

# Approximation Algorithms for Restless Bandit Problems

Sudipto Guha\*

Kamesh Munagala†

Peng Shi‡

## Abstract

In this paper, we consider the *restless bandit* problem, which is one of the most well-studied generalizations of the celebrated stochastic multi-armed bandit problem in decision theory. In its ultimate generality, the restless bandit problem is known to be PSPACE-Hard to approximate to any non-trivial factor, and little progress has been made on this problem despite its significance in modeling activity allocation under uncertainty. We make progress on this problem by showing that for an interesting and general subclass that we term MONOTONE bandits, a surprisingly simple and intuitive greedy policy yields a factor 2 approximation. Such greedy policies are termed *index* policies, and are popular due to their simplicity and their optimality for the stochastic multi-armed bandit problem. The MONOTONE bandit problem strictly generalizes the stochastic multi-armed bandit problem, and naturally models multi-project scheduling where the state of a project becomes increasingly uncertain when the project is not scheduled. We develop several novel techniques in the design and analysis of the index policy. Our algorithm proceeds by introducing a novel “balance” constraint to the dual of a well-known LP relaxation to the restless bandit problem. This is followed by a structural characterization of the optimal solution by using both the exact primal as well as dual complementary slackness conditions. This yields an interpretation of the dual variables as potential functions from which we derive the index policy and the associated analysis.

## 1 Introduction

The multi-armed bandit (MAB) problems are fundamental to stochastic decision theory. These problems model activity allocation under uncertainty and have

numerous applications and a vast literature (see [5] and references therein). The most well-known variant is the *stochastic* MAB problem, which is stated as follows: There is a bandit with  $n$  *independent* arms (think of these as different projects or jobs). Each arm  $i$  can be in one of several states denoted  $\mathcal{S}_i$ . At any time step, the player can play one arm. If arm  $i$  in state  $k \in \mathcal{S}_i$  is played, it transitions in a Markovian fashion to state  $j \in \mathcal{S}_i$  w.p.  $q_{kj}^i$  and yields reward  $r_k^i \geq 0$ . The states of arms which are not played stay the same. There is a discount factor  $\beta \in (0, 1)$ . Given the initial states of the arms, the goal is to find a policy for playing the arms in order to maximize one of the following infinite horizon quantities:  $\sum_{t=0}^{\infty} R_t \beta^t$  (discounted reward), or  $\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{t=0}^{\infty} R_t$  (average reward), where  $R_t$  is the expected reward of the policy at time step  $t$ .

The input to an algorithm specifies the rewards and transition probabilities for each arm, and has size linear in  $n$ . The output is a *policy*: A (possibly implicit) specification *fixing upfront* which arm (or distribution over arms) to play for every possible joint state of the arms. We seek poly-time algorithms (in terms of the input size) that output (near-) optimal policies with poly-size specifications, with the property that for each execution step, the action for the current joint state can be computed from the specification in poly-time.

Since a policy is a fixed (possibly randomized) mapping from the *exponential (or possibly infinite) size* joint state space of  $n$  arms to actions, ensuring poly-time computation and execution often requires simplifying the description of the optimal policy using the problem structure. The stochastic MAB problem is the most well-known decision problem for which such a structure is known: The *optimal* policy is a greedy policy termed the GITTINS index policy [11, 29, 5]. An index policy specifies a single number called “index” for each state  $k \in \mathcal{S}_i$  for each arm  $i$ , and at every time step, plays the arm whose current state has the highest index. In addition to allowing for poly-time computation and execution, index policies are also optimal for several generalizations of the stochastic MAB, such as arm-acquiring bandits [32] and branching bandits [31]; in fact, a general characterization of problems for which index policies are optimal is now known [6].

Not all variants of the stochastic MAB problem

\*Department of Computer and Information Sciences, University of Pennsylvania, Philadelphia PA 19104-6389. Email: [sudipto@cis.upenn.edu](mailto:sudipto@cis.upenn.edu). Research supported in part by an Alfred P. Sloan Research Fellowship, an NSF CAREER Award, and NSF Award CCF-0644119.

†Department of Computer Science, Duke University, Durham NC 27708-0129. Email: [kamesh@cs.duke.edu](mailto:kamesh@cs.duke.edu). Research supported by NSF via a CAREER award and grant CNS-0540347.

‡Duke University, Durham NC 27708. Email: [peng.shi@duke.edu](mailto:peng.shi@duke.edu). This research was supported by the Duke University Work-Study Program and by NSF award CNS-0540347.

admit to optimal index policies or efficient solutions – the most fundamental (and well known) of these is the *restless bandits* problem proposed by Whittle [33]. This problem is the same as the stochastic MAB problem, except that when arm  $i$  in state  $k \in \mathcal{S}_i$  is not played, it’s state evolves to  $j \in \mathcal{S}_i$  with probability  $\tilde{q}_{kj}^i$ . Therefore, the state of the arm varies according to an *active* transition matrix  $q$  when the arm is played, and according to a *passive* transition matrix  $\tilde{q}$  if the arm is not played. Unlike the stochastic MAB problem which is interesting only in the discounted reward setting<sup>1</sup>, the restless bandit problem is interesting even in the *infinite horizon average reward* setting – this is the setting in which this problem has been typically studied, and which we consider in this paper. For these problems, it is relatively straightforward to show that in general, any index policy can be a poor approximation to optimal; in fact, Papadimitriou and Tsitsiklis [27] show that for  $n$  arms, computing the optimal policy is PSPACE-hard. Their proof also rules out any poly-time algorithm to decide if the optimal reward is more than zero, hence ruling out any approximation algorithm as well.

On the positive side, Whittle [33] presents a poly-size LP relaxation of the problem that has variables for *single arms*. In this relaxation, the constraint that exactly one arm is played per time step is replaced by the constraint that one arm *in expectation* is played per time step ; this is the only constraint connecting the arms. (Such decision problems have been termed *weakly coupled* systems [19, 1].) The *Whittle index* [33] is based on the Lagrangean of this relaxation, and generalizes the Gittins index. However, this and subsequent works [7, 19, 1] present little in terms of performance analysis.

**Monotone Bandits.** We revisit the restless bandit problem, and ask: *Is there a natural subclass for which index policies are provably near-optimal? For the same subclass, can Whittle’s LP relaxation be shown to have small gap w.r.t. the optimal policy?* We show positive answers to both these questions for a general subclass which we term MONOTONE bandits. The problem formulation is similar to the stochastic MAB – there are  $n$  arms, state of an arm remains the same if the arm is not played, and when arm  $i$  in state  $k$  is played, it yields reward  $r_k^i$ . However, there is an “escape probability”  $f_k^i(t) \in [0, 1]$ , so that if arm  $i$  in state  $k$  is played after  $t$  steps, its transition probability to  $j \neq k$  is  $q_{kj}^i f_k^i(t)$ . (With the remaining probability it remains at state  $k$ .) The crucial property we enforce is that  $f_k^i(t)$  is *monotonically non-decreasing* in  $t$ .

<sup>1</sup>Playing the arm with the highest long-term average reward exclusively is the trivial optimal policy for stochastic MAB in the infinite-horizon average reward setting.

Since stochastic MAB corresponds to  $f_k^i(t) = 1$ , the MONOTONE bandit problem strictly generalizes it. The MONOTONE bandit problem naturally models scheduling scenarios where the arm is a resource/machine which *recovers* strength when rested, so that when rested for more steps and played, the probability that the arm is actually played,  $f_k^i(t)$ , increases. When actually played, the state evolves according to transition matrix  $q$ , else nothing happens. The MONOTONE bandit problem also models certain *Partially Observable Markov Decision Processes* (or POMDPs) where the state of each arm is continuously changing and unobservable except when the arm is played. The non-decreasing property of  $f$  corresponds to saying that the longer the time elapsed since last play, the more likely the arm is to be in a different state. A special case of this is the FEEDBACK MAB problem discussed below.

**Our Results:** We provide a 2 approximate index policy for MONOTONE bandits based on a novel duality-based framework using a subtle modification to Whittle’s LP. We further show that this relaxation has an  $\Omega(1)$  gap, so that the above result is nearly best possible. In a full version [18], we use our techniques to show  $O(1)$  guarantees for Whittle-type indexes in such contexts, and finally extend our techniques to solve other related restless bandit problems. Except one (the FEEDBACK MAB), these problems did not have any previous performance guarantees. We discuss the previous work first to explain the generalizations and improvements.

The FEEDBACK MAB, studied independently by [15, 35, 20, 26], is a special case of the MONOTONE bandit problem. (Refer Appendix A for the reduction.) In this problem, there is a bandit with  $n$  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 state of each arm evolves according to a bursty 2-state Markov process (with transition probabilities specified as input); the evolution does not depend on whether the arm is played or not at a time slot. The evolution of states for different arms are independent. Any policy chooses at most one arm to play every time slot. Each play yields reward depending on the state of the arm, and in addition, reveals to the policy the current state of that arm. When the arm is not played, the underlying state cannot be observed and must be inferred from the last time the arm was played. As motivation, in wireless channel selection [15, 35, 20], the bandit is a wireless node with access to multiple noisy channels (arms). The state of the arm is the state (good/bad) of the channel, which varies in a Markovian fashion. Playing the arm corresponds to transmitting on the channel, yielding reward if the transmission is successful (good channel state), and furthermore, revealing to the transmitter

the current state of the channel. In Unmanned Aerial Vehicle (UAV) routing [26], the arms are locations where interesting events follow 2-state Markov processes (interesting/uninteresting); visiting a location by the UAV corresponds to playing the arm, and yields reward if an interesting event is detected.

The previous result for this problem was a 68 approximation [15]. As a consequence of our identification of the MONOTONE bandit problem as the correct abstraction, we improve the approximation to  $2 + \epsilon$ .

Furthermore, since we have a different solution technique, we can gracefully handle several generalizations motivated by the applications described above: Costs for switching into a bandit arm that are subtracted from the rewards; costs for observing the state of an arm without utilizing its reward; and multiple simultaneous plays of varying duration. We note that these generalizations lead to negative rewards and blocking plays, and *cannot* be handled by the solution technique in [15]. We discuss all these extensions in the full version [18]. Finally, our techniques also extend to similar restless bandit problems which do not fall into the MONOTONE bandits framework: We show a 2-approximation for non-preemptive machine replenishment in [18].

**The Index Policy.** Our 2-approximation for the MONOTONE bandits problem is achieved by an index policy with surprisingly simple (and in hindsight, intuitive) structure. The description of this policy is as follows: Solve the *dual* of Whittle’s relaxation with an added “balance” constraint. The optimal solution yields a classification of the states of each arm into high reward “good” states and low reward “bad” states. When an arm is in a good state, the policy plays the arm repeatedly until it becomes bad. In a bad state, the policy leaves the arm alone for a fixed period of time, allowing it to “recover” and gain a sufficiently high probability of transitioning into good states. When no arm is in a good state, the policy plays any bad arm that has waited long enough to “recover”. This corresponds to the following index: Add a dummy arm that corresponds to doing nothing, with index 0. For any real arm, when it is in bad state and recovering, its index is  $-1$ . When it has recovered, its index becomes 1. When in good state, its index is 2. The policy therefore exploits every good state as much as it can, and if no arm is in a good state, it explores the bad arms that have recovered, until one of them transitions to a good state.

Our index construction depends on the LP relaxation, akin to the index constructed by Bertsimas and Nino-Mora [7]. This is unlike the Gittins and Whittle indexes, where the index for an arm does not depend at all on the parameters of the other arms. Despite this difference, we show in the full paper [18] that for FEED-

BACK MAB, our policy is equivalent to a slight but natural modification of Whittle’s index to favor myopic exploitation. This yields one of the first analyses of such indexes used in practice in this context.

**Assumptions and Lower Bounds.** In order to ensure poly-time computation and execution, we insist that the  $f_k^i(t)$  be either piece-wise linear with poly-size specification, or certain compactly specified differentiable functions (as with FEEDBACK MAB). Given the infinite-horizon setting, we restrict ourselves to ergodic policies. Notice that since the state of an arm depends on how long it has rested, *the state space even for a single arm could be exponential in size. Despite this, our results show the existence of a poly-time computable and executable index policy that is a 2-approximation*<sup>2</sup>. Our index depends on time elapsed since that arm was last played; for FEEDBACK MAB, it was shown in [15] that such indexes are necessarily some constant factor worse than the optimal policy, so a constant-factor approximation is best-possible given our solution structure.

We show in the full paper [18] that even when  $f_k^i(t)$ ’s are piece-wise linear with poly-size specification, the MONOTONE bandit problem is NP-HARD. We further show that the requirement that  $f$  is monotone in  $t$  is necessary; relaxing this makes the problem  $n^\epsilon$  hard to approximate, where  $n$  is the number of arms.

In Appendix A, we show that the integrality gap of Whittle’s LP is at least  $e/(e-1) \approx 1.58$ , and therefore our factor 2 analysis is almost the best possible against LP bounds. Interestingly, relaxing the separability of transition probabilities, so that each  $k$  to  $j$  transition is an arbitrary (but still monotone) function,  $q_{kj}^i(t)$ , of time, leads to unbounded integrality gap (see [18]).

**Techniques.** The chief technical highlight of the paper is the novel “dual balancing” condition. In more detail, for MONOTONE bandits, we first re-write Whittle’s LP so that the dual has an interpretation in terms of rewards and potentials for states (Section 3). Next, we add a “dual balancing” condition and solve the dual with this added constraint (Section 4). We then show using complementary slackness conditions on the optimal solution that the rewards obtained by playing the arms are tightly related to the dual objective via the potential variables (Section 4.1). In fact, the dual naturally splits the states of each arm into *exploration* or “bad” states, and *exploitation* or “good” states. We then use the tight constraints in the dual to construct an index policy (Section 5), and we analyze its performance by carefully constructing a piece-wise linear potential function using the variables from the dual solution

<sup>2</sup>We note that a  $1 + \epsilon$  approximation can be found in time exponential in the input specification by dynamic programming.

(Section 5.1). The analysis crucially depends on the MONOTONE property of  $f$  as well as the balancing condition that we add.

Our solution technique differs from primal-dual approximation algorithms [30] and online algorithms [34] that relax either the primal or the dual complementary slackness conditions using a careful dual-growing procedure. Our index policy and associated potential function analysis crucially exploit the structure of the *optimal* dual solution that is gleaned using both the *exact* primal as well as dual complementary slackness conditions. Further, our notion of dual balancing is very different from that used by Levi *et al* [22] for designing online algorithms for stochastic inventory management.

**Roadmap.** For the MONOTONE bandit problem, we show a 2 approximation in the main body of the paper. Since the FEEDBACK MAB is a special case of MONOTONE bandits, the above results in a  $2+\epsilon$  approximation, due to a  $1+\epsilon$  approximation to the LP solution, while preserving the required duality structure. This, along with the  $e/(e-1)$  integrality gap for Whittle’s LP is discussed in Appendix A. Due to lack of space the results on generalizations of MONOTONE bandits, analysis of Whittle-type indexes for FEEDBACK MAB, and the approximation algorithm for non-preemptive machine replenishment are presented in the full version [18].

**Related Work.** The FEEDBACK MAB problem is perhaps the simplest instance of a multi-armed bandit problem where a changing environment influences the evolution of the rewards of the arms. This general paradigm has received significant interest recently in computational learning [2, 4, 9, 21, 28]. In closely related but independent work, Slivkins and Upfal [28] consider the modification of FEEDBACK MAB where the underlying state of the arms vary according to a reflected Brownian motion with bounded variance. As discussed in [28], these classes of problems are very different, even requiring different performance metrics.

The results in [14, 12, 16, 17] consider variants of the stochastic MAB where the environment does not change and only a limited time is allotted to learning about this environment. Although several of these results use LP rounding, they have little connection to the duality based framework considered here.

A different restless bandit problem of pre-emptive machine replenishment [5, 13] is considered in [25], where Whittle’s index is shown to be a 1.51 approximation. However, the techniques used to show that result are very different from the framework developed here. In fact, our techniques show a 2 approximation for the more general problem of non-preemptive machine replenishment, for which Whittle’s index is an arbitrarily poor approximation (refer the full paper [18]).

## 2 MONOTONE Bandits: Preliminaries

We repeat the problem statement: There are  $n$  bandit arms. Each arm  $i$  can be in one of  $K$  states denoted  $\mathcal{S}_i = \{\sigma_1^i, \sigma_2^i, \dots, \sigma_K^i\}$ . Each state  $\sigma_k^i$  is associated with a set of transition probability values  $q^i(k, j)$  so that  $\sum_{j \neq k} q^i(k, j) \leq 1$ . Furthermore, the state  $\sigma_k^i \in \mathcal{S}_i$  is associated with an “escape probability”  $f_k^i(t) \in [0, 1]$  for positive integers  $t$ . When the arm is not played, its state remains the same and it does not fetch reward. Suppose the arm is in state  $\sigma_k^i$  and is played next after  $t \geq 1$  steps. Then, it gains reward  $r_k^i \geq 0$ , and transitions to one of the states  $\sigma_j^i \neq \sigma_k^i$  w.p.  $q^i(k, j)f_k^i(t)$ , and with the remaining probability stays in state  $\sigma_k^i$ . For notational convenience, we denote  $\sigma_k^i$  simply as  $k$ ; the arm it refers to will be clear from the context.

The transition probabilities for different arms are independent. At most one arm is played per step. The goal is to find a policy for playing the arms so that the infinite horizon time-average reward is maximized.

**Assumptions.** For simplicity of exposition, we assume that for each arm  $i$ , the graph, where the vertices are  $k \in \mathcal{S}_i$  and a directed edge  $(j, k)$  exists if  $q^i(j, k) > 0$ , is strongly connected. Since we consider the infinite horizon time average reward, assume that the policy is ergodic and can choose the start state of each arm.

We need the following key property about the transition probabilities for designing the index policy.

**Monotone Property:** For every arm  $i$  and state  $k \in \mathcal{S}_i$ , we have:  $f_k^i(t) \leq f_k^i(t+1)$  for every  $t$ . For polynomial input size, we assume these monotone functions are piece-wise linear with poly-size specification.

**DEFINITION 1.** Given  $i, k \in \mathcal{S}_i$ ,  $f_k^i(t)$  is specified as the piece-wise linear function that passes through breakpoints  $(t_1 = 1, f_k^i(1)), (t_2, f_k^i(t_2)), \dots, (t_m, f_k^i(t_m))$ . Denote the set  $\{t_1, t_2, \dots, t_m\}$  as  $\mathcal{W}_k^i$ . Therefore, for two consecutive points  $t_1, t_2 \in \mathcal{W}_k^i$  with  $t_1 < t_2$ , the function  $f_k^i$  is specified at  $t_1$  and  $t_2$ . For  $t \in (t_1, t_2)$ , we have  $f_k^i(t) = ((t_2 - t)f_k^i(t_1) + (t - t_1)f_k^i(t_2))/(t_2 - t_1)$ . For  $t \geq t_m$ , we have  $f_k^i(t) = f_k^i(t_m)$ . We assume that  $\mathcal{W}_k^i$  has poly-size specification.

Even with these assumptions, we show NP-HARDNESS in the full paper [18], and further show that when the monotone property is relaxed, the problem becomes  $n^\epsilon$ -hard to approximate. As shown in Appendix A, our algorithms can also be extended to certain compactly specified differentiable functions  $f$ .

## 3 Whittle’s LP and its Dual

We first present the linear programming relaxation due to Whittle [33]. We *do not* solve this relaxation. Instead, we actually solve the dual of a slightly different relaxation which we present in the next section. In this



section, we simply present the relaxation, its dual, and poly-size equivalent versions.

For each arm  $i$  and  $k \in \mathcal{S}_i$ , we have variables  $\{x_{kt}^i, t \geq 1\}$  and  $\{y_{kt}^i, t \geq 1\}$ . These variables capture the probabilities in the optimal policy that when the arm is in state  $k$  and was last played  $t$  steps ago, it is played and not played respectively. These quantities are well-defined for ergodic policies. Therefore, the linear program (WHITTLE) in Fig. 1 is clearly a relaxation of the optimal policy. Let its optimal value be denoted  $OPT$ . The program effectively encodes the constraints on the evolution of the state of each arm separately, connecting them only by the constraint that at most one arm is played *in expectation* every step. This LP has infinite size, and we will fix that aspect in this section.

We first eliminate the  $y_{kt}^i$  variables by substitutions to obtain the following equivalent formulation.

$$\begin{aligned} \text{Maximize} \quad & \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \geq 1} r_k^i x_{kt}^i \\ & \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \geq 1} x_{kt}^i \leq 1 \\ & \sum_{k \in \mathcal{S}_i} \sum_{t \geq 1} t x_{kt}^i \leq 1 \quad \forall i \\ & \sum_{j \in \mathcal{S}_i, j \neq k} \sum_{t \geq 1} x_{kt}^i q^i(k, j) f_k^i(t) \\ & = \sum_{j \in \mathcal{S}_i, j \neq k} \sum_{t \geq 1} x_{jt}^i q^i(j, k) f_j^i(t) \quad \forall k \in \mathcal{S}_i \\ & x_{kt}^i \geq 0 \quad \forall i, k, t \end{aligned}$$

We now show that the LP has polynomial size when the  $f_k^i$  are piece-wise linear with poly-size specification. In Appendix A, we show that the LP can also be solved to arbitrary accuracy for many differentiable functions  $f$  (e.g., the FEEDBACK MAB problem [15]). Take the dual of the above relaxation. The first constraint has multiplier  $\lambda$ , the second set of constraints have multipliers  $h_i$ , and the final equality constraints have multipliers  $p_k^i$ . For notational convenience, define:

$$\Delta P_k^i = \sum_{j \in \mathcal{S}_i, j \neq k} (q^i(k, j)(p_j^i - p_k^i))$$

We obtain the following dual:

$$\begin{aligned} \text{Minimize} \quad & \lambda + \sum_{i=1}^n h_i \\ \lambda + t h_i & \geq r_k^i + f_k^i(t) \Delta P_k^i \quad \forall i, k \in \mathcal{S}_i, t \geq 1 \\ \lambda, h_i & \geq 0 \quad \forall i \end{aligned}$$

Recall from Definition 1 that  $\mathcal{W}_k^i$  is the set of  $t$ 's for which  $f_k^i$  is specified in the input. Since  $f_k^i(t)$  is piece-wise linear, for two consecutive break-points  $t_1 < t_2$  in  $\mathcal{W}_k^i$ , the constraint  $\lambda + t h_i \geq r_k^i + f_k^i(t) \Delta P_k^i$  is true for all  $t \in [t_1, t_2]$  iff it is true at  $t_1$  and at  $t_2$ . This means that the constraints for  $t \notin \mathcal{W}_k^i$  are redundant. Therefore, the above dual is equivalent to the following:

$$\text{Minimize} \quad \lambda + \sum_{i=1}^n h_i \quad (D1)$$

$$\begin{aligned} \lambda + t h_i & \geq r_k^i + f_k^i(t) \Delta P_k^i \quad \forall i, k \in \mathcal{S}_i, t \in \mathcal{W}_k^i \\ \lambda, h_i & \geq 0 \quad \forall i \end{aligned}$$

Taking the dual of the above program, we finally obtain the polynomial size relaxation for MONOTONE bandits, which we denote (WHITTLE-POLY). This is shown in Figure 2, and its value is precisely  $OPT$ .

#### 4 The Balanced Linear Program

We do not solve Whittle's relaxation. Instead, we solve the modification of the dual (D1), which we denote (BALANCE). The additional constraint in (BALANCE) is the constraint  $\lambda = \sum_{i=1}^n h_i$ , which is a "dual balancing" condition that makes our later analysis possible.

$$\text{Minimize} \quad \lambda + \sum_{i=1}^n h_i \quad (\text{BALANCE})$$

$$\begin{aligned} \lambda + t h_i & \geq r_k^i + f_k^i(t) \Delta P_k^i \quad \forall i, k \in \mathcal{S}_i, t \in \mathcal{W}_k^i \\ \lambda & = \sum_{i=1}^n h_i \\ \lambda, h_i & \geq 0 \quad \forall i \end{aligned}$$

The primal linear program corresponding to (BALANCE) is the following (where we place an unconstrained multiplier  $\omega$  to the final constraint of (BALANCE)):

$$\begin{aligned} \text{Maximize} \quad & \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} r_k^i x_{kt}^i \quad (\text{LPSCALE}) \\ & \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} x_{kt}^i \leq 1 - \omega \\ & \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} t x_{kt}^i \leq 1 + \omega \quad \forall i \\ & \sum_{j \neq k} \sum_{t \in \mathcal{W}_k^i} x_{kt}^i q^i(k, j) f_k^i(t) \\ & = \sum_{j \neq k} \sum_{t \in \mathcal{W}_j^i} x_{jt}^i q^i(j, k) f_j^i(t) \quad \forall i, k \\ & x_{kt}^i \geq 0 \quad \forall i, k, t \end{aligned}$$

**4.1 Using Complementary Slackness** As noted above, the first step of the algorithm is to solve the linear program (BALANCE). Clearly the value of this LP is at least  $OPT$ . We first interpret the dual along with the balancing condition.

The dual can be seen as a debtor rationing a steady-stream income in order to pay-off the reward of the original system at every step. Suppose that an adversary is controlling the original system, and demand reward  $r_k^i$  when he plays arm  $i$  in state  $k$ . To pay-off the adversary, the debtor gets income  $\lambda + \sum_i h_i$  each time step. Of this, the debtor stores an amount  $h_i$  in arm  $i$ , which he can access later. The debtor uses the

$$\begin{aligned}
& \text{Maximize} \quad \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \geq 1} r_k^i x_{kt}^i && \text{(WHITTLE)} \\
& \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \geq 1} x_{kt}^i \leq 1 \\
& \sum_{k \in \mathcal{S}_i} \sum_{t \geq 1} (x_{kt}^i + y_{kt}^i) \leq 1 && \forall i = 1, 2, \dots, n \\
& \sum_{t \geq 2} x_{kt}^i + \sum_{j \in \mathcal{S}_i, j \neq k} \sum_{t \geq 1} (x_{jt}^i q^i(j, k) f_j^i(t) - x_{kt}^i q^i(k, j) f_k^i(t)) = y_{k1}^i && \forall i, k \in \mathcal{S}_i, t \geq 1 \\
& x_{kt}^i, y_{kt}^i \in [0, 1] && \forall i, k \in \mathcal{S}_i, t \geq 1
\end{aligned}$$

Figure 1: The linear program (WHITTLE).

$$\begin{aligned}
& \text{Maximize} \quad \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} r_k^i x_{kt}^i && \text{(WHITTLE-POLY)} \\
& \sum_{i=1}^n \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} x_{kt}^i \leq 1 \\
& \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} t x_{kt}^i \leq 1 && \forall i \\
& \sum_{j \in \mathcal{S}_i, j \neq k} \sum_{t \in \mathcal{W}_k^i} x_{kt}^i q^i(k, j) f_k^i(t) = \sum_{j \in \mathcal{S}_i, j \neq k} \sum_{t \in \mathcal{W}_j^i} x_{jt}^i q^i(j, k) f_j^i(t) && \forall k \in \mathcal{S}_i \\
& x_{kt}^i \geq 0 && \forall i, k \in \mathcal{S}_i, t \in \mathcal{W}_k^i
\end{aligned}$$

Figure 2: The linear program (WHITTLE-POLY).

additional  $\lambda$  to pay-off the adversary for the current time step. Thus, when the adversary plays arm  $i$  in state  $k$ , the debtor has a total of  $\lambda + th_i$  at his disposal. This corresponds to the LHS of the dual constraint.

Since the rewards differ for every state, the debtor maintains a certain amount of money, or “potential” at each state: When arm  $i$  enters state  $k$ , the debtor pays  $p_k^i$  towards the state. When the adversary plays the arm  $i$  in state  $k$ , the expected amount the debtor has to pay for the new state minus the  $p_k^i$  he can remove from the old state, is exactly  $f_k^i(t) \sum_{j \neq k} q^i(k, j) (p_j^i - p_k^i)$ . With this accounting scheme, the expected amount the debtor has to pay out is  $r_k^i$  plus the above quantity, which is the RHS of the dual constraint.

Therefore, the dual finds a the minimum income that the debtor needs in order to stay solvent over the long term. The balancing condition we impose in addition to the dual advises the debtor to equally exploit the two ways of paying off the adversary: paying the adversary now with  $\lambda$  or storing  $h_i$  in the arm to pay the adversary later. This intuitive condition will be essential in our 2-approximation analysis.

We now show the following properties of the optimal solution to (BALANCE) using complementary slackness conditions between (BALANCE) and (LPSCALE). *From now on, we only deal with the optimal solutions to the above programs, so all variables correspond to the optimal setting.*

LEMMA 4.1. *Recall that  $OPT$  is the optimal value to (WHITTLE-POLY). Since any feasible solution to (BAL-*

*ANCE) is feasible to (D1), in the optimal solution to (BALANCE),  $\lambda = \sum_{i=1}^n h_i \geq OPT/2$ .*

The next lemma is the crux of the analysis, where for any arm being played in any state, we use complementary slackness to explicitly relate the dual variables to the reward obtained. Note that unlike the analyses of primal-dual algorithms, our proof needs to use both the *exact* primal as well as dual complementary slackness conditions. This aspect requires us to actually solve the dual optimally.

LEMMA 4.2. *One of the following is true for the optimal solution to (BALANCE): Either there is a trivial 2-approximation by repeatedly playing the same arm; or for every arm  $i$  with  $h_i > 0$  and for every state  $k \in \mathcal{S}_i$ , there exists  $t \in \mathcal{W}_k^i$  such that the following LP constraint is tight with equality.*

$$(4.1) \quad \lambda + th_i \geq r_k^i + f_k^i(t) \Delta P_k^i$$

*Proof.* Note that if  $\omega \leq -1$  or  $\omega \geq 1$ , then the values of (LPSCALE) is 0, but the optimal value of (LPSCALE) is at least  $OPT > 0$ . Thus, in the optimal solution to (LPSCALE),  $\omega \in (-1, 1)$ .

The optimal solutions to (BALANCE) and (LPSCALE) satisfy the following complementary slackness conditions (recall from above that  $\omega > -1$  so that  $1 + \omega > 0$ ):

$$(4.2) \quad h_i > 0 \quad \Rightarrow \quad \sum_{k \in \mathcal{S}_i} \sum_{t \in \mathcal{W}_k^i} t x_{kt}^i = 1 + \omega > 0$$

$$(4.3) \quad \lambda + th_i > r_k^i + f_k^i(t)\Delta P_k^i \quad \Rightarrow \quad x_{kt}^i = 0$$

Suppose that for some  $i$  such that  $h_i > 0$ , and for some  $k \in \mathcal{S}_i$ , we have  $\lambda + th_i > r_k^i + f_k^i(t)\Delta P_k^i$  for every  $t \in \mathcal{W}_k^i$ . By condition (4.3),  $x_{kt}^i = 0 \forall t \in \mathcal{W}_k^i$ , which trivially implies that  $x_{kt}^i f_k^i(t) = 0 \forall t \in \mathcal{W}_k^i$ .

Now, given that for a certain arm  $i$  and state  $k$ ,  $x_{kt}^i f_k^i(t) = 0 \forall t$ . Therefore, in the following constraint in (LPSCALE):

$$\sum_{l \neq k} \sum_{t \in \mathcal{W}_k^i} x_{kt}^i f_k^i(t) q^i(k, l) = \sum_{l \neq k} \sum_{t \in \mathcal{W}_k^i} x_{lt}^i f_l^i(t) q^i(l, k)$$

the LHS is zero because  $x_{kt}^i f_k^i(t) = 0$ , which means the RHS is zero. Since all variables are non-negative, this implies that for any  $j \in \mathcal{S}_i$  with  $q^i(j, k) > 0$ , we have  $x_{jt}^i f_j^i(t) = 0$  for all  $t \in \mathcal{W}_j^i$ .

Recall (from Section 2) that we assumed the graph on the states with edges from  $j$  to  $k$  if  $q^i(j, k) > 0$  is strongly connected. Therefore, by repeating the above argument, we get  $\forall j, t \in \mathcal{W}_j^i, x_{jt}^i f_j^i(t) = 0$ .

By Condition (4.2), since  $h_i > 0$ , there exists  $j \in \mathcal{S}_i$  and  $t \in \mathcal{W}_j^i$ , such that  $x_{jt}^i > 0$  (or else the sum in Condition (4.2) is zero). By what we proved in the previous paragraph, this implies that  $f_j^i(t) = 0$ , which implies that  $f_j^i(1) = 0$  by the MONOTONE property. Since  $x_{jt}^i > 0$ , using Condition (4.3) and plugging in  $f_j^i(t) = 0$ , we get  $\lambda + th_i = r_j^i$ . Moreover, by plugging in  $f_j^i(1) = 0$  into the  $t = 1$  constraint of (BALANCE), we get  $\lambda + h_i \geq r_j^i$ . These two facts imply that  $\lambda + h_i = r_j^i$ . The above implies that the policy that starts with arm  $i$  in state  $j$  and always plays this arm obtains per-step reward  $\lambda + h_i > OPT/2$ .

In the remaining discussion, we assume that the above lemma does not find an arm  $i$  that yields reward at least  $OPT/2$ . This means that  $\forall i, k$ , there exists some  $t \in \mathcal{W}_k^i$  that makes Inequality (4.1) tight.

LEMMA 4.3. *For any arm  $i$  such that  $h_i > 0$ , and state  $k \in \mathcal{S}_i$ , if  $\Delta P_k^i < 0$ , then:*

$$\lambda + h_i = r_k^i + f_k^i(1)\Delta P_k^i$$

*Proof.* By Lemma 4.2 and our assumption above, Inequality (4.1) in Lemma 4.2 is tight for some  $t \in \mathcal{W}_k^i$ . If it is not tight for  $t = 1$ , then since  $f_k^i(t)$  is non-decreasing in  $t$  and since  $\Delta P_k^i < 0$ , it will not be tight for any  $t$ . Thus, we have a contradiction.

## 5 The Index Policy

Start with the optimal solution to (BALANCE). First throw away the arms for which  $h_i = 0$ . By Lemma 4.1, for the remaining arms,  $\sum_i h_i \geq OPT/2$ . Define the following quantities for each of these arms.

DEFINITION 2. *For each  $i$  ( $h_i > 0$  by assumption) and state  $k \in \mathcal{S}_i$ , let  $t_k^i$  be the smallest value of  $t \in \mathcal{W}_k^i$  for which  $\lambda + th_i = r_k^i + f_k^i(t)\Delta P_k^i$  in the optimal solution to (BALANCE). By Lemma 4.2,  $t_k^i$  is well-defined for every  $k \in \mathcal{S}_i$ .*

DEFINITION 3. *For arm  $i$ , partition the states  $\mathcal{S}_i$  into states  $\mathcal{G}_i, \mathcal{I}_i$  as follows:*

1.  $k \in \mathcal{G}_i$  if  $\Delta P_k^i < 0$ . (By Lemma 4.3,  $t_k^i = 1$ .)
2.  $k \in \mathcal{I}_i$  if  $\Delta P_k^i \geq 0$ .

With the notation above, the policy is now presented in Figure 3. In this policy, if arm  $i$  has been in state  $k \in \mathcal{I}_i$  for less than  $t_k^i$  steps, it is defined to be “not ready” for play. Once it has waited for  $t_k^i$  steps, it becomes “ready” and can be played. Moreover, if arm  $i$  moves to a state in  $k \in \mathcal{G}_i$ , it is continuously played until it moves to a state in  $\mathcal{I}_i$ .

### INDEX Policy

1. **Exploit:** Some arm  $i$  moves to a state  $k \in \mathcal{G}_i$ :
  - (a) Play this arm exclusively as long as it remains in a state in  $\mathcal{G}_i$ .
  - (b) **Goto** step (2).
2. **Explore:**
  - (a) Play any “ready” arm  $i$  in state  $k \in \mathcal{I}_i$ . (If no arm is “ready”, do not play at this step.)
  - (b) **If** the arm moves to state in  $\mathcal{G}_i$ :  
**goto** Step (1), **else goto** Step (2a).

Figure 3: The INDEX Policy.

Intuitively, the states in  $\mathcal{G}_i$  are the “exploitation” or “good” states. (In section 5.1, we analyze the policy using a potential function, and in these states, the potentials decrease in expectation on playing. This means that the analysis is using the high rewards earned in these states to balance out the lower rewards in other states.) On the contrary, the states in  $\mathcal{I}_i$  are “exploration” or “bad” states, so the policy waits until it has a high enough probability of exiting these states before playing them. In both cases,  $t_k$  corresponds to the “recovery time” of the state, which is 1 in a “good” state but could be large in a “bad” state.

Although we have not explicitly defined our policy as an index policy, we can easily describe it using the following indices. Place a dummy arm yielding no reward and having index 0. When the state of arm  $i$  is good state  $k \in \mathcal{G}_i$ , the index is 2. When the arm is in bad state  $k \in \mathcal{I}_i$  and not “ready”, its index is  $-1$ , and when it gets “ready”, the index is 1. Ties are broken arbitrarily.

**5.1 Lyapunov Function Analysis** We use a Lyapunov (potential) function argument to show that the policy described in Figure 3 is a 2-approximation. Define the potential for each arm at any time as follows. (Recall the definition of  $t_k^i$  from Definition 2, as well as the quantities  $\lambda$ ,  $h_i$  from the optimal solution of (BALANCE).)

DEFINITION 4. *If arm  $i$  moved to state  $k \in \mathcal{S}_i$  some  $y$  steps ago ( $y \geq 1$  by definition), the potential is  $p_k^i + h_i(\min(y, t_k^i) - 1)$ .*

Therefore, whenever the arm  $i$  enters state  $k$ , its potential is  $p_k^i$ . If  $k \in \mathcal{I}_i$ , the potential then increases at rate  $h_i$  for  $t_k^i - 1$  steps, after which it remains fixed until the arm is played. Our policy plays arm  $i$  only if its current potential is  $p_k^i + h_i(t_k^i - 1)$ .

We finally complete the analysis in the following lemma. The proof crucially uses the ‘‘balance’’ property of the dual, which states that  $\lambda = \sum_i h_i \geq OPT/2$ . Let  $\Phi_T$  denote the total potential at any step  $T$  and let  $R_T$  denote the total reward accrued until that step. Define the function  $\mathcal{L}_T = t \cdot OPT/2 - R_T - \Phi_T$ . Let  $\Delta R_T = R_{T+1} - R_T$  and  $\Delta \Phi_T = \Phi_{T+1} - \Phi_T$ .

LEMMA 5.1.  $\mathcal{L}_T$  is a Lyapunov function. i.e.,  $\mathbf{E}[\mathcal{L}_{T+1} | \mathcal{L}_T] \leq \mathcal{L}_T$ . Equivalently, at any step:

$$\mathbf{E}[\Delta R_T + \Delta \Phi_T | R_T, \Phi_T] \geq OPT/2$$

*Proof.* At a given step, suppose the policy does nothing, then all arms are ‘‘not ready’’. The total increase in potential is precisely  $\Delta \Phi_T = \sum_i h_i \geq OPT/2$ .

On the other hand, suppose that the policy plays arm  $i$ , which is currently in state  $k$  and has been in that state for  $y \geq t_k^i$  steps. The change in reward  $\Delta R_T = r_k^i$ . Moreover, the current potential of the arm must be  $\Phi_T = p_k^i + h_i(t_k^i - 1)$ . The new potential follows the following distribution:

$$\Phi_{T+1} = \begin{cases} p_j^i, & \text{with probability } f_k^i(y)q^i(k, j) \quad \forall j \neq k \\ p_k^i, & \text{with probability } 1 - \sum_{j \neq k} f_k^i(y)q^i(k, j) \end{cases}$$

Therefore, if arm  $i$  is played, the change in potential is:

$$\mathbf{E}[\Delta \Phi_T] = f_k^i(y) \sum_{j \in \mathcal{S}_i, j \neq k} (q^i(k, j)(p_j^i - p_k^i)) - h_i(t_k^i - 1)$$

From the description of the INDEX policy,  $y = t_k^i = 1$  if  $k \in \mathcal{G}_i$ . Therefore,  $y$  might be strictly greater than  $t_k^i$  only when  $k \in \mathcal{I}_i$ . In that case  $\Delta P_k^i \geq 0$  by Definition 3, so that  $f_j^i(y)\Delta P_k^i \geq f_j^i(t_k^i)\Delta P_k^i$  by the MONOTONE property (since  $y \geq t_k^i$ ).

Therefore, for the arm  $i$  being played, regardless of whether  $k \in \mathcal{G}_i$  or  $k \in \mathcal{I}_i$ ,

$$\begin{aligned} \Delta R_T + \mathbf{E}[\Delta \Phi_T] &= r_k^i + f_k^i(y)\Delta P_k^i - h_i(t_k^i - 1) \\ &\geq r_k^i + f_k^i(t_k^i)\Delta P_k^i - h_i t_k^i + h_i \\ &= \lambda + h_i > OPT/2 \end{aligned}$$

where the last equality follows from the definition of  $t_k^i$  (Definition 2). Since the potentials of the arms not being played do not decrease (since all  $h_i > 0$ ), the total change in reward plus potential is at least  $OPT/2$ . Refer Figure 4 for a ‘‘picture proof’’ when  $k \in \mathcal{I}_i$ . This completes the proof.

By their definition, the potentials  $\Phi_T$  are bounded independent of the time horizon, by telescoping summation, the above lemma implies that  $\lim_{T \rightarrow \infty} \frac{\mathbf{E}[R_T]}{T} \geq OPT/2$ . We finally have:

THEOREM 5.1. *The INDEX policy is a 2 approximation for MONOTONE bandits.*

**Acknowledgment.** We thank Saswati Sarkar, Shivnath Babu, Jerome Le Ny, and Alex Slivkins for helpful discussions concerning parts of this work.

## References

- [1] D. Adelman and A. J. Mersereau. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research*, 2008.
- [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-armed bandit problem. In *Proc. 36<sup>th</sup> Annual Symp. on Foundations of Computer Science*, 1995.
- [5] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, second edition, 2001.
- [6] 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.
- [7] D. Bertsimas and J. Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Oper. Res.*, 48(1):80–90, 2000.
- [8] 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.
- [9] D. P. de Farias and N. Megiddo. Combining expert advice in reactive environments. *J. ACM*, 53(5):762–799, 2006.



- [10] A. Flaxman, A. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proc. 16<sup>th</sup> Annual ACM-SIAM Symp. on Discrete Algorithms*, 2005.
- [11] 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.
- [12] A. Goel, S. Guha, and K. Munagala. Asking the right questions: Model-driven optimization using probes. In *Proc. 25<sup>th</sup> ACM Symp. on Principles of Database Systems*, 2006.
- [13] K. Goseva-Popstojanova and K. S. Trivedi. Stochastic modeling formalisms for dependability, performance and performability. In *Performance Evaluation: Origins and Directions*, pages 403–422, 2000.
- [14] S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *Proc. 39<sup>th</sup> ACM Symp. on Theory of Computing (STOC)*, 2007.
- [15] S. Guha and K. Munagala. Approximation algorithms for partial-information based stochastic control with Markovian rewards. In *Proc. 48<sup>th</sup> IEEE Symp. on Foundations of Computer Science (FOCS)*, 2007.
- [16] S. Guha and K. Munagala. Model-driven optimization using adaptive probes. In *Proc. 18<sup>th</sup> ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2007.
- [17] S. Guha, K. Munagala, and S. Sarkar. Information acquisition and exploitation in multichannel wireless systems. *CoRR*, arXiv:0804.1724, 2008.
- [18] S. Guha, K. Munagala, and P. Shi. Approximation algorithms for restless bandit problems. *CoRR*, arXiv:0711.3861, 2008.
- [19] J. T. Hawkins. *A Lagrangian decomposition approach to weakly coupled dynamic optimization problems and its applications*. PhD thesis, Operations Research Center, Massachusetts Institute of Technology, 2003.
- [20] T. Javidi, B. Krishnamachari, Q. Zhao, and M. Liu. Optimality of myopic sensing in multi-channel opportunistic access. In *Proc. IEEE International Conf. in Communications (ICC '08)*, 2008.
- [21] S. M. Kakade and M. J. Kearns. Trading in markovian price models. In *COLT*, pages 606–620, 2005.
- [22] R. Levi, M. Pál, R. Roundy, and D. B. Shmoys. Approximation algorithms for stochastic inventory control models. In *IPCO*, pages 306–320, 2005.
- [23] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.
- [24] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *Proc. 20<sup>th</sup> Intl. Colloq. on Automata, Languages and Programming (ICALP)*, pages 40–51, 1993.
- [25] K. Munagala and P. Shi. The stochastic machine replenishment problem. In *IPCO*, 2008.
- [26] J. Le Ny, M. Dahleh, and E. Feron. Multi-UAV dynamic routing with partial observations using restless bandits allocation indices. In *Proceedings of the 2008 American Control Conference*, 2008.
- [27] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queuing network control. *Math. Oper. Res.*, 24(2):293–305, 1999.
- [28] A. Slivkins and E. Upfal. Adapting to a changing environment: The Brownian restless bandits. In *COLT*, 2008.
- [29] J. N. Tsitsiklis. A short proof of the Gittins index theorem. *Annals of Appl. Probab.*, 4(1):194–199, 1994.
- [30] V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- [31] G. Weiss. Branching bandit processes. *Probab. Engng. Inform. Sci.*, 2:269–278, 1988.
- [32] P. Whittle. Arm acquiring bandits. *Ann. Probab.*, 9:284–292, 1981.
- [33] P. Whittle. Restless bandits: Activity allocation in a changing world. *Appl. Probab.*, 25(A):287–298, 1988.
- [34] N. E. Young. The  $k$ -server dual and loose competitiveness for paging. *Algorithmica*, 11(6):525–541, 1994.
- [35] Q. Zhao, B. Krishnamachari, and K. Liu. On myopic sensing for multi-channel opportunistic access, 2007. *CoRR*, arXiv:0712.0035, 2007.

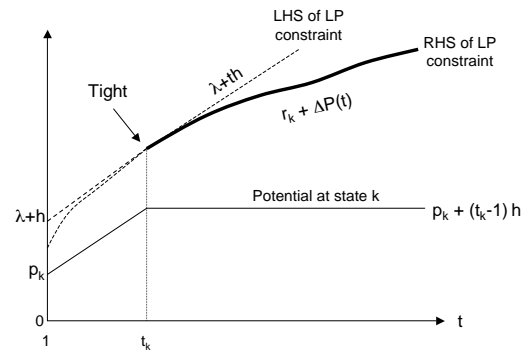


Figure 4: Proof of Lemma 5.1. Define  $\Delta P(t) = f_k^i(t) \sum_{j \in S_i, j \neq k} (q^i(k, j)(p_j^i - p_k^i))$ . The growth of the potential is shown on the lower piece-wise linear function. The upper set of curves represent the LHS and RHS of the LP constraints for given  $i, k$ . The tight point  $t_k^i$  is where the potential switches to being constant.

## A FEEDBACK MAB Problem

In this problem, which was studied independently by [15, 35, 20, 26] in the contexts of scheduling transmissions across multiple wireless channels and routing Unmanned Aerial Vehicles (UAVs), there is a 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 bursty Markovian process which does not depend on whether the arm is played or not at a time slot. Let  $s_{it}$  denote the state of arm  $i$  at time  $t$ . Then denote the transition probabilities of the Markov chain as follows:  $\Pr[s_{i(t+1)} = g_i | s_{it} = b_i] = \alpha_i$  and

$\Pr[s_{i(t+1)} = b_i | s_{it} = g_i] = \beta_i$ . The  $\alpha_i, \beta_i, r_i$  values are specified as input. To ensure ‘‘burstiness’’, assume  $\alpha_i + \beta_i \leq 1 - \delta$  for some small  $\delta > 0$  specified as part of the input. The evolution of states for different arms are independent. Any policy chooses at most one arm to play every time slot. Each play 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.

In this section, we show that this problem is a special case of MONOTONE bandits. We further show how to solve the LP approximately in polynomial time. Finally, we show that the gap of Whittle’s relaxation is  $e/(e-1) \approx 1.58$ , indicating that our analysis for MONOTONE bandits is reasonably tight.

For integer  $t \geq 1$ , define:  $v_{it} = \Pr[s_{it} = g_i | s_{i0} = b_i]$  and  $u_{it} = \Pr[s_{it} = g_i | s_{i0} = g_i]$ . Focus on a particular arm  $i$ , and omit the subscript  $i$ . For integer  $t \geq 1$ , define  $\gamma_t = (1 - \alpha - \beta)^t$ . We have:  $v_t = \frac{\alpha}{\alpha + \beta}(1 - \gamma_t)$  and  $1 - u_t = \frac{\beta}{\alpha + \beta}(1 - \gamma_t)$ . It follows that  $1 - u_t$  and  $v_t$  are increasing concave functions of  $t$ .

**Reduction to MONOTONE bandits:** The FEEDBACK MAB problem is a special case of MONOTONE bandits. Each arm  $i$  has two states  $\mathcal{S}_i = \{g_i, b_i\}$ , where  $r_g^i = r_i$  and  $r_b^i = 0$ . Let  $q^i(g, b) = q^i(b, g) = 1$ . Let  $f_g^i(t) = 1 - u_{it}$  and  $f_b^i(t) = v_{it}$ . This is a valid reduction since  $v_{it}$  is the probability that if the arm was observed in state  $b_i$  during the last play, the current play will find it in state  $g_i$ . Similarly,  $1 - u_{it}$  is the probability if the arm was observed to be in state  $g_i$  during the last play, the current play will find it in state  $b_i$ . Both  $v_{it}$  and  $1 - u_{it}$  increase with  $t$ , which completes the reduction.

**A.1 Solving (BALANCE) in Polynomial Time** In the case of FEEDBACK MAB, the  $f_k^i(t)$ ’s in (BALANCE) are specified using a closed-form expression rather than breakpoints. This implies (BALANCE) does not have polynomial size. However, as we outline below, we can still solve (BALANCE) to any degree of accuracy in polynomial time, and complementary slackness is approximately preserved as well.

First observe that (BALANCE) can be written as follows, where  $p_i = p_g^i - p_b^i$ . It is easy to see that  $p_i \geq 0$ .

$$\text{Minimize} \quad \lambda + \sum_{i=1}^n h_i \quad (\text{BALANCE})$$

$$\begin{aligned} \lambda + th_i &\geq r_i - (1 - u_{it})p_i && \forall i, t \geq 1 \\ \lambda + th_i &\geq v_{it}p_i && \forall i, t \geq 1 \\ \lambda &= \sum_i h_i \\ \lambda, h_i, p_i &\geq 0 && \forall i \end{aligned}$$

The above LP can be solved by performing a binary search over  $\lambda$ . Note that if a certain  $\lambda$  is feasible, so are all larger values. Fix some  $\lambda$ . For this  $\lambda$ , for each  $i$  in turn, compute the smallest  $h_i$  satisfying the first two constraints by binary search. For some  $h_i$ , choose  $p_i$  as the largest value so that  $v_{it}p_i$  touches  $\lambda + th_i$ . This can be computed by simple function maximization since the LHS is a straight line and the RHS is concave and increasing. For this  $p_i$ , if  $\lambda + h_i < r_i - \beta p_i$ , then the given  $h_i$  cannot be feasible (and neither can smaller values). On the other hand, if  $\lambda + h_i > r_i - \beta p_i$ , then both  $h_i$  and  $p_i$  can be reduced, so that a smaller  $h_i$  is also feasible. Continue in this fashion until either  $h_i = 0$ , or there exists  $p_i$  so that both the first and second set of constraints are tight for some (possibly different)  $t$ . This is the smallest feasible  $h_i$ .

Once all  $h_i$  are computed for this  $\lambda$ , check if  $\lambda > \sum_i h_i$ . If so, then  $\lambda$  can be reduced, else this choice of  $\lambda$  is not feasible for the LP. Finally perform binary search on  $\lambda$  until the third constraint is approximately satisfied. This yields a  $(1 + \epsilon)$  approximate solution to the LP in polynomial time for any  $\epsilon > 0$ . Note that this process also guarantees that if  $h_i > 0$ , then both the first and second set of constraints are tight for some  $t$ , hence ensuring complementary slackness.

**A.2 Gap of the LP Relaxation** Consider  $n$  i.i.d. arms with  $n\beta \ll 1$ ,  $\alpha = \beta/(n-1)$  and  $r = 1$ . Each arm is in state  $g$  w.p.  $1/n$ , so that all arms are in state  $b$  w.p.  $1/e$  and the maximum possible reward is  $1 - 1/e$  even with complete information.

LEMMA A.1. *Whittle’s LP has value  $1 - O(\sqrt{n\beta})$  for  $n\beta \ll 1$ ,  $\alpha = \beta/(n-1)$  and  $r = 1$ .*

*Proof.* Using the results in [33, 15], it is easy to show that Whittle’s LP will construct  $n$  identical single-arm policies such that each policy always plays in state  $g$ , and plays in state  $b$  after some  $t$  steps. The value  $t$  is chosen so that the rate of play for each arm is  $1/n$ . The rate of play is given by the formula:  $\frac{1/\beta + 1/v_t}{1/\beta + t/v_t}$ . Since this is  $1/n$ , we have  $v_t = \beta(t-n)/(n-1)$ . The reward of each arm is  $\frac{1/\beta}{1/\beta + t/v_t} = \frac{t-n}{n(t-1)}$ , so that the objective of Whittle’s LP is  $1 - \Theta(n/t)$ . From  $v_t = \beta(t-n)/(n-1)$ , we obtain  $1 - (1 - \beta')^t = \beta'(t-n)$ , where  $\beta' = \alpha + \beta = \beta \frac{n}{n-1}$ . This holds for  $t = \Theta(\sqrt{n/\beta})$  provided  $n\beta \ll 1$ . Plugging this value of  $t$  into the value of the LP completes the proof.

COROLLARY A.1. *The integrality gap of Whittle’s LP is arbitrarily close to  $e/(e-1)$  as  $\beta \rightarrow 0$ .*