Whittle index based Q-learning for restless bandits with average reward

Konstantin E. Avrachenkov*, Vivek S. Borkar†

*Inria Sophia Antipolis, 2004 Route des Lucioles, Valbonne 06902, France, *k.avrachenkov@inria.fr*
†Indian Institute of Technology, Powai, Mumbai, 400076, India, *borkar.vs@gmail.com*

***Abstract*—A novel reinforcement learning algorithm is introduced for multiarmed restless bandits with average reward, using the paradigms of Q-learning and Whittle index. Specifically, we leverage the structure of the Whittle index policy to reduce the search space of Q-learning, resulting in major computational gains. Rigorous convergence analysis is provided, supported by numerical experiments. The numerical experiments show excellent empirical performance of the proposed scheme.**

***Keywords*—reinforcement learning; restless bandits; Whittle index; Q-learning; average reward**

I. Introduction

Restless bandits have found numerous applications for various scheduling and resource allocation problems, such as wireless communication [1], [31], [40], web crawling [6], [9], [36], congestion control [5], [7], [8], queueing systems [4], [18], [23], [29], cluster and cloud computing [20], [35], machine maintenance [24], target tracking [37] and clinical trials [46]. See [22], [26], [43] for book-length accounts of theory and applications of restless bandits. While restless bandits can be viewed as a special case of classical Markov decision processes, this suffers from curse of dimensionality because the state space grows exponentially in the number of arms. In fact, the problem is provably hard in the sense of belonging to the complexity class PSPACE [38]. One very successful heuristic in this context has been the celebrated Whittle index policy [49], which firstly relaxes the 'hard' constraint of using a certain number of arms at each time, to doing so on the average. Thereby it allows a decoupling of the problem into multiple individual controlled Markov chains via the Lagrange multiplier, using the fact that both the reward and the constrained functional are separable. This leads to a state space that grows linearly in the number of arms. Then, secondly, these chains are coupled through the control policy based on ordinal comparison of a scalar function of their individual states, viz., the so-called Whittle index. While this is known not to be optimal in general, it works very well in practice and is asymptotically optimal in a certain sense [29], [48].

The use of Whittle index policy, however, requires full knowledge of the system, both in the relatively few cases where it is known explicitly (e.g., in [9]) or when it has to be numerically calculated. This is often not the case in practice, sometimes called the 'curse of modeling'. The uncertainties can be either parametric or structural, or both. In either case, the classical adaptive control schemes (e.g., [25]) or off-the-shelf reinforcement learning schemes (e.g., [10], [11], [45]) become computationally unmanageable if applied directly to restless bandits. These schemes typically do not exploit the special structure available in the problem, in this case the Whittle indexability.

In this work, we combine the Q-learning algorithm for average reward [2] with a tuning scheme for the Whittle indices. This yields a provably convergent learning algorithm with excellent empirical performance on test cases. In case the arms are statistically identical, the algorithm is particularly economical because it learns the common Q-values and Whittle index. The algorithm has a notably simple form compared to above works, and can be executed in both on-line and off-line modes, the latter allowing for off-policy iterations.

The main novelty of our work lies in the fact that our Q-learning algorithm is tuned to a specific policy class, viz., the Whittle index policy. This drastically reduces the search space for policy by exploiting a specific additional structure in the problem, viz., its Whittle indexability. The task is now to learn the Whittle indices. If $d_1, \cdots, d_N$ are the cardinalities of the state spaces of the N bandit's arms, then our scheme needs $\sum_{\alpha=1}^{N}(2d_\alpha^2 + d_\alpha)$ updates per iteration, whereas a vanilla Q-learning would need $2^N \prod_{\alpha=1}^{N} d_\alpha$ updates. For $d_\alpha \equiv 5 \ \forall i$ and $N = 100$, this number is 10^{100} which already exceeds the number of atoms in the universe. Compare this with our scheme where the corresponding number is 5500 and the scheme is efficient on most standard computational devices. In fact this issue persists even after using function approximation, e.g., DQN [34], because the maximization operator requires searching a large action space. We elaborate on this in Section 4. On the other hand, compared to earlier works along similar lines, we have the advantage that our scheme does not assume additional structure such as an optimal threshold policy for decoupled chains [17] or explicit formula for the Whittle index [6]. In addition, [17] lacks a completely rigorous convergence proof whereas [6] has numerical issues and is useful only when the model is known but its parameters are not. Another precursor is [21] which does not have a rigorous

Work is supported in part by the DST-Inria project "Machine Learning for Network Analytics" IFC/DST-Inria-2016-01/448. VB is also supported in part by a J. C. Bose Fellowship from the Government of India and KA is also supported in part by Nokia Bell Labs and ANSWER project PIA FSN2 (P15 9564-266178 \DOS0060094). This is the author version of the paper in *Automatica* 139(3):110186, 2022.

proof, systematically underestimates the reward, and converges to a random limit.

A related line of work [42], [41] develops learning schemes for threshold policies. The Whittle index itself, however, is not a simple threshold, but a function of the state. So the problems are quite distinct. Furthermore, [42], [41] use policy gradient based methods as in [32] which are better suited when a low dimensional parameter such as a threshold is involved. In contrast, Whittle index is a function on the state space for which they are ill suited. At the same time, Whittle index is defined in terms of an equality. So a much simpler scheme is used here, which makes incremental changes towards forcing this equality.

It is worth noting that both the present work as well as some others such as primal-dual type schemes for constrained Markov decision processes [15] fall within the ambit of the larger paradigm of using two time scale iterations when both control and parametric optimization are simultaneously present, and it makes sense to perform the latter on a slower time scale to reduce the problem to a bi-level optimization, which makes it computationally more tractable.

The paper is organized as follows. The next section summarizes the Whittle index formalism. Section 3 describes the algorithm in detail. Section 4 presents numerical experiments. Section 5 provides convergence analysis, which relies upon [2], [12], [16], [27] and [28].

The notation will be as follows. The subscripts n, m, resp., t, will stand for discrete and continuous time indices. For the Q-learning iterates, we use superscript α to indicate the α-th arm. Another superscript (c) denotes a scaling factor. Letters i, j, k etc. indicate elements of the state space. When we want to keep a state variable fixed during an iteration, we distinguish it as $\hat{i}, i^*$ etc. We use letters u, v to denote the control variables. We use $[[q_{ij}]]$ to denote a matrix whose (i, j)th entry is q_{ij}. Letter λ will denote the Whittle subsidy, also used as a subscript when required.

II. Whittle index for restless bandits

We first recall briefly the Whittle paradigm [49]. The restless bandit problem consists of a finite collection of (say) N Markov chains, each with two possible modes of evolution, active and passive. These correspond to two possibly distinct transition matrices and reward functions that may depend on the particular chain under consideration. The problem is to keep active exactly (or at most) $M < N$ out of N chains so as to maximize the reward. The passive chains also evolve, albeit with a different transition matrix, instead of remaining frozen in the state they occupied at the time of turning passive. This makes this scenario distinct from the classical multiarmed rested bandits. The latter problem has an explicit optimal solution in terms of the Gittins index policy [22], which assigns to each chain a function of its state called the Gittins index. At every time instant, given the current state profile, the corresponding indices are sorted in a decreasing order and the top M rendered active, ties being resolved arbitrarily. The restless bandit problem, however, is much harder, provably so as already noted. Whittle replaced it by a more tractable relaxation wherein the per stage constraint of M out of N chains is relaxed to an average constraint that requires only the asymptotic fraction of active chains to be M (this is made precise below). This alone is not enough. Motivated by the classical Lagrange multiplier formulation for this problem, Whittle introduced a 'subsidy' for passivity and defined the problem to be (Whittle) indexable if the set of passive states increases monotonically from the empty set to the whole state space when the subsidy is increased from $-\infty$ to ∞. In this case, he defined the (Whittle) index to be the value of the subsidy, as a function of the current state, for which both active and passive modes are equally desirable. The Whittle policy then is to sort these indices in a decreasing order for the current state profile of the chains, and render the top M active. While not necessarily optimal, the policy is known to do well in practice and is provably optimal in a certain limiting sense as mentioned above[1].

Specifically, we consider $N > 1$ controlled Markov chains $\{X_n^\alpha, n \geq 0\}$, $1 \leq \alpha \leq N$, on a finite state space $S = \{1, 2, \cdots, d\}$, $1 < d < \infty$ with control space $\mathcal{U} := \{0, 1\}$. The controlled transition kernel

$$(i, j, u) \in S^2 \times \mathcal{U} \mapsto p^\alpha(j|i, u) \in [0, 1]$$

for the α-th chain satisfies $\sum_j p^\alpha(j|i, u) = 1 \ \forall \ i, u$, and has the interpretation of 'probability of going from state i to state j under control u'. The control variable u is binary, corresponding to two modes of operation, active ($u = 1$) and passive ($u = 0$). Define the increasing family of σ-fields $\mathcal{F}_n := \sigma(X_m^\alpha, U_m^\alpha, 1 \leq \alpha \leq N, m \leq n)$, $n \geq 0$. The 'controlled Markov property' is

$$P(X_{n+1}^\alpha = j|\mathcal{F}_n) = p^\alpha(j|X_n^\alpha, U_n^\alpha), \ \forall \ n \geq 0, \ 1 \leq \alpha \leq N,$$

where $\{U_n^\alpha\}_{n \geq 0}, 1 \leq \alpha \leq N$, are the $\mathcal{U}$-valued control processes, called 'admissible controls'. A special subclass denoted SP is that of stationary policies wherein $U_n^\alpha = \varphi^\alpha(X_n^\alpha)$ for some $\varphi^\alpha : S \mapsto \{0, 1\}$, $n \geq 0$. The individual chains are called 'arms' of the restless bandit. Let $r^\alpha : (i, u) \in S \times \mathcal{U} \mapsto \mathcal{R}$ denote prescribed per stage reward function for the α-th chain. These controlled Markov chains are assumed to satisfy:
(C0) (Unichain property) There exists a distinguished state $i_0 \in S$ that is reachable with strictly positive probability from any other state under any stationary policy (SP).

Since $d = |S| < \infty$, this implies in particular that for $\tau := \min\{n \geq 0 : X_n^\alpha = i_0\}$,

$$\max_{k \in S, SP} E[\tau | X_0^\alpha = k] < \infty. \tag{1}$$

The objective is to maximize the long run average reward

$$\liminf_{n \uparrow \infty} \frac{1}{n} E\left[\sum_{m=0}^{n-1} \sum_{\alpha=1}^{N} r^\alpha(X_m^\alpha, U_m^\alpha)\right], \tag{2}$$

[1] Note that this work is in the domain of 'Markov bandits', a branch of Markov decision processes, distinct from the current activity on multiarmed bandits in machine learning that deals with rewards independent across arms and time, or their dependent variants quite distinct from Markov bandits [30]. Their objective is also different, viz., to bound the asymptotic regret, unlike the Markov bandits which seek optimality for the classical criteria of Markov decision theory. Work on Markov bandits in machine learning community is only recently beginning to pick up, see, e.g., [33].

subject to the constraint: for a prescribed $M < N$,

$$\sum_{\alpha=1}^{N} U_n^\alpha = M, \quad \forall n. \tag{3}$$

That is, at each time instant, only M arms are activated.

The Whittle relaxation is to replace the 'per time instant' constraint (3) by a 'time-averaged constraint'

$$\liminf_{n\uparrow\infty} \frac{1}{n} E\left[\sum_{m=0}^{n-1}\sum_{\alpha=1}^{N} U_m^\alpha\right] = M. \tag{4}$$

This renders it a classical 'constrained Markov decision process' [3]. While this is a significant simplification, the problem is still unwieldy. Whittle's ingenious observation was to use the fact that it is a problem with separable cost and constraint and invoke the Lagrangian relaxation to decouple it into individual control problems given the Lagrange multiplier λ. That is, we consider now the unconstrained control problem of maximizing

$$\liminf_{n\uparrow\infty} \frac{1}{n} E\left[\sum_{m=0}^{n-1}(r^\alpha(X_m^\alpha, U_m^\alpha) + \lambda(1 - U_m^\alpha))\right] \tag{5}$$

separately for each α.[2] The dynamic programming equation then is

$$\begin{aligned} V^\alpha(i) = \max\Big(&r^\alpha(i,1) + \sum_j p^\alpha(j|i,1)V^\alpha(j), \\ &r^\alpha(i,0) + \lambda + \sum_j p^\alpha(j|i,0)V^\alpha(j)\Big) - \beta^\alpha \end{aligned} \tag{6}$$

$$\begin{aligned} = \max_{u\in\mathcal{U}} \Big(&u(r^\alpha(i,1) + \sum_j p^\alpha(j|i,1)V^\alpha(j)) + (1-u) \times \\ &(r^\alpha(i,0) + \lambda + \sum_j p^\alpha(j|i,0)V^\alpha(j))\Big) - \beta^\alpha, \end{aligned} \tag{7}$$

with $(V^\alpha(\cdot), \beta^\alpha) \in \mathcal{R}^d \times \mathcal{R}$ the unknown variables. Under **(C0)**, β^α is unique and equals the optimal reward. V is unique up to an additive constant. The optimal decision $u^*(i)$ in state i then is given by the maximizer in the right hand side of (6) [39]. If we consider individual arms separately, the superscript α will be dropped henceforth, used only when it is needed. Note, however, that this does not mean that all arms are assumed statistically identical.

Define the Q-value as

$$\begin{aligned} Q(i,u) \;:=\; &u(r(i,1) + \sum_j p(j|i,1)V(j)) + (1-u) \times \\ &(r(i,0) + \lambda + \sum_j p(j|i,0)V(j))) - \beta. \end{aligned} \tag{8}$$

This satisfies the equation

$$\begin{aligned} Q(i,u) \;=\; &ur(i,1) + (1-u)(\lambda + r(i,0)) - \\ &\beta + \sum_j p(j|i,u)\max_v Q(j,v), \end{aligned} \tag{9}$$

for $i \in S, u \in U$. Under **(C0)**, this has a solution (Q,β) where β is uniquely specified as the optimal reward and Q is unique up to an additive scalar, just as for (6). The set $\{j \in S : u^*(j) = 1\}$ is the set of states when the arm is active, its complement being the set of states when it is passive. Whittle's insight was to view the Lagrange multiplier as a 'subsidy' for passivity. He defined the problem to be indexable when the set of passive states increases monotonically from the empty set to all of S as the subsidy is increased from $-\infty$ to ∞. In this case, he defines the (Whittle) index for state $\hat{k}$ to be the value $\lambda(\hat{k})$ of λ for which both active and passive modes are equally preferred in state $\hat{k}$. That is,

$$\begin{aligned} \lambda(\hat{k}) \;:=\; &r(\hat{k},1) + \sum_j p(j|\hat{k},1)V(j) - r(\hat{k},0) \\ &- \sum_j p(j|\hat{k},0)V(j). \end{aligned} \tag{10}$$

This is equivalent to solving

$$Q(\hat{k},1) - Q(\hat{k},0) = 0, \tag{11}$$

for $\lambda = \lambda(\hat{k})$, where the λ-dependence of the left hand side is not rendered explicit as per our convention thus far.

Our algorithm is a two time scale iteration wherein the faster timescale performs Q-learning for a 'static' subsidy λ_n, the latter in reality changing on a slower time scale. Thus it tracks the Q-value corresponding to the slowly changing subsidy, which in turn is updated on a slower timescale by a simple tuning scheme suggested by (11). The Whittle index is a function of $\hat{k} \in S$, so for large state spaces, one may compute it for a suitably chosen subset of S and interpolate.

III. Q-learning for Whittle index

Q-learning is one of the oldest and most popular reinforcement learning scheme for approximate dynamic programming, due to Watkins [47]. Originally developed for infinite horizon discounted rewards, we shall be using a variant for average reward from [2]. For the controlled Markov chain $\{X_n^\alpha\}$ above with average reward (5), the 'RVI Q-learning' algorithm of (2.7) in [2] is as follows (with a key difference we highlight later). Fix a stepsize sequence $\{a(n)\}$ satisfying $\sum_n a(n) = \infty$ and $\sum_n a(n)^2 < \infty$. For each $i \in S,\ u \in \mathcal{U}$, do:

$$\begin{aligned} Q_{n+1}(i,u) \;=\; &Q_n(i,u) + a(\nu(i,u,n))I\{X_n = i, U_n = u\} \\ &\times\Big((1-u)(r(i,0)+\lambda) + ur(i,1) + \\ &\max_{v\in\mathcal{U}} Q_n(X_{n+1}, v) - f(Q_n) - \; Q_n(i,u)\Big), \end{aligned} \tag{12}$$

where[3]

$$f(Q) = \frac{1}{2d}\sum_{i\in S}(Q(i,0) + Q(i,1)). \tag{13}$$

[2] We have dropped a constant factor involving M from the total reward so as to match it with Whittle's set-up. This does not affect the optimization problem because we finally do an ordinal comparison that is unaffected by this.

[3] This is not the unique choice of $f(\cdot)$, see [2].

Here for $i \in S, u \in \mathcal{U}$,

$$\nu(i,u,n) = \sum_{m=0}^{n} I\{X_m = i, U_m = u\},$$

is the 'local clock' for the pair (i,u) counting the updates of the (i,u)-th component.

Our objective is to learn the Whittle index, i.e., the value $\lambda(\hat{k})$ of λ defined in (10), equivalently in (11), for which active and passive modes are equally desirable for a given $\hat{k} \in S$. Hence we also have an updating scheme for λ, leading to a coupled iteration for each $\hat{k} \in S$. The first component is the same as (12) except for the replacement of λ by the estimated Whittle index $\lambda_n(\hat{k})$. Thus for each $\hat{k} \in S$, we perform the iteration

$$\begin{aligned} Q_{n+1}(i,u;\hat{k}) &= Q_n(i,u;\hat{k}) + a(\nu(i,u,n))\times \\ &I\{X_n = i, U_n = u\}\Big((1-u)(r(i,0) + \lambda_n(\hat{k})) + ur(i,1) \\ &+ \max_{v \in \mathcal{U}} Q_n(X_{n+1}, v; \hat{k}) - f(Q_n(\hat{k})) - \ Q_n(i,u;\hat{k})\Big) \end{aligned} \tag{14}$$

along with an update for learning the Whittle index $\lambda(\hat{k})$ for state $\hat{k}$ given by: with a prescribed stepsize sequence $\{b(n)\}$ satisfying $\sum_n b(n) = \infty$, $\sum_n b(n)^2 < \infty$ and $b(n) = o(a(n))$, do

$$\lambda_{n+1}(\hat{k}) = \lambda_n(\hat{k}) + b(n)\left(Q_n(\hat{k},1;\hat{k}) - Q_n(\hat{k},0;\hat{k})\right). \tag{15}$$

We use the 'hat notation' $\hat{k}$ to emphasize that $\hat{k}$ is the $\hat{k}$-th component of the Whittle index estimation evolving on the slow time scale.

Note that we need to estimate $Q(i,u;\hat{k})$ for each arm α. However, if some arms are statistically identical, we can take advantage of this and collect statistics simultaneously from statistically identical arms. For instance, in the case of homogeneous arms and shared memory architecture, we need to update only $2d^2 + d$ variables, whereas they would have been $(2d)^N$ with Q-learning applied directly without the Whittle scheme.

The control actions at time n are defined as follows: Let $0 < \epsilon < 1$ be prescribed. With probability $(1-\epsilon)$, we sort arms in the decreasing order of the estimated Whittle indices $\lambda_n(X_n^\alpha), 1 \leq \alpha \leq N$, and render the top M arms active, the remaining arms are passive. Ties are broken according to some pre-specified convention. With probability ϵ, we render active M random arms, chosen uniformly and independently, the rest passive. This uniform randomization with probability ϵ is essential in order to ensure that all state-action pairs are sampled 'frequently', i.e.,

$$\liminf_{n \uparrow \infty} \frac{\nu(i,u,n)}{n+1} \geq \Delta \quad \text{a.s.} \tag{16}$$

for some $\Delta > 0$ and all i, u. This is because by the martingale law of large numbers,

$$\begin{aligned} \lim_{n\uparrow\infty} \frac{1}{n}\Big(\sum_{m=0}^{n}(I\{X_{m+1} = i, U_{m+1} = u\} \\ - P(U_{m+1} = u|i)p(i|X_m, U_m))\Big) = 0 \text{ a.s.} \end{aligned}$$

and therefore the l.h.s. of (16) equals

$$\begin{aligned} &\liminf_{n\uparrow\infty} \frac{\sum_{m=0}^{n} I\{X_{m+1} = i, U_{m+1} = 0\}}{n} \\ \geq\ & \frac{\sum_{m=0}^{n} \epsilon p(i|X_m, U_m)}{n} \\ \geq\ & \epsilon \min\{p(k|j,v) : p(k|j,v) > 0, k, j \in S, v \in \mathcal{U}\} > 0. \end{aligned}$$

This ensures (16), i.e., adequate exploration, though at the expense of settling for near-optimality rather than optimality. The prospect of slowly decreasing ϵ with n has been explored in literature, see, e.g., [44].

We used following stepsize sequences, which gave good performance in practice:

$$a(n) = \frac{C}{\lceil \frac{n}{500} \rceil}, \ b(n) = \frac{C'}{1 + \lceil \frac{n \log n}{500} \rceil} I\{n (\text{mod } N) \equiv 0\}. \tag{17}$$

Define $h(Q,\lambda) = [[h(Q,\lambda)_{iu}]]_{i \in S, u \in \{0,1\}} : \mathcal{R}^{2d} \times \mathcal{R} \mapsto \mathcal{R}^{2d}$ by

$$\begin{aligned} h(Q,\lambda)_{iu} \ :=\ & (1-u)(r(i,0) + \lambda) \ + ur(i,1) \\ & + \sum_j p(j|i,u) \max_{v \in \{0,1\}} Q(j,v) - f(Q). \end{aligned} \tag{18}$$

Letting $Q(\hat{k})$ denote $Q(\cdot,\cdot;\hat{k})$ suitably vectorized, we also define $M_n(\hat{k}) := [[M_n(\hat{k})_{iu}]]_{i \in S, u \in \{0,1\}}$ by

$$\begin{aligned} M_{n+1}(\hat{k})_{iu} &:= (1-u)(r(i,0) + \lambda_n(\hat{k})) + ur(i,1) + \\ & \max_{v \in \mathcal{U}} Q_n(X_{n+1}, v; \hat{k}) - f(Q_n(\hat{k})) - h(Q_n(\hat{k}), \lambda_n(\hat{k}))_{iu}\Big). \end{aligned} \tag{19}$$

Then $\{M_n(\hat{k})\}$ are martingale difference sequences w.r.t. $\{\mathcal{F}_n\}$, i.e., they are adapted to $\{\mathcal{F}_n\}$ and satisfy $E[M_{n+1}(\hat{k})_{iu}|\mathcal{F}_n] = 0 \ \forall\ i, \hat{k}, u, n$. Rewrite (14) as

$$\begin{aligned} & Q_{n+1}(i,u;\hat{k}) = Q_n(i,u;\hat{k}) \\ & + \ a(\nu(i,u,n))I\{X_n = i, U_n = u\} \\ & \times(h(Q_n(\hat{k}), \lambda_n(\hat{k}))_{iu} - Q_n(i,u;\hat{k}) + M_{n+1}(\hat{k})_{iu}) \end{aligned} \tag{20}$$

In view of the fact $b(n) = o(a(n))$, the coupled iterates (20), (15) form a two time scale stochastic approximation algorithm in the sense of [16], section 6.1, with (20) operating on the faster time scale and (15) on the slower time scale. We exploit this fact later in the convergence analysis.

IV. Numerical examples

Let us illustrate the proposed scheme with two examples, both with statistically identical arms (i.e., the transition matrix and reward do not depend on the arm).

A. Example with circulant dynamics

We first test our scheme on the example from [21]. The example has four states and the dynamics are circulant: when an arm is passive ($u = 0$), resp. active ($u = 1$), the state evolves according to the transition probability matrices

$$P_0 = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}, \quad \text{and} \quad P_1 = P_0^T,$$

respectively. The rewards do not depend on the action and are given by $r(1,0) = r(1,1) = -1$, $r(2,0) = r(2,1) = 0$, $r(3,0) = r(3,1) = 0$, and $r(4,0) = r(4,1) = 1$. Intuitively, there is a preference to activate an arm when the arm is in state 3. Indeed, the exact values of the Whittle indices, calculated in [21], are as follows: $\lambda(1) = -1/2$, $\lambda(2) = 1/2$, $\lambda(3) = 1$, and $\lambda(4) = -1$, which give priority to state 3. Consider a scenario with $N = 100$ arms, out of which $M = 20$ are active at each time. We initialize our algorithm with $\lambda_0(i) = 0$, and $Q(i,u) = r(i,u)$, $\forall i \in S$.

In this example, we assumed the shared memory architecture and took full advantage of the fact that the arms are statistically identical. This helps to collect the statistics very quickly and results in a rapid convergence of the algorithm. We first set the exploration parameter as $\epsilon = 0.1$.

In Figure 1 we present the convergence of the estimated values of the Whittle indices (see (15)) to the exact values. In Figure 2, we present the comparison of the running average reward obtained by our algorithm with that of the algorithm based on the use of the exact Whittle indices from the beginning. We see that the average rewards stabilize in both approaches already after 250 iterations. The 10% loss of efficiency of our scheme with respect to the approach using the exact Whittle indices is due to the fact that we spend 10% of effort on pure exploration. This actually can be mitigated by decreasing the exploration parameter with time. We notice that as predicted by the theory and confirmed by Figure 1 the estimated Whittle indices in our algorithm converge to the true values.

We also note that the convergence of the running average reward is significantly faster than the convergence of the estimates of the Whittle indices (compare Figure 2 vs Figure 1). This is because the control policy used depends only on the ordinal comparison of the estimated Whittle indices and their order settles much faster than their actual numerical values.

If we set the exploration parameter as $\epsilon = 0.01$, there is hardly any loss of efficiency of our scheme with respect to the scheme using the exact Whittle indices (see Figure 3). Remarkably, the convergence of the running time averaged reward does not seem to suffer. Of course, the convergence of the estimated Whittle indices to the exact values is now slower. However, since the Whittle indices form a discrete set with generous spacing, what matters is actually the ordinal ranking produced by the estimated Whittle indices, which is quite robust, and not their proximity to the exact values.

This controlled chain in fact is not unichain, as under some stationary policies, it splits into two communicating classes. However, any state is reachable from any other under some control, as a result of which the optimal cost does not depend on the initial state and the dynamic programming equation (6) remains valid.

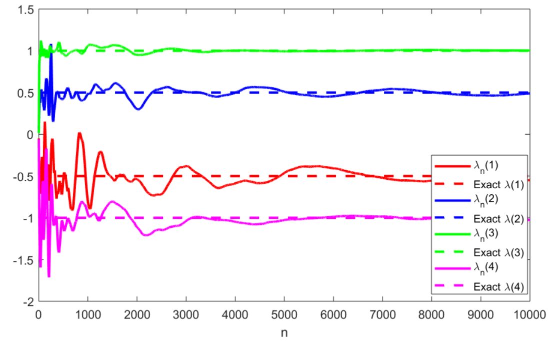

Fig. 1. Estimated (solid lines) and exact (dash lines) Whittle indices in the example with circulant dynamics.

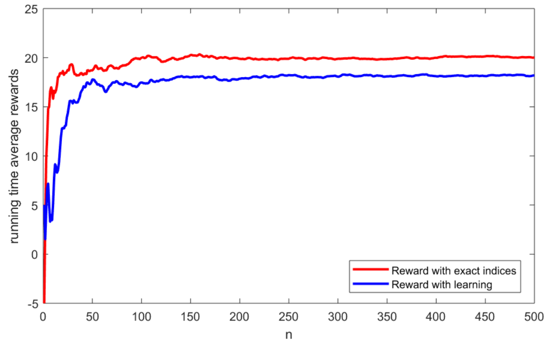

Fig. 2. Rewards comparison in the circulant dynamics ($\epsilon = 0.1$).

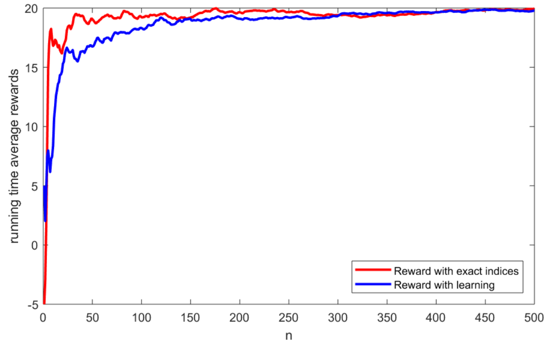

Fig. 3. Rewards comparison in the circulant dynamics ($\epsilon = 0.01$).

B. Example with restart

Now we consider an example where the active action forces an arm to restart from some state. Specifically, we consider an example with 5 states, where in the passive mode ($u = 0$) an arm has tendency to go up the state space, i.e.,

$$P_0 = \begin{bmatrix} 1/10 & 9/10 & 0 & 0 & 0 \\ 1/10 & 0 & 9/10 & 0 & 0 \\ 1/10 & 0 & 0 & 9/10 & 0 \\ 1/10 & 0 & 0 & 0 & 9/10 \\ 1/10 & 0 & 0 & 0 & 9/10 \end{bmatrix},$$

whereas in the active mode ($u = 1$) the arm restarts from state 1 with probability 1, i.e.,

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The rewards in the passive mode are given by $r(k, 0) = a^k$ (in our numerical experiments, we have taken $a = 0.9$) and the rewards in the active mode are all zero.

At least three facts have motivated us to choose this example. Bandits with restarting dynamics have several applications such as congestion control [5], [7], web crawling [9], [6], [36] and machine maintenance [24]. Their Whittle indices can be easily calculated, see e.g., [26], [29]. The upper states are much less visited, if at all, which poses a challenge for learning.

As in the previous example, we consider the scenario with $N = 100$ arms out of which $M = 20$ are active at each time step. The exact Whittle indices are given by: $\lambda(1) = -0.9$, $\lambda(2) = -0.73$, $\lambda(3) = -0.5$, $\lambda(4) = -0.26$, and $\lambda(5) = -0.01$. We initialize the algorithm with $\lambda_0(i) = 0$, and $Q(i, u) = r(i, u)$, $\forall i \in S$.

In Figure 4 we plot the evolution of the estimated Whittle indices with $\epsilon = 0.1$. As expected in this example, the non-homogeneous structure of the state space poses some problems for learning in comparison with the more symmetric example with circulant dynamics. It takes noticeably longer time to learn the Whittle indices for the upper states 4 and 5 in comparison with the lower states 1, 2 and 3.

So far, we have applied decreasing stepsizes recommended in (17). In practice one could also apply constant stepsizes. For instance, in Figure 5 we used constant stepsizes $a = 0.02$, $b = 0.005$. The results are fairly good for all the states except the top state 5. However, the top state is visited rarely and thus the value of its Whittle index is not really relevant for good control of the system. One clear practical advantage of the constant stepsize is the possibility of using this variant for tracking a slowly varying environment.

As a final remark for this section, we would like to mention that the application of both standard Q-learning and neural network based reinforcement learning (e.g., DQN [34]) to the system as a whole is simply not feasible, since in the operation $\max_v Q(i, v)$ we need to search through the space of size 2^{100} during each learning step. We have also tried Monte Carlo approach, where we sample a significant number of possible actions and compared Q-values, or their approximations in the DQN algorithm, for those actions. We could not observe any noticeable improvement in the empirical average reward even after a very large number (10^5) of iterations.

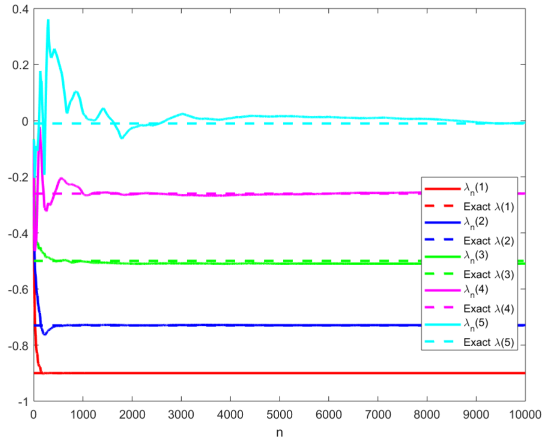

Fig. 4. Estimated (solid lines) and exact (dash lines) Whittle indices in the example with restart.

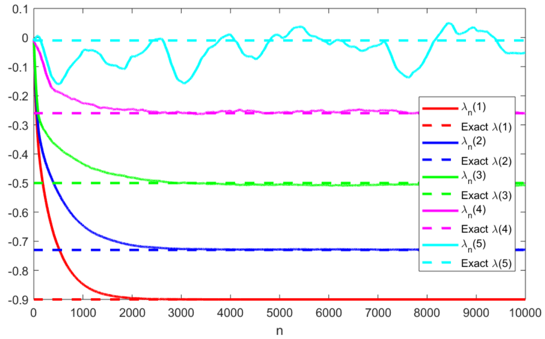

Fig. 5. Estimated (solid lines) and exact (dash lines) Whittle indices in the example with restart. Constant step sizes: $a = 0.02$, $b = 0.005$.

V. Convergence analysis

In addition to **(C0)**, we make the following assumptions:

- **(C1)** The stepsizes $\{a(n)\}$ satisfy: $a(n+1) \leq a(n)$ for sufficiently large n, $\sum_n a(n)^{1+v} < \infty$ for some $v > 0$, and, for $x \in (0, 1)$,

$$\sup_n \frac{a(\lfloor xn \rfloor)}{a(n)} < \infty,$$

$$\sup_{y \in [x,1]} \left| \frac{\sum_{m=0}^{\lfloor yn \rfloor} a(m)}{\sum_{m=0}^{n} a(m)} - 1 \right| \to 0,$$

and for all $x > 0$ and

$$N(n, x) := \min\{m \geq n : \sum_{k=n}^{m} a(k) \geq x\},$$

the limit

$$\lim_{n \uparrow \infty} \frac{\sum_{k=\nu(i,u,n)}^{\nu(i,u,N(n,x))} a(k)}{\sum_{k=\nu(j,v,n)}^{\nu(j,v,N(n,x))} a(k)}$$

exists a.s., for $i, j \in S,\ u, v \in \mathcal{U}$.
These are satisfied, e.g., by $a(n) = \frac{1}{n}$ or $\frac{1}{n \log n}$ from some n on.

- **(C2)** The problem is Whittle indexable.

We prove convergence of the above scheme to the desired limit using a combination of results from the theory of stochastic approximation, in conjunction with [2]. We call the iteration (14) *synchronous* if all components of $Q^{\alpha}_n(\hat{k})$ are updated at the same time, i.e., the indicator $I\{X_n = i, U_n = u\}$ in (14) is dropped and $\nu(i,u,n) = n\ \forall i, u$. Also, for updating the (i,u)-th component, X_{n+1} is replaced by $\mathcal{X}_{n+1}(i,u)$, a simulated random variable independent of all else, with law $p(\cdot|i,u)$. The iterate becomes

$$\begin{aligned} & Q^{\alpha}_{n+1}(i,u;\hat{k}) = Q^{\alpha}_n(i,u;\hat{k}) \ + \\ & \quad a(n)\Big((1-u)(r(i,0)+\lambda_n(\hat{k})) + \ ur(i,1) + \\ & \quad \max_{v\in\{0,1\}} Q^{\alpha}_n(\mathcal{X}_{n+1}(i,u),v;\hat{k}) - \\ & \quad f(Q^{\alpha}_n(\hat{k})) - Q^{\alpha}_n(i,u;\hat{k})\Big). \qquad (21) \end{aligned}$$

This is legitimate only for off-line and therefore off-policy learning. It does, however, provide a step towards analyzing the fully *asynchronous* update (14) based on a single run $\{(X_n, U_n)\}$, which updates only the (X_n, U_n)th component at the n-th time step[4].

Our analysis of the coupled iterations (14)-(15) uses the two time scale analysis of [16], Section 6.1. To facilitate this, we first need the well-posedness of the limiting o.d.e.s. Lemma 1 paves the way for it. We also need a.s. boundedness of the iterates. Lemmas 2 and 3 establish this using the criterion of [28]. This requires verification of the assumptions of [28], which is carried out in Lemma 2 for the synchronous case first for ease of exposition. Indeed, [28] deals with synchronous iterates. Lemma 3 provides the link to extend this to asynchronous iterates where our choice of stepsizes plays a key role. Theorem 1 then provides the convergence argument using the methodology of Section 6.1 of [16] combined with [27].

Recall the function h defined in (18). The limiting o.d.e. for (12) with λ_n frozen at λ is ((3.4) in [2])

$$\dot{Q}_t = h(Q_t, \lambda). \qquad (22)$$

This has as its globally asymptotically stable equilibrium the solution $Q^*_\lambda = [[Q^*_\lambda(i,u)]]$ of (9) with $f(Q^*_\lambda) = \beta_\lambda$ ($:= \beta$ with its λ-dependence made explicit), see Theorem 3.4 of [2]. Define $g : \mathcal{R}^{|S|} \mapsto \mathcal{R}^{|S|}$ by:

$$g_i(\lambda) = Q^*_\lambda(i,1) - Q^*_\lambda(i,0).$$

Lemma 1 The map $\lambda \mapsto Q^*_\lambda$ is Lipschitz.

Proof We have $f(\hat{Q}^*_\lambda) = \hat{\beta}_\lambda := \lambda\times$ the stationary probability of the set of passive states $+$ the stationary expectation of $r(X_n, U_n)$, under the optimal policy. For any stationary policy φ, the reward for this problem would be likewise, i.e., affine in λ with slope $\in [0,1]$ and the constant offset bounded uniformly in φ. Since $\hat{\beta}_\lambda$ is the maximum thereof over all policies for each λ, $\lambda \mapsto \hat{\beta}_\lambda$ is Lipschitz with Lipschitz constant ≤ 1. Fix $i_0 \in S, u_0 \in \mathcal{U}$ and let $\tau :=$ the first time $X_n = i_0, U_n = u_0$. Then for a fixed stationary policy $\varphi : S \mapsto \mathcal{U}$, letting $Q^{(\varphi)}_\lambda$ denote the corresponding vector of Q-values, we have the representation (see, e.g., the arguments of Lemma 2.5, pp. 79-80, [13])

$$\begin{aligned} Q^{(\varphi)}_\lambda(i,u) \quad = \quad & (1-u)(\lambda + r(i,0)) + ur(i,1) + \hat{\beta}_\lambda + \\ & \sum_j p(j|i,\varphi(i)) E_j\left[\sum_{n=0}^{\tau-1}((1-\varphi(X_n))\times\right. \\ & \left.(\lambda + r(X_n,0)) + \varphi(X_n)r(X_n,1) - \beta_\lambda)\right]. \end{aligned}$$

Using the fact that $E_j[\tau]$ is bounded uniformly in j, φ under our hypotheses, it follows that the above is Lipschitz in λ with a Lipschitz constant independent of φ. Hence $Q^*_\lambda(i,u) := \sup_\varphi Q^{(\varphi)}_\lambda(i,u)$ (see *ibid.*) is Lipschitz. □

Lemma 2 Under the hypotheses (C0), (C1) and (C2), the updates of (15),(21) remain a.s. bounded.

Proof This essentially follows from the results of [28]. Let us verify the assumptions (A1)-(A5) of [28], pp. 109-110, one by one.

1. (A1) of [28] requires that h, g are Lipschitz. This is obvious for h and follows from Lemma 1 for g.

2. In the notation of [28], $M^{(1)}_n, M^{(2)}_n$ correspond to resp., $M_n(\hat{k})$ in (19) and the process that is identically zero. Both of these are martingale difference sequences (the latter trivially so). Furthermore, $\forall n$,

$$E\left[\|M_{n+1}(\hat{k})\|^2|\mathcal{F}_n\right] \leq K\left(1 + \|Q_n(\hat{k})\|^2 + \|\lambda_n(\hat{k})\|^2\right)$$

a.s. by the Lipschitz property of the functions involved. The zero process trivially satisfies such an inequality. This is precisely (A2) of [28].

3. (A3) of [28] requires that $\sum_n a(n) = \sum_n b(n) = \infty$, $\sum_m (a(n)^2 + b(n)^2) < \infty$, and $b(n) = o(a(n))$, which hold here by assumption.

4. For (A4), consider the o.d.e. tracked by the iterates (14) in the synchronous case. We need to use the analysis of [2] for the *synchronous* case for which Assumptions 2.1 and 2.2 of *ibid.* suffice, Assumptions 2.3 and 2.4 are not needed. Of these, the first is simply our assumption (C0), whereas the second is satisfied by the function f defined in (13) by construction. The limiting o.d.e. with λ_n frozen at λ is (22) has the globally asymptotically stable equilibrium $Q^*_\lambda = [[Q^*_\lambda(i,u)]]$. (Here and until the end of the lemma proof we omit the notation for $\hat{k}$

[4] One can consider more general forms of asynchrony where some but not all, and not necessarily only one, components are updated at each time. The analysis will be similar.

from Q^* and λ to make the equations more transparent. We keep in mind that Q^* and λ depend on $\hat{k}$ throughout the proof.) The limit

$$h_\infty(Q, \lambda) := \lim_{c \uparrow \infty} \frac{h(cQ, c\lambda)}{c}$$

then corresponds to the Q-learning problem for average reward control with constant running reward $\equiv \lambda$ for passive states and zero reward for active states. As above, by Theorem 3.4 of [2], this converges to the unique $\hat{Q}^*_\lambda$ for which $f(\hat{Q}^*_\lambda) = \hat{\beta}_\lambda := \lambda \times$ the stationary probability of the set of passive states under the optimal policy. By Lemma 1, $\hat{Q}^*_\lambda(i, u)$ is Lipschitz in λ. Furthermore, for $\lambda = 0$, both the active and passive running rewards are zero, and therefore $\hat{\beta}_0 = 0$. So the unique solution to (9) with $f(\hat{Q}^*_0) = \hat{\beta}_0$ is the zero vector. It follows that the o.d.e.

$$\dot{Q}_t = h_\infty(Q_t, \lambda) \tag{23}$$

has $\hat{Q}^*_\lambda$ as its unique asymptotically stable equilibrium, which reduces to the origin when $\lambda = 0$. This is precisely (A4) of [28].

5. Consider the limit

$$g_\infty(\lambda) = \lim_{c \uparrow \infty} \frac{g(\hat{Q}^*_{c\lambda}, c\lambda)}{c}.$$

Letting

$$\hat{r}_c(i, u) := \Big(ur(1, i) + (1 - u)(c\lambda + r(0, i))\Big)/c$$

denote the scaled running reward and $\beta^{(c)} := \beta/c$ the scaled optimal reward, both are seen to be uniformly bounded for $c \geq 1$. Divide both sides of equation (9) by c and let $c \uparrow \infty$. For each $c \geq 1$, it becomes the counterpart of (9) for running reward r_c that remains uniformly bounded over $c \in [1, \infty)$. Let τ be the first hitting time of a fixed state $i_0 \in S$ accessible from every other state as per (C0) and V_c the value function for the average reward problem with running reward r_c and $V_c(i_0) = 0$. For now, we write Q-values as $Q^{(c)}(\cdot, \cdot)$ to show the c-dependence explicitly. Using a standard representation for the value function ([13], p. 79),

$$\begin{aligned}
& Q^{(c)}(i, u)/c \\
= \quad & \hat{r}_c(i, u) - \beta^{(c)} + \sum_j p(j|i, u) V_c(j) \\
= \quad & \hat{r}_c(i, u) - \beta^{(c)} + \sum_j p(j|i, u) \times \\
& \max_{v \in SP} E\Big[\sum_{m=0}^{\tau} (r_c(X_m, v(X_m)) \\
& - \beta^{(c)}) | X_0 = j\Big] \\
\leq \quad & C\left(1 + \max_{SP,\ j \in S} E\left[\tau | X_0 = j\right]\right) < \infty
\end{aligned}$$

by (1), for a suitable constant C. Thus $Q^{(c)}(\cdot, \cdot)/c, \beta^{(c)}$ remain bounded as $c \uparrow \infty$. Any limit point $(Q^{(\infty)}_\lambda(\cdot, \cdot),\ \beta^{(\infty)}_\lambda)$ thereof (with the λ-dependence rendered explicit again) satisfies

$$\begin{aligned}
Q^{(\infty)}_\lambda(i, 0) \quad &= \quad \lambda - \beta^{(\infty)}_\lambda + \sum_j p(j|i, 0) \times \\
& \qquad \max_v Q^{(\infty)}_\lambda(j, v), \qquad (24) \\
Q^{(\infty)}_\lambda(i, 1) \quad &= \quad -\beta^{(\infty)}_\lambda + \sum_j p(j|i, 1) \times \\
& \qquad \max_v Q^{(\infty)}_\lambda(j, v). \qquad (25)
\end{aligned}$$

Consider three distinct cases:

- *Case 1:* For $\lambda > 0$ as $c \uparrow \infty$, eventually $u = 0$ is optimal for all states. Then $\beta^{(\infty)}_\lambda = \lambda$ and $Q^{(\infty)}(i, 0) = \max_v Q^{(\infty)}(i, v)\ \forall i$. By (24),

  $$Q^{(\infty)}_\lambda(i, 0) = \sum_j p(j|i, 0) Q^{(\infty)}_\lambda(j, 0)\ \forall i,$$

  implying

  $$Q^{(\infty)}_\lambda(i, 0) \equiv \text{ a constant } = \max_v Q^{(\infty)}(i, v)\ \forall i.$$

  Substracting (24) from (25),

  $$Q^{(\infty)}_\lambda(i, 1) - Q^{(\infty)}_\lambda(i, 0) = -\beta^{(\infty)}_\lambda = -\lambda.$$

- *Case 2:* For $\lambda < 0$ as $c \uparrow \infty$, eventually $u = 1$ is optimal for all states. Equation (25) then implies that $\beta^{(\infty)}_\lambda = 0$, otherwise the equation does not have a solution: Iterating (25) leads to $Q^{(\infty)}_\lambda(i, 1) = -n\beta^{(\infty)}_\lambda +$ a bounded quantity. This becomes unbounded unless $\beta^{(\infty)}_\lambda = 0$. In turn, (25) with $\beta^{(\infty)}_\lambda = 0$ leads to $Q^{(\infty)}_\lambda(i, 1) \equiv$ a constant independent of i. From (24), we then have

  $$Q^{(\infty)}_\lambda(i, 1) - Q^{(\infty)}_\lambda(i, 0) = -\lambda + \beta^{(\infty)}_\lambda = -\lambda.$$

- *Case 3:* For $\lambda = 0$, $\beta^{(\infty)}_\lambda = 0$ and the zero vector trivially satisfies (24), (25). The solution thereof is unique up to an additive scalar. This leads to

  $$Q^{(\infty)}_\lambda(i, 1) - Q^{(\infty)}_\lambda(i, 0) = 0 = -\lambda.$$

We have proved that $g_\infty(\lambda) = -\lambda\ \forall \lambda$. The limiting o.d.e. $\dot{\lambda}_t = g_\infty(\lambda_t) = -\lambda_t$ has zero as its unique globally asymptotically stable equilibrium. This verifies (A5) of [28].

Theorem 10 (iv) of [28] then implies a.s. boundedness of the iterates, i.e.,

$$\sup_n |\lambda_n(\hat{k})| < \infty,\ \sup_n |Q^\alpha_n(i, u; \hat{k})| < \infty\ \forall\ i, \hat{k}, u, \quad \text{a.s.}$$

□

Lemma 3 Under the hypotheses (C0), (C1) and (C2), the updates of (14)-(15) remain a.s. bounded.

Proof The only difference with Lemma 2 here is that the fast iterate (14) is asynchronous. First we consider it in isolation, as in [28], section 5.1. From the arguments leading to Theorem 8 therein, the crux is the behavior of the limiting o.d.e. From [14] (see equation (2.11) therein), an asynchronous variant tracks a limiting o.d.e. of the form

$$\dot{Q}_t = \Gamma_t h(Q_t, \lambda_t), \tag{26}$$

where $t \mapsto \Gamma_t$ is a diagonal matrix valued trajectory with non-negative entries on the diagonal of Γ_t for each $t \geq 0$. (These reflect the relative frequencies with which different components get updated, see, e.g., [16], Chapter 7.) Note that we have allowed time dependence of λ so that the framework of section 4 of [28] becomes applicable. Our assumptions (C0) and (C1), together with condition (16) verify resp., Assumptions 2.1, 2.3, 2.4 of [2]), whereas, as already observed, our function f in (13) was explicitly chosen to satisfy Assumption 2.2 therein, viz., for $\mathbf{1} :=$ the vector of all 1's, $f(\mathbf{1}) = 1$ and $f(x + c\mathbf{1}) = f(x) + c)$. Thus as in [14], we have $\Gamma_t \equiv \frac{1}{2d}\times$ the identity matrix. See, e.g., the concluding remark of section 3, p. 850, in *ibid.*, where this claim follows from the proof of Theorem 3.2[5] therein. This makes (26) merely a time-scaled version of (22). That (22) has a unique asymptotically stable equilibrium is already established in Theorem 3.4, p. 689, of [2]. Hence the arguments of [28] apply and the claim follows. □

This brings us to our main result.

Theorem 1 Under the hypotheses (C0), (C1) and (C2), $\lambda_n(\hat{k}) \to$ the Whittle index $\lambda(\hat{k})$ for all $\hat{k} \in S$, a.s.

Proof The claim follows from Theorem 2 of [27]. We begin by verifying the assumptions of [27].

1. The process $\{X_n\}$ takes values in a finite, hence compact state space, and the probability $P(X_{n+1} = i|\mathcal{F}_n)$ for $i \in S$ depends only on X_n, U_n for each n. This verifies (A1) of [27]. Furthermore, this dependence is trivially continuous because the latter take values in finite sets. This verifies (A5) of *ibid.*

2. (A2), (A3) and (A4) of [27] are verified as follows: (A2) requires h defined in (18) to be Lipschitz, which it is. (A3) requires $\{M_n(\hat{k})\}$ to be $\{\mathcal{F}_n\}$-martingales satisfying

$$E\left[\|M_{n+1}(\hat{k})\|^2|\mathcal{F}_n\right] \leq K\left(1 + \|Q_n(\hat{k})\|^2 + \|\lambda_n(\hat{k})\|^2\right).$$

This is also easily verified as in the beginning of the proof of Lemma 2 when we verified (A2) of [28]. Finally, (A4) imposes the standard conditions on $\{a(n)\}, \{b(n)\}$, viz., $\sum_n a(n) = \sum_n b(n) = \infty, \sum_n (a(n)^2 + b(n)^2) < \infty$ and $b(n) = o(a(n))$, that we have already assumed[6].

3. Here we use the stronger condition (A6') of [27] as opposed to (A6). For us, it reduces to the fact that the o.d.e. (23) has a unique globally asymptotically stable equilibrium $Q^*_\lambda :=$ the unique solution of (9) corresponding to $f(Q^*_\lambda) = \beta_\lambda :=$ the optimal cost corresponding to $\lambda_n \equiv \lambda$ (Theorem 3.4, p. 689, [2]). In the notation of [27], this corresponds to $\lambda(\theta)$.

4. (A7) assumes a.s. boundedness of the iterates, which we established in Lemmas 2 and 3.

The final observation we need is the fact that for each fixed $\hat{k}$, the slow iterates $\{\lambda_n(\hat{k})\}$ a.s. track the o.d.e.

$$\dot{\Lambda}_t = Q^*_{\Lambda_t}(\hat{k}, 1) - Q^*_{\Lambda_t}(\hat{k}, 0). \tag{27}$$

This is a straightforward consequence of two timescale analysis of [16], section 6.1, leading to Theorem 2, pp. 66-67, of *ibid.* If $\Lambda_t >$ the Whittle index $\lambda(\hat{k})$ of $\hat{k}$ (excess subsidy), the passive mode is preferred, i.e., $Q^*_{\Lambda_t}(\hat{k}, 0) > Q^*_{\Lambda_t}(\hat{k}, 1)$. Then the r.h.s. is < 0 and Λ_t decreases. Likewise, if the opposite (strict) inequality holds, the r.h.s. is > 0 and Λ_t increases. Thus the trajectory $\Lambda_t, t \geq 0$, remains bounded. Since any well-posed scalar o.d.e. with bounded trajectories must converge to an equilibrium, Λ_t converges to the Λ satisfying $Q^*_\Lambda(\hat{k}, 1) = Q^*_\Lambda(\hat{k}, 0)$, i.e., the Whittle index $\lambda(\hat{k})$. This is unique by hypothesis. By theory of two time scale stochastic approximation (Theorem 2, section 6.1, [16]), we have $\lambda_n(\hat{k}) \to \lambda(\hat{k}),\ \forall \hat{k} \in S$, a.s.

□

VI. Conclusions

We have presented a novel Q-learning algorithm for Whittle indexable restless bandits and justified it both analytically and through numerical experiments. The general philosophy extends easily to related problems such as discounted rewards and related algorithms such as SARSA. An interesting future direction is to combine function approximation with the present scheme in order to handle large state spaces. Another open issue is to obtain convergence rates and regret bounds for this scheme. While some results are available for two time scale stochastic approximations in general (see, e.g., [19]), none seems available for two time scale algorithms with asynchronous iterates.

[5] See also the correction note of *ibid.*

[6] Note that in [27], $a(n) = o(b(n))$, so that the roles of $\{a(n)\}, \{b(n)\}$ are reversed.

References

[1] S. Aalto, P. Lassila, and I. Taboada. Whittle index approach to opportunistic scheduling with partial channel information. *Performance Evaluation*, 136:102052, 2019.

[2] J. Abounadi, D. Bertsekas, and V.S. Borkar. Learning algorithms for Markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2001.

[3] E. Altman. *Constrained Markov decision processes*. CRC Press, 1999.

[4] T.W. Archibald, D.P. Black, and K.D. Glazebrook. Indexability and index heuristics for a simple class of inventory routing problems. *Operations Research*, 57(2):314–326, 2009.

[5] K. Avrachenkov, U. Ayesta, J. Doncel, and P. Jacko. Congestion control of TCP flows in internet routers by means of index policy. *Computer Networks*, 57(17):3463–3478, 2013.

[6] K. Avrachenkov and V.S. Borkar. A learning algorithm for the Whittle index policy for scheduling web crawlers. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1001–1006. IEEE, 2019.

[7] K. Avrachenkov, V.S. Borkar, and S. Pattathil. Controlling G-AIMD using index policy. In *The 56th IEEE Conference on Decision and Control, Melbourne, December*, pages 12–15, 2017.

[8] K. Avrachenkov, A. Piunovskiy, and Y. Zhang. Impulsive control for G-AIMD dynamics with relaxed and hard constraints. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 880–887. IEEE, 2018.

[9] K.E. Avrachenkov and V.S. Borkar. Whittle index policy for crawling ephemeral content. *IEEE Transactions on Control of Network Systems*, 5(1):446–455, 2016.

[10] D.P. Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[11] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[12] S. Bhatnagar. The Borkar–Meyn theorem for asynchronous stochastic approximations. *Systems & Control Letters*, 60(7):472–478, 2011.

[13] V.S. Borkar. *Topics in controlled Markov chains*. Longman Scientific & Technical Harlow, 1991.

[14] V.S. Borkar. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization*, 36(3):840–851, 1998. (Correction note in *ibid.*, 38(2), 662-663, 2000.).

[15] V.S. Borkar. An actor-critic algorithm for constrained Markov decision processes. *Systems & Control Letters*, 54(3):207–213, 2005.

[16] V.S. Borkar. *Stochastic approximation: A dynamical systems viewpoint*, volume 48. Springer, 2009.

[17] V.S. Borkar and K. Chadha. A reinforcement learning algorithm for restless bandits. In *2018 Indian Control Conference (ICC)*, pages 89–94. IEEE, 2018.

[18] V.S. Borkar and S. Pattathil. Whittle indexability in egalitarian processor sharing systems. *Annals of Operations Research*, pages 1–21, 2017.

[19] V.S. Borkar and S. Pattathil. Concentration bounds for two time scale stochastic approximation. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 504–511. IEEE, 2018.

[20] V.S. Borkar, K. Ravikumar, and K. Saboo. An index policy for dynamic pricing in cloud computing under price commitments. *Applicationes Mathematicae*, 44:215–245, 2017.

[21] J. Fu, Y. Nazarathy, S. Moka, and P.G. Taylor. Towards Q-learning the Whittle index for restless bandits. In *2019 Australian & New Zealand Control Conference (ANZCC)*, pages 249–254. IEEE, 2019.

[22] J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

[23] K.D. Glazebrook, C. Kirkbride, and J. Ouenniche. Index policies for the admission control and routing of impatient customers to heterogeneous service stations. *Operations Research*, 57(4):975–989, 2009.

[24] K.D. Glazebrook, H.M. Mitchell, and P.S. Ansell. Index policies for the maintenance of a collection of machines by a set of repairmen. *European Journal of Operational Research*, 165(1):267–284, 2005.

[25] O. Hernández-Lerma. *Adaptive Markov control processes*, volume 79. Springer Science & Business Media, 2012.

[26] P. Jacko. *Dynamic priority allocation in restless bandit models*. Lambert Academic Publishing Saarbrücken, Germany, 2010.

[27] P. Karmakar and S. Bhatnagar. Two time-scale stochastic approximation with controlled Markov noise and off-policy temporal-difference learning. *Mathematics of Operations Research*, 43(1):130–151, 2018.

[28] C. Lakshminarayanan and S. Bhatnagar. A stability criterion for two timescale stochastic approximation schemes. *Automatica*, 79:108–114, 2017.

[29] M. Larrañaga, U. Ayesta, and I.M. Verloop. Asymptotically optimal index policies for an abandonment queue with convex holding cost. *Queueing Systems*, 81(2-3):99–169, 2015.

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

[31] K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of Whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2010.

[32] P. Marbach and J. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 2001.

[33] A. Mete, R. Singh, and P.R. Kumar. Reward biased maximum likelihood estimation for reinforcement learning. *Learning in Dynamics and Control, PLMR*, 2021.

[34] V. Mnih *et al.* Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[35] J. Niño-Mora. Admission and routing of soft real-time jobs to multiclusters: Design and comparison of index policies. *Computers & Operations Research*, 39(12):3431–3444, 2012.

[36] J. Niño-Mora. A dynamic page-refresh index policy for web crawlers. In *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pages 46–60. Springer, 2014.

[37] J. Niño-Mora and S.S. Villar. Sensor scheduling for hunting elusive hiding targets via Whittle's restless bandit index policy. In *International Conference on Network Games, Control and Optimization (NetGCooP 2011)*, pages 1–8. IEEE, 2011.

[38] C.H. Papadimitriou and J.N. Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.

[39] M.L. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[40] V. Raghunathan, V. Borkar, M. Cao, and P.R. Kumar. Index policies for real-time multicast scheduling for wireless broadcast systems. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 1570–1578. IEEE, 2008.

[41] A. Roy, V. Borkar, P. Chaporkar, and A. Karandikar. Low complexity online radio access technology selection algorithm in lte-wifi hetnet. *IEEE Transactions on Mobile Computing*, 19(2):376–389, 2019.

[42] A. Roy, V. Borkar, A. Karandikar, and P. Chaporkar. A structure-aware online learning algorithm for Markov decision processes. In *Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools*, pages 71–78, 2019.

[43] D. Ruiz-Hernandez. *Indexable restless bandits: Index policies for some families of stochastic scheduling and dynamic allocation problems*. VDM Publishing, 2008.

[44] S. Singh, T. Jaakkola, M.L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.

[45] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction (2nd ed.)*. MIT press, 2018.

[46] S.S. Villar, J. Bowden, and J. Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical Science*, 30(2):199, 2015.

[47] C.J.C.H. Watkins. *Learning from delayed rewards*. PhD Thesis, King's College, Cambridge University, 1989.

[48] R.R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.

[49] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25(A):287–298, 1988.