(\sigma^*|\mathbf{D})) \quad (10) \\ &= f(\sigma|\mathbf{D}) - f(\sigma|\mathbf{E}) + \alpha \cdot (\mathtt{OPT}(\mathbf{E}) - \mathtt{OPT}(\mathbf{D})) \\ &\leq \ f(\sigma|\mathbf{D}) - f(\sigma|\mathbf{E}). \quad (11) \end{aligned}$$

Inequality (10) uses the fact that σ is an α-approximate policy to the instance with distribution $\mathbf{E}$. In (11), the equality uses the fact that σ^* is an optimal policy for the instance with distribution $\mathbf{D}$, and the final inequality is by up-monotonicity (Definition I.1) and $\mathbf{D} \preceq_{\mathsf{SD}} \mathbf{E}$.

a) Equivalent view of adaptive policy σ: Let N denote the number of nodes in decision tree σ (this may be exponential, but it is only used in the analysis). Note that there is a *partial ordering* of the nodes of σ based on ancestor-descendent relationships in the tree. We index the nodes in σ from 1 to N according to this partial order, so that $u < v$ for any node v that is a child of node u. Recall that each node $v \in \sigma$ is labeled by one of the n items. Note that the same item $i \in [n]$ may label multiple nodes of σ; however, in any policy execution (i.e., root-leaf path in σ) we will encounter at most one node labeled by item i. Hence, we can equivalently view policy σ as having an item X_v with *independent* distribution $\mathcal{D}_v$ at each node $v \in \sigma$. (This involves making several independent copies of each item, which does not affect the policy execution as at most one copy of each item is seen on any root-leaf path.)

b) Bounding $|f(\sigma|\mathbf{D}) - f(\sigma|\mathbf{E})|$: Based on the above view of σ, we use $\mathcal{D}_v$ (resp. $\mathcal{E}_v$) to denote the independent distribution at each node $v \in \sigma$ under the joint distribution $\mathbf{D}$ (resp. $\mathbf{E}$). Using the above indexing of the nodes in σ, we define the following *hybrid* product distributions: for each $v \in [N]$ let $\mathbf{H}^v = \mathcal{D}_1 \times \cdots \times \mathcal{D}_v \times \mathcal{E}_{v+1} \times \cdots \times \mathcal{E}_N$. Observe that $f(\sigma \mid \mathbf{H}^N) = f(\sigma \mid \mathbf{D})$ and $f(\sigma \mid \mathbf{H}^0) = f(\sigma \mid \mathbf{E})$. Using a telescoping sum, we can write:

$$f(\sigma \mid \mathbf{D}) - f(\sigma \mid \mathbf{E}) = \sum_{v=1}^{N} f(\sigma \mid \mathbf{H}^v) - f(\sigma \mid \mathbf{H}^{v-1}).$$

Crucially, we now show that for every node $v \in [N]$,

$$| f(\sigma \mid \mathbf{H}^v) - f(\sigma \mid \mathbf{H}^{v-1}) | \leq f_{\max} \cdot Q_v(\sigma) \cdot \epsilon_v, \quad (12)$$

where $Q_v(\sigma)$ is the probability that σ reaches node v under distribution $\mathbf{D}$ and $\epsilon_v = \mathtt{TV}(\mathcal{D}_v, \mathcal{E}_v)$. We first complete the proof of the lemma using (12). Note that for any node v labeled by item $i \in [n]$, we have $\epsilon_v = \mathtt{TV}(\mathcal{D}_i, \mathcal{E}_i) \leq \epsilon_i$. We get

$$\begin{aligned}
&|f(\sigma \mid \mathbf{D}) - f(\sigma \mid \mathbf{E})| \\
&\quad \leq \sum_{v=1}^{N} |f(\sigma \mid \mathbf{H}^v) - f(\sigma \mid \mathbf{H}^{v-1})| \\
&\quad \leq f_{max} \cdot \sum_{v=1}^{N} Q_v(\sigma) \cdot \epsilon_v \\
&\quad = f_{max} \cdot \sum_{i=1}^{n} \epsilon_i \sum_{v \in [N]:\ \text{labeled } i} Q_v(\sigma) \\
&\quad = f_{max} \cdot \sum_{i=1}^{n} \epsilon_i \cdot Q_i(\sigma). \qquad (13)
\end{aligned}$$

The last equality uses the fact that the probability $Q_i(\sigma)$ of probing i equals the probability of reaching some node $v \in [N]$ labeled i.

Towards proving (12), we introduce the following notation. Let $Q_v(\sigma \mid \mathbf{U})$ denote the probability that σ reaches node v under any distribution $\mathbf{U}$. Note that the event "σ reaches node v" only depends on the random realizations at the ancestor nodes of v, which are all contained in $\{u \in [N] : u < v\}$ (by our indexing of nodes). Hence,

$$\begin{aligned}
Q_v(\sigma \mid \mathbf{H}^v) &= Q_v(\sigma \mid \mathbf{H}^{v-1}) \\
&= Q_v(\sigma \mid \mathbf{D}) = Q_v(\sigma), \qquad (14)
\end{aligned}$$

because each node $\{u \in [N] : u < v\}$ has the same distribution ($\mathcal{D}_u$) under $\mathbf{D}$, $\mathbf{H}^v$ and $\mathbf{H}^{v-1}$.

Below, let $\mathcal{R}_v$ denote the event that σ reaches node v. Using the fact that each node $w \in [N] \setminus \{v\}$ has the same distribution (either $\mathcal{D}_w$ or $\mathcal{E}_w$) under *both* $\mathbf{H}^v$ and $\mathbf{H}^{v-1}$, we get:

$$\mathbb{E}_{\mathbf{x} \sim \mathbf{H}^v} [f(\sigma, \mathbf{x}) \mid \neg \mathcal{R}_v] - \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^{v-1}} [f(\sigma, \mathbf{x}) \mid \neg \mathcal{R}_v] = 0. \quad (15)$$

We now bound the difference in expectation conditional on event $\mathcal{R}_v$. Let ρ denote the subtree of σ rooted at v (including v itself). We also use ρ to denote the set of nodes in this subtree. Let L denote all leaf nodes in subtree ρ, and for each $\ell \in L$ let f_ℓ be the function value accrued at leaf node ℓ. For any realization $\{x_w : w \in \rho\}$, we use $\ell(\{x_w : w \in \rho\})$ to denote the (unique) leaf that is reached when subtree ρ is executed under this realization: this corresponds to following the branch labeled x_w out of each node $w \in \rho$ (starting from node v). We now define function h that maps any realization $\{x_w : w \in \rho\}$ to the function value f_k at the leaf $k = \ell(\{x_w : w \in \rho\})$. By our assumption on the objective function of the stochastic problem, h is bounded between 0 and f_{max}. We now define two product distributions on nodes of ρ:

$$\begin{aligned}
\mathbf{V}_1 &= \langle \mathcal{D}_v, \{\mathcal{E}_w : w \in \rho \setminus v\} \rangle \quad \text{and} \\
\mathbf{V}_2 &= \langle \mathcal{E}_v, \{\mathcal{E}_w : w \in \rho \setminus v\} \rangle.
\end{aligned}$$

Note that

$$\begin{aligned}
h(\mathbf{V}_1) &= \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^v} [f(\sigma, \mathbf{x}) \mid \mathcal{R}_v] \quad \text{and} \\
h(\mathbf{V}_2) &= \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^{v-1}} [f(\sigma, \mathbf{x}) \mid \mathcal{R}_v]
\end{aligned}$$

Applying Lemma I.3 to function h and product distributions $\mathbf{V}_1$ and $\mathbf{V}_2$,

$$\begin{aligned}
&\left| \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^v} [f(\sigma, \mathbf{x}) \mid \mathcal{R}_v] - \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^{v-1}} [f(\sigma, \mathbf{x}) \mid \mathcal{R}_v] \right| \\
&\leq f_{max} \cdot \mathtt{TV}(\mathbf{V}_1, \mathbf{V}_2) \leq f_{max} \cdot \mathtt{TV}(\mathcal{D}_v, \mathcal{E}_v). \qquad (16)
\end{aligned}$$

The last inequality is by Lemma I.2 and the fact that $\mathbf{V}_1$ and $\mathbf{V}_2$ only differ at node v.

We now combine the conditional expectations from (15) and (16). Using (14), the probability of event $\mathcal{R}_v$ under *both* distributions $\mathbf{H}^{v-1}$ and $\mathbf{H}^v$ is the same, which equals $Q_v(\sigma)$. So,

$$\begin{aligned}
&| f(\sigma \mid \mathbf{H}^v) - f(\sigma \mid \mathbf{H}^{v-1}) | \\
&= \left| \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^v} [f(\sigma, \mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim \mathbf{H}^{v-1}} [f(\sigma, \mathbf{x})] \right| \\
&= Q_v(\sigma) \cdot \left| \mathbb{E}_{\mathbf{H}^v} [f(\sigma, \mathbf{x}) \mid \mathcal{R}_v] - \mathbb{E}_{\mathbf{H}^{v-1}} [f(\sigma, \mathbf{x}) \mid \mathcal{R}_v] \right| \\
&\leq Q_v(\sigma) \cdot f_{max} \cdot \mathtt{TV}(\mathcal{D}_v, \mathcal{E}_v) = f_{max} \cdot Q_v(\sigma) \cdot \epsilon_v.
\end{aligned}$$

This completes the proof of (12) and Lemma II.3.

The algorithmic framework described so far assumed that the stochastic problem has a minimization objective and is up-monotone. The changes required to handle down-monotone problems and problems with a maximization objective are described in the full version.

B. Framework for Continuous Distributions

The algorithmic framework described above assumed that the random variables are discrete. Here, we show how to handle stochastic problems with arbitrary (continuous or discrete) r.v.s as long as the policies are based on a finite number of thresholds. We first define a natural discretization of continuous r.v.s.

Definition II.1 (Thresholded r.v.)**.** *For any $[0, U]$-bounded r.v. X (possibly continuous) and list $\mathbf{b} = \langle b_1, \cdots, b_{k-1} \rangle$ of increasing threshold values, let $\theta(\mathsf{X}, \mathbf{b})$ be the r.v. such that*

$$\theta(\mathsf{X}, \mathbf{b}) = i \text{ if } b_{i-1} \leq \mathsf{X} < b_i, \qquad \forall i \in [k],$$

where we use $b_0 := 0$ and $b_k := U$. This corresponds to thresholding X according to $\mathbf{b}$.

Note that $\theta(\mathsf{X}, \mathbf{b})$ is a discrete random variable with support size k, even if X is continuous. We will assume that the α-approximation algorithm ALG for the stochastic problem $\mathcal{P}$ always finds a *k-threshold policy*, defined as follows. Such a policy is given by a decision tree σ, where each node is labeled by an item j to probe and list $\mathbf{b} = \langle b_1, \cdots, b_{k-1} \rangle$ of thresholds, such that branch $i \in \{1, \cdots k\}$ is followed when $\mathsf{X}_j \in [b_{i-1}, b_i)$. This limits the number of branches out of each node to at most k, whereas an arbitrary policy may have an unbounded number of branches (corresponding to each possible outcome). We emphasize that the thresholds at different nodes may be completely different. This restriction allows us to analyze policies involving continuous r.v.s in the same manner as before (for discrete r.v.s). In particular, all other definitions related to the policy (terminal/leaf nodes, policy execution, cost etc) remain the same as for discrete r.v.s. Moreover, for problems like Prophet inequality and Pandora's Box, there are optimal 1-threshold policies: so this restriction is without loss of generality for these problems.

Our main result is the following (the details are left to the full version).

Theorem II.4. *Suppose that stochastic problem $\mathcal{P}$ has an α-approximation algorithm via k-threshold policies, and $\mathcal{P}$ is either up-monotone or down-monotone. Then, there is a polynomial-time semi-bandit learning algorithm for $\mathcal{P}$ with α-regret $O(knf_{max}\sqrt{T\log(knT)})$. Here, n is the number of items, f_{max} is the maximal value of the objective function and T is the number of periods.*

III. Applications

In this section, we show that several stochastic optimization problems are covered by our framework, resulting in $\sqrt{T \log T}$ regret online learning algorithm for all these problems. In each of these problems, the distributions of the random parameters are unknown to the online algorithm; all other (deterministic) parameters are known.

a) Series Testing: We start with a simple problem: there are n components, where each component i is "working" independently with some known probability p_i. To determine if any component $i \in [n]$ is working, we need to perform a test, which costs c_i. All n components must be working for the system to be functional. The goal is to test components sequentially to determine whether/not the system is functional, at the minimum expected cost. Note that testing stops once a failed component is found: so we do not observe all the outcomes and only have semi-bandit feedback. It is easy to show that this problem is up-monotone. It is well-known that the natural greedy policy achieves the optimal cost [16]. So, using Theorem I.1 with $k = 2$ (all r.v.s are binary), we obtain a polynomial time online learning algorithm for series testing having 1-regret $O(nC\sqrt{T \log T})$ where $C = \sum_{i=1}^{n} c_i$ is the total cost.

b) Prophet Inequality: The Prophet Inequality [3], [53] is a fundamental problem in optimal stopping, which has also been used extensively in algorithmic game theory. The input consists of n rewards which arrive in a given fixed sequence, say $\mathsf{X}_1, \cdots \mathsf{X}_n$. Each reward X_i is drawn independently from a known distribution $\mathcal{D}_i$. We are interested in online policies, that upon observing each reward, selects or discards it immediately. The policy can select at most one reward, and it terminates right after making a selection (without observing any future reward). Note that we have semi-bandit feedback because only some of the rewards are observed in any policy execution. The goal is to maximize the expected selected reward. The classical results obtain a $\frac{1}{2}$-approximate policy relative to the "clairvoyant" optimal value $\mathbb{E}[\max_{i=1}^{n} \mathsf{X}_i]$; there are also instances where no policy can achieve a better approximation to this benchmark. Here, we will compare to a more realistic non-clairvoyant benchmark: the optimal policy which is also constrained to make selection decisions in the given order (same as an algorithm). It is known that there is an optimal threshold-based policy: given thresholds $\{\tau_i\}_{i=1}^{n}$, the policy selects i if and only if $\mathsf{X}_i > \tau_i$. This is a 1-threshold policy, as defined in §II-B. Moreover, the prophet inequality problem is strongly monotone (see Lemma 28 in [20]), which implies that it is down-monotone. Using Theorem II.4, we get a polynomial time online learning algorithm for the the prophet inequality problem with unknown distributions that has 1-regret $O(nU\sqrt{T \log T})$ where all r.v.s are $[0, U]$ bounded. This improves upon the $O(n^3 U\sqrt{T} \log T)$ bound from [19], although the previous result holds in the stronger bandit-feedback model. We note that there are other learning-based results [54], [55] based on limited number of samples, that imply $\frac{1}{2}$-regret algorithms by comparing to the clairvoyant benchmark. Note that our guarantee and that of [19] are stronger because they do not incur any multiplicative approximation factor.

c) Combinatorial Prophet Inequalities: The basic prophet inequality concept has also been extended to settings where there is some combinatorial feasibility constraint on the selected items. Here, we have n items with reward $\mathsf{X}_i \sim \mathcal{D}_i$ for each $i \in [n]$. In addition, there is a *downward-closed*[4] set family $\mathcal{F} \subseteq 2^{[n]}$ that represents a generic feasibility constraint; the selected

[4]Set family $\mathcal{F}$ is downward-closed if, for any $S \in \mathcal{F}$ we have $S' \in \mathcal{F}$ for every $S' \subseteq S$.

subset must be in $\mathcal{F}$. The n items arrive in a given fixed sequence. When item i arrives, if $S \cup \{i\} \notin \mathcal{F}$ where S is the set of previously selected items then item i is not considered for selection (and we do not observe X_i); otherwise, the policy observes the value of X_i and selects/discards item i. Again, note that we only have semi-bandit feedback because only a subset of items is observed by the policy. The performance of an online policy is compared to the clairvoyant optimum $\texttt{OPT}^* = \mathbb{E}\left[\max_{S\in\mathcal{F}} \sum_{i\in S} \mathsf{X}_i\right]$. Many specific problems can be modeled in this manner:

- When $\mathcal{F} = \{S : |S| \leq 1\}$, we get the classic prophet inequality, which has a $\frac{1}{2}$ approximation.
- When $\mathcal{F} = \{S : |S| \leq k\}$, there is a $1 - \frac{1}{\sqrt{k+3}}$ approximation [56], [57].
- When $\mathcal{F}$ corresponds to independent sets in a matroid, we obtain the matroid prophet inequality where again a $\frac{1}{2}$ approximation is known [17].
- When $\mathcal{F}$ is given by the intersection of p matroids, a $\frac{1}{e(p+1)}$ approximation is known [5].
- When $\mathcal{F}$ is given by matchings in a graph, a 0.337 approximation is known [58].

The online policies in all these results are 1-threshold policies (as in §II-B). For the monotone property, note that we are comparing to the clairvoyant optimum $\texttt{OPT}^*$ (not the optimal policy). So, it suffices to prove down-monotonicity for $\texttt{OPT}^*$, which is immediate by stochastic dominance. Therefore, using Theorem II.4, we obtain α-regret $O(nU\sqrt{T\log T})$ for all the above combinatorial prophet inequalities where α is the best (offline) approximation ratio; we assume that the r.v.s are $[0, U]$ bounded.

Some approximate regret bounds can also be obtained from previous work on single-sample prophet inequalitites [54], [55], [59]. While the regret bounds via this approach are better (just $O(n)$), they need to compare to approximation ratios that are often much worse than the usual (known distribution) setting. For example, no constant-factor approximation is known for single-sample matroid prophet inequality, whereas our results imply $\frac{1}{2}$-regret of $\sqrt{T\log T}$ for general matroids, matching the approximation ratio from [17].

d) Pandora's Box.: In this problem [1], we are given distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ such that r.v. $\mathsf{X}_i \sim \mathcal{D}_i$. The realization of X_i can be ascertained by paying a known inspection cost c_i. Now, the goal is to find a policy to (adaptively) inspect a subset $S \subseteq [n]$ of the random variables to maximize $\mathbb{E}\left[\max_{i\in S} \mathsf{X}_i - \sum_{i\in S} c_i\right]$. Note that any policy only inspects a subset of items and we only receive feedback from these items, which corresponds to semi-bandit feedback. [1] obtained an optimal policy based on the "reservation value" for each item and probing items according to this value until the reward for an item exceeds all remaining reservation values. The reservation value r_i for an item i is such that $\mathbb{E}[(\mathsf{X}_i - r_i)_+] = c_i$. We note that this optimal policy is 1-threshold, according to the definition in §II-B.

For the online setting, we assume that the r.v.s X_i are $[0, U]$ bounded; the distributions may be discrete or continuous. It was shown in [20] (see Lemma 31 in that paper) that the Pandora's box problem is strongly monotone, which implies that it is down-monotone. Combined with Theorem II.4, we get a polynomial time online learning algorithm for the Pandora's box problem with unknown distributions that has 1-regret $O(n(C+U)\sqrt{T\log T})$ where $C = \sum_{i\in[n]} c_i$ is the total cost. Our regret bound improves upon the $O(n^{4.5}(C + U)\sqrt{T}\log T)$ bound from [19], although the previous result holds in the stronger bandit-feedback model.

e) Variants of Pandora's Box.: Our framework also applies to more general versions of Pandora's problem that have been studied in prior work. We mention two such variants here.

In Pandora's box with *order constraints*, in addition to the n r.v.s, there are precedence constraints that enforce that any r.v. X_i may be selected only after all its predecessors have been selected. Although the original policy [1] does not apply to this extension, [25] obtained a different optimal policy when the precedence constraints form a directed tree: this policy is also a 1-threshold policy. Hence, Theorem II.4 implies an online learning algorithm for Pandora's box with tree order constraints having 1-regret $O(n(C+U)\sqrt{T\log T})$, where $C = \sum_{i\in[n]} c_i$ is the total cost and U is the bound on the r.v.s.

In the matroid Pandora's box problem [2], [24], in addition to the n r.v.s $\{\mathsf{X}_i\}_{i=1}^n$, there is a matroid $\mathcal{M}$ with groundset $[n]$. The goal is to inspect a subset $S \subseteq [n]$ of r.v.s and select a subset $B \subseteq S$ such that B is independent in matroid $\mathcal{M}$. The objective is to maximize $\mathbb{E}\left[\sum_{j\in B} \mathsf{X}_j - \sum_{i\in S} c_i\right]$, the difference between the total selected value and inspection cost. We recover the original Pandora's box problem when $\mathcal{M}$ is a rank-1 uniform matroid. There is an optimal 1-threshold policy known for this variant [2]. This policy is also non-adaptive. So, by Theorem II.4, we obtain an online learning algorithm for matroid Pandora's box having 1-regret $O(n(C+U)\sqrt{T\log T})$, where $C = \sum_{i\in[n]} c_i$ is the total cost and U is the bound on the r.v.s.

f) Stochastic Knapsack: This is a classic problem in stochastic optimization, which was introduced in [13] and has been studied extensively [18], [60], [61]. There are n items with deterministic rewards $\{r_i\}_{i=1}^n$ and random costs $\{C_i \sim \mathcal{D}_i\}_{i=1}^n$. The realized cost C_i of item i is only known after selecting it. Given a knapsack budget B, a policy selects items sequentially until the total cost exceeds B. The objective is to maximize the expected total reward from items that fit in the knapsack. (If there is an item that overflows the budget then it does

not contribute to the objective.) Note that only some subset of items is selected by a policy, and we only observe those costs as feedback. In Lemma III.1 below, we show that this problem is down-monotone (assuming that the costs are discrete r.v.s).

There is a *non-adaptive* $\frac{1}{4}$-approximation algorithm for stochastic knapsack [13], which probes items in a fixed sequence until the budget is exhausted; so it is a 1-threshold policy. Theorem II.4 then implies a polynomial time online learning algorithm for stochastic knapsack having $\frac{1}{4}$-regret of $O(nR\sqrt{T\log T})$ where $R=\sum_{i=1}^{n} r_i$ is the total reward. There is also a better *adaptive* algorithm for stochastic knapsack, which is a $(\frac{1}{2}-\epsilon)$ approximation (for any constant $\epsilon>0$) [18]. Theorem I.1 then implies an online learning algorithm with $(\frac{1}{2}-\epsilon)$-regret of $O(nkR\sqrt{T\log(kT)})$ where k is the maximum support size.

Our results also apply to the more general *stochastic orienteering* problem, where items are located at vertices in a metric space and we want to find a *path* with budget B on the total distance (from the edges in the path) plus cost (of the visited items). There is a non-adaptive $\Omega(\frac{1}{\log\log B})$ approximation algorithm for this problem [26], which is a 1-threshold policy. Here, we obtain $\Omega(\frac{1}{\log\log B})$-regret of $O(nR\sqrt{T\log T})$.

Lemma III.1. *The stochastic (maximum) knapsack problem is up-monotone.*

We defer the proof of this lemma to the full version of the paper.

g) Stochastic Matching and Probing: In the *stochastic matching* problem [14], [15], there is an undirected graph $G=(V,E)$ with edge-weights $\{w_e\}_{e\in E}$, edge-probabilities $\{p_e\}_{e\in E}$, and vertex bounds $\{t_v\}_{v\in V}$. Each edge e is *active* independently with probability e. However, the status (active/inactive) of an edge can only be determined by *probing* it. There is also a constraint on the set of probed edges: for any vertex v, the number of probed edges incident to v must be at most t_v. A solution/policy needs to to probe a subset of edges and select a matching M consisting of active edges. The objective is to maximize the expected weight of M. Finally, there is a "query commit" requirement that any probed edge which is active *must* be included in the selected matching M. Observe that in any policy execution, we only see the status of some subset of edges, which corresponds to semi-bandit feedback. This problem has been extensively studied, see e.g. [14], [15], [62], [63]. The current best approximation ratio is 0.5 for the unweighted case [62] and 0.382 for the weighted case [63]. We show in Lemma III.2 below (see full version for the proof) that the stochastic matching problem is down-monotone. So, Theorem I.1 with $k=2$ (as all r.v.s are binary) implies an online learning algorithm for stochastic matching having 0.382-regret of $O(nW\sqrt{T\log T})$ where $W=\sum_{e\in E} w_e$ is the total weight.

In fact, our result applies to the much more general *stochastic probing problem*, as defined in [9]. Here, we have a set E of stochastic items with weights $\{w_e\}_{e\in E}$ and probabilities $\{p_e\}_{e\in E}$. Each item is active independently with probability e, and this status can only be determined by probing e. We now have two downward-closed constraints: an *inner* constraint $\mathcal{F}_{in}$ and an *outer* constraint $\mathcal{F}_{out}$. We want to probe a set Q of items subject to the outer constraint (i.e. $Q\in\mathcal{F}_{out}$) and select a subset $S\subseteq Q$ of active (probed) items satisfying the inner constraint (i.e. $S\in\mathcal{F}_{in}$). The objective is to maximize the expected weight of the selected items S. We again have the query-commit requirement that any active probed item must be selected into the solution S. When both inner/outer constraints are k_{in} and k_{out} systems (see [9] for the definition of k-systems), there is a $\frac{1}{k_{in}+k_{out}}$ approximation for the unweighted case and an $\Omega(\frac{1}{(k_{in}+k_{out})^2})$ approximation for the weighted case [9]. When the k-systems are intersections of matroids, [10] gave an improved adaptive algorithm with approximation ratio $\frac{1}{k_{in}+k_{out}}$, even for the weighted case.

Lemma III.2. *The stochastic probing problem is down-monotone.*

APPENDIX A
SAMPLING-BASED ALGORITHMS

A stronger monotonicity condition is that of *strong monotonicity* [20], defined next.

Definition A.1 (Strong Monotonicity)**.** *A stochastic problem is* strongly up-monotone *if for any instance $\mathcal{I}$ and probability distributions $\mathbf{D}$ and $\mathbf{E}$ with $\mathbf{D}\preceq_{\mathsf{SD}}\mathbf{E}$, we have $\mathbb{E}_{\mathbf{x}\sim\mathbf{E}}[f(\sigma_{\mathbf{D}},\mathbf{x})]\leq \mathtt{OPT}_{\mathcal{I}}(\mathbf{D})$, where $\sigma_{\mathbf{D}}$ is the optimal policy for instance $\mathcal{I}$ under distribution $\mathbf{D}$. The problem is said to be* strongly down-monotone *if $\mathbb{E}_{\mathbf{x}\sim\mathbf{E}}[f(\sigma_{\mathbf{D}},\mathbf{x})]\geq \mathtt{OPT}_{\mathcal{I}}(\mathbf{D})$ under the same conditions as above.*

We note that strong monotonicity implies monotonicity. In prior work, [20] gave optimal sample complexity bounds for stochastic optimization problems that exhibit the strong monotonicity condition. They also proved this property for problems including Prophet inequality and Pandora's box. We make use of this property in §III, when we apply our result to these problems.

Below, we assume that $f_{max}=1$ by scaling. The sample-complexity result in [20] is:

Theorem A.1 (Theorem 17 [20])**.** *For any strongly monotone stochastic problem, suppose the number of*

samples is at least:

$$C \cdot \frac{n}{\epsilon^2} \log\left(\frac{n}{\epsilon}\right) \log\left(\frac{nT}{\epsilon}\right)$$

where $C > 1$ is a sufficiently large constant. Then, there is an algorithm that gets an ϵ-additive approximation to the optimum with probability at least $1 - \frac{1}{T^2}$.

Below, we discuss what regret bounds can be achieved via this sampling-based approach, in order to compare to our results. In the full-feedback model, one can obtain regret bounds of $\sqrt{nT \log T}$, which is nearly optimal. However, for *semi-bandit* feedback that we consider (and for bandit feedback), such a "reduction" from sample-complexity bounds only provides a sub-optimal $\widetilde{O}(T^{2/3})$ regret. Therefore, we need new ideas to get the optimal $\widetilde{O}(\sqrt{T})$ regret bound in the semi-bandit model, which we do in this paper.

Full-feedback. Recall that, in the full-feedback model, we get one sample from each r.v. in every time-step. We now describe the algorithm. The algorithm is straightforward: for each $t = 1, 2, \ldots T$, we use estimates from the prior $t-1$ time steps to obtain a policy to use for the t^{th} time-step. See Algorithm 2 for a formal description.

Algorithm 2 EXPLOIT

1: **Input:** one sample $\mathbf{x} = (x_1, \cdots, x_n) \sim \mathbf{D}$,
2: **for** $t = 1, 2, \ldots, T$ **do**
3: let $\widehat{\mathbf{D}}_{\mathbf{t}} \leftarrow$ frequentist estimate from previous $t-1$ time-steps
4: run policy $\texttt{OPT}(\widehat{\mathbf{D}}_{\mathbf{t}})$ for the t^{th} time-step

As a consequence of Theorem A.1, with probability at least $1 - \frac{1}{T^2}$, we have that $\texttt{OPT}(\widehat{\mathbf{D}}_{\mathbf{t}})$ is an ϵ_t-additive approximation to $\texttt{OPT}(\mathbf{D})$ where $\epsilon_t \leq \sqrt{\frac{Cn \log(nT)}{t}}$. (To keep calculation simple, we ignore the $\log \frac{1}{\epsilon}$ dependence in Theorem A.1; so we are actually assuming a slightly stronger sample-complexity bound.) By union bound, this is true for all $t = 1, \ldots, T$ with probability at least $1 - \frac{1}{T}$. We consider this to be a *good* event, G. Under this event, the total regret, say R_T, can be bounded as follows.

$$\mathbb{E}[R_T \mid G] \leq \sum_{t=1}^{T} \sqrt{\frac{Cn \log(nT)}{t}} = O\left(\sqrt{nT \log(nT)}\right).$$

By law of total expectation we have

$$\begin{aligned}\mathbb{E}[R_T] &\leq \mathbb{E}[R_T \mid G] \cdot \mathbf{P}(G) + \mathbb{E}[R_T \mid \overline{G}] \cdot \mathbf{P}(\overline{G}) \\ &\leq O\left(\sqrt{nT \log(nT)}\right) + T \cdot \frac{1}{T} \\ &= O\left(\sqrt{nT \log(nT)}\right).\end{aligned}$$

Semi-bandit feedback. A significant challenge in the semi-bandit feedback model arises from our lack of control over the r.v.s from which we get samples. One potential strategy to address this issue is to artificially generate samples for an item i by probing item i first in the algorithm's policy for that period. However, this approach comes with an inherent drawback – probing item i first may result in a poor policy, and so we suffer a high regret in such periods. The standard *explore-then-exploit* algorithm that first gets $T^{2/3}$ samples for each r.v., and then plays the optimal policy (for the empirical distribution) for the remaining time steps only achieves regret $\widetilde{O}(T^{2/3})$.

In the hope of obtaining improved regret guarantees, we consider a different algorithm where we pair periods of exploration with some periods of exploitation, wherein we leverage the learned distribution. To do this, we divide the time horizon T into meta periods, each of length $n + \beta n$ (we optimize β later). In meta-period h, we first gather n samples (one for each r.v.) and then use policy $\texttt{OPT}(\widehat{\mathbf{D}}_{\mathbf{h}})$ for βn time-steps. Here $\widehat{\mathbf{D}}_{\mathbf{h}}$ denotes the frequentist distribution obtained from the h samples (until meta-period h) of $\mathbf{D}$. Note that we have no control over the r.v.s probed by $\texttt{OPT}(\widehat{\mathbf{D}}_{\mathbf{h}})$, and thus cannot use these samples in our analysis.

By Theorem A.1, with probability at least $1 - \frac{1}{T^2}$, we have that $\texttt{OPT}(\widehat{\mathbf{D}}_{\mathbf{h}})$ is an ϵ_h-additive approximation to $\texttt{OPT}(\mathbf{D})$ where $\epsilon_h \leq \sqrt{\frac{Cn \log(nT)}{h}}$. By union bound, this is true for all meta-periods $h = 0, 1, \ldots,$ with probability at least $1 - \frac{1}{T}$ (since $h \leq T$). As before, we consider this to be a *good* event, G. Under this event, the regret in meta-period h is at most $n + \beta n \cdot \sqrt{\frac{Cn \log(nT)}{h}}$, and the total regret under G is at most

$$\begin{aligned}&nH + C\beta n^{3/2} \sqrt{\log(nT)} \sum_{h=1}^{H} \sqrt{\frac{1}{h}} \\ &\quad = nH + C\beta n^{3/2} \sqrt{H \log(nT)}\end{aligned}$$

where $H = \frac{T}{(1+\beta)n}$. The optimal choice for $\beta = O\left(\left(\frac{T}{n^2 \log T}\right)^{1/3}\right)$, and so we again end up with a regret of $O\left((nT)^{2/3} (\log T)^{1/3}\right)$. To summarize, we are not aware of any generic approach that reduces sample-complexity bounds to regret minimization in the semi-bandit have $o(T^{2/3})$ regret.

Remark 1. The sampling approach for semi-bandit feedback also applies to many problems (e.g., prophet inequality, Pandora's box and series testing) in the *bandit* feedback model. Basically, for each item i we need a policy where its objective corresponds to the value of r.v. X_i. So, we can directly get $\widetilde{O}(T^{2/3})$ regret for these problems even with bandit feedback.

Remark 2. We note that [20] also give (slightly worse) sample complexity bounds for a broader class of prob-

lems that need not satisfy any monotonicity property, but have a finite support-size k. Using the approaches described above, we can convert these sample-complexity guarantees to obtain $O\left(\sqrt{nkT\log(nT)}\right)$ regret under full-feedback and $O\left((nkT)^{2/3}(\log T)^{1/3}\right)$ regret under semi-bandit (and bandit) feedback.

REFERENCES

[1] M. L. Weitzman, "Optimal Search for the Best Alternative," *Econometrica*, vol. 47, no. 3, pp. 641–654, May 1979.

[2] S. Singla, "The price of information in combinatorial optimization," in *Proceedings of the twenty-ninth annual ACM-SIAM symposium on discrete algorithms*. SIAM, 2018, pp. 2523–2532.

[3] E. Samuel-Cahn, "Comparison of threshold stop rules and maximum for independent nonnegative random variables," *the Annals of Probability*, pp. 1213–1216, 1984.

[4] R. Kleinberg and S. M. Weinberg, "Matroid prophet inequalities," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 123–136.

[5] M. Feldman, O. Svensson, and R. Zenklusen, "Online contention resolution schemes with applications to bayesian selection problems," *SIAM J. Comput.*, vol. 50, no. 2, pp. 255–300, 2021.

[6] A. Asadpour and H. Nazerzadeh, "Maximizing stochastic monotone submodular functions," *Management Science*, vol. 62, no. 8, pp. 2374–2391, 2016.

[7] D. Golovin and A. Krause, "Adaptive submodularity: A new approach to active learning and stochastic optimization," *CoRR*, vol. abs/1003.3967, 2017.

[8] S. Im, V. Nagarajan, and R. V. D. Zwaan, "Minimum latency submodular cover," *ACM Transactions on Algorithms (TALG)*, vol. 13, no. 1, pp. 1–28, 2016.

[9] A. Gupta and V. Nagarajan, "A stochastic probing problem with applications," in *Integer Programming and Combinatorial Optimization - 16th International Conference*, 2013, pp. 205–216.

[10] M. Adamczyk, M. Sviridenko, and J. Ward, "Submodular stochastic probing on matroids," *Math. Oper. Res.*, vol. 41, no. 3, pp. 1022–1038, 2016.

[11] A. Gupta, V. Nagarajan, and S. Singla, "Adaptivity gaps for stochastic probing: Submodular and xos functions," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2017, pp. 1688–1702.

[12] A. Deshpande, L. Hellerstein, and D. Kletenik, "Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack," *ACM Transactions on Algorithms (TALG)*, vol. 12, no. 3, pp. 1–28, 2016.

[13] B. C. Dean, M. X. Goemans, and J. Vondrák, "Approximating the stochastic knapsack problem: The benefit of adaptivity," *Math. Oper. Res.*, vol. 33, no. 4, pp. 945–964, 2008.

[14] N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra, "Approximating matches made in heaven," in *36th International Colloquium on Automata, Languages and Programming (ICALP*, pp. 266–278.

[15] 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*, vol. 63, no. 4, pp. 733–762, 2012.

[16] R. Butterworth, "Some reliability fault-testing models," *Operations Research*, vol. 20, no. 2, pp. 335–343, 1972.

[17] R. Kleinberg and S. M. Weinberg, "Matroid prophet inequalities and applications to multi-dimensional mechanism design," *Games Econ. Behav.*, vol. 113, pp. 97–115, 2019.

[18] W. Ma, "Improvements and generalizations of stochastic knapsack and markovian bandits approximation algorithms," *Math. Oper. Res.*, vol. 43, no. 3, pp. 789–812, 2018.

[19] K. Gatmiry, T. Kesselheim, S. Singla, and Y. Wang, "Bandit algorithms for prophet inequality and pandora's box," in *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2024.

[20] C. Guo, Z. Huang, Z. G. Tang, and X. Zhang, "Generalizing complex hypotheses on product distributions: Auctions, prophet inequalities, and pandora's problem," in *Conference on Learning Theory*. PMLR, 2021, pp. 2248–2288.

[21] T. L. Lai and H. Robbins, "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Mathematics*, vol. 6, pp. 4–22, 1985.

[22] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[23] A. Rubinstein and S. Singla, "Combinatorial prophet inequalities," in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2017, pp. 1671–1687.

[24] R. D. Kleinberg, B. Waggoner, and E. G. Weyl, "Descending price optimally coordinates search," in *ACM Conference on Economics and Computation*. ACM, 2016, pp. 23–24.

[25] S. Boodaghians, F. Fusco, P. Lazos, and S. Leonardi, "Pandora's box problem with order constraints," *Math. Oper. Res.*, vol. 48, no. 1, pp. 498–519, 2023.

[26] A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi, "Running errands in time: Approximation algorithms for stochastic orienteering," *Math. Oper. Res.*, vol. 40, no. 1, pp. 56–79, 2015.

[27] M. Goemans and J. Vondrák, "Stochastic covering and adaptivity," in *Latin American symposium on theoretical informatics*. Springer, 2006, pp. 532–543.

[28] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[29] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[30] E. Hazan, *Introduction to online convex optimization*. MIT Press, 2022.

[31] M. G. Azar, I. Osband, and R. Munos, "Minimax regret bounds for reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, vol. 70. PMLR, 2017, pp. 263–272.

[32] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is q-learning provably efficient?" in *Advances in Neural Information Processing Systems*, 2018, pp. 4868–4878.

[33] C. Jin, T. Jin, H. Luo, S. Sra, and T. Yu, "Learning adversarial markov decision processes with bandit feedback and unknown transition," in *Proceedings of the 37th International Conference on Machine Learning, (ICML)*, vol. 119. PMLR, 2020, pp. 4860–4869.

[34] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *J. Mach. Learn. Res.*, vol. 11, pp. 1563–1600, 2010.

[35] I. Osband and B. V. Roy, "On lower bounds for regret in reinforcement learning," *CoRR*, vol. abs/1608.02732, 2016.

[36] V. Gabillon, B. Kveton, Z. Wen, B. Eriksson, and S. Muthukrishnan, "Adaptive submodular maximization in bandit setting," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[37] S. Chawla, E. Gergatsouli, Y. Teng, C. Tzamos, and R. Zhang, "Pandora's box with correlations: Learning and approximation," in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2020, pp. 1214–1225.

[38] E. Gergatsouli and C. Tzamos, "Online learning for min sum set cover and pandora's box," in *International Conference on Machine Learning*. PMLR, 2022, pp. 7382–7403.

[39] H. Esfandiari, M. HajiAghayi, B. Lucier, and M. Mitzenmacher, "Online pandora's boxes and bandits," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1885–1892.

[40] R. Cole and T. Roughgarden, "The sample complexity of revenue maximization," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, 2014, pp. 243–252.

[41] T. Roughgarden and O. Schrijvers, "Ironing in the dark," in *Proceedings of the 2016 ACM Conference on Economics and Computation*, 2016, pp. 1–18.

[42] J. Correa, P. Dütting, F. Fischer, and K. Schewior, "Prophet inequalities for iid random variables from an unknown distribution," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 3–17.

[43] B. Kveton, Z. Wen, A. Ashkan, H. Eydgahi, and B. Eriksson, "Matroid bandits: Fast combinatorial optimization with learning," in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014, pp. 420–429.

[44] B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvári, "Tight regret bounds for stochastic combinatorial semi-bandits," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[45] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *J. Mach. Learn. Res.*, vol. 17, pp. 50:1–50:33, 2016.

[46] W. Chen, W. Hu, F. Li, J. Li, Y. Liu, and P. Lu, "Combinatorial multi-armed bandit with general reward functions," in *Advances in Neural Information Processing Systems*, 2016, pp. 1651–1659.

[47] A. T. Kalai and S. S. Vempala, "Efficient algorithms for online decision problems," *J. Comput. Syst. Sci.*, vol. 71, no. 3, pp. 291–307, 2005.

[48] S. M. Kakade, A. T. Kalai, and K. Ligett, "Playing games with approximation algorithms," *SIAM J. Comput.*, vol. 39, no. 3, pp. 1088–1106, 2009.

[49] R. Niazadeh, N. Golrezaei, J. R. Wang, F. Susan, and A. Badanidiyuru, "Online learning via offline greedy algorithms: Applications in market design and optimization," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, 2021, pp. 737–738.

[50] G. Nie, Y. Y. Nadew, Y. Zhu, V. Aggarwal, and C. J. Quinn, "A framework for adapting offline algorithms to solve combinatorial multi-armed bandit problems with bandit feedback," in *International Conference on Machine Learning (ICML)*, vol. 202. PMLR, 2023, pp. 26 166–26 198.

[51] S. Ghosal and A. Van der Vaart, *Fundamentals of nonparametric Bayesian inference*. Cambridge University Press, 2017, vol. 44.

[52] A. Agarwal, R. Ghuge, and V. Nagarajan, "Semi-bandit learning for monotone stochastic optimization," *arXiv preprint arXiv:2312.15427*, 2023.

[53] U. Krengel and L. Sucheston, "Semiamarts and finite values," *Bulletin of the American Mathematical Society*, vol. 83, pp. 745–747, 1977.

[54] P. D. Azar, R. Kleinberg, and S. M. Weinberg, "Prophet inequalities with limited information," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2014, pp. 1358–1377.

[55] A. Rubinstein, J. Z. Wang, and S. M. Weinberg, "Optimal single-choice prophet inequalities from samples," in *11th Innovations in Theoretical Computer Science Conference, (ITCS)*, ser. LIPIcs, vol. 151, 2020, pp. 60:1–60:10.

[56] S. Alaei, "Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers," *SIAM J. Comput.*, vol. 43, no. 2, pp. 930–972, 2014.

[57] J. Jiang, W. Ma, and J. Zhang, "Tight guarantees for multi-unit prophet inequalities and online stochastic knapsack," in *ACM-SIAM Symposium on Discrete Algorithms, (SODA)*. SIAM, 2022, pp. 1221–1246.

[58] T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang, "Prophet matching with general arrivals," *Math. Oper. Res.*, vol. 47, no. 2, pp. 878–898, 2022.

[59] C. Caramanis, P. Dütting, M. Faw, F. Fusco, P. Lazos, S. Leonardi, O. Papadigenopoulos, E. Pountourakis, and R. Reiffenhäuser, "Single-sample prophet inequalities via greedy-ordered selection," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2022, pp. 1298–1325.

[60] A. Bhalgat, A. Goel, and S. Khanna, "Improved approximation results for stochastic knapsack problems," in *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011, pp. 1647–1665.

[61] A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi, "Approximation algorithms for correlated knapsacks and non-martingale bandits," in *IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, R. Ostrovsky, Ed., 2011, pp. 827–836.

[62] M. Adamczyk, "Improved analysis of the greedy algorithm for stochastic matching," *Inf. Process. Lett.*, vol. 111, no. 15, pp. 731–737, 2011.

[63] B. Brubach, N. Grammel, W. Ma, and A. Srinivasan, "Improved guarantees for offline stochastic matching via new ordered contention resolution schemes," in *Advances in Neural Information Processing Systems*, 2021, pp. 27 184–27 195.