

Approximation Algorithms for Partial-information based Stochastic Control with Markovian Rewards

Sudipto Guha*

Department of Computer and Information Sciences,
University of Pennsylvania,
Philadelphia PA 19104-6389.

Email: sudipto@cis.upenn.edu.

Kamesh Munagala[†]

Department of Computer Science,
Duke University,
Durham NC 27708-0129.

Email: kamesh@cs.duke.edu.

Abstract

We consider a variant of the classic multi-armed bandit problem (MAB), which we call FEEDBACK MAB, where the reward obtained by playing each of n independent arms varies according to an underlying on/off Markov process with known parameters. The evolution of the Markov chain happens irrespective of whether the arm is played, and furthermore, the exact state of the Markov chain is only revealed to the player when the arm is played and the reward observed. At most one arm (or in general, M arms) can be played any time step. The goal is to design a policy for playing the arms in order to maximize the infinite horizon time average expected reward. This problem is an instance of a Partially Observable Markov Decision Process (POMDP), and a special case of the notoriously intractable “restless bandit” problem. Unlike the stochastic MAB problem, the FEEDBACK MAB problem does not admit to greedy index-based optimal policies. The state of the system at any time step encodes the beliefs about the states of different arms, and the policy decisions change these beliefs – this aspect complicates the design and analysis of simple algorithms.

We design a constant factor approximation to the FEEDBACK MAB problem by solving and rounding a natural LP relaxation to this problem. As far as we are aware, this is the first approximation algorithm for a POMDP problem.

1 Introduction

In this paper, we design approximately optimal policies for a natural variant of the classic multi-armed bandit

(MAB) problem which we term FEEDBACK multi-armed bandits. This variant is motivated by the following application scenario: Suppose a transmitter has access to n wireless channels, and can choose any one of these channels to transmit on every time-slot. The quality of each channel varies with time according to a bursty on/off process whose transition probabilities are known to the transmitter. The transmission rate for using a channel depends on the quality of the channel. The transmitter gets feedback about the instantaneous quality of a channel when it transmits on the channel, since the achievable rate of transmission depends on the quality of the channel. The transmitter can use this feedback to guide future transmission decisions. The goal of the transmitter is to come up with a policy for deciding which channel to transmit on every time slot based on the outcomes of past decisions, to maximize the infinite horizon time average transmission rate.

We model the above problem as a variant of the multi-armed bandits problem as follows: There is a bandit with n independent arms. At each discrete time step, playing an arm yields reward. The reward of arm i evolves according to a two-state bursty Markovian process. Denote the two states “good” and “bad”. The reward from playing the arm when it is in “good” state is $r_i > 0$, and when it is in “bad” state is 0. When the reward is r_i , at the next time step, the reward is 0 w.p. β_i and remains r_i with the remaining probability. Similarly, when the reward is 0, the reward at the next time step is r_i w.p. α_i , and remains 0 with the remaining probability. Note that different arms could have different rewards and burst lengths. When the arm is played, the observed reward (r_i or 0) reveals to the player the current state of the arm. We emphasize that the state of an arm is observed only when the arm is played; at other time instants, the player has a belief about possible states of the arm based on past observations about its state. The goal is to design a policy which plays *at most one* arm per time step, so that

*Research supported in part by an Alfred P. Sloan Research Fellowship and by an NSF Award CCF-0644119

[†]Research supported in part by NSF CNS-0540347.

the infinite horizon time averaged reward is maximized. We term this new variant the FEEDBACK MAB problem, since playing the arm yields feedback about the current state of the arm. We note that with some technical changes, our results can be extended to the case where $M \geq 1$ arms are allowed to be played every time step.

Related Problems and Technical Hurdles: MAB problems model the trade-off between exploration, *i.e.*, learning about the state of the system, and exploitation, *i.e.*, using the information about the state of the system to perform better actions. In the FEEDBACK MAB problem, playing an arm which was just observed to be in good state (exploitation) trades off with playing an arm which was observed a while ago to be bad, but which may have become good now and yield larger reward.

The most well-studied “classical” variant of the MAB problem is the *stochastic* MAB problem [7, 3]. One arm can be played in each time slot, and the random reward obtained depends on the state of the arm played. The state of an arm changes *only* when the arm is played, and otherwise remains the same. The player therefore knows the current states of all the arms. The goal is to maximize the infinite horizon discounted reward. The ingeniously simple and elegant greedy Gittins index technique [7] computes the optimal policy. An index policy computes an *index* for each arm based on just its current state, and plays the arm with the highest index. The FEEDBACK MAB problem has two fundamental differences: (1) The state of the arm evolves irrespective of whether it is played; and (2) The process of playing reveals information about the state of the arm, but does not change the state evolution of the arm itself – the play is merely an observation. Our model is therefore more appropriate for situations as the example of wireless channels, where the arm corresponds to a channel which the player can only observe but not control.

Designing policies for the FEEDBACK MAB problem poses interesting challenges. At any point in time, each arm has been played a certain number of time steps ago. When an arm was last played, the player knew its reward state (or simply, state) precisely. However, since the time of last play, the state of the arm has evolved according to its Markov process. *The exact trajectory of evolution is unknown to the player.* The partial information with the player yields a belief over the possible current states of the arm, which in our case simply means a probability distribution over the arm currently being in good or bad state. The player uses this partial information in making the decision about which arm to play next, which in turn affects the information at future time steps. This problem is therefore a special case of Partially Observable Markov Decision Problems (POMDPs), which are in general notoriously intractable [3].

A natural question in this regard is: Does the optimal policy for FEEDBACK MAB have a simple characterization? For instance, is an index policy analogous to the Gittins index scheme optimal? (Index policies are desirable since they can be compactly represented, and are the heuristic method of choice for several MDP problems.) The following three-arm example shows the contrary: $(\alpha_1, \beta_1, r_1) = (0.4, 0, 1)$, and $(\alpha_2, \beta_2, r_2) = (\alpha_3, \beta_3, r_3) = (0.1, 0.1, 2)$. Note that arm 1 yields a deterministic reward of 1, while arms 2 and 3 are *i.i.d.* Any policy is symmetric w.r.t. arms 2 and 3. The optimal policy exhibits the following non-index behavior where given the beliefs about arms 1 and 2, the decision to play switches between these arms depending on the belief about arm 3. If arm 2 was observed to be “bad” 4 steps ago, then: (i) If arm 3 was “bad” 2 steps ago, the policy plays arm 1; (ii) If arm 3 was “bad” 3 steps ago, the policy plays arm 2. This policy has average reward 1.4622. The policy that flips the former decision to play arm 2 has reward 1.4617, and the policy that flips the latter decision to play arm 1 has reward 1.4616, implying the “non-index” part affects the solution value by a constant factor. In this sense, the FEEDBACK MAB is structurally different from the stochastic MAB problem. The obvious question in the wake of the above discussion is: *Are there simple policies which are provably near optimal?* This is the central question we focus on, and we show that simple “index-type” policies that arise from a natural linear program relaxation are indeed near optimal.

The FEEDBACK MAB problem is indeed a special case of a well-studied generalization of the stochastic MAB problem, called the *restless* bandits problem [15, 4, 17]. In this problem, the state of the arm changes according to a passive transition matrix when the arm is not being played, and according to an active transition matrix when the arm is played. In the former case, the arm yields a state-dependent passive reward, and in the latter case, a state-dependent active reward. Papadimitriou and Tsitsiklis [16] show that this problem is PSPACE-hard; their reduction in fact shows that attaining any non-trivial approximation factor is also intractable. The FEEDBACK MAB problem is a special case of restless bandits where the active and passive matrices have the following relationship: The active matrix is the actual transition matrix for the underlying Markov chain, and the passive matrix corresponds to the evolution of the belief about the state of the arm when it is not being played. The passive rewards are all zero.

A widely used heuristic approach for the restless bandits problem is to convert the solution of poly-size linear programming relaxations into index policies. This approach was pioneered by Whittle [17], and extensively studied by Bertsimas and Niño-Mora [4]. However, in

view of the intractability of the restless bandits problem and the difficulty in analyzing index policies, no performance guarantees are known for this approach. The key contribution of this paper is to study the performance of the linear programming relaxation for the FEEDBACK MAB problem, and show that an extremely simple index-type policy has a constant factor approximation ratio against the LP bound.

In previous work [8, 10, 11, 9], we considered a very different version of the MAB problem, which we termed budgeted learning, where the exact reward distribution of any arm is unknown to the player, but a prior on the possible distributions is specified. The reward distribution is *time-invariant*, and although a greedy policy based on the solution of a linear programming relaxation achieves constant approximation ratio, the basic problem and the required techniques are very different from the version considered here. Several researchers have considered the question of performing learning and prediction tasks when the underlying rewards are adversarial and drifting. These problems are referred to as the *adversarial* MAB problems or *experts* problems [14, 1, 5, 6, 12], and the goal is to compete with the algorithm which always plays one arm, but with the benefit of hindsight. The FEEDBACK MAB problem is *not* a learning problem since the parameters of the underlying Markov process are known, and the uncertainty is the result of partial information about the states of the arms, which in turn depends on previous policy actions.

Our Contribution: We show that for the FEEDBACK MAB problem, there is a natural poly-time solvable linear programming relaxation whose solution can be converted to a feasible policy whose reward is within a constant factor of the optimal reward of any ergodic strategy. As far as we are aware, this is the first provable analysis of a linear programming relaxation for a POMDP or for a restless bandit problem.

Analysis of policies for FEEDBACK MAB is made difficult because the plays for different arms happen at mutually exclusive time instants so that the beliefs about the states of the arms tend to get correlated. One solution is to make the plays memoryless, as in the rounding schemes for machine replenishment [2, 13]. However, making the exploitation memoryless is not desirable since it should obviously depend on the state of the arm. We get around this difficulty by developing a rounding scheme which in some sense preserves independent beliefs about the states of the arms by performing the exploration in a memoryless fashion. Our scheme finds the middle ground in having just the right amount of correlation to obtain a large fraction of the reward, while at the same time, keeping the process analytically tractable. The analysis involves extracting a so-

lution with specific structure from the LP optimum, and then performing a sequence of modifications on the LP solution to extract a policy with the desired properties. Specially **the roadmap** we follow is:

- We present the definitions and the natural LP formulation in Section 2. Our LP formulation does not have polynomial size. We show via the Lagrangean that the formulation is equivalent to a poly-time solvable non-linear function maximization.
- Subsequently in Section 3 we show that either (i) we have identified a single arm which approximates the entire problem or (ii) we can identify a subset of arms and rates at which they should be played which together give us an infeasible but near optimal solution.
- In Section 4 we show how to modify the processes found in step (ii) to nice memoryless variants. In Section 5 we combine these to obtain a constant factor approximation.

Our policy, though not an index policy, has a natural interpretation in terms of index policies [17, 4]. Furthermore, it has a linear size specification, which is desirable when the policy needs to be executed with severe computational resource constraints, such as in a wireless node. We leave open the question of whether our algorithm can be derandomized to extract an index policy with similar performance guarantees.

2 Problem Statement and LP formulation

The bandit has n independent arms. Arm i has two states: The good state g_i yields reward r_i , and the bad state b_i yields no reward. The evolution of state of the arm follows a Markovian process which does not depend on whether the arm is played or not at a time slot. Let s_{ik} denote the state of arm i at time k . Then denote $\Pr[s_{i(k+1)} = g_i | s_{ik} = b_i] = \alpha_i$ and $\Pr[s_{i(k+1)} = b_i | s_{ik} = g_i] = \beta_i$. The evolution of states for different arms are independent. Any policy chooses exactly *one arm* to play every time slot. This yields reward depending on the state of the arm, and in addition, reveals to the policy the current state of that arm. The goal of the policy is to judiciously play the arms to maximize the infinite horizon time average reward.

Assumptions: To ensure ergodicity and “burstiness”, assume $\alpha_i, \beta_i \leq 1/2(1 - \delta)$ for some small $\delta > 0$ specified as part of the input. For convenience, assume $\beta_i, \alpha_i + \beta_i \geq \delta$ (our proofs go through with minor changes even without this assumption).

Definition 1. For integer $k \geq 1$, define: $v_{ik} = \Pr[s_{ik} = g_i | s_{i0} = b_i]$ $u_{ik} = \Pr[s_{ik} = g_i | s_{i0} = g_i]$

Lemma 2.1. *Focus on a particular arm i , and omit the subscript i . For integer $k \geq 1$, we have $v_{k+1} = (1 - \alpha - \beta)v_k + \alpha$, and $u_{k+1} = (1 - \alpha - \beta)u_k + \alpha$. Combined with $v_1 = \alpha$ and $u_1 = 1 - \beta$ we have*

$$v_k = \frac{\alpha}{\alpha + \beta}(1 - (1 - \alpha - \beta)^k)$$

$$1 - u_k = \frac{\beta}{\alpha + \beta}(1 - (1 - \alpha - \beta)^k)$$

It follows that u_k is a monotonically non-increasing function of k while v_k is monotonically non-decreasing. Furthermore, $u_\infty = v_\infty = \frac{\alpha}{\alpha + \beta}$.

The proof of the above lemma is simple, and omitted. We now state a useful fact, which will be used extensively later on in the paper:

Fact 2.2. *For any $t \geq 1$ and for any $x \in [0, 1]$, we have $(1 + tx)(1 - x)^t \leq 1$.*

2.1 The Linear Program

We present a constant factor approximation to the above problem by rounding a linear programming relaxation. The LP as stated has infinitely many variables and constraints; however, we show in the next section that the optimal solution has a very simple structure, and furthermore, a near-optimal solution with the same structure can be constructed in polynomial time by performing elementary function maximization. This structure in the constructed solution will be critically used in the rounding scheme later.

For any feasible policy, suppose at any time instant, arm i was observed to be in state b_i some $k \geq 1$ steps ago. Then, if the policy plays arm i , this arm will be observed to be in state g_i with probability u_{ik} and in state b_i with probability $1 - u_{ik}$. Therefore, the expected reward from this play is $r_i u_{ik}$, and the belief about the state of the arm collapses to either b_i or g_i . This motivates the following notation: Let x_{ik-1} denote the probability that at the beginning of a random time instant, it is the case that the policy played and observed arm i in state g_i exactly $k \geq 1$ steps ago. Let x_{ik}^c denote the probability that the previously mentioned event happens and arm i is played. Let y_{ik-1} denote the probability that at the beginning of a random time instant, it is the case that the policy observed arm i in state b_i exactly $k \geq 1$ steps ago. Let y_{ik}^c denote the probability that the state of arm i was observed k steps ago to be in state b_i , and arm i is played. Note that these probability values are well-defined since the optimal policy is ergodic. The following linear program MAINLP upper bounds the value of the optimal policy.

$$\text{Maximize } \sum_{i=1}^n r_i \sum_{k \geq 1} (u_{ik} x_{ik}^c + v_{ik} y_{ik}^c)$$

$$\begin{aligned} \sum_{i=1}^n \sum_{k \geq 1} (x_{ik}^c + y_{ik}^c) &= 1 && \forall i \\ \sum_{k \geq 0} x_{ik} + y_{ik} &= 1 && \forall i \\ \sum_{k \geq 1} (x_{ik}^c u_{ik} + y_{ik}^c v_{ik}) &= x_{i0} && \forall i \\ \sum_{k \geq 1} (x_{ik}^c (1 - u_{ik}) + y_{ik}^c (1 - v_{ik})) &= y_{i0} && \forall i \\ x_{ik-1} &= x_{ik}^c + x_{ik} && \forall i, k \geq 1 \\ y_{ik-1} &= y_{ik}^c + y_{ik} && \forall i, k \geq 1 \\ x_{ik}, y_{ik}, x_{ik}^c, y_{ik}^c &\in [0, 1] && \forall i, k \geq 0 \end{aligned}$$

We will next show that an arbitrarily good approximation to the above can be obtained in polytime.

2.2 Solving the Linear Program

To solve MAINLP, we will consider a Lagrangean relaxation whose optimal solution replaces the large number of variables and constraints in the LP with a simple function maximization. We subsequently perform binary search on the Lagrange multiplier to find the optimal solution to MAINLP.

Lagrangean Formulation: In MAINLP, the only constraint which runs across different arms is the constraint:

$$\sum_{i=1}^n \sum_{k \geq 1} (x_{ik}^c + y_{ik}^c) = 1$$

The above constraint effectively replaces the hard constraint of playing one arm per time step with the same constraint in expectation over time. In that sense, MAINLP is a relaxation of the optimal policy. We absorb this constraint into the objective via Lagrange multiplier $\lambda \geq 0$ to obtain the following Lagrangean objective:

$$\text{Max. } \lambda + \sum_{i=1}^n \sum_{k \geq 1} ((r_i u_{ik} - \lambda) x_{ik}^c + (r_i v_{ik} - \lambda) y_{ik}^c)$$

This maximization is subject to the remaining constraints in the above LP. Denote this Lagrangean formulation LPLAGRANGE(λ). Let $G(\lambda)$ denote its optimal solution value. The optimal solution for LPLAGRANGE(λ) now yields n separate maximization problems, one for each arm. The maximization problem for arm i exactly encodes the following policy design question: At any time step, the arm can be played (and reward obtained from it), or not played. Whenever the arm is played, we incur a penalty λ in addition to the reward. The goal is to maximize the difference between the expected reward and the expected penalty. Note that if the penalty is sufficiently large, the optimal solution would be to never play the arm. Let $L(i, \lambda)$ denote the optimal policy for arm i , with value $W(i, \lambda)$, so that $G(\lambda) = \sum_{i=1}^n W(i, \lambda)$.

We first show that the optimal policy $L(i, \lambda)$ for any arm i belongs to the class of policies $\mathcal{P}_i(k)$ for $k \geq 1$, whose specification is presented in Figure 1.

Policy $\mathcal{P}_i(k)$:

1. If the arm was just observed to be in state g_i , then play the arm.
2. If the arm was just observed to be in state b_i , wait $k - 1$ steps and play the arm.

Figure 1. The Policy $\mathcal{P}_i(k)$.

Intuitively, *step (1) corresponds to exploitation, and step (2) to exploration*. Set $\mathcal{P}_i(\infty)$ to be the policy that never plays the arm. The policy $\mathcal{P}_i(k)$ for arm i corresponds to setting the variables as follows: $x_{i0} = x_{i1}^c$, $y_{ij}^c = 0$ for $j = 0, 1, \dots, k - 1$, and $y_{ik}^c = y_{i0}$.

Lemma 2.3. $L(i, \lambda) \in \{\mathcal{P}_i(k), k \geq 1\}$.

Proof. The policy $L(i, \lambda)$ has deterministic actions for each possible state of the arm since it encodes the optimal solution to a MDP [3]. If such a policy, on observing the arm in state g_i , waits k' steps and plays the arm for the first time, a better solution would be to play the arm immediately. This is intuitively clear and can be rigorously proved as follows. A Markov chain analysis similar to the proof of Lemma 2.4 below shows that the value of the solution (omitting subscript i) would be:

$$\frac{\frac{(r-\lambda)v_k}{1-u_{k'}} - \lambda}{v_k \frac{k'}{1-u_{k'}} + k}$$

The function $1/(1 - u_\ell)$ is monotonically decreasing in ℓ , and the function $\ell/(1 - u_\ell)$ is monotonically increasing in ℓ . The latter follows by an application of Fact 2.2 using $t = \ell, x = \alpha + \beta$. Therefore, setting $k' = 1$ maximizes the above expression.

Therefore, the only parameter that is needed to specify the policy completely is the number k^* of steps that the policy waits on observing a b_i before playing the arm. Such a policy is precisely $\mathcal{P}_i(k^*)$ \square

Solving LPLAGRANGE(λ): For policy $\mathcal{P}_i(k)$ executed for arm i , let $R_i(k)$ denote the expected reward (ignoring the penalty λ) from playing the arm, and let $Q_i(k)$ denote the steady-state probability (or rate) with which the arm is played. Let $F_i(\lambda, k) = R_i(k) + \lambda(1 - Q_i(k))$. Then it is easy to see the following:

$$W(i, \lambda) = \max_{k \geq 1} F_i(\lambda, k)$$

We will now derive a closed form expression for the quantity $F_i(\lambda, k)$. A similar derivation yields the expression used in the proof of Lemma 2.3 as well. *In the lemmas below, we omit the subscript i for notational simplicity.* Recall that v_k is the probability that the arm is in state g at the current step given that it was in state b some k steps ago.

Lemma 2.4. $R(k) = r \frac{v_k}{v_k + k\beta}$ and $Q(k) = \frac{v_k + \beta}{v_k + k\beta} \geq \frac{1}{k}$.

Proof. The Markov chain describing the policy $\mathcal{P}(k)$ is shown in Figure 4(a), and has $k + 1$ states which we denote $s, 0, 1, 2, \dots, k - 1$. The state s corresponds to the arm being observed to be in state g , and the state j corresponds to the arm being observed in state b exactly j steps ago. The transition probability from state j to state $j + 1$ is 1, from state s to state 0 is β , from state $k - 1$ to state s is v_k , and from state k to state 0 is $1 - v_k$. Let $\pi_s, \pi_0, \pi_1, \dots, \pi_{k-1}$ denote the steady state probabilities of being in states $s, 0, 1, \dots, k - 1$ respectively. This Markov chain is easy to solve. We have $\pi_0 = \pi_1 \dots = \pi_{k-1}$, so that the first identity is: $\pi_s + k\pi_0 = 1$. Furthermore, by considering transitions into and out of s , we obtain: $\beta\pi_s = v_k\pi_{k-1} = v_k\pi_0$. Combining these, we obtain: $\pi_s = \frac{v_k}{v_k + k\beta}$, and $\pi_0 = \frac{\beta}{v_k + k\beta}$. Now we have:

$$R(k) = r[(1 - \beta)\pi_s + v_k\pi_0] = r\pi_s = r \frac{v_k}{v_k + k\beta}$$

$$Q(k) = \pi_s + \pi_{k-1} = \frac{v_k + \beta}{v_k + k\beta}$$

\square

Note: In the above $k = 1$ implies $Q(k) = 1$. We now have the following lemma, which implies a poly-time solution for LPLAGRANGE(λ) to arbitrary precision.

Lemma 2.5.

$$W(\lambda) = \max_{k \geq 1} F(\lambda, k) = \max_{k \geq 1} \left(\lambda + \frac{(r - \lambda)v_k - \lambda\beta}{v_k + k\beta} \right)$$

The maximum value $k^* = \operatorname{argmax}_{k \geq 1} F(\lambda, k)$ satisfies the following:

1. If $\lambda \geq r \left(\frac{\alpha}{\alpha + \beta(\alpha + \beta)} \right)$, then $k^* = \infty$, and $W(\lambda) = \lambda$.
2. If $\lambda = r \left(\frac{\alpha}{\alpha + \beta(\alpha + \beta)} \right) - \rho$ for some $\rho > 0$, then k^* can be computed in time polynomial in the input size and in $\log(1/\rho)$ by binary search.

Proof. The expression for $W(\lambda)$ follows easily from Lemma 2.4.

Case 1: $\lambda \geq r \left(\frac{\alpha}{\alpha + \beta(\alpha + \beta)} \right)$. Consider the subcase $\lambda \geq r$. The function $F(\lambda, k)$ is maximized by driving the second term in the summation (which is always

non-positive) to zero. This happens when $k = \infty$. Otherwise, when $r > \lambda$ observe (using the upper bound of v_k) that

$$F(\lambda, k) = \lambda + \frac{(r - \lambda)v_k - \lambda\beta}{v_k + k\beta} \leq \lambda + \frac{(r - \lambda)\frac{\alpha}{\alpha + \beta} - \lambda\beta}{v_k + k\beta}$$

In the above, the second term is now non-positive, and it follows again that $k = \infty$ is the optimum solution.

Case 2: In this case let $\lambda = r \left(\frac{\alpha}{\alpha + \beta(\alpha + \beta)} \right) - \rho$ for some $\rho > 0$. Rewrite the above expression as

$$F(\lambda, k) = r - \beta \frac{\lambda + k(r - \lambda)}{v_k + k\beta}$$

Define the following quantities (independent of k):

$$\nu = (1 - \alpha - \beta) \quad \eta = \frac{\alpha}{\alpha + \beta} \log \frac{1}{\nu}$$

$$\phi = \eta\lambda + \frac{\alpha}{\alpha + \beta}(r - \lambda)$$

$$\mu = \eta(r - \lambda) \quad \omega = \lambda\beta - \alpha \frac{r - \lambda}{\alpha + \beta} = -\rho \frac{\alpha + \beta(\alpha + \beta)}{\alpha + \beta}$$

Observe $r - \lambda > \rho$. Note that $\phi, \mu \geq 0$. By assumption, the value $\nu \in (\delta, 1 - \delta)$ has polynomial bit complexity. The same holds for η, ϕ, μ and ρ . If we consider the partial derivative of F w.r.t k (relaxing k to be a real),

$$\frac{\partial F(\lambda, k)}{\partial k} = \frac{\beta}{(v_k + k\beta)^2} ((\phi + \mu k)\nu^k + \omega)$$

Since the denominator of $\partial F/\partial k$ is always non-negative, the value of k^* is either $k^* = 1$, or the point where the sign of the numerator $g(k) = (\phi + \mu k)\nu^k + \omega$ changes from $+$ to $-$. We observe that $g(k)$ has a unique maximum at $k_3 = \frac{1}{\log(1/\nu)} - \frac{\phi}{\mu}$. If $g(k_3)$ is negative then the numerator of $\frac{\partial F(\lambda, k)}{\partial k}$ is always negative and the optimum solution is at $k^* = 1$.

If $g(k_3)$ is positive, then it cannot change sign from $+$ to $-$ in the range $[1, k_3]$ since it has a unique maximum. Therefore in this range $k = 1$ or $k = k_3$ are the optimum solutions. Note that we should check both $\lfloor k_3 \rfloor, \lceil k_3 \rceil$.

But for $k \geq k_3$ since $g(k)$ is decreasing, $\partial F/\partial k$ changes sign once from $+$ to $-$ as k increases, and $\rightarrow 0$ as $k \rightarrow \infty$. This behavior is illustrated in Figure 4(b). Therefore, we find a $k_4 > k_3$ such that $g(k_4) < 0$, and perform binary search in the range $[k_3, k_4]$ to find the point where F is maximized. It is easy to compute k_4 with polynomial bit complexity in the bit complexities of ν, η, ϕ, μ and ρ . We would finally have to compare this maximal value of F to the values of F at $1, \lfloor k_3 \rfloor, \lceil k_3 \rceil$. Thus we can solve $W(\lambda)$ and obtain k^* in polytime. \square

Solving MAINLP: We now perform a parametric search on λ to solve MAINLP. Let $k_i^*(\lambda) = \operatorname{argmax}_{k \geq 1} F_i(\lambda, k)$. Let $R_i^*(\lambda) = R_i(k_i^*(\lambda))$ and $Q_i^*(\lambda) = Q_i(k_i^*(\lambda))$. Therefore, if the policy $\mathcal{P}_i(k_i^*(\lambda))$ is executed, the expected per step reward is $R_i^*(\lambda)$, and the probability that the arm is played at a random time step is $Q_i^*(\lambda)$.

Let $\mathcal{Q}(\lambda) = \sum_i Q_i^*(\lambda)$ and $\mathcal{R}(\lambda) = \sum_i R_i^*(\lambda)$. Note that the optimal objective of $\text{LPLAGRANGE}(\lambda)$ is given by $\mathcal{R}(\lambda) + \lambda(1 - \mathcal{Q}(\lambda))$. Intuitively, for given λ , if the n policies $\mathcal{P}_i(k_i^*(\lambda))$, one for each arm i , were executed *independently*, the total expected reward is $\mathcal{R}(\lambda)$, and the expected number of plays per time step is $\mathcal{Q}(\lambda)$. Note that multiple arms could be played per time step, so such execution does not represent a feasible policy.

Lemma 2.6. *Let $R_{\min} = \max_i \frac{r_i \alpha_i}{\alpha_i + \beta_i}$. Let γ^* be the optimum solution of MAINLP. Then, for any constant $\epsilon > 0$, there exist poly-time computable λ_*^-, λ_*^+ s.t.:*

1. $\mathcal{Q}(\lambda_*^-) = \mathcal{Q}_1 \geq 1$, and $\mathcal{Q}(\lambda_*^+) = \mathcal{Q}_2 < 1$.
2. $\lambda_*^+ - \lambda_*^- \leq \epsilon R_{\min}$.
3. If $\gamma_1 = \mathcal{R}(\lambda_*^-)$, $\gamma_2 = \mathcal{R}(\lambda_*^+)$, and $a = \frac{1 - \mathcal{Q}_2}{\mathcal{Q}_1 - \mathcal{Q}_2}$, then:

$$a\gamma_1 + (1 - a)\gamma_2 \geq (1 - \epsilon)\gamma^*$$

Proof. Recall $R_{\min} = \max_i \frac{r_i \alpha_i}{\alpha_i + \beta_i}$. Since using just one arm yields this reward, the optimal solution γ^* to MAINLP has value at least R_{\min} .

When $\lambda = 0$, then $k_i^*(\lambda) = 1$ for all i , implying $Q_i^* = 1$, so that $\mathcal{Q}(\lambda) = n$. Similarly, when $\lambda = \lambda_{\max} \geq \max_i r_i$, $k_i^*(\lambda) = \infty$ for all i , so that $\mathcal{Q}(\lambda) = 0$. Therefore, as λ is increased from 0 to λ_{\max} , there is a transition value λ_* such that $\mathcal{Q}(\lambda_*^-) = \mathcal{Q}_1 \geq 1$, and $\mathcal{Q}(\lambda_*^+) = \mathcal{Q}_2 < 1$.

We choose these values so that $\lambda_*^+ - \lambda_*^- \leq \epsilon R_{\min}$ for constant $\epsilon \geq 0$. To compute λ_*^+ and λ_*^- , perform a binary search on the interval $[0, \lambda_{\max}]$. If $\epsilon > 0$ is a constant, it is easy to show using Lemma 2.5 with $\rho = \frac{\epsilon R_{\min}}{n^2}$ that a suitably fine-grained binary search terminates in polynomial time in the bit complexity of the input and computes the desired λ_*^+, λ_*^- .

The optimum solution to MAINLP is feasible for $\text{LPLAGRANGE}(\lambda)$ for all λ , and has objective value precisely γ^* . Therefore,

$$\mathcal{R}(\lambda) + \lambda(1 - \mathcal{Q}(\lambda)) \geq \gamma^* \quad \forall \lambda$$

Plugging in $\lambda = \lambda_*^-$ and $\lambda = \lambda_*^+$, and multiplying the two inequalities by $a = \frac{1 - \mathcal{Q}_2}{\mathcal{Q}_1 - \mathcal{Q}_2}$ and $1 - a = \frac{\mathcal{Q}_1 - 1}{\mathcal{Q}_1 - \mathcal{Q}_2}$ respectively, and adding them yields:

$$a\gamma_1 + (1 - a)\gamma_2 \geq \gamma^* - (\lambda_*^+ - \lambda_*^-)(1 - a)(1 - \mathcal{Q}_2)$$

Since $\gamma^* \geq R_{\min}$, and $\lambda_*^+ - \lambda_*^- \leq \epsilon R_{\min}$, the RHS is at least $\gamma^*(1 - \epsilon)$. \square

Observe now that the solution obtained by taking a convex combination of the optimal solutions to $\text{LPLAGRANGE}(\lambda_*^-)$ and $\text{LPLAGRANGE}(\lambda_*^+)$ with weights $a = \frac{1-Q_2}{Q_1-Q_2}$ and $1-a$ respectively is feasible for MAINLP, since $aQ_1 + (1-a)Q_2 = 1$. Therefore, the above lemma implies that a near-optimal poly-time computable solution to MAINLP is a convex combination of the optimal solutions to $\text{LPLAGRANGE}(\lambda)$ for at most two values of λ given by λ_*^+ and λ_*^- . Each solution in the convex combination has a simple structure, being made up of n policies, where the policy for arm i is of the type $\mathcal{P}_i(k)$ (as described in Lemma 2.3). This completes our discussion on solving MAINLP to arbitrary accuracy in polynomial time. We ignore the $(1-\epsilon)$ factor in Lemma 2.6 from the rest of the discussion.

3 The Infeasible Start Solution

The solution for MAINLP constructed in the previous section represents a convex combination of two solutions S_1 and S_2 for $\text{LPLAGRANGE}(\lambda)$ corresponding to $\lambda = \lambda_*^-$ and $\lambda = \lambda_*^+$ respectively. Each of these two solutions is an ensemble of n policies, one per arm, where the policy for arm i is of the type $\mathcal{P}_i(k)$. It is easy to see that the solution S_1 (resp. S_2) has the following property. If the n single-arm policies in this solution are executed *independently*, then: (1) The total expected reward per time step is γ_1 (resp. γ_2), and (2) The expected number of plays per time step is at most Q_1 (resp. Q_2).

We first extract a single policy of type $\mathcal{P}_i(k)$ per arm i , and use this ensemble as an infeasible start solution. This infeasible solution will have the property that either we already have a simple 68 approximation, or we can restrict our attention to a subset S of arms and a corresponding ensemble of single-arm policies such that if the policies in the subset were executed independently, the expected reward is at least $0.445\gamma^*$, and the expected number of plays per time step is at most 1. (Recall that γ^* is the optimal value to MAINLP.) Executing the policies in the ensemble independently is clearly infeasible since multiple arms could be played at some time step; however, *in expectation*, at most one arm will be played per time step. Intuitively, the choice of a subset is already a ‘‘rounding step’’ – this is critical, we cannot deal with fractional choices later.

Recall that $Q_1 \geq 1$ and $Q_2 < 1$. Let $a_1 = \frac{1-Q_2}{Q_1-Q_2}$ and $a_2 = 1 - a_1$. From Lemma 2.6, either $a_1\gamma_1 \geq \frac{1}{2}\gamma^* + \gamma^*/100$ or $a_2\gamma_2 \geq \frac{1}{2}\gamma^* - \gamma^*/100$. Furthermore, $a_1Q_1 + a_2Q_2 = 1$ by definition.

Case 1: If $a_1\gamma_1 \geq \frac{1}{2}\gamma^* + \gamma^*/100$, then consider the n single-arm policies corresponding to $\text{LPLAGRANGE}(\lambda_*^-)$. Let $m_i = Q_i^*(\lambda_*^-)$ and $w_i = R_i^*(\lambda_*^-)$. Choose a subset S of arms with $\sum_{i \in S} m_i \leq 1$ so that $\sum_{i \in S} w_i$

is maximized. To achieve this, pick greedily from the single-arm policies obtained from $\text{LPLAGRANGE}(\lambda_*^-)$ in decreasing order of w_i/m_i till the next arm l violates the constraint $\sum_{i \in S} m_i \leq 1$. If $w_l \geq \gamma^*/50$, we simply use the policy for arm l specified by $\text{LPLAGRANGE}(\lambda_*^-)$ as the final solution – this yields a 50-approximation. Otherwise the set S we have chosen (excluding l) has reward $\sum_{i \in S} w_i \geq \frac{1}{2}\gamma^* - \gamma^*/100$ and enforces $\sum_{i \in S} m_i \leq 1$. In this case the chosen subset S is the infeasible start solution.

Case 2: If $a_2\gamma_2 \geq \frac{1}{2}\gamma^* - \gamma^*/100$, then the policies corresponding to $\text{LPLAGRANGE}(\lambda_*^+)$ yields the start solution S . We have $Q_2 = \sum_{i=1}^n Q_i^*(\lambda_*^+) < 1$, and $\gamma_2 = \sum_{i=1}^n R_i^*(\lambda_*^+) \geq \frac{1}{2}\gamma^* - \gamma^*/100$.

Pruning: Consider now the set S constructed above. It consists of a subset of arms, and for each arm $i \in S$, a policy $\mathcal{P}_i(k_i)$. The guarantee on this ensemble is that $\sum_{i \in S} Q_i(k_i) \leq 1$, and $\sum_{i \in S} R_i(k_i) \geq \frac{49}{100}\gamma^*$.

By Lemma 2.4, we have $\sum_{i \in S} \frac{1}{k_i} \leq 1$. Suppose for $i \in S$, we have $k_i \leq 3$. Note that there can be at most 3 such arms. If any of these $\mathcal{P}_i(k_i)$ had a reward $R_i(k_i)$ of at least $\gamma^*/68$, we simply use that one arm as the final solution, and get a 68 approximation. Otherwise, we discard these arms and for the remaining arms, $\sum_i R_i(k_i) \geq (0.49 - 3/68)\gamma^* \geq 0.445\gamma^*$. To summarize, we have the following theorem.

Theorem 3.1. *In polynomial time, we can either output a single arm l and a policy $\mathcal{P}_l(k_l)$, such that executing this policy has expected reward at least $\frac{\gamma^*}{68}$, or find a subset S of arms with one policy $\mathcal{P}_i(k_i)$ for each arm $i \in S$, such that: (i) All $k_i \geq 4$, and (ii) If the policies were executed independently, we have:*

1. *The expected number of arms played per time step is at most one, i.e. $\sum_{i \in S} Q_i(k_i) \leq 1$.*
2. *The expected reward is at least $0.445\gamma^*$, i.e., $\sum_{i \in S} R_i(k_i) \geq 0.445\gamma^*$.*

4 Modifying the Single-arm Policies

In this part, we present a general scheme to convert a policy of the type $\mathcal{P}_i(k)$ to a new policy $\text{GEOMOPT}(i, k)$. This new policy has comparable reward and probability of playing the arm as the old policy, and in addition, executes in a reasonably memoryless fashion. In the next section, we show how to use these new policies to convert the infeasible start solution constructed in Theorem 3.1 into an overall feasible solution.

For this part, we focus attention on a single arm i . We drop the subscript denoting the arm; that is, throughout this section $k = k_i, r = r_i, \alpha = \alpha_i$, and $\beta = \beta_i$.

Consider any policy $\mathcal{P}(k)$ associated with this arm, with $k \geq 4$. Let R denote the expected reward of $\mathcal{P}(k)$ and let q denote the probability (over time) that the arm is played. From Lemma 2.4, we have $R = r/(1 + \frac{k\beta}{v_k})$ and $q = (v_k + \beta)/(v_k + k\beta)$.

We first show upper bound on the reward and the lower bound on the probability of playing an arm. Looking ahead, we will be modifying $\mathcal{P}(k)$ such that we will play the arm (possibly) more but may get less reward, which mandates the two different bounds for R, q .

Lemma 4.1. $R \leq r \frac{1}{\max\{1, k\beta\}} \frac{\alpha}{\alpha + \beta}$ and $q \geq \frac{1}{k} \left(1 + \frac{(k-1)\alpha}{(\alpha + \beta)(k\beta + 1)}\right)$.

Proof. We first consider R . Suppose $k\beta \leq 1$ then we observe that $R \leq r\alpha/(\alpha + \beta)$ which simply bounds the expected number of good states seen by the arm. For $k\beta \geq 1$ we observe that

$$\begin{aligned} R &= \frac{r}{1 + \frac{k\beta(\alpha + \beta)}{\alpha} \frac{1}{1 - (1 - \alpha - \beta)^k}} \\ &\leq \frac{r\alpha}{(\alpha + \beta)} \frac{1}{k\beta} (1 - (1 - \alpha - \beta)^k) \leq \frac{r\alpha}{(\alpha + \beta)} \frac{1}{k\beta} \end{aligned}$$

This proves the claim for R . To derive a lower bound on q , observe that:

$$q = \frac{v_k + \beta}{v_k + k\beta} = \frac{1}{k} + \frac{(k-1)v_k}{k(v_k + k\beta)}$$

Applying Fact 2.2 with $t = k, x = \alpha + \beta$, we have $v_k \geq \frac{\alpha k}{1 + (\alpha + \beta)k}$. Thus:

$$\begin{aligned} q &\geq \frac{1}{k} + \frac{(k-1)\alpha}{k} \frac{1}{(1 + (\alpha + \beta)k)\beta + \alpha} \\ &= \frac{1}{k} \left(1 + \frac{(k-1)\alpha}{(\alpha + \beta)(k\beta + 1)}\right) \end{aligned}$$

□

4.1 The Nice Policy GEOMOPT

We now modify the policy $\mathcal{P}_i(k)$, and call this new policy GEOMOPT(i, k). We drop the subscript i in the remaining description. This new policy is presented in Figure 2. There are two distinct phases in the policy: An *explore* phase which proceeds in a memoryless fashion, and an *exploit* phase which happens for at most $\ell = \lfloor \frac{2(k-1)}{k\beta + 1} \rfloor$ steps if the explore phase observed the state to be g . Note that $\ell \geq 1$ because $k \geq 2$.

The crucial aspect of this policy is that the ‘‘exploit’’ plays in Step (2) are *not used for updating knowledge of the state of the arm*. The only time when the outcome is used to decide the next play is when the arm is explored

in Step (3). Furthermore, Step (3) is executed in a memoryless fashion independent of the evolution of the state of the arm. In short, *the updates about the knowledge of the state of the arm define a memoryless process independent of the state of the arm itself*. This will be crucial in the next section.

Lemma 4.2. For policy $\mathcal{P}(k)$, let R and q denote the expected reward and the probability with which the arm is played respectively. If $k \geq 4$, the reward of GEOMOPT(k) is at least $\frac{R}{15}$, and the probability (over time) that the arm is played according to this policy is at most $\frac{1}{3}q$.

Proof. We analyze the Markov Chain (Fig. 5) corresponding to GEOMOPT(k). This has two infinite sets of states: $A = \{s_0, s_1, s_2, \dots\}$ and $B = \{t_0, t_1, t_2, \dots\}$. State s_0 is reached if the arm is observed to be in state g in Step (3), and t_0 is reached if the state is observed to be in state b .

Define $p = \frac{1}{6k}$. The transition probability from state s_j to state s_0 is pu_{j+1} , to state s_{j+1} is $1 - p$, and to state t_0 is $p(1 - u_{j+1})$. The transition probability from state t_j to state s_0 is pv_{j+1} , to state t_{j+1} is $1 - p$, and to state t_0 is $p(1 - v_{j+1})$. The arm is played in states s_0, s_1, \dots, s_ℓ (this corresponds to Step 2(b) under the **else** condition), but the results of the play are ignored for belief updates unless this play corresponds to Step (3).

Let π_1 denote the probability of being in state s_0 and π_0 denote the probability of being in state t_0 . A simple analysis shows that $\pi_0 + \pi_1 = p$, and furthermore, $\pi_1 = p\alpha/(\alpha + \beta) = \alpha/(6k(\alpha + \beta))$.

The expected reward \hat{R} of GEOMOPT(k) is at least (since $u_{j+1} \geq (1 - \beta)^{j+1}$ for all j):

$$\begin{aligned} \hat{R} &= r \left(\pi_1 + \sum_{j=0}^{\ell-1} \pi_1 (1 - p)^j u_{j+1} \right) \\ &\geq r\pi_1 \left(1 + \sum_{j=0}^{\ell-1} (1 - p)^j (1 - \beta)^{j+1} \right) \end{aligned}$$

Set $z = 1 - (1 - p)(1 - \beta) \geq \beta$. The above implies:

$$\begin{aligned} \frac{\hat{R}}{r\pi_1} &\geq 1 + (1 - \beta) \frac{1 - (1 - z)^\ell}{z} \\ &\geq 1 + (1 - z) \frac{1 - (1 - z)^\ell}{z} = \frac{1 - (1 - z)^{\ell+1}}{z} \end{aligned}$$

Using fact 2.2 with $t = \ell + 1, x = z$, we have $\hat{R} \geq r\pi_1/(z + 1/(\ell + 1))$. This increases in $\ell + 1$; since $2(k - 1)/(k\beta + 1) \leq \ell + 1$ we get

$$\begin{aligned} \hat{R} &\geq \frac{r\pi_1}{\frac{k\beta + 1}{2(k-1)} + \frac{1}{6k} + \beta - \frac{\beta}{6k}} \geq \frac{r\pi_1}{\frac{k\beta + 1}{2(k-1)} + \frac{1}{6k} + \beta} \\ &= \frac{\pi_1 r 6k}{6(k\beta + 1) \frac{k}{2(k-1)} + (1 + 6k\beta)} \end{aligned}$$

Policy GEOMOPT(k)

1. Choose T from `Geometric`($\frac{1}{6k}$) independently.
2. /* Lasts exactly $T - 1$ steps */
If the current state of the arm is b :
 - (a) Wait for $T - 1$ steps.
 - (b) **Goto** step (3).**else:** /* The current state of the arm is g */
 - (a) **Exploit:** Play the arm for $\delta = \min(T - 1, \ell)$ steps where $\ell = \lfloor \frac{2(k-1)}{k\beta+1} \rfloor$.
 - (b) Do not update the knowledge of the state of the arm during these plays.
 - (c) Wait an additional $T - 1 - \delta$ steps, and **goto** step (3).
3. **Explore:** Play the arm, update the knowledge of its state and **goto** Step (1).

Figure 2. The Policy GEOMOPT(k).

For $k \geq 4$ we get (using Lemma 4.1)

$$\begin{aligned}
 \hat{R} &\geq \frac{r\alpha}{\alpha + \beta} \frac{1}{6(k\beta + 1)^{\frac{4}{6}} + (1 + 6k\beta)} \\
 &= \frac{r\alpha}{\alpha + \beta} \frac{1}{10k\beta + 5} \\
 &\geq \frac{1}{15} \frac{r\alpha}{\alpha + \beta} \frac{1}{\max\{k\beta, 1\}} = \frac{R}{15}
 \end{aligned}$$

The total probability with which the arm is played is at most:

$$\begin{aligned}
 p + \ell\pi_1 &\leq \frac{1}{6k} \left(1 + \frac{2(k-1)}{k\beta+1} \frac{\alpha}{\alpha+\beta} \right) \\
 &\leq \frac{1}{6k} \left(2 + \frac{2(k-1)}{k\beta+1} \frac{\alpha}{\alpha+\beta} \right) \leq \frac{1}{3}q
 \end{aligned}$$

□

5 The Final Policy GLOBAL

Consider the set S of arms constructed in Theorem 3.1, and the associated single arm policies. Recall that $\mathcal{P}_i(k_i)$ denotes the policy for arm i . For policy $\mathcal{P}_i(k_i)$, let the probability with which this policy plays the arm be denoted $q_i = Q_i(k_i)$. Recall from Theorem 3.1 that $\sum_{i \in S} q_i \leq 1$. Since $q_i \geq \frac{1}{k_i}$ from Lemma 2.4, this implies $\sum_{i \in S} \frac{1}{6k_i} \leq \frac{1}{6}$. For $i \in S$, consider GEOMOPT(i, k_i) constructed from $\mathcal{P}_i(k_i)$. From Lemma 4.2 and Theorem 3.1, it is clear that if the GEOMOPT(i, k_i) policies for different arms $i \in S$ are executed independently, the following hold:

1. The expected number of arms played at any time step is at most $\frac{1}{3} \sum_{i \in S} q_i \leq \frac{1}{3}$.
2. The expected total reward is at least $\frac{1}{15} 0.445\gamma^*$.

We now convert this ensemble of GEOMOPT policies into a single feasible policy. The final policy GLOBAL

(Figure 3) executes the GEOMOPT policies for the different arms independently. This could lead to multiple arms being played at once – this conflict is resolved as shown in Figure 3. Note that the policy gives priority to the executions of Step (3) over Step (2). This preference will be crucial for the analysis.

Theorem 5.1. *The expected reward of the GLOBAL policy is at least $\frac{1}{68}\gamma^*$.*

Proof. Focus attention on any arm $i \in S$. Define $p = \frac{1}{6k_i}$. In the Markov chain corresponding to GEOMOPT(i, k_i), let f_j denote the presence of either state s_j or state t_j , i.e., step (3) was executed j steps ago (recall the notation from the proof of Lemma 4.2). We have $\Pr[f_j] = p(1-p)^{j-1}$.

Now, the behavior of the policy GEOMOPT(i, k_i) and GLOBAL for arm i are coupled since they execute Step (3) in a memoryless fashion at the same rate. This implies the state evolution of f_j can be coupled in the two executions. Consider a random time instant, conditioned on being in state f_j in either policy. The arm has just executed Step (2) if $j > 0$ and Step (3) if $j = 0$. The arm i is said to be *active* if the arm did not see conflict in the just executed step of GLOBAL. This corresponds to the following conditions:

1. $\mathcal{E}_i = 1$, i.e., the immediately previous attempt to play the arm in Step (3) succeeded.
2. No other arm just attempted play (if $j > 0$).

Let Ω denote the event “No other arm executes Step (3)”. Denote the event that the condition (1) holds by Ψ_j and the event that the condition (2) holds by Δ_j . Condition (1) concerns the event Ω happening at the immediately previous time Step (3) was executed in GEOMOPT(i, k_i). Since the event Ω defines an *i.i.d* process independent of the execution of GEOMOPT(i, k_i), and since the executions of Step (3) for arm i define a

Policy GLOBAL for arm i

1. Choose T from $\text{Geometric}(\frac{1}{6k_i})$ independently.
- 2 /* Lasts exactly $T - 1$ steps */
 - If** the current state of the arm is b_i or $\mathcal{E}_i = 0$:
 - (a) Wait for $T - 1$ steps.
 - (b) **Goto** step (3).
 - else**: /* The current state of the arm is g_i and $\mathcal{E}_i = 1$ */
 - (a) **Exploit**: For $\delta = \min(T - 1, \ell_i)$ steps, play arm iff no other arm attempts to be played that step. Recall $\ell_i = \lfloor \frac{2(k_i - 1)}{k_i \beta_i + 1} \rfloor$.
 - (b) Do not update the knowledge of the state of the arm during these plays.
 - (c) Wait an additional $T - 1 - \delta$ steps, and **goto** step (3).
3.
 - (a) Attempt to play the arm.
 - (b) The attempt succeeds iff this is the only arm executing Step (3).
 - (c) **If** attempt succeeds, set $\mathcal{E}_i = 1$, play the arm and update state. (**explore**)
else Set $\mathcal{E}_i = 0$ and do not play the arm.
 - (d) **Goto** Step (1).

Figure 3. The Final Policy GLOBAL restricted to arm i .

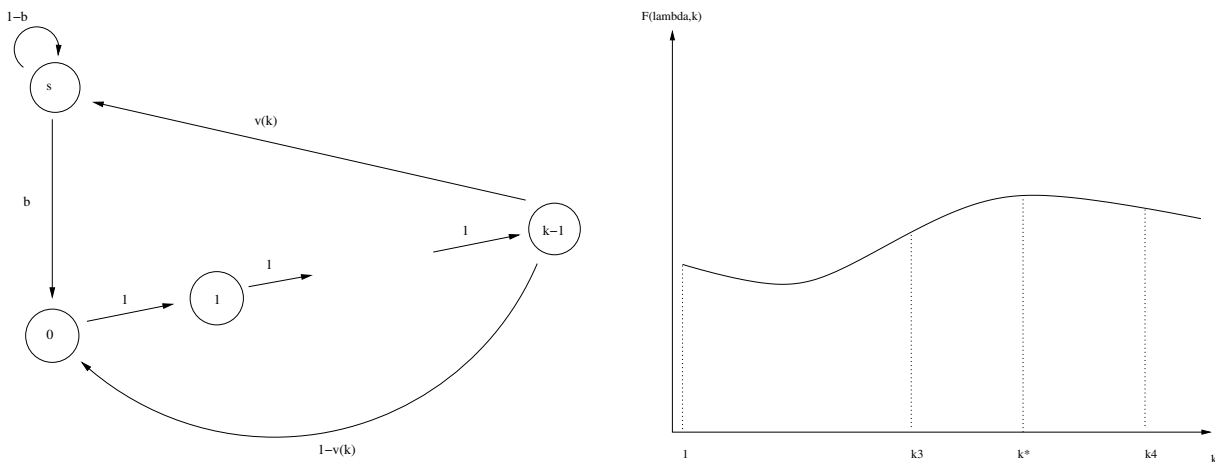


Figure 4. (a) Markov Chain for policy $\mathcal{P}(k)$.

(b) Behavior of the function $F(\lambda, k)$.

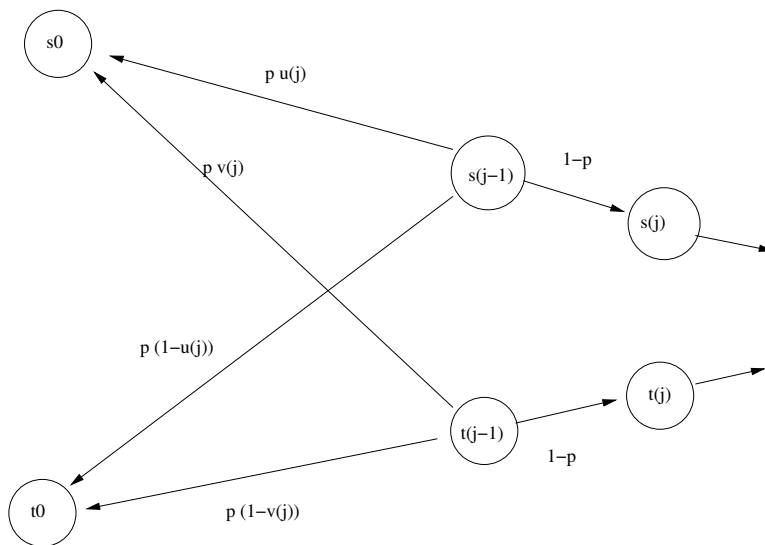


Figure 5. Markov Chain for $\text{GEOMOPT}(k)$ showing transitions for s_{j-1} and t_{j-1} . Here, $p = \frac{1}{6k}$.

memoryless process, the probability of event Ψ_j is exactly the same as the probability that the event Ω happens at a random time instant. Therefore, $\Pr[\Psi_j] = \Pr[\Omega] \geq 1 - \sum_{m \neq i} \frac{1}{6k_m} \geq \frac{5}{6}$.

If $j = 0$, the event Δ_j is irrelevant. Else, we argue about $\Pr[\Delta_j]$ as follows. Let Π_j denote the event that if the GEOMOPT policies were indeed executing independently with multiple plays allowed per time step, then no other arm besides arm i attempts to play when the state of GEOMOPT(i, k_i) is f_j . It is easy to see that if the event Π_j occurs, then the event Δ_j also occurs. Furthermore, the event Π_j is independent of the execution of GEOMOPT(i, k_i). Since at most $1/3$ fraction of the arms are playing at any given time, we have $\Pr[\Delta_j] \geq \Pr[\Pi_j] \geq 1 - \frac{1}{3} = \frac{2}{3}$. Therefore, by union bounds, the probability that arm i is active is given by $\Pr[\Psi_j \wedge \Delta_j] \geq 1 - \frac{1}{3} - \frac{1}{6} = \frac{1}{2}$.

Here is the key observation: Given that the state of GEOMOPT(i, k_i) is f_j , the event that arm i is active (the event $\Psi_j \wedge \Delta_j$) is independent of whether the state in GEOMOPT(i, k_i) is s_j or t_j . To see this, note that the event of arm i being active depends on \mathcal{E}_m for the arms $m \neq i$, and on their respective Markov chains. The \mathcal{E}_m for $m \neq i$ only depend on conflicts with arm i encountered in step (3), which in turn are independent of the underlying Markov chain for arm i . Furthermore, if the arm is active, the knowledge about the state of arm i is the same in both GEOMOPT(i, k_i) and GLOBAL because step (3) was executed in both policies. This implies the expected reward GLOBAL obtains from arm i is at least $\frac{1}{2}$ of the expected reward of GEOMOPT(i, k_i). By linearity of expectation over the arms in S , the total reward of GLOBAL is at least $\frac{1}{2} \cdot \frac{1}{15} \cdot 0.445\gamma^* \geq \frac{1}{68}\gamma^*$. \square

6 Extensions

The results can also be extended to the case where some $M \geq 1$ arms can be played every time step, and a constant factor approximation obtained. The technical details remain almost the same as the single arm case, with the following differences: For the ensemble of GEOMOPT policies, in expectation $M/3$ attempt to play any time step. If more than M arms attempt to execute Step (3), none of these plays succeeds. A play at Step (2) succeeds if no more than $M - 1$ other arms are attempting to play simultaneously. It is easy to see that with these modifications, the analysis of GLOBAL yields a constant factor approximation. The result can also be extended to obtain a constant factor approximation to the case where the reward of the “bad” state b_i is $s_i \in [0, r_i)$. The construction of the policy remains the same; the only additional detail is that for policy $\mathcal{P}_i(k)$, we need to show both lower and upper bounds for the probability q^* of playing the arm in Lemma 4.1. These

bounds can be shown to be within constant factors of each other, which ensures an overall constant factor approximation. The technical details will be presented in the full version.

Acknowledgment: We thank Ashish Goel and Saswati Sarkar for several helpful discussions.

References

- [1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *FOCS*, 1995.
- [2] A. Bar-Noy, R. Bhatia, J. Naor, and B. Schieber. Minimizing service and operation costs of periodic scheduling. In *SODA*, pages 11–20, 1998.
- [3] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2001.
- [4] D. Bertsimas and J. Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Oper. Res.*, 48(1):80–90, 2000.
- [5] 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.
- [6] A. Flaxman, A. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proc. of the 2005 Annual ACM-SIAM Symp. on Discrete Algorithms*, 2005.
- [7] 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.
- [8] 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.
- [9] S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *Proc. ACM Symp. on Theory of Computing*, 2007.
- [10] S. Guha and K. Munagala. Model-driven optimization using adaptive probes. In *Proc. ACM-SIAM Symp. on Discrete Algorithms*, 2007.
- [11] S. Guha, K. Munagala, and S. Sarkar. Information acquisition and exploitation in multi-channel wireless systems. *Submitted to IEEE Transactions on Information Theory*, 2007.
- [12] S. M. Kakade and M. J. Kearns. Trading in markovian price models. In *COLT*, pages 606–620, 2005.
- [13] C. Kenyon and N. Schabanel. The data broadcast problem with non-uniform transmission times. In *Proc. ACM-SIAM Symp. on Discrete Algorithms*, 1999.
- [14] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.
- [15] J. Niño-Mora. Restless bandits, partial conservation laws and indexability. *Adv. in Appl. Probab.*, 33(1):76–98, 2001.
- [16] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queuing network control. *Math. Oper. Res.*, 24(2):293–305, 1999.
- [17] P. Whittle. Restless bandits: Activity allocation in a changing world. *Appl. Prob.*, 25(A):287–298, 1988.