Multi-armed Bandits with Limited Exploration*

Sudipto Guha† Kamesh Munagala‡

Abstract

A central problem to decision making under uncertainty is the trade-off between exploration and exploitation: between learning from and adapting to a stochastic system and exploiting the current best-knowledge about the system. A fundamental decision-theoretic model that captures this trade-off is the celebrated stochastic Multi-arm Bandit Problem. In this paper, we consider scenarios where the exploration phase corresponds to designing experiments, and the exploration phase has the following restrictions: (1) it must necessarily precede the exploitation phase; (2) it is expensive in terms of some resource consumed, so that only a limited amount of exploration can be performed; and (3) switching from one experiment to another incurs a setup cost. Such a model, which is termed *budgeted learning*, is relevant in scenarios such as clinical trials and sensor network data acquisition.

Though the classic multi-armed bandit problem admits to a polynomial time greedy optimal solution termed the *Gittins index* policy, the budgeted learning problem does not admit to such a greedy optimal solution. In fact, the problem is NP-Hard even in simple settings. Our main contribution is in presenting constant factor approximation algorithms for this problem via a novel linear program rounding technique based on stochastic packing.

*This combines work from two papers [23, 22] appearing in the ACM-SIAM Symposium on Discrete Algorithms (SODA), 2007 and the 39^{th} Annual ACM Symposium on Theory of Computing (STOC), 2007 respectively.

†Department of Computer and Information Sciences, University of Pennsylvania. Email: sudipto@cis.upenn.edu. Research supported in part by an Alfred P. Sloan Research Fellowship and by an NSF Award CCF-0430376.

‡Department of Computer Science, Duke University. Email: kamesh@cs.duke.edu. Research supported in part by NSF CNS-0540347.

1 Introduction

In this paper, we study variants of the celebrated stochastic multi-armed bandit problem when exploration is costly. We first present the multi-armed bandit problem (due to Robbins [39]) in a simple setting. Suppose we are given n coins, so that each coin i has an unknown probability of heads p_i. We can toss any coin any number of times – but we are limited to C tosses in all. If we wish to maximize the number of times we observe heads, how should we proceed? Such a problem is termed the multi-armed bandit problem. The answer clearly depends on what information is available upfront about the p_i. If an adversary is assumed to choose these probabilities, this is the adversarial multi-armed bandit problem or experts problem [4, 2, 3, 10, 18, 30]. If a *prior* probability distribution $\mathcal{R}_i$ is available over the possible values of p_i, then the goal becomes to find a *policy* to maximize the expected number of heads – this is the model-based stochastic multi-armed bandit problem that was formulated by Gittins and Jones [19] and extensively studied since then in decision theory [6, 44, 37, 45]. In the infinite horizon discounted reward setting, the stochastic version admits to an elegant greedy optimal solution termed the *Gittins index* policy [19]. In the bandit literature, the coins are termed "arms", and the process of tossing a coin is termed "playing the arm". This play yields an *i.i.d.* observation or "reward" from the underlying unknown reward distribution. The goal then becomes to find a playing strategy to maximize reward.

The multi-armed bandit problem models two fundamental aspects in decision making under uncertainty – *exploration* plays to learn more about the rewards of different arms, and *exploitation* plays of the current best arm. These two types of plays trade-off with each other, and the algorithms for this problem attempt to optimally balance these two types. In this paper, we consider scenarios where exploration corresponds to designing experiments to understand the reward distributions better, and has one or more of the following constraints: (1) it must necessarily precede the exploitation phase; (2) it is expensive in terms of some resource consumed, so that only a limited amount of exploration can be performed; and (3) switching from one experiment to another incurs a setup cost. These aspects make the problem algorithmically very different from the standard models, and the goal of this work is to design algorithms for variants incorporating these constraints.

As a motivating example, consider the clinical trial setting which was one of the original motivations for the multi-armed bandit problem [19]. We have competing treatments and little data on the effectiveness of the individual treatments. Each treatment can be modeled as a bandit arm, and the effectiveness of the treatment is the reward distribution that is unknown a priori. Playing an arm corresponds to running a trial of the treatment on a patient. In this setting, the clinical trial phase must necessarily precede the exploitation phase of marketing the best treatment. Further, each trial costs a lot of money, and for an uncommon enough disease, there are not enough patients. This imposes a cost constraint on the number of trials that can be performed. In other cases, for instance, testing system parameter settings for optimal software performance, each system configuration is an experiment setting (or bandit arm). Running repeated experiments with this setting incurs less cost than switching between configurations, which incurs setup cost. These considerations motivate the budget and switching cost constraints on playing the arms. Finally, such a framework extends to any classification scenario where we are interested in finding the most successful classifier from a few trials where we can verify the classifier by human experts.

We term the above class of problems as *budgeted learning*; this is a generalization of the budgeted multi-armed bandit problems formulated in [33, 40]. For these types of problems, we argue in Section 1.5 that it is more reasonable to work in the stochastic setting as opposed to the adversarial setting. Before proceeding further, we present the formal problem statements.

1.1 Problem Formulation

As in the classic stochastic multi-armed bandit problem [19, 6], we are given a bandit with n independent arms. The set of possible states of arm i is denoted $\mathcal{S}_i$, and the initial state is $\rho_i \in \mathcal{S}_i$. Assume all $|\mathcal{S}_i| \leq m$. When the arm is played $u \in \mathcal{S}_i$, the play yields reward r_u, and transitions to state $v \in \mathcal{S}_i$ w.p. $\mathbf{p}_{uv}$. The cost of a play depends on whether the previous play was for the same arm or not. If the previous play was for the same arm, the play at $u \in \mathcal{S}_i$ costs c_u, else it costs $c_u + h_i$, where h_i is the setup cost for switching into arm i.

Martingale Property. We now explain what we mean by "state" of an arm. The arm has a true underlying reward distribution, and we have prior distributional knowledge over possible reward distributions of an arm. Each play yields a reward from the underlying distribution that resolves somewhat the prior knowledge. At any step, the "state" u of the arm encodes the current distributional knowledge about the possible underlying reward distribution, *i.e.*, it encodes the uncertainty about the true reward distribution conditioned on the outcomes of the plays so far. The reward r_u of the state is simply the expected reward that would be observed on playing the arm given the current distributional knowledge. We therefore assume the evolution of rewards r_u satisfy the following *Martingale* property: $r_u = \sum_{v \in \mathcal{S}_i} \mathbf{p}_{uv} r_v$. We discuss this in more detail via examples below. From now on, unless otherwise stated, the term "reward" would mean the reward of a state (expected reward given the distributional knowledge in that state).

A policy π consists of two phases: First, the policy performs a possibly adaptive sequence of plays. We term this phase *exploration*. Since the state evolutions are stochastic, the exploration phase leads to a probability distribution over outcomes, $\mathcal{O}(\pi)$. In outcome $o \in \mathcal{O}(\pi)$, each arm i is in some final state u_i^o. Let the probability of outcome $o \in \mathcal{O}(\pi)$ be $q(o, \pi)$. The second phase, *exploitation* begins after the exploration, *i.e.*, after the outcome of the policy execution becomes known. For outcome o, the exploitation phase chooses the "best" arm, which is the arm i with maximum reward of final state, $r_{u_i^o}$, which will be the reward of the policy for this outcome.

Let $R(\pi)$ denote the expected reward of the policy π over the outcomes of exploration, which is simply $\sum_{o \in \mathcal{O}(\pi)} q(o, \pi) \max_i r_{u_i^o}$. Let $C(\pi)$ denote the expected cost of the plays made by the policy. We have two related problem formulations:

Budgeted Learning: There is a cost budget C. A policy π is *feasible* if for any sequence of plays made by the policy, the cost is at most C. The goal is to find the feasible policy π with maximum $R(\pi)$.

Non-Adaptive Version: Any feasible policy π fixes the number of plays for each arm in advance, so that the total cost of the plays is at most C. Again, the goal is to find the feasible policy π with maximum $R(\pi)$. Non-adaptive strategies are desirable since the plays can be executed in parallel.

Lagrangean Version: Find the policy π with maximum $R(\pi) - C(\pi)$.

In this paper, we focus on algorithms whose running time is polynomial in n, m (recall that each $|\mathcal{S}_i| \leq m$), which corresponds to the input size. The problem of finding the optimal policy in either case is a Markov Decision Problem (MDP). The optimal MDP solution would compute an action for each joint state of all n arms. The size of the joint state space is $O(m^n)$, which implies a similar running time of the standard dynamic programming algorithm to compute the optimal policy. The immediate question is therefore whether the optimal solution has specification and computation time polynomial in m, n. The following theorem answers this in the negative.

Theorem 1.1. *[33, 13, 20] The budgeted learning problem is* NP-HARD *when the costs c_i are arbitrary. Further, computing the optimal* non-adaptive *policy is* NP-HARD *even with unit costs.*

We have not been able to show APX-Hardness for budgeted learning, and we believe it to be a challenging open question. In [20], a connection is shown between this and the APX-Hardness for a certain kind of covering integer programs. Further evidence to this effect is given by the fact that the optimal policy is not an *index* policy. This is shown in [33] for $n = 3$ arms. An index policy computes a number a_u (called the index) for every $u \in \mathcal{S}_i$ based just on the characteristics of arm i, and always plays the arm whose current state has highest index. Note that the classic stochastic multi-armed bandit problem admits to an optimal index scheme termed the *Gittins index* [19].

1.2 Types of State Spaces and Representative Examples

We now present two representative scenarios in order to better motivate the abstract problem formulation. In the first scenario, the underlying reward distribution is deterministic, and the distributional knowledge is specified as a distribution over the possible deterministic values; this implies that the uncertainty about an arm is completely resolved in one play by observing the reward. In the second scenario, the uncertainty resolves gradually over time.

Two-level State Space. Consider a sensor network where the root server monitors the maximum value [5, 42]. The probability distributions of the values at various nodes are known to the server via past observations. However, at the current step, probing all nodes to find out their actual values is undesirable since it requires transmissions from all nodes, consuming their battery life. Consider the simple setting where the network connecting the nodes to the server is a one-level tree, and probing a node consumes battery power of that node. Given a bound on the total battery life consumed, the goal of the root server is to maximize (in expectation) its estimate of the maximum value. Formally, each node corresponds to a distribution X_i with mean μ_i; the exact value sensed at the node can be found by paying a "transmission cost" c_i. The goal of the server is to adaptively probe a subset S of nodes with total transmission cost at most C in order to maximize the estimate of the largest value sensed, *i.e* maximize $\mathbf{E}[\max(\max_{i\in S} X_i, \max_{i\notin S} \mu_i)]$, where the expectation is over the adaptive choice of S and the outcome of the probes. The term $\max_{i\notin S} \mu_i$ incorporates the mean of the unprobed nodes into the estimate of the maximum value.

We map this problem to budgeted learning as follows. Each sensor node is a bandit arm, and the cost for playing it is c_i. The node senses one of the values, $\{a_1^i, a_2^i, \ldots, a_m^i\}$; the prior distributional knowledge X_i is a discrete distribution over these values, so that $\Pr[X_i = a_j^i] = p_j^i$ for $j = 1, 2, \ldots, m$. The state space $\mathcal{S}_i$ of the arm is as follows: The root node ρ_i has $r_{\rho_i} = \mathbf{E}[X_i] = \mu_i$, and $c_{\rho_i} = c_i$. For $j = 1, 2, \ldots, m$, state i_j has $r_{i_j} = a_j^i$, and $\mathbf{p}_{\rho_i i_j} = p_j^i$. Since the underlying reward distribution is simply a deterministic value, the state space is 2-level, defining a star graph with ρ_i being the root, and $i_1, i_2, \ldots, i_m$ being the leaves.

We give an example to showing benefit of probing.

Example 1.1. *If only one probe is allowed, the root server's estimate is* $\max_i \mathbf{E}[X_i]$. *If the cost constraint is sufficient to probe all nodes, this estimate improves to* $\mathbf{E}[\max_i X_i]$, *since the server can find the exact values at all nodes and return the maximum value. If all X_i are Bernoulli $B(1, p)$ with $p < 1/n$, the former value is p and the latter is $1 - (1 - p)^n \approx np$. Therefore, probing nodes can improve the expected value returned by a factor of $\Omega(n)$.*

In this context, it is desirable for the sensor node to probe the nodes in parallel, *i.e.*, use a non-adaptive strategy. The question then becomes how good is such a strategy compared to the optimal adaptive strategy. We show positive results for the context of 2-level states in Section 2.6.

Multi-level State Spaces. Consider next the clinical trial setting mentioned above. Here, each experimental drug is a bandit arm, and the goal is to devise a clinical trial phase to maximize the belief about the effectiveness of the drug finally chosen for marketing. Each drug has an effectiveness that is unknown a priori. The effectiveness can be modeled as a coin whose bias, θ, is unknown a priori – the outcomes of tossing the coin (running a trial) are 0 and 1 which correspond to a trial being ineffective and effective respectively. The uncertainty in the bias is specified by a *prior* distribution (or belief) on the possible values it can take. Since the underlying distribution is Bernoulli, its conjugate prior is the Beta distribution. A Beta distribution with parameters $\alpha_1, \alpha_2 \in \{1, 2, \ldots\}$, which we denote $B(\alpha_1, \alpha_2)$ has p.d.f. of the form $c\theta^{\alpha_1 - 1}(1-\theta)^{\alpha_2 - 1}$, where c is a normalizing constant. $B(1,1)$ is the uniform distribution, which corresponds to having no a priori information. The distribution $B(\alpha_1, \alpha_2)$ corresponds to the current (posterior) distribution over the possible values of the bias θ after having observed $(\alpha_1 - 1)$ 0's and $(\alpha_2 - 1)$ 1's. Given this distribution as our belief, the expected value of the bias or effectiveness is $\frac{\alpha_1}{\alpha_1+\alpha_2}$.

The state space $\mathcal{S}_i$ is a DAG, whose root ρ_i encodes the initial belief about the bias, $B(\alpha_1, \alpha_2)$, so that $r_{\rho_i} = \frac{\alpha_1}{\alpha_1^{\rho}+\alpha_2}$. When the arm is played in this state, the state evolves depending on the outcome observed – if the outcome is 1, which happens w.p. $\frac{\alpha_1}{\alpha_1+\alpha_2}$, the child u has belief $B(\alpha+1, \alpha_2)$, so that $r_u = \frac{\alpha_1+1}{\alpha_1+\alpha_2+1}$, and $\mathbf{p}_{\rho u} = \frac{\alpha_1}{\alpha_1+\alpha_2}$; if the outcome is 0, the child v has belief $B(\alpha_1, \alpha_2+1)$, $r_v = \frac{\alpha_1}{\alpha_1+\alpha_2+1}$, and $\mathbf{p}_{\rho v} = \frac{\alpha_2}{\alpha_1+\alpha_2}$. In general, if the DAG $\mathcal{S}_i$ has depth C (corresponding to playing the arm at most C times), it has $O(C^2)$ states. We do not explain Beta priors in more detail, since it is standard in the bandit setting (refer [19]).

The key constraint in clinical trial design, and experiment design in general is that each trial or experiment is difficult and expensive in terms of some resource to run. Finding a patient to run a trial on consumes time and money. In other cases, for instance, testing system parameter settings for optimal software performance, each system configuration is an experiment setting (or bandit arm). Running repeated experiments with this setting incurs less cost than switching between configurations, which incurs setup cost. These considerations motivate the budget and switching cost constraints on playing the arms. The Beta priors described above are a good model when an experiment has two outcomes; a larger set of outcomes motivates the multi-nomial generalization which are termed Dirichlet priors.

1.3 Results and Technical Contributions

In this paper, we present approximation algorithms for multi-armed bandit problems with exploration constraints. In particular, we show the following results. We first present a 4-approximation to the budgeted learning problem (Section 2), and a bicriteria $2(1+\alpha)$ approximation with the cost constraint relaxed by a factor $\frac{1}{\alpha}$. We also present a 2 approximation for the Lagrangean variant (Section 3).

In Section 2.6, we show that for the budgeted learning problem with multi-level state spaces, an adaptive strategy can yield $\Omega(\sqrt{n})$ times more reward than the optimal non-adaptive strategy that allocates plays upfront to different arms. However, in the case of 2-level state spaces and the budgeted version, we show that any adaptive probing strategy has at most a factor 7 benefit over the optimal non-adaptive probing strategy that chooses the set S to probe upfront, ahead of the outcomes of any of the probes. Recall the sensor network scenario described above. There, a non-adaptive strategy would be desirable since the root server can probe all desired nodes in parallel and ignore the outcomes of the plays in deciding about further exploration.

In Section 4, finally present unified solution framework for general budgeted learning problems where the exploitation objective could involve any separable non-negative concave utility function

over the rewards of the arms subject to a packing constraint. The concave utility assumption models decreasing marginal utilities in resource allocation, and captures natural extensions such as choosing the top-m best arms for exploitation, etc. As another example, consider a pharmaceutical company that wants to decide how to allocate a different fixed resource for production (which is separate from the C for trials) among the competing treatments. The benefit (to the company) of a treatment i is a non-negative function $g(x_i, r_i)$ of the alloted resource x_i, and effectiveness r_i. This function is concave and non-decreasing in x_i (decreasing marginal benefit), and non-decreasing in r_i. The distribution of the r_i can be resolved during exploration (clinical trial phase) before deciding how to allocate x_i among competing treatments. Our framework handles these type of extensions which arise frequently in optimizing system performance (e.g. resource allocation for fairness [32]), as long as the arms are independent.

Our algorithms are based on rounding natural LP relaxations; we further show constant factor gaps of these relaxations against the optimal policy, thereby showing the limits of our approach. The policies produced by our algorithm are natural – they are semi-adaptive policies, where the play of an arm depends on the outcomes for that arm, but the arms are explored in a fixed order and never revisited.

Techniques: Our policy design is via a linear programming formulation over the state space of individual arms, and we achieve polynomial sized formulation in the size of each individual state space. We then bring to bear techniques from stochastic packing literature, particularly the work on adaptivity gaps [16, 17, 15] to develop a novel LP rounding scheme. Our overall technique can be thought of as "LP rounding via stochastic packing" – finding this connection between budgeted learning and stochastic packing by designing simple LP rounding policies for a very general class of budgeted learning problems represents the key contribution of this work.

The work of Dean, Goemans, and Vondrák [16] shows constant factor adaptivity gaps for knapsack and related packing problems with stochastic item sizes when items need to be *irrevocably* committed to the knapsack before their true size is revealed. In contrast, the budgeted multi-armed bandit problem does not have irrevocable commitment – after an arm has been explored many times, it can be discarded if no benefit is observed; an arm can be revisited multiple times during exploration, etc. In fact, the strategy needs to be adaptive (refer Section 2.6). Despite these significant differences between the problems, we show that the idea of irrevocable commitment is a powerful rounding technique, yielding simple to analyze semi-adaptive policies.

1.4 Related Work

The budgeted learning problem is a special case of *active learning* [12, 1, 26, 35], and also a special case of the Design of Experiments (DOE) problem in Statistics literature (refer [36] for a survey). This problem also falls in the framework of *action-evaluation* in meta-reasoning [13]. As mentioned in Theorem 1.1, the optimal solution of the budgeted learning problem is NP-Hard even when at most one play is allowed per arm [33, 13], and with unit costs [20].

Several heuristics have been proposed for the budgeted learning problem by Schneider and Moore [40]. Extensive empirical comparison of these heuristics in the context of Beta priors is presented by Madani *et al.* [33], and it appears that an adaptive policy which makes the next play decision based on both (1) the remaining budget, and (2) the outcomes of the plays so far, outperform heuristics which base their decisions on lesser information.

The Lagrangean variant with 2-level state space and the net profit being the difference between the value of the highest posterior expected reward arm and the exploration cost is considered in [24], where a 1.25 approximation is presented. The techniques are specific to 2-level state spaces and do

not extend to either general state spaces.

There is a rich literature on policies for stochastic packing [28, 21, 16, 17] as well as stochastic scheduling [34, 43]. LP formulations over the state space of outcomes exist for multi-stage stochastic optimization with recourse [29, 41, 11]. However, the significant difference is that in order to achieve poly-sized formulation, we do not sample the joint space of outcomes – since our budget constraint runs across stages, our scenarios would be trajectories [27] which are hard to sample. We instead use the independence of the arms to decompose the state space. In that sense, our LP relaxation is akin to the LP relaxations for multi-class queueing systems [7, 8, 9]; however, unlike our relaxation, there are no provable guarantees on the quality of the multi-class queueing relaxations.

1.5 Contrast with the Experts Framework

Our problem formulations are *model-based*, meaning that we assume prior distributional knowledge of the underlying distributions and values, akin to the stochastic bandit setting [6, 19, 44, 45, 37]. The availability of some distributional knowledge is a reasonable assumption for scenarios such as sensor networks and experiment design. This is in contrast with the *model-free* approach of the adversarial multi-armed problems. As mentioned above, these problems, also known as *experts* problems [31, 4, 2, 3, 14, 10, 18, 25, 30], model performing learning and prediction tasks when the bandit rewards are unknown and chosen by an adversary. The goal here is to minimize the regret with respect to the optimal algorithm which performs exploitation with complete information.

Our model captures two key aspects that are difficult to capture in the experts setting. First, our exploration phase is expensive and very limited in duration. For instance, in the sensor network setting, exploring all nodes is a bad option; similarly, in the clinical trial setting, the budget is much smaller than $O(n \log n)$, which is needed to even coarsely estimate the effectiveness of individual drugs. In such scenarios, the regret measure is not very meaningful when the exploration phase has small duration, and it is more reasonable to assume distributional models of prior information and optimize for the expected exploitation reward. Secondly, our model incorporates setup cost for an experiment, or the cost of switching into an arm, which is hard to incorporate into standard experts algorithms (refer [38] for a related heuristic attempt).

The main drawback of our approach is that when exploration is relatively cheap so that a large number of plays are allowed (which is *not* the setting of interest in this paper), then both the experts framework and the Gittins index strategy [19] yield near-optimal algorithms, while our approach will still yield only a 2 approximation. It is an interesting open question to merge our work with the experts and the Gittins index settings to obtain a unified solution.

2 Budgeted Learning

We first consider the budgeted learning problem, where the exploration policy is feasible only if the total cost of playing in any decision trajectory is at most C. We describe the linear programming formulation and rounding technique that yields a 4-approximation. We note that the formulation and solution are polynomial in n, the number of arms, and m, the number of states per arm.

2.1 Linear Programming Formulation

Recall the notation from Section 1.1. Consider any adaptive policy π. For some arm i and state $u \in \mathcal{S}_i$, let: (1) w_u denote the probability that during the execution of the policy π, arm i enters state $u \in \mathcal{S}_i$; (2) z_u denote the probability that the state of arm i is u and the policy plays arm i in this state; and (3) x_u denote the probability that the policy π chooses the arm i in state u during

the exploitation phase. Note that since the latter two correspond to mutually exclusive events, we have $x_u + z_u \leq w_u$. The following LP which has three variables w_u, x_u, and z_u for each arm i and each $u \in \mathcal{S}_i$. A similar LP formulation was proposed for the multi-armed bandit problem by Whittle [45] and Bertsimas and Nino-Mora [37].

$$\text{Maximize} \quad \sum_{i=1}^{n} \sum_{u \in \mathcal{S}_i} x_u r_u$$

$$\begin{aligned}
\textstyle\sum_{i=1}^{n} \left(h_i z_{\rho_i} + \sum_{u \in \mathcal{S}_i} c_u z_u \right) &\leq C \\
\textstyle\sum_{i=1}^{n} \sum_{u \in \mathcal{S}_i} x_u &\leq 1 \\
\textstyle\sum_{v \in \mathcal{S}_i} z_v \mathbf{p}_{vu} &= w_u \qquad \forall i, u \in \mathcal{S}_i \setminus \{\rho_i\} \\
x_u + z_u &\leq w_u \qquad \forall u \in \mathcal{S}_i, \forall i \\
x_u, z_u, w_u &\in [0,1] \quad \forall u \in \mathcal{S}_i, \forall i
\end{aligned}$$

Let γ^* be the optimal LP value, and OPT be the expected reward of the optimal adaptive policy.

Claim 2.1. *$OPT \leq \gamma^*$.*

Proof. We show that the w_u, z_u, x_u as defined above, corresponding to the optimal policy π^*, are feasible for the constraints of the LP. Since each possible outcome of exploration leads to choosing one arm i in some state $u \in \mathcal{S}_i$ for exploitation, in expectation over the outcomes, one arm in one state is chosen for exploitation. This is captured by the first constraint. Further, since on each decision trajectory, the cost of playing and switching into the arm is at most C, over the entire decision tree, the expected cost of switching into the root states ρ_i plus the expected cost of play is at most C. This is captured by the second constraint. Note that the LP only takes into account the cost of switching into an arm the very first time this arm is explored, and ignores the rest of the switching costs. This is clearly a relaxation, though the optimal policy might switch multiple times into any arm. However, our rounding procedure switches into an arm at most once, preserving the structure of the LP relaxation.

The third constraint simply encodes that the probability of reaching a state $u \in \mathcal{S}_i$ during exploration is precisely the probability with which it is played in some state $v \in \mathcal{S}_i$, times the probability $\mathbf{p}_{vu}$ that it reaches u conditioned on that play. The constraint $x_u + z_u \leq w_u$ simply captures that playing an arm is a disjoint event from exploiting it in any state. The objective is precisely the expected reward of the policy. Hence, the LP is a relaxation of the optimal policy. □

2.2 The Single-arm Policies

The optimal LP solution clearly does not directly correspond to a feasible policy since the variables do not faithfully capture the joint evolution of the states of different arms. Below, we present an interpretation of the LP solution, and show how it can be converted to a feasible approximately optimal policy.

Let $\langle w_u^*, x_u^*, z_u^* \rangle$ denote the optimal solution to the LP. Assume w.l.o.g. that $w_{\rho_i}^* = 1$ for all i. Ignoring the first two constraints of the LP for the time being, the remaining constraints encode a separate policy for each arm as follows: Consider any arm i in isolation. The play starts at state ρ_i. The arm is played with probability $z_{\rho_i}^*$, so that state $u \in \mathcal{S}_i$ is reached with probability $z_{\rho_i}^* \mathbf{p}_{\rho_i u}$. This play incurs cost $h_i + c_{\rho_i}$, which captures the cost of switching into this arm, and the cost of playing at the root. At state ρ_i, with probability $x_{\rho_i}^*$, the play stops and arm i is chosen for exploitation. The events involving playing the arm and choosing for exploitation are disjoint. Similarly, conditioned on reaching state $u \in \mathcal{S}_i$, with probabilities z_u^*/w_u^* and x_u^*/w_u^*, arm i is played and chosen for exploitation respectively. This yields a policy ϕ_i for arm i which is described in

Figure 1. For policy ϕ_i, it is easy to see by induction that if state $u \in \mathcal{S}_i$ is reached by the policy with probability w_u^*, then state $u \in \mathcal{S}_i$ is reached *and* arm i is played with probability z_u^*.

The policy ϕ_i sets $\mathcal{E}_i = 1$ if on termination, arm i was chosen for exploitation. If $\mathcal{E}_1 = 1$ at state $u \in \mathcal{S}_i$, then exploiting the arm in this state yields reward r_u. Note that $\mathcal{E}_i$ is a random variable that depends on the execution of policy ϕ_i. Let R_i, C_i denote the random variables corresponding to the exploitation reward, and cost of playing and switching, respectively.

Policy ϕ_i: If arm i is currently in state u, then choose $q \in [0, w_u^*]$ uniformly at random:

1. If $q \in [0, z_u^*]$, then play the arm (**explore**).
2. If $q \in (z_u^*, z_u^* + x_u^*]$, then stop executing ϕ_i, set $\mathcal{E}_i = 1$ (**exploit**).
3. If $q \in (z_u^* + x_u^*, w_u^*]$, then stop executing ϕ_i, set $\mathcal{E}_i = 0$.

Figure 1: The Policy ϕ_i.

For policy ϕ_i, define the following quantities:

1. $P(\phi_i) = \mathbf{E}[\mathcal{E}_i] = \sum_{u \in \mathcal{S}_i} \Pr[\mathcal{E}_i = 1 | u] = \sum_{u \in \mathcal{S}_i} x_u^*$: Probability the arm is exploited.
2. $R(\phi_i) = \mathbf{E}[R_i] = \sum_{u \in \mathcal{S}_i} r_u \Pr[\mathcal{E}_i = 1 | u] = \sum_{u \in \mathcal{S}_i} x_u^* r_u$: Expected reward of exploitation.
3. $C(\phi_i) = \mathbf{E}[C_i] = h_i z_i^* + \sum_{u \in \mathcal{S}_i} c_u z_u^*$: Expected cost of switching into and playing this arm.

Let ϕ denote the policy that is obtained by executing each ϕ_i independently in succession. Since policy ϕ_i is obtained by considering arm i in isolation, ϕ is **not a feasible policy** for the following reasons: (i) The cost $\sum_i C_i$ spent exploring all the arms need not be at most C in every exploration trajectory, and (ii) It could happen that for several arms i, $\mathcal{E}_i$ is set to 1, which implies several arms could be chosen simultaneously for exploitation.

However, all is not lost. First note that the r.v. $R_i, C_i, \mathcal{E}_i$ for different i are independent. Furthermore, it is easy to see using the first two constraints and objective of the LP formulation that ϕ is feasible in the following expected sense: $\sum_i \mathbf{E}[C_i] = \sum_i C(\phi_i) \leq C$. Secondly, $\sum_i \mathbf{E}[\mathcal{E}_i] = \sum_i P(\phi_i) \leq 1$. Finally, $\sum_i \mathbf{E}[R_i] = \sum_i R(\phi_i) = \gamma^*$.

Based on the above, we show that policy ϕ can be converted to a feasible policy using ideas from the adaptivity gap proofs for stochastic packing problems [16, 17, 15]. We treat each policy ϕ_i as an item which takes up cost C_i, has size $\mathcal{E}_i$, and profit R_i. These items need to be placed in a knapsack – placing item i corresponds to exploring arm i according to policy ϕ_i. This placement is an irrevocable decision, and after the placement, the values of $C_i, \mathcal{E}_i, R_i$ are revealed. We need $\sum_i C_i$ for items placed so far should be at most C. Furthermore, the placement (or exploration) stops the first time some $\mathcal{E}_i$ is set to 1, and uses arm i is used for exploitation (obtaining reward or profit R_i). Since only one $\mathcal{E}_i = 1$ event is allowed before the play stops, this yields the "size constraint" $\sum_i \mathcal{E}_i \leq 1$. The knapsack therefore has both cost and size constraints, and the goal is to sequentially and irrevocably place the items in the knapsack, stopping when the constraints would be violated. The goal is to choose the order to place the items in order to maximize the expected profit, or the exploitation gain. This is a two-constraint stochastic packing problem. The LP solution implies that the expected values of the random variables satisfy the packing constriants.

We show that the "start-deadline" framework in [15] can be adapted to show that there is a fixed order of exploring the arms according to the ϕ_i which yields gain at least $\gamma^*/4$. There is one subtle point – the profit (or gain) is also a random variable correlated with the size and cost. Furthermore, the "start deadline" model in [15] would also imply the final packing could violate the constraints by a small amount. We get around this difficulty by presenting an algorithm GreedyOrder that

explicitly obeys the constraints, but whose analysis will be coupled with the analysis of a simpler policy GreedyViolate which exceeds the budget. The central idea would be that although the benefit of the current arm has not been "verified", the alternatives have been ruled out.

2.3 The Rounding Algorithm

Algorithm GreedyOrder

1. Order the arms in decreasing order of $\frac{R(\phi_i)}{P(\phi_i)+\frac{C(\phi_i)}{C}}$ and choose the arms to play in this order.
2. For each arm j in sorted order, play arm j according to ϕ_j as follows until ϕ_j terminates:
   (a) If the next play according to ϕ_j would violate the budget constraint, then **stop** exploration and **goto** step (3).
   (b) If ϕ_j has terminated and $\mathcal{E}_j = 1$, then **stop** exploration and **goto** step (3).
   (c) Else, play arm j according to policy ϕ_j and **goto** step (2a).
3. Choose the *last arm* played in step (2) for exploitation.

Figure 2: The GreedyOrder policy.

The GreedyOrder policy is shown in Figure 2. Note that step (3) ensures that no arm is ever revisited. For the purpose of analysis, we first present an infeasible policy GreedyViolate which is simpler to analyze. The algorithm is the same as GreedyOrder except for step (2), which we outline in Figure 3.

In GreedyViolate, the cost budget is checked only *after* fully executing a policy ϕ_j. Therefore, the policy could violate the budget constraint by at most the exploration cost $c_{\max}$ of one arm.

Theorem 2.2. GreedyViolate *spends cost at most* $C + c_{\max}$ *and yields reward at least* $\frac{OPT}{4}$.

Proof. We have $\gamma^* = \sum_i R(\phi_i)$, and $\sum_i P(\phi_i) \leq 1$. We note that the random variables corresponding to different i are independent.

For notational convenience, let $\nu_i = R(\phi_i)$, and let $\mu_i = P(\phi_i) + C(\phi_i)/C$. We therefore have $\sum_i \mu_i \leq 2$. The sorted ordering is decreasing order of ν_i/μ_i. Re-number the arms according to the sorted ordering so that the first arm played is numbered 1. Let k denote the smallest integer such that $\sum_{i=1}^{k} \mu_i \geq 1$. By the sorted ordering property, it is easy to see that $\sum_{i=1}^{k} \nu_i \geq \frac{1}{2}\gamma^*$.

Arm i is reached and played by the policy iff $\sum_{j<i} \mathcal{E}_j = 0$, and $\sum_{j<i} C_j < C$. This translates to $\sum_{j<i} \left(\mathcal{E}_j + \frac{C_j}{C}\right) < 1$. Note that $\mathbf{E}[\mathcal{E}_j + \frac{C_j}{C}] = P(\phi_j) + C(\phi_j)/C = \mu_j$. Therefore, by Markov's

Step 2 (GreedyViolate) For each arm j in sorted order, do the following:

(a) Play arm j according to policy ϕ_j until ϕ_j terminates.

(b) When the policy ϕ_j terminates execution, if event $\mathcal{E}_j = 1$ is observed or the cost budget C is exhausted or exceeded, then **stop** exploration and **goto** step (3).

Figure 3: The GreedyViolate policy.

inequality, $\Pr\left[\sum_{j<i}\left(\mathcal{E}_j + \frac{C_j}{C}\right) < 1\right] \geq \max(0, 1 - \sum_{j<i}\mu_j)$. Note further that for $i \leq k$, we have $\mu_i \leq 1$.

If arm i is played, it yields reward ν_i that directly contributes to the exploitation reward. Since the reward is independent of the event that the arm is reached and played. Therefore, the expected reward of GreedyViolate can be bounded by linearity of expectation as follows.

$$\text{Reward of GreedyViolate} = \mathcal{G} \geq \sum_{i=1}^{k}(1 - \sum_{j<i}\mu_j)\nu_i$$

We now follow the proof idea in [15]. Consider the arms $1 \leq i \leq k$ as deterministic items with item i having profit ν_i and size μ_i. We therefore have $\sum_{i=1}^{k}\nu_i \geq \gamma^*/2$ and $\sum_{i=1}^{k-1}\mu_i \leq 1$.

Suppose these items are placed into a knapsack of size 1 in decreasing order of $\frac{\nu_i}{\mu_i}$ with the last item possibly being fractionally placed. This is the same ordering that the algorithm uses to play the arms. Let $\Phi(q)$ denote the profit when size of the knapsack filled is $q \leq 1$. We have $\Phi(1) \geq \gamma^*/2$. Plot the function $\Phi(q)$ as a function of q. This plot connects the points $\{(0,0), (\mu_1, v_1), (\mu_1 + \mu_2, v_1 + v_2), \ldots (1, \Phi(1))\}$. This function is concave, therefore the area under the curve is at least $\frac{\Phi(1)}{2} \geq \gamma^*/4$. However, the area under this curve is at most

$$v_1 + v_2(1 - \mu_1) + \ldots + v_k(1 - \sum_{j<k}\mu_j) \leq \mathcal{G}$$

Therefore, $\mathcal{G} \geq \gamma^*/4$. Since $OPT \leq \gamma^*$, $\mathcal{G}$ is at least $\frac{OPT}{4}$. □

Theorem 2.3. *The* GreedyOrder *policy with cost budget C achieves reward at least $\frac{OPT}{4}$.*

Proof. Consider the GreedyViolate policy. This policy could exceed the cost budget because the budget was checked only at the end of execution of policy ϕ_i for arm i. Now suppose the play for arm i reaches state $u \in \mathcal{S}_i$, and the next decision of GreedyViolate involves playing arm i and this would exceed the cost budget. The GreedyViolate policy continues to play arm i according to ϕ_i and when the play is finished, it checks the budget constraint, realizes that the budget is exhausted, stops, and chooses arm i for exploitation. Suppose the policy was modified so that instead of the decision to play arm i further at state u, the policy instead checks the budget, realizes it is not sufficient for the next play, stops, and chooses arm i for exploitation. This new policy is precisely GreedyOrder.

Note now that conditioned on reaching node u with the next decision of GreedyViolate being to play arm i, so that the policies GreedyViolate and GreedyOrder diverge in their next action, both policies choose arm i for exploitation. By the martingale property of the rewards, the reward from choosing arm i for exploitation at state u is the same as the expected reward from playing the arm further and then choosing it for exploitation. Therefore, the expected reward of both policies is identical, and the theorem follows. □

2.4 Bi-criteria Result

Suppose we allow the cost budget to be exceeded by a factor $\alpha \geq 1$, so that the cost budget is αC. Consider the GreedyOrder policy where the arms are ordered in decreasing order of $\frac{R(\phi_i)}{\alpha P(\phi_i) + C(\phi_i)/C)}$, and the budget constraint is relaxed to αC. We have the following theorem:

Theorem 2.4. *For any $\alpha \geq 1$, if the cost budget is relaxed to αC, the expected reward of the modified* GreedyOrder *policy is $\frac{\alpha}{2(1+\alpha)}\gamma^*$.*

Proof. We mimic the proof of Theorem 2.2, and define $\nu_i = R(\phi_i)$, and let $\mu_i = P(\phi_i) + \frac{1}{\alpha}C(\phi_i)/C$. Note that the LP satisfies the constraint $\sum_i \left(P(\phi_i) + \frac{1}{\alpha}\frac{C(\phi_i)}{C}\right) \leq \frac{1+\alpha}{\alpha}$. We therefore have $\sum_i \mu_i \leq \frac{1+\alpha}{\alpha}$. Let k denote the smallest integer such that $\sum_{i=1}^{k} \mu_i \geq 1$. By the sorted ordering property, we have $\sum_{i=1}^{k} \nu_i \geq \frac{\alpha}{1+\alpha}\gamma^*$. The rest of the proof remains the same, and we show that the reward of the new policy, $\mathcal{G}$, satisfies: $\mathcal{G} \geq \frac{1}{2}\Phi(1)$, and $\Phi(1) \geq \frac{\alpha}{2(1+\alpha)}\gamma^*$. This completes the proof. □

2.5 Integrality Gap of the Linear Program

We now show via a simple example that the linear program has an integrality gap of at least $e/(e-1) \approx 1.58$. All arms $i = 1, 2, \ldots, n$ have identical 2-level state spaces. Each $\mathcal{S}_i$ has $c_\rho = 1$, $r_\rho = 1/n$, switching cost $h_i = 0$, and two other states u_0 and u_1. We have $\mathbf{p}_{\rho u_0} = 1 - 1/n$, $\mathbf{p}_{\rho u_1} = 1/n$, $r_{u_0} = 0$, $r_{u_1} = 1$. Set $C = n$, so that any policy can play all the arms. The expected reward of such a policy is precisely $1 - (1 - 1/n)^n \approx 1 - 1/e$. The LP solution will set $z^*_\rho = 1$ and $x^*_{u_1} = 1/n$ for all i, yielding an LP objective of 1. This shows that the linear program cannot yield better than a constant factor approximation. It is an interesting open question whether the LP can be strengthened by other convex constraints to obtain tighter bounds (refer for instance [15]).

2.6 Adaptivity Gaps

Recall that a non-adaptive strategy allocates a fixed budget to each arm in advance. It then explores the arms according to these budgets (ignoring the outcome of the plays in choosing the next arm to explore), and at the end of exploration, chooses the best arm for exploitation. This is termed an *allocational strategy* in [33]. Such strategies are desirable since they allow the experimenter to consider various competing arms in parallel. We show two results in this case: For general state spaces, we show that such a non-adaptive strategy can be arbitrarily worse than the optimal adaptive strategy. On the positive side, we show that for 2-level state spaces, which correspond to deterministic underlying rewards (refer Section 1.2), a non-adaptive strategy is only a factor 7 worse than the performance of the optimal adaptive strategy.

We first present an example with unit costs where an adaptive strategy that dynamically allocates the budget achieves far better exploitation gain than a non-adaptive strategy. Note that we can ignore switching costs in such strategies.

Theorem 2.5. *The adaptivity gap of the budgeted learning problem is $\Omega(\sqrt{n})$. Furthermore, even if we allow the non-adaptive exploration to use $\gamma > 1$ times the exploration budget, the adaptivity gap remains $\Omega(\sqrt{n/\gamma})$.*

Proof. Each arm has an underlying reward distribution over the three values $a_1 = 0$, $a_2 = 1/n^9$ and $a_3 = 1$. Let $q = 1/\sqrt{n}$. The underlying distribution could be one of 3 possibilities: R_1, R_2, R_3. R_1 is the deterministic value a_1, R_2 is deterministically a_2 and R_3 is a_3 w.p. q and a_2 w.p. $1-q$. For each arm, we know in advance that $\Pr[R_1] = 1 - q$, $\Pr[R_2] = q(1-q)$ and $\Pr[R_3] = q^2$. Therefore, the knowledge for each arm is a prior over the three distributions R_1, R_2, R_3. The priors for different arms are i.i.d. All $c_i = 1$ and the total budget is $C = 5n$.

We first show that the adaptive policy chooses an arm with underlying reward distribution R_3 with constant probability. This policy first plays each arm once and discards all arms with observed reward a_1. With probability at least $1/2$, there are at most $2/q$ arms which survive, and at least one of these arms has underlying reward distribution R_3. If more arms survive, choose any $2/q$ arms. The policy now plays each of the $2/q$ arms $2\sqrt{n}$ times. The probability that an arm with distribution R_3 yields reward a_3 on some play is at least once is $1 - (1-q)^{2/q} \approx \Theta(1)$. In this

case, it chooses the arm with reward distribution R_3 for exploitation. Since this happens w.p. at least a constant, the expected exploitation reward is $\Theta(q)$. Note that this is best possible to within constant factors, since $\mathbf{E}[R_3] = \Theta(q)$.

Now consider any non-adaptive policy. With probability $1 - 1/n^{\Theta(1)}$, there are at most $2\log n$ arms with reward distribution R_3, and at least $1/(2q)$ arms with reward distribution R_2. Let $r \gg 2\log n$. The strategy allocates at most $5r$ plays to at least $n(1-1/r)$ arms – call this set of arms T. With probability $(1-1/r)^{2\log n} = \Omega(1-(2\log n)/r)$, all arms with reward distribution R_3 lie in this set T. For any of these arms played $O(r)$ times, with probability $1-O(qr)$, all observed rewards will have value a_2. This implies with probability $1 - O(qr)$, all arms with distribution R_3 yield rewards a_2, and so do $\Omega(1/(2q))$ arms with distributions R_2. Since these appear indistinguishable to the policy, it can at best choose one of these at random, obtaining exploitation reward $\frac{q\log n}{2(1/q)} = O(q^2\log n)$. Since this situation happens with probability $1 - O(\log n/r)$, and with the remaining probability the exploitation reward is at most q, the strategy therefore has expected exploitation reward $O(q\log n(\frac{1}{r}+q))$. This implies the adaptivity gap is $\Omega(1/q) = \Omega(\sqrt{n})$ if we set $r = 1/q$.

Now suppose we allow the budget to be increased by a factor of $\gamma > 1$. Then the strategy would allocate at most $5\gamma r$ plays to at least $n(1-1/r)$ arms. By following the same argument as above, the expected reward is $O(q\log n(\frac{1}{r}+q\gamma))$. This proves the second part of the theorem. □

We next show that for 2-level state spaces, which correspond to deterministic underlying rewards (refer Section 1.2), the adaptivity gap is at most a factor of 7.

Theorem 2.6. *If each state space $\mathcal{S}_i$ is a directed star graph with ρ_i as the root, then there is a non-adaptive strategy that achieves reward at least $1/7$ the LP bound.*

Proof. In the case of 2-level state spaces, a non-adaptive strategy chooses a subset S of arms and allocates zero/one plays to each of these so that the total cost of the plays is at most C. We consider two cases based on the LP optimal solution.

In the first case, suppose $\sum_i r_{\rho_i} x_{\rho_i} \geq \gamma^*/7$, then not playing anything but simply choosing the arm with highest r_{ρ_i} directly for exploitation is a 7-approximation.

In the remaining proof, we assume the above is not the case, and compare against the optimal LP solution that sets $x_{\rho_i} = 0$ for all i. This solution has value at least $6\gamma^*/7$. For simplicity of notation, define $z_i = z_{\rho_i}$ as the probability that the arm i is played. Define $X_i = \frac{1}{z_i}\sum_{u\in\mathcal{S}_i} x_u$ as the probability that the arm is exploited conditioned on being played, and $R_i = \frac{1}{z_i}\sum_{u\in S_i} x_u r_u$ as the expected exploitation reward conditioned on being played. Also define $c_i = c_{\rho_i}$. The LP satisfies the constraint: $\sum_i z_i\left(\frac{c_i}{C} + X_i\right) \leq 2$, and the LP objective is $\sum_i z_i R_i$, which has value at least $6\gamma^*/7$.

A better objective for the LP can be obtained by considering the arms in decreasing order of $\frac{R_i}{\frac{c_i}{C}+X_i}$, and increasing z_i in this order until the constraint $\sum_i z_i\left(\frac{c_i}{C} + X_i\right) \leq 1$ becomes tight. Set the remaining $z_i = 0$. It is easy to see $\sum_i z_i R_i \geq \frac{3}{7}\gamma^*$. At this point, let k denote the index of the last arm which could possibly have $z_k < 1$, and let S denote the set of arms with $z_i = 1$ for $i \in S$. There are again two cases.

In the first case, if $z_k R_k > \gamma^*/7$, then choosing just this arm for exploitation has reward at least $\gamma^*/7$, and is a 7-approximation.

In the second and final case, we have a subset of arms $\sum_{i\in S}\left(\frac{c_i}{C} + X_i\right) \leq 1$, and $\sum_{i\in S} R_i \geq \frac{3}{7}\gamma^* - \gamma^*/7 = \frac{2}{7}\gamma^*$. If all these arms are played, the expected number of arms that are exploited is $\sum_{i\in S} X_i \leq 1$, and the expected reward is $\sum_{i\in S} R_i \geq \frac{2}{7}\gamma^*$. The proof of Theorem 2.2 can be adapted to show that choosing the best arm for exploitation yields at least half the reward, *i.e.*, reward at least $\gamma^*/7$. □

3 Lagrangean Version

Recall from Section 1.1 that in the Lagrangean version of the problem, there are no budget constraints on the plays, the goal is to find a policy π such that $R(\pi) - C(\pi)$ is maximized. Denote this quantity as the *profit* of the strategy.

The linear program relaxation is below. The variables are identical to the previous formulation, but there is no budget constraint.

$$\text{Maximize} \quad \sum_{i=1}^{n} \left(\sum_{u \in \mathcal{S}_i} (x_u r_u - c_u z_u) - h_i z_{\rho_i} \right)$$

$$\begin{array}{rcll} \sum_{i=1}^{n} \sum_{u \in \mathcal{S}_i} x_u & \leq & 1 & \\ \sum_{v \in \mathcal{S}_i} z_v \mathbf{p}_{vu} & = & w_u & \forall i, u \in \mathcal{S}_i \setminus \{\rho_i\} \\ x_u + z_u & \leq & w_u & \forall u \in \mathcal{S}_i, \forall i \\ x_u, z_u, w_u & \in & [0,1] & \forall u \in \mathcal{S}_i, \forall i \end{array}$$

Let OPT = optimal net profit and γ^* = optimal LP solution. The next is similar to Claim 2.1.

Claim 3.1. *$OPT \leq \gamma^*$.*

From this LP optimum $\langle w_u^*, x_u^*, z_u^* \rangle$, the policy ϕ_i is constructed as described in Figure 1, and the r.v.'s $\mathcal{E}_i, C_i, R_i$ and their respective expectations $P(\phi_i), C(\phi_i)$, and $R(\phi_i)$ are obtained as described in the beginning of Section 2.2. Let r. v. $Y_i = R_i - C_i$ denote the profit of playing arm i according to ϕ_i. Note that $\mathbf{E}[Y_i] = \left(\sum_{u \in \mathcal{S}_i} (x_u r_u - c_u z_u) - h_i z_{\rho_i} \right)$.

The nice aspect of the proof of Theorem 2.2 is that it does not necessarily require the r.v. corresponding to the reward of policy ϕ_i, R_i to be non-negative. As long as $\mathbf{E}[R_i] = R(\phi_i) \geq 0$, the proof holds. This will be crucial for the Lagrangean version.

Claim 3.2. *For any arm i, $\mathbf{E}[Y_i] = R(\phi_i) - C(\phi_i) \geq 0$.*

Proof. For each i, since all $r_u \geq 0$, setting $x_{\rho_i} \leftarrow \sum_{u \in \mathcal{S}_i} x_u$, $w_{\rho_i} \leftarrow 1$, and $z_u \leftarrow 0$ for $u \in \mathcal{S}_i$ yields a feasible non-negative solution. The LP optimum will therefore guarantee that the term $\sum_{u \in \mathcal{S}_i} (x_u r_u - c_u z_u) - h_i z_{\rho_i} \geq 0$. Therefore, $\mathbf{E}[Y_i] \geq 0$ for all i. □

The GreedyOrder policy orders the arms in decreasing order of $\frac{R(\phi_i) - C(\phi_i)}{P(\phi_i)}$, and plays them according to their respective ϕ_i until some $\mathcal{E}_i = 1$.

Theorem 3.3. *The expected profit of* GreedyOrder *is at least $OPT/2$.*

Proof. Let $\mu_i = P(\phi_i)$ and $\nu_i = \mathbf{E}[Y_i]$ for notational convenience. The LP solution yields $\sum_i \mu_i \leq 1$ and $\sum_i \nu_i = \gamma^*$. Re-number the arms according to the sorted ordering of $\frac{\nu_i}{\mu_i}$ so that the first arm played is numbered 1.

The event that GreedyOrder plays arm i corresponds to $\sum_{j<i} \mathcal{E}_j = 0$. By Markov's inequality, we have $\Pr[\sum_{j<i} \mathcal{E}_j = 0] = \Pr[\sum_{j<i} \mathcal{E}_j < 1] \geq 1 - \sum_{j<i} \mu_j$.

If arm i is played, it yields profit Y_i. This implies the profit of GreedyOrder is $\sum_i Y_i (1 - \sum_{j<i} \mathcal{E}_j)$. Since Y_i is independent of $\sum_{j<i} \mathcal{E}_j$, and since Claim 3.2 implies $\mathbf{E}[Y_i] \geq 0$, the expected profit $\mathcal{G}$ of GreedyOrder can be bounded by linearity of expectation as follows.

$$\mathcal{G} = \sum_i \Pr \left[\sum_{j<i} \mathcal{E}_j < 1 \right] \mathbf{E}[Y_i] \geq \sum_i \nu_i \left(1 - \sum_{j<i} \mu_j \right)$$

We now follow the proof idea in [15]. Consider the arms $1 \leq i \leq n$ as deterministic items with item i having profit μ_i and size μ_i. We therefore have $\sum_i \nu_i \geq \gamma^*$ and $\sum_i \mu_i \leq 1$. Using the same proof idea as in Theorem 2.2, it is easy to see that $\mathcal{G} \geq \frac{\gamma^*}{2}$. Since $OPT \leq \gamma^*$, $\mathcal{G}$ is at least $\frac{OPT}{2}$. □

4 Concave Utility Functions

The above framework in fact solves the more general problem of maximizing any concave stochastic objective function over the rewards of the arms subject to a (deterministic) packing constraint. In what follows, we extend our arguments in the previous section to develop approximation algorithms for all positive concave utility maximization problems in this exploration-exploration setting. Suppose arm i in state $u \in \mathcal{S}_i$ has a value function $g_u(y)$ where $y \in [0,1]$ denotes the weight assigned to it in the exploitation phase. We enforce the following properties on the function $g_u(y)$:

Concavity. $g_u(y)$ is an arbitrary positive non-decreasing concave function of y.

Super-Martingale. $g_u(y) \geq \sum_{v \in \mathcal{S}_i} \mathbf{p}_{uv} g_v(y)$.

Given an outcome $o \in \mathcal{O}(\pi)$ of exploration, suppose arm i ends up in state u, and is assigned weight y_i in the exploitation phase, the contribution of this arm to the exploitation value is $g_u(y_i)$. The assignment of weights is subject to a deterministic packing constraint $\sum_i \sigma_i y_i \leq B$, where $\sigma_i \in [0, B]$. Therefore, for a given outcome $o \in \mathcal{O}(\pi)$, the value of this outcome is given by the convex program:

$$\max \sum_{i=1}^{n} g_u(y_i) \qquad \text{s.t.} \qquad \sum_{i=1}^{n} \sigma_i y_i \leq B, \forall i \quad y_i \in [0,1]$$

The goal as before is to design an adaptive exploration phase π so that the expected exploitation value is maximized, where the expectation is over the outcomes $\mathcal{O}(\pi)$ of exploration and cost of exploration is at most C.

- For the maximum reward problem, $g_u(y) = r_u$, $\sigma_i = 1$, and $B = 1$.
- Suppose we wish to choose the m best rewards, we simply set $B = m$. Note that we can also conceive of a scenario where the c_i correspond to cost of "pilot studies" and each treatment i requires cost σ_i for large scale studies. This would lead us to a KNAPSACK type problem where σ_i are now the "sizes".

4.1 Linear Program

The state space $\mathcal{S}_i$ and the probabilities $\mathbf{p}_{uv}$ are defined just as in Section 1.1. For small constant $\epsilon > 0$, let $L = \frac{n}{\epsilon}$. Discretize the domain $[0,1]$ in multiples of $1/L$. For $l \in \{0, 1, \ldots, L\}$, let $\zeta_u(l) = g_u(l/L)$. This corresponds to the contribution of arm i to the exploitation value on allocating weight $y_i = l/L$. Define the following linear program:

$$\text{Max} \quad \sum_{i=1}^{n} \sum_{u \in \mathcal{S}_i} \sum_{l=0}^{L} x_{ul} \zeta_u(l)$$

$$\begin{aligned}
\textstyle\sum_{i=1}^{n} \left(h_i z_{\rho_i} + \sum_{u \in \mathcal{S}_i} c_u z_u \right) &\leq C \\
\textstyle\sum_{i=1}^{n} \sigma_i \left(\sum_{u \in \mathcal{S}_i} \sum_{l=0}^{L} l x_{ul} \right) &\leq BL(1+\epsilon) \\
\textstyle\sum_{v : u \in D(v)} z_v \mathbf{p}_{vu} &= w_u \quad \forall i, u \in \mathcal{S}_i \setminus \{\rho_i\} \\
\textstyle z_u + \sum_{l=0}^{L} x_{ul} &\leq w_u \quad \forall u \in \mathcal{S}_i, \forall i \\
w_u, x_{ul}, z_u &\in [0,1] \quad \forall u \in \mathcal{S}_i, \forall i, l
\end{aligned}$$

Policy ϕ_i: If arm i is currently in state u, choose $q \in [0, w_u^*]$ u.a.r. and do one of the following:

1. If $q \in [0, z_u^*]$, **then** play the arm.
2. **else** Stop executing ϕ_i.
   Find the smallest $l \geq 0$ such that $q \leq z_u^* + \sum_{k=0}^{l} x_{uk}^*$. Set $\mathcal{E}_i = \frac{l}{L}$ and $R_i = \zeta_u(l)$.

Figure 4: The policy ϕ_i for concave value functions.

Let γ^* be the optimal LP value and OPT = value of the optimal adaptive exploration policy.

Lemma 4.1. *$OPT \leq \gamma^*$.*

Proof. In the optimal solution, let w_u denote the probability that the policy reaches state $u \in \mathcal{S}_i$, and let z_u denote the probability of reaching state $u \in \mathcal{S}_i$ and playing arm i in this state. For $l \geq 1$, let x_{ul} denote the probability of stopping exploration at $u \in \mathcal{S}_i$ and allocating weight $y_i \in (\frac{l-1}{L}, \frac{l}{L}]$ to arm i. All the constraints are straightforward, except the constraint involving B. Observe that if the weight assignments y_i in the optimal solution were rounded up to the nearest multiple of $1/L$, then the total size of any assignment increases by at most ϵB since all $s_i \leq B$. Therefore, this constraint is satisfied. Using the same rounding up argument, if the weight satisfies $y_i \in (\frac{l-1}{L}, \frac{l}{L}]$, then the contribution of arm i to the exploitation value is upper bounded by $\zeta_u(l)$ since the function $g_u(y)$ is non-decreasing in y. Therefore, the proof follows. □

4.2 Exploration Policy

Let $\langle w_u^*, x_{ul}^*, z_u^* \rangle$ denote the optimal solution to the LP. Assume $w_{\rho_i}^* = 1$ for all i. Also w.l.o.g, $z_u^* + \sum_{l=0}^{L} x_{ul}^* = w_u^*$ for all $u \in \mathcal{S}_i$. The LP solution yields a natural (infeasible) exploration policy ϕ consisting of one independent policy ϕ_i per arm i. Policy ϕ_i is described in Figure 4.

The policy ϕ_i is independent of the states of the other arms. It is easy to see by induction that if state $u \in \mathcal{S}_i$ is reached by the policy with probability w_u^*, then state $u \in \mathcal{S}_i$ is reached *and* arm i is played with probability z_u^*. Let random variable C_i denote the cost of executing ϕ_i, and let $C(\phi_i) = \mathbf{E}[C_i]$. Denote this overall policy ϕ – this corresponds to one independent decision policy ϕ_i (determined by $\langle w_u^*, x_{ul}^*, z_u^* \rangle$) per arm. It is easy to see that the following hold for ϕ:

1. $C(\phi_i) = \mathbf{E}[C_i] = h_i z_{\rho_i}^* + \sum_{u \in \mathcal{S}_i} c_u z_u^*$ so that $\sum_i C(\phi_i) \leq C$.
2. $P(\phi_i) = \mathbf{E}[\mathcal{E}_i] = \frac{1}{L} \sum_{u \in \mathcal{S}_i} \sum_{l=0}^{L} l x_{ul}^* \quad \Rightarrow \quad \sum_i \sigma_i P(\phi_i) \leq B(1+\epsilon)$.
3. $R(\phi_i) = \mathbf{E}[R_i] = \sum_{u \in \mathcal{S}_i} \sum_{l=0}^{L} x_{ul}^* \zeta_u(l) \quad \Rightarrow \quad \sum_i R(\phi_i) = \gamma^*$.

The GreedyOrder policy is presented in Figure 5. We again use an infeasible policy GreedyViolate which is simpler to analyze. The algorithm is the same as GreedyOrder except for step (2), where violation of the cost constraint is only checked after the policy ϕ_j terminates.

Theorem 4.2. *Let $c_{\max}$ denote the maximum cost of exploring a single arm. Then* GreedyViolate *spends cost at most $C + c_{\max}$ and has expected value $\frac{OPT}{8}(1-\epsilon)$.*

Proof. Let $\nu_i = R(\phi_i)$ and let $\mu_i = \frac{\sigma_i}{B} P(\phi_i) + \frac{1}{C} C(\phi_i)$. The LP constraints imply that $\gamma^* = \sum_i \nu_i$, and $\sum_i \mu_i \leq 2 + \epsilon$. Now using the same proof as Theorem 2.2, we obtain the value $\mathcal{G}$ of GreedyViolate according to the weight assignment $\mathcal{E}_i$ at the end of Step (2) is at least

Algorithm GreedyOrder

1. Order the arms in decreasing order of $\frac{R(\phi_i)}{\frac{\sigma_i}{B}P(\phi_i)+\frac{1}{C}C(\phi_i)}$.
2. For each arm j in sorted order, play it according to ϕ_j as follows until ϕ_j terminates:
   (a) If the next play would violate the cost constraint, then set $\mathcal{E}_j \leftarrow 1$, **stop** exploration, and **goto** step (3).
   (b) If ϕ_j terminates and $\sum_i \sigma_i \mathcal{E}_i \geq B$, then **stop** exploration and **goto** step (3).
   (c) Else, play arm j according to policy ϕ_j and **goto** step (2a).
3. **Exploitation:** Scale down $\mathcal{E}_i$ by a factor of 2.

Figure 5: The GreedyOrder policy for concave functions.

$\frac{OPT}{4}(1-\epsilon)$. This weight assignment could be infeasible because of the last arm, so that the $\mathcal{E}_i$ only satisfy $\sum_i \sigma_i \mathcal{E}_i \leq 2B$. This is made feasible in Step (3) by scaling all $\mathcal{E}_i$ down by a factor of 2. Since the functions $g_i(y)$ are concave in y, the exploitation value reduces by a factor of $1/2$ because of scaling down. □

Theorem 4.3. GreedyOrder *policy with budget C achieves expected value at least $\frac{OPT}{8}(1-\epsilon)$.*

Proof. Consider the GreedyViolate policy. Now suppose the play for arm i reaches state $u \in \mathcal{S}_i$, and the next decision of GreedyViolate involves playing arm i and this would exceed the cost budget. Conditioned on this next decision, GreedyOrder sets $\mathcal{E}_i = 1$ and stops exploration. In this case, the exploitation value of GreedyOrder from arm i is at least the expected exploitation gain of GreedyViolate for this arm by the super-martingale property of the value function g. Therefore, for the assignments at the end of Step (2), the gain of GreedyOrder is at least $\frac{OPT}{4}(1-\epsilon)$. Since Step (3) scales the $\mathcal{E}$'s down by a factor of 2, the theorem follows. □

5 Conclusions

We studied a variant of the classical stochastic multi-armed bandit problem where the exploration phase is expensive and necessarily precedes the exploitation phase. We showed that this model is relevant to settings involving data acquisition and design of experiments, and can be extended to incorporate costs of switching into an arm. We showed that the Gittins index policy for the classic multi-armed bandit problem does not yield an optimal algorithm for this variant; in fact, this variant is NP-Hard. We considered several extensions of this basic problem and presented constant factor approximation algorithms. These algorithms proceed via LP rounding and show a surprising connection to stochastic packing algorithms.

There are two key aspects incompletely addressed in this work, and which form challenging open questions. First is to show APX-Hardness for the budgeted learning problem. Secondly, our approach does not converge to the optimal strategy when either exploration becomes very cheap or the budget constraint is relaxed. It would therefore be interesting to analyze non LP-based policies, such as adaptations of the Gittins index scheme and experts algorithms, in our setting.

Acknowledgement: We would like to thank Jen Burge, Vincent Conitzer, Ashish Goel, Ronald Parr, Fernando Pereira, and Saswati Sarkar for helpful discussions.

References

[1] Q. An, H. Li, X. Liao, and L. Carin. Active feature acquisition with POMDP models. *Submitted to Pattern Recognition Letters*, 2006.

[2] P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[4] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *Proc. of the 1995 Annual Symp. on Foundations of Computer Science*, pages 322–331, 1995.

[5] B. Babcock and C. Olston. Distributed top-k monitoring. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 28–39, 2003.

[6] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, second edition, 2001.

[7] D. Bertsimas, D. Gamarnik, and J. Tsitsiklis. Performance of multiclass markovian queueing networks via piecewise linear Lyapunov functions. *Annals of Applied Probability*, 11(4):1384–1428, 2002.

[8] D. Bertsimas and J. Nino-Mora. Conservation laws, extended polymatroids and multi-armed bandit problems: A unified polyhedral approach. *Math. of Oper. Res.*, 21(2):257–306, 1996.

[9] D. Bertsimas, I. Paschalidis, and J. N. Tsitsiklis. Optimization of multiclass queueing networks: Polyhedral and nonlinear characterizations of achievable performance. *Annals of Applied Probability*, 4(1):43–75, 1994.

[10] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997.

[11] M. Charikar, C. Chekuri, and M. Pál. Sampling bounds for stochastic optimization. In *APPROX-RANDOM*, pages 257–269, 2005.

[12] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. In *Advances in Neural Information Processing Systems*, pages 705–712, 1995.

[13] V. Conitzer and T. Sandholm. Definition and complexity of some basic metareasoning problems. In *IJCAI*, pages 1099–1106, 2003.

[14] D. P. de Farias and N. Megiddo. Combining expert advice in reactive environments. *J. ACM*, 53(5):762–799, 2006.

[15] B. Dean. *Approximation Algorithms for Stochastic Scheduling Problems*. PhD thesis, MIT, 2005.

[16] B. C. Dean, M. X. Goemans, and J. Vondrak. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 208–217, 2004.

[17] B. C. Dean, M. X. Goemans, and J. Vondrák. Adaptivity and approximation for stochastic packing problems. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 395–404, 2005.

[18] A. Flaxman, A. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Annual ACM-SIAM Symp. on Discrete Algorithms*, 2005.

[19] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. *Progress in statistics (European Meeting of Statisticians)*, 1972.

[20] A. Goel, S. Guha, and K. Munagala. Asking the right questions: Model-driven optimization using probes. In *Proc. of the 2006 ACM Symp. on Principles of Database Systems*, 2006.

[21] A. Goel and P. Indyk. Stochastic load balancing and related problems. In *Proc. of the 1999 Annual Symp. on Foundations of Computer Science*, 1999.

[22] S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *Proc. ACM Symp. on Theory of Computing (STOC)*, 2007.

[23] S. Guha and K. Munagala. Model driven optimization using adaptive probes. In *Proc. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2007.

[24] S. Guha, K. Munagala, and S. Sarkar. Jointly optimal probing and transmission strategies for multi-channel wireless systems. *Submitted to IEEE Transactions on Information Theory*, 2006. Available at `http://www.cs.duke.edu/~kamesh/partialinfo.pdf`.

[25] S. M. Kakade and M. J. Kearns. Trading in markovian price models. In *COLT*, pages 606–620, 2005.

[26] R. Karp and R. Kleinberg. Noisy binary search and its applications. In *Proc. 18th ACM-SIAM Symp. on Discrete Algorithms (SODA 2007)*, pages 881–890, 2007.

[27] M. J. Kearns, Y. Mansour, and A. Y. Ng. Approximate planning in large POMDPs via reusable trajectories. In *NIPS*, pages 1001–1007, 1999.

[28] J. Kleinberg, Y. Rabani, and É. Tardos. Allocating bandwidth for bursty connections. *SIAM J. Comput*, 30(1), 2000.

[29] A. J. Kleywegt, A. Shapiro, and T. Homem de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM J. on Optimization*, 12(2):479–502, 2002.

[30] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[31] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

[32] S. H. Low and D. E. Lapsley. Optimization flow control-I: Basic algorithm and convergence. *IEEE/ACM Trans. Netw.*, 7(6):861–874, 1999.

[33] O. Madani, D. J. Lizotte, and R. Greiner. Active model selection. In *UAI '04: Proc. 20th Conf. on Uncertainty in Artificial Intelligence*, pages 357–365, 2004.

[34] R. H. Mohring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of LP-based priority policies. *J. ACM*, 46(6):924–942, 1999.

[35] A. Moore, J. Schneider, J. Boyan, and M. Lee. Q2: Memory-based active learning for optimizing noisy continuous functions. *International Conference on Machine Learning*, 1998.

[36] R. H. Myers and D. C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments (2^{nd} ed.)*. Wiley, 2002.

[37] J. Niño-Mora. Restless bandits, partial conservation laws and indexability. *Adv. in Appl. Probab.*, 33(1):76–98, 2001.

[38] J. Nińo-Mora. Computing an index policy for bandits with switching penalties. In *ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools*, pages 1–10, 2007.

[39] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin American Mathematical Society*, 55:527–535, 1952.

[40] J. Schneider and A. Moore. Active learning in discrete input spaces. In 34^{th} *Interface Symp.*, 2002.

[41] D. Shmoys and C. Swamy. Stochastic optimization is (almost) as easy as discrete optimization. In *Proc.* 45^{th} *IEEE Symp. on Foundations of Computer Science*, pages 228–237, 2004.

[42] A. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang. A sampling based approach to optimizing top-k queries in sensor networks. In *Proc. of the Intl. Conf. on Data Engineering*, 2006.

[43] M. Skutella and M. Uetz. Scheduling precedence-constrained jobs with stochastic processing times on parallel machines. In *Proc.* 12^{th} *ACM-SIAM Symp. on Discrete algorithms*, pages 589–590, 2001.

[44] J. N. Tsitsiklis. A short proof of the Gittins index theorem. *Annals of Appl. Prob.*, 4(1):194–199, 1994.

[45] P. Whittle. Restless bandits: Activity allocation in a changing world. *Appl. Prob.*, 25(A):287–298, 1988.