A Novel Implementation of Q-Learning for the Whittle Index

Lachlan J. Gibson[1][0000−0001−7345−7878], Peter Jacko[2,3][0000−0003−3376−0260], and Yoni Nazarathy[1][0000−0003−3071−8462]

[1] The University of Queensland, School of Mathematics and Physics, Australia.
{l.gibson1,y.nazarathy}@uq.edu.au
[2] Lancaster University, UK. p.jacko@lancaster.ac.uk
[3] Berry Consultants, UK.

Abstract. We develop a method for learning index rules for multi-armed bandits, restless bandits, and dynamic resource allocation where the underlying transition probabilities and reward structure of the system is not known. Our approach builds on an understanding of both stochastic optimisation (specifically, the Whittle index) and reinforcement learning (specifically, Q-learning). We propose a novel implementation of Q-learning, which exploits the structure of the problem considered, in which the algorithm maintains two sets of Q-values for each project: one for reward and one for resource consumption. Based on these ideas we design a learning algorithm and illustrate its performance by comparing it to the state-of-the-art Q-learning algorithm for the Whittle index by Avrachenkov and Borkar. Both algorithms rely on Q-learning to estimate the Whittle index policy, however the nature in which Q-learning is used in each algorithm is dramatically different. Our approach seems to be able to deliver similar or better performance and is potentially applicable to a much broader and more general set of problems.

Keywords: Multi-Armed Bandits · Restless Bandits · Reinforcement Learning · Q-learning · Markov Decision Processes

1 Introduction

Problems of dynamic resource allocation are ubiquitous in fields such as public health, business, communications, engineering, and agriculture as they focus on making an efficient dynamic use of limited amounts of resources. Examples include: allocation of patients to treatments in a clinical trial in order to learn about their effectiveness as quickly as possible; allocation of physicians (or beds) to patients in a hospital in order to improve their health; allocation of production lines (or manpower) in a business in order to satisfy demand of customers; allocation of shelf space to products in order to maximise revenue; allocation of online advertisements to users in order to maximise the click-through rate; allocation of webpage designs with different cognitive styles to customers in order to improve their understanding; allocation of agents' time to process intelligence information in order to minimise damage caused by attacks; allocation of frequency spectrum to users in wireless networks in order to achieve a desirable quality of service; allocation of computer power/memory to simulation tasks in order to improve an existing solution. Indeed in almost every technological, societal, or logistical setting one can think of many scenarios where dynamic resource allocation plays a key role.

1.1 Problem Formulation

We consider the following formulation of the problem of dynamic resource allocation. We present it as a Markov decision process (MDP) framework, which is sufficiently versatile to allow for a variety of observability settings. A set of K heterogeneous alternative *projects* compete for a resource with the expected one-period *capacity* of M units. Each resource can be allocated to at most one project, but each project requires to be attended by a certain number of resource units. At the beginning of every time period $t \in \mathcal{T} := \{0, 1, \ldots, T-1\}$ (where T, possibly infinite, is the problem *horizon*), a decision-maker can choose which projects will be allocated to the resource units during the period. Let us denote by $\mathcal{A} := \{0, 1\}$ the *action space* of each project, where action 1 means being allocated to the resource, while action 0 means the opposite.

Each project $k \in \mathcal{K} := \{1, 2, \ldots, K\}$ is formulated as a work-reward MDP given by a tuple $(\mathcal{N}_k, (\boldsymbol{W}_k^a)_{a \in \mathcal{A}}, (\boldsymbol{R}_k^a)_{a \in \mathcal{A}}, (\boldsymbol{P}_k^a)_{a \in \mathcal{A}})$, where $\mathcal{N}_k$ is the *state space*; $\boldsymbol{W}_k^a := (W_{k,n}^a)_{n \in \mathcal{N}_k}$, where $W_{k,n}^a$ is the expected one-period capacity consumption, or *work* required by the project at state n if action a is chosen; $\boldsymbol{R}_k^a := (R_{k,n}^a)_{n \in \mathcal{N}_k}$, where $R_{k,n}^a$ is the expected one-period *reward* earned from the project at state n if action a is chosen; $\boldsymbol{P}_k^a := (p_{k,n,m}^a)_{n,m \in \mathcal{N}_k}$ is the one-period *transition probability matrix*, where $p_{k,n,m}^a$ is the probability for project k evolving from state n to state m if action a is chosen. The dynamics of project k are captured by *state process* $n_k(\cdot)$ and *action process* $a_k(\cdot)$, which correspond to state $n_k(t)$ and action $a_k(t)$ at the beginning of every time period t.

At each time period t, the choice of action $a_k(t)$ for project k in state $n_k(t)$ entails the consumption of allocated capacity (work) $W_{k,n_k(t)}^{a_k(t)}$, the gain of reward $R_{k,n_k(t)}^{a_k(t)}$, and the evolution of the state to $n_k(t+1)$. A policy's outcome is a mapping $t \mapsto \boldsymbol{a}(t)$ for all t, where $\boldsymbol{a}(t) := (a_k(t))_{k \in \mathcal{K}}$ is the vector of project actions. We are interested in the characterisation of policies with desired properties (detailed below) from the set of *admissible policies* Π, which are randomized, history-dependent, non-anticipative, and which satisfy the sample-path resource constraint $\sum_{k \in \mathcal{K}} W_{k,n_k(t)}^{a_k(t)} = M$ at every time period t. Notice that considering the resource constraint as an equality is without loss of generality, because the problem with an inequality ($\leq M$) can always be reformulated by adding a sufficient number of single-state projects, labelled, say, ℓ, with $\mathcal{N}_\ell := \{*\}, W_{\ell,*}^a := a, R_{\ell,*}^a := 0, p_{\ell,*,*}^a := 1$, to be virtually attended by the unallocated resource units.

When all the parameters of the model are known (fully observable setting) or are assumed using an observational, e.g. Bayesian model (partially observable setting), the typical objective in the field of *stochastic optimisation* is to find/characterise a policy π^{ETA} maximising the expected time-average (ETA) reward rate,

$$\mathbb{R}^{\mathrm{ETA}}(T) := \max_{\pi \in \Pi} \lim_{\tau \to T} \mathbb{E}_0^{\pi} \left[\frac{1}{\tau} \sum_{t=0}^{\tau-1} \sum_{k \in \mathcal{K}} R_{k,n_k(t)}^{a_k(t)} \right], \tag{1}$$

where $\mathbb{E}_0^{\pi}$ denotes the expectation over state processes $n_k(\cdot)$ and action processes $a_k(\cdot)$ for all k conditioned on initial states $n_k(0)$ for all k and on policy π. Technical assumptions are required for the maximum and the limit to exist, however for simplicity we use notation as above. Variants of (1), e.g. total discounted reward, are also of interest.

When parameters are not known or it is not desirable to assume any observational model (limited or non-observable/non-parametric setting) and when an analytic solution to a problem is not available, the typical objective in the field of *reinforcement learning* is to find/characterise a policy minimising the expected cumulative regret (ECR) of the reward which captures the lost reward due to the limited knowledge of parameters or observations with respect to the expected time-average reward obtainable in the fully observable setting over an infinite horizon. Several variants of this problem can be considered, depending on the assumptions made about which parameters are known by the decision-maker and what is being observed at each period.

By allowing for a more general set of resources and more general dynamics, the above problem formulation covers the classic multi-armed bandit problem, a popular model in the field of *design of sequential experiments*, which is one of the fundamental topics in statistics and machine learning. The classic *multi-armed bandit* problem can be obtained by setting $M := 1$, $W_{k,n}^{a} := a$, $R_{k,n}^{0} := 0$ for all k, n, a, and taking $\boldsymbol{P}_k^0$ as an identity operator/matrix (i.e. unattended projects do not change state). When unattended projects are allowed to change state (i.e., $\boldsymbol{P}_k^0$ is not necessarily an identity operator) and the resource capacity M is allowed to be any integer between 1 and $K-1$, this more general problem is known as the *restless multi-armed bandit.*

On the other hand, our problem formulation belongs to a broader, multidisciplinary field of *decision making under uncertainty* (or, recently suggested to be called *sequential decision analytics*). The problem as formulated above thus covers an intermediate family of problems: rich enough to cover a range of important real-world problems, while the explicit structure of these problems allows to develop tractable and effective solution methods. The development of methods, algorithms, and the design of low-complexity policies with provable performance guarantees for automatically solving such problems is an *important research challenge.*

1.2 Related Work

Stochastic optimisation focusses on solving a problem by finding an exact or approximate solution of a well specified optimisation model, which captures the essential features of the problem. The objective is typically maximisation of the expected time-average or total discounted reward over a time horizon, which can be finite or infinite. Probabilistic assumptions must be made about the nature of uncertainty, which may lead to misspecified models. Nevertheless, solutions to such models are often good enough for practical purposes. Classical solution approaches based on direct use of dynamic programming occasionally allow for characterising the structure of the optimal policy. However, exact algorithms quickly become intractable because of the *curse of dimensionality.* A much more viable theoretical approach is therefore to develop and study methods that convert a problem into a simpler, tractable one, either by decomposition or by considering it in an asymptotic regime. A plethora of general methods and algorithms is available, e.g. dynamic programming, Lagrangian relaxation and decomposition, approximate dynamic programming, asymptotically optimal policies, look-ahead approximations, myopic policies, etc.

Using stochastic optimisation approaches, structural results have been obtained for certain families of resource allocation problems. In his celebrated result, Gittins established [16, 13] that a particular type of single-resource allocation problem (known as the *classic multi-armed bandit problem*, described above, with geometrically discounted rewards, and an infinite horizon) can be optimally solved by decomposing it into a collection of single-project parametric optimal-stopping problems, which in turn can be solved by assigning certain dynamic (state-dependent) quantities, now called the Gittins index values. These indices define an optimal policy, the *Gittins index rule*, which prescribes to allocate the resource at every period to the project with currently highest index value. This classic problem in Bayesian setting, under finite horizon, and with non-geometric discounting was thoroughly studied in Berry and Fristedt [6]; see also Russo and van Roy [33], Kaufmann [27].

Index rules and their generalisations became an important concept in addressing sequential resource allocation problems, which are PSPACE-hard in general [32], for their simplicity to implement, economic interpretation, and asymptotic (or near-) optimality. See, e.g. Whittle [40], Gittins [14], Weber and Weiss [39], Niño-Mora [30], Glazebrook et al. [20], Archibald et al. [2], Glazebrook et al. [21], Hauser et al. [23], Jacko [25], Ayesta et al. [4], Gittins et al. [15], Ayesta et al. [5], Villar et al. [36, 37], Verloop [35], Larrañaga et al. [28] for a palette of models restricted to $W^a_{k,n} := a$ for all k, n, a. The key observation allowing to derive index rules in such more general models was made by Whittle [40], where he proposed a more general *Whittle index* definition. Subsequent work established asymptotic optimality, index characterisation, algorithms, and performance evaluation of the Whittle index rule. Further index generalisations for problems with unrestricted resource requirements $W^a_{k,n}$ were introduced in Glazebrook and Minty [17], Glazebrook et al. [18], Jacko [26], Graczová and Jacko [22], Glazebrook et al. [19], Hodge and Glazebrook [24]. It is important to note that the decomposition technique leading to index rules is not applicable when the projects are not mutually independent.

Reinforcement learning provides an alternative approach to tackling dynamic problems under uncertainty. It has gained popularity due to not requiring an observational model. Reinforcement learning focusses on solving a problem by starting from very limited or no assumptions with the intention to learn efficiently about the nature of the problem by interacting with a system to be controlled. Information may be limited either because of decision-maker's lack of historical knowledge of observations or conscious lack of model assumptions. Various methods and algorithms are available to improve decisions over time through interaction with the system, e.g., temporal-difference methods, Monte Carlo simulation, gradient-based methods, randomised greedy algorithms, etc. Some reinforcement learning methods work with an ETA objective while others work with ECR objective.

One of the earliest learning approaches is *Q-learning* [38, 34], which has become very popular as a powerful tool useful in a variety of real-life problems. Q-learning is model-free and is suited for learning an optimal policy in general MDPs with the ETA objective where the states, rewards and transition probabilities are unknown. Q-learning estimates the optimal solution by maintaining estimates of the conditional expected total reward given the current history and assuming that (what is currently estimated as) optimal actions are chosen in all future periods, known as Q-values (Q stands for 'Quantity' or more recently 'Quality'). Improvements and extensions of Q-learning are currently a very active research area, with recent advances such as deep Q-learning, which was successfully implemented in software to play games at superhuman level, including AlphaGo where a computer beat the world's best players of Go. Other techniques for learning general MDPs have been developed as well, e.g. Ortner et al. [31] proposed a general algorithm for learning certain structured MDPs, while Burnetas and Katehakis [8] developed a technique to learn an optimal policy for general MDPs where the rewards and transition probabilities are unknown.

In this context, the ECR minimization problem has been heavily studied in the literature in the i.i.d. version of the classic multi-armed bandit problem, i.e. with single resource ($M = 1$), independent and single-state projects ($\mathcal{N}_k = \{*\}, W^a_{k,*} := a, p^a_{k,*,*} := 1$), zero-reward unattended projects ($R^0_{k,*} = 0$), unknown expected one-period rewards $R^1_{k,*}$, and where observed rewards are only samples from distributions with these unknown means $R^1_{k,*}$. For the non-parametric problem setting see e.g. Burnetas and Katehakis [7], Cowan and Katehakis [10]; for parametric Gaussian problem setting see e.g. Cowan et al. [9] and the references therein. Note that problem (1) is trivial in this i.i.d. case.

Combining optimisation and learning. Although the two fields have common roots, they have evolved into separate theoretical research fields with an extremely limited interaction. For problems of dynamic resource allocation, there are only a few works, and typically only very special cases. The Q-learning method for the Gittins index was proposed in Duff [11]. Near-optimal regret guarantees for the finite-horizon Gittins index rule in the Gaussian problem setting are proven in Lattimore [29]. The ECR objective was studied and a learning algorithm was proposed for the setting of Whittle [40] in Ortner et al. [31].

While it is immediate that a policy for optimizing ECR is suboptimal if implemented in a fully observable setting, (1), methods for learning the policy π^{ETA} in the limited or non-observable setting are not well understood yet. For general MDPs, the Q-learning algorithm maintains $Q(n, a)$ for each state-action combination of the MDP, which is an estimate of the expected time-average (or total discounted) reward if action a is chosen in state n in the current period and the optimal policy is followed thereafter. For the problems of dynamic resource allocation, direct implementation of Q-learning would suffer from the curse of dimensionality in the same way as standard stochastic optimisation approaches.

1.3 Learning the Whittle index rule

Most related to our paper are works that aim at using Q-learning for learning of index rules rather than of general optimal policies for bandit problems. Specifically, we look at techniques that estimate the Whittle index policy, whereby unit values of work are assigned to the M projects with the highest estimated Whittle indices. The usual relaxation approach [40] results in the problem decomposition allowing to consider each project independently with a wage $\lambda \in \mathbb{R}$ for assigning every unit of work, thus resulting in a *profit* objective calculated as the reward less the work wage paid. Then, the quality of an action in a particular state of the kth project satisfies the dynamic programming equation

$$\beta_k(\gamma, \lambda)+Q_k(\gamma, \lambda, n, a) = R^a_{k,n} - \lambda W^a_{k,n} + \gamma \sum_{m \in \mathcal{N}_k} p^a_{k,n,m} \max_{a' \in \mathcal{A}} Q_k(\gamma, \lambda, m, a'), \quad (2)$$

where γ is the discount factor and $Q_k(\gamma, \lambda, n, a)$ is the profit quantity relative to $\beta_k(\gamma, \lambda)$ which is the optimal expected time-average profit. Under regularity conditions, the system of equations (2) for fixed γ, λ and k implies that the solution $Q_k(\gamma, \lambda, n, a)$ is unique up to an additive constant [1]. Note that when $\gamma < 1$, $\beta_k(\gamma, \lambda) = 0$ and $Q_k(\gamma, \lambda, n, a)$ is the expected total discounted quantity, while when $\gamma = 1$, $Q_k(\gamma, \lambda, n, a)$ is the expected relative time-average quantity.

The policy for control on an individual project maps the set of states available to that project to either a passive (0) or active (1) action $\phi : \mathcal{N}_k \to \mathcal{A}$. An optimal policy (for an individual project) is one such mapping that chooses the action which maximises $Q_k(\gamma, \lambda, n, a)$ for all $n \in \mathcal{N}_k$. Optimal policies depend on λ. For example, choosing a sufficiently high wage will ensure that the passive policy, whereby the passive action is always chosen, becomes optimal. Conversely, choosing a sufficiently negative wage will ensure that the active policy, whereby the active action is always chosen, becomes optimal. In each extreme there exists bounds where the wage dominates the rewards and the optimal policy is fixed as either passive or active for all λ outside the bounds. Between these bounds the optimal policy can change. If the optimal policy transitions at specific values of λ, such that the optimal policy at each state transitions exactly once, then the project is indexable and wage values at each point of transition are the Whittle index values for their corresponding state. Under this formulation, the Whittle index for an indexable project k in state n ($\lambda_{k,n}$) can be defined as the wage on the active action required to make both actions equally desirable. Namely,

$$Q_k(\gamma, \lambda_{k,n}, n, 1) - Q_k(\gamma, \lambda_{k,n}, n, 0) = 0. \tag{3}$$

Although indexability (meaning that the project instance satisfies the law of diminishing marginal returns) in theory restricts the applicability of index rules, in practice it is rare to have non-indexable problems [30].

Learning the Whittle index values involves learning both $Q_k(\gamma, \lambda, n, a)$ and $\lambda_{k,n}$ such that (3) is satisfied. Fu et al. [12] accomplish this by storing separate tables of estimated Q values across a fixed search grid of λ values. All Q values are updated each time step using standard Q-learning updates based on a variant of equation (2). The Whittle indices are estimated to be the λ grid values that minimise $|Q_k(\gamma, \lambda, n, 1) - Q_k(\gamma, \lambda, n, 0)|$ in each state. A key issue with this approach is that the true Whittle index values are generally excluded from the fixed λ search grid, and the computational requirements scale with the size of the search space.

Avrachenkov and Borkar [3] resolve this issue by using a dynamic search space of $\lambda_{k,n}$ values, which represent estimates of the Whittle index of each state. At each time step after updating the Q values, the Whittle index estimates are also updated at a slower rate, such that the current estimates of $Q_k(\gamma, \lambda_{k,n}, n, 1) - Q_k(\gamma, \lambda_{k,n}, n, 0)$ move closer to 0. In this way, estimated Whittle index values slowly converge to the true Whittle index values (under certain conditions) without increasing the search space size.

We present an alternative approach which is fundamentally different from the approaches in Fu et al. [12] and Avrachenkov and Borkar [3]. These approaches aim at iteratively learning the index values, which results in continuous modification of the index value estimates. On the contrary, our approach aims at iteratively learning the index-induced policies.

We develop a method for problems of dynamic resource allocation built on an understanding of both stochastic optimisation (specifically, the Whittle index) and reinforcement learning (specifically, Q-learning). We propose a *novel implementation of Q-learning*, which exploits the structure of the problem considered, in which the algorithm maintains two sets of Q-values for each project: one for reward and one for work.

2 A New Q-Learning Method for the Whittle Index

We now present a new scheme for Q-learning the Whittle index rule that exploits the policy structure of indexable bandits to remove λ from the Q-learning updates, and significantly constrain the policy space. Our approach can be interpreted as iteratively learning the policy induced by the Whittle index values, instead of iteratively learning the Whittle index values directly which was focus of the previous literature. We believe that in certain cases, this can reduce the error introduced by incorrect estimates of the Whittle index values, thereby improving the convergence rate.

In the rest of the paper, we consider the case $\gamma = 1$, focusing on the time-average criterion, and drop the dependency on γ to simplify the notation. However, an analogous approach for the discounted case can also be formulated. Consider a policy evaluation version of the Bellman equation (2),

$$Q_k^{\phi}(\lambda, n, a) = R_{k,n}^a - \lambda W_{k,n}^a + \sum_{m \in \mathcal{N}_k} p_{k,n,m}^a Q_k^{\phi}(\lambda, m, \phi(m)) - \overline{Q}_k^{\phi}(\lambda). \tag{4}$$

There are three key differences between (2) and (4). Firstly, this is a "learning" version of (2), in which the Q values should be interpreted as estimates of the true Q values that satisfy (2). Secondly, the Q values are computed under policy ϕ, which, unlike in (2), might not be optimal. Thirdly, following Avrachenkov and Borkar [3], the $\overline{Q}_k^{\phi}(\lambda)$ term, where

$$\overline{Q}_k^{\phi}(\lambda) = \frac{1}{2|\mathcal{N}_k|} \sum_{m \in \mathcal{N}_k} \left[Q_k^{\phi}(\lambda, m, 1) + Q_k^{\phi}(\lambda, m, 0) \right], \tag{5}$$

is the time-average relative profit quantity across all states and actions for the given policy ϕ and wage λ, is included instead of $\beta_k(\lambda)$ to ensure that (4) has a unique solution in the case $\gamma = 1$ [1]. A key observation is that for a fixed policy ϕ, (4) forms a linear system. Therefore, the profit Q-value $Q_k^{\phi}(\lambda, n, a)$ depends linearly on λ and can be decomposed into two parts

$$Q_k^{\phi}(\lambda, n, a) = Q_k^{\phi}(n, a) - \lambda H_k^{\phi}(n, a), \tag{6}$$

where $Q_k^{\phi}(n, a) := Q_k^{\phi}(0, n, a)$ represents the profit Q-value when wage $\lambda = 0$, i.e. can be interpreted as the reward Q-value, and $H_k^{\phi}(n, a)$ can be interpreted as the work Q-value, both under policy ϕ. Policy evaluation equations for both $Q_k^{\phi}(n, a)$ and $H_k^{\phi}(n, a)$ can be derived by substituting (6) into (4) and (5) and applying separation of variables

$$Q_k^{\phi}(n, a) = R_{k,n}^a + \sum_{m \in \mathcal{N}_k} p_{k,n,m}^a Q_k^{\phi}(m, \phi(m)) - \overline{Q}_k^{\phi}, \tag{7}$$

$$H_k^{\phi}(n, a) = W_{k,n}^a + \sum_{m \in \mathcal{N}_k} p_{k,n,m}^a H_k^{\phi}(m, \phi(m)) - \overline{H}_k^{\phi}, \tag{8}$$

$$\overline{Q}_k^{\phi} = \frac{1}{2|\mathcal{N}_k|} \sum_{m \in \mathcal{N}_k} \left[Q_k^{\phi}(m, 1) + Q_k^{\phi}(m, 0) \right], \tag{9}$$

$$\overline{H}_k^{\phi} = \frac{1}{2|\mathcal{N}_k|} \sum_{m \in \mathcal{N}_k} \left[H_k^{\phi}(m, 1) + H_k^{\phi}(m, 0) \right]. \tag{10}$$

With this representation, direct dependence on λ is eliminated, and replaced by dependence on ϕ. Solving these new policy evaluation equations and combing the results with (6), yields an evaluation of the policy for all λ, even when the policy is not optimal.

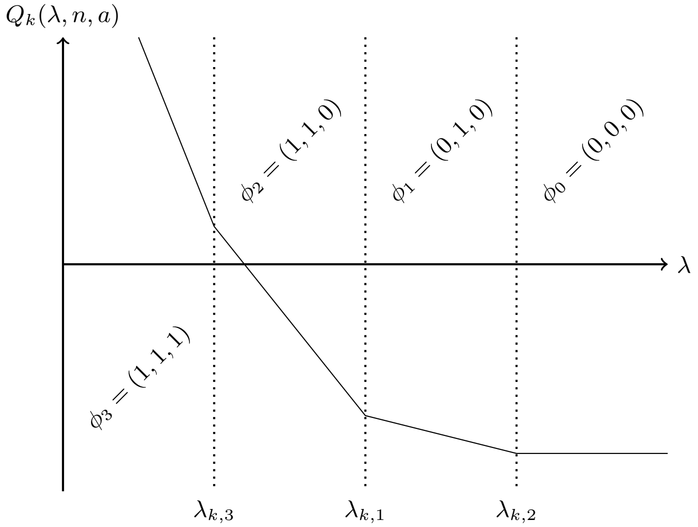

Fig. 1: A 3-state indexable bandit exemplifying the structure of $Q_k(\lambda, n, a)$. $Q_k(\lambda, n, a)$ is a piecewise linear function of λ with corners occurring at the Whittle indices. The optimal policies are independent of λ within each interval of each line segment, but change at the corners where the policy transitions and λ equals a Whittle index of the state that transitions. For example, the optimal policy changes between $(1, 1, 1)$ and $(1, 1, 0)$ on the left, so the Whittle index of the third state equals the wage at the interception point.

If the policy ϕ is optimal, then the Whittle index value for each project k in state n can be computed by combining (3) and (6)

$$\lambda_{k,n} = \frac{Q_k^\phi(n, 1) - Q_k^\phi(n, 0)}{H_k^\phi(n, 1) - H_k^\phi(n, 0)}. \tag{11}$$

Notice that during the learning process (in which policy ϕ may not be optimal), the right-hand side of (11) can be interpreted as the current estimate of the marginal productivity rate, which is the same interpretation as the Whittle index has [30]. One can therefore expect that (11) converges to the Whittle index value when ϕ is optimal, and thus might be a pragmatic choice for learning the Whittle index values and employing such an index rule in the multi-project problem. This is despite the fact that the Whittle index exists only under indexability conditions [30], and the Whittle index rule is asymptotically optimal only under certain technical assumptions [39, 35].

2.1 Algorithmic Learning Approach

Noting that $Q_k^\phi(\lambda, n, a)$ is a linear function of λ, and in cases where the space of possible policies ϕ is finite, the optimal quantity, $Q_k(\lambda, n, a)$, must be a continuous piecewise linear function of λ. If the project is indexable with a unique Whittle index for each state, then the optimal policy is fixed within each piece. Furthermore, the set of states that the optimal policies activate in indexable projects decreases monotonically with λ, so the set of optimal policies can be represented by an ordering of the states. Figure 1 illustrates an example of this structure.

Algorithm 1: Learning Whittle index control policy via policy iteration

```
for $k \in \mathcal{K}$ do
    for $l \in \{0, 1 \ldots, |\mathcal{N}_k|\}$ do
        Initialise $Q_k^l$ and $H_k^l$ tables;
    end
    Initialise optimal policy and Whittle index estimates;
end
for $t \in \{1, 2, \ldots, T\}$ do
    Randomly decide to explore based on an exploration schedule;
    if Exploring then
        Activate $M$ projects randomly with equal probability;
    else
        Activate $M$ projects with highest estimated Whittle indices (breaking
          ties randomly);
    end
    for $k \in \mathcal{K}$ do
        for $l \in \{0, 1 \ldots, |\mathcal{N}_k|\}$ do
            Update $Q_k^l$ and $H_k^l$ for observed project transition and reward via
              equations (12) and (13);
        end
        Update policy and Whittle index estimates via algorithm 2;
    end
end
```

In practice, our method stores $|\mathcal{N}_k|+1$ tables for Q and H and for each project type. These correspond to the optimal policies that exist between the Whittle indices of an indexable project which occur at the policy transitions where λ is a Whittle index. We denote these policies as ϕ_l and the Q and H values as $Q_k^l(n,a) := Q_k^{\phi_l}(n,a)$ and $H_k^l(n,a) := H_k^{\phi_l}(n,a)$, where $l \in \{0, 1, \ldots, |\mathcal{N}_k|\}$ indicates the size of the set of states that the policy activates.

Our method of learning works by iteratively estimating the optimal policies ϕ_l from the $Q_k^l(n,a)$ and $H_k^l(n,a)$ estimates, and updating these estimates using the following Q updates for all l and for all observed transitions and rewards at each time step t

$$Q^{\phi}_{k,t+1}(n,a) = Q^{\phi}_{k,t}(n,a) + \alpha\left(R^{a}_{k,n} + Q^{\phi}_{k,t}(s',\phi(s')) - \overline{Q}^{\phi}_{k,t} - Q^{\phi}_{k,t}(n,a)\right), \tag{12}$$

$$H^{\phi}_{k,t+1}(n,a) = H^{\phi}_{k,t}(n,a) + \alpha\left(W^{a}_{k,n} + H^{\phi}_{k,t}(s',\phi(s')) - \overline{H}^{\phi}_{k,t} - H^{\phi}_{k,t}(n,a)\right), \tag{13}$$

where,

$$\overline{Q}^{\phi}_{k,t} = \frac{1}{2|\mathcal{N}_k|}\sum_{m\in\mathcal{N}_k}\left[Q^{\phi}_{k,t}(m,1) + Q^{\phi}_{k,t}(m,0)\right], \tag{14}$$

$$\overline{H}^{\phi}_{k,t} = \frac{1}{2|\mathcal{N}_k|}\sum_{m\in\mathcal{N}_k}\left[H^{\phi}_{k,t}(m,1) + H^{\phi}_{k,t}(m,0)\right], \tag{15}$$

where $\alpha \in (0,1)$ is a Q-learning learning rate. Algorithm 1 outlines these steps in more detail.

The policies ϕ_l, and Whittle index values are estimated using algorithm 2. It works by beginning at $l = 0$ where ϕ_0 is the passive policy and is known to be optimal for all λ above the maximum Whittle index, and then choosing the state which maximises the marginal productivity rate computed by equation (11). The Whittle index of this state is estimated to be the marginal productivity rate and the next policy at $l = 1$ is estimated to be the same as the previous with the exception of this state. This process is repeated where the marginal productivity rate is maximised from the set of states available, and the subsequent optimal policy estimate ϕ_l follows

$$\phi_{l+1}(n) = \begin{cases} 1 - \phi_l(n), & n = n', \\ \phi_l(n), & \text{otherwise.} \end{cases} \tag{16}$$

Here n' is the available state that maximises the marginal productivity rate. Notice that estimating the optimal policies in this way constrains the space of available policy options to those of indexable projects. In doing so, we anticipate convergence to the true optimal policies could be accelerated.

The algorithm could be equivalently formulated in the reverse direction starting from $l = |\mathcal{N}_k|$ where the optimal policy $\phi_{|\mathcal{N}_k|}$ is known to be the active policy, and then choosing states that minimise the marginal productivity rate. Actually, each step of the algorithm can proceed from either the “top” or the “bottom”, and we alternate between these in practice.

3 Illustrative Numerical Comparison

A comprehensive empirical comparison between control algorithms would involve controlling a broad range of restless multiarmed bandits of different sizes and types, and comparing a range of performance metrics. However, in this paper, we begin with a single test case to demonstrate that the algorithm can work in practice.

Algorithm 2: Whittle indices and policies from Q and H estimates

```
input  : $Q^l_k(n,a), H^l_k(n,a) \quad \forall l \in \{0,1,\dots,|\mathcal{N}_k|\}, \forall n \in \mathcal{N}_k, \forall a \in \mathcal{A}$
output: Estimated Whittle index ordering and values
$\Lambda^l_k(n) \leftarrow \frac{Q^l_k(n,1)-Q^l_k(n,0)}{H^l_k(n,1)-H^l_k(n,0)}$;
$(l_{\text{top}},\ l_{\text{bottom}}) \leftarrow (0,\ |\mathcal{N}_k|)$;
$(\lambda_{\text{top}},\ \lambda_{\text{bottom}}) \leftarrow (+\infty,\ -\infty)$;
remaining $\leftarrow \{1,\dots,|\mathcal{N}_k|\}$;
forward order $\leftarrow$ empty tuple;
backward order $\leftarrow$ empty tuple;
while $|remaining| > 0$ do
    decide to iterate from the top or bottom remaining index;
    if iterate from top then
        $n' \leftarrow \arg\max_{n \in \text{remaining}} \Lambda^{l_{\text{top}}}_k(n)$;
        // clamp index between previous values
        $\lambda_{\text{top}} \leftarrow \max\left\{\lambda_{\text{bottom}}, \min\left\{\Lambda^{l_{\text{top}}}_k(n'), \lambda_{\text{top}}\right\}\right\}$;
        append $n'$ to end of forward order;
        $\lambda_{k,n'} \leftarrow \lambda_{\text{top}}$;
        $l_{\text{top}} \leftarrow l_{\text{top}} + 1$;
        remaining $\leftarrow$ remaining $\setminus\{n'\}$;
    else
        $n' \leftarrow \arg\min_{n \in \text{remaining}} \Lambda^{l_{\text{bottom}}}_k(n)$;
        // clamp index between previous values
        $\lambda_{\text{bottom}} \leftarrow \min\left\{\lambda_{\text{top}}, \max\left\{\Lambda^{l_{\text{bottom}}}_k(n'), \lambda_{\text{bottom}}\right\}\right\}$;
        append $n'$ to beginning of backward order;
        $\lambda_{k,n'} \leftarrow \lambda_{\text{bottom}}$;
        $l_{\text{bottom}} \leftarrow l_{\text{bottom}} - 1$;
        remaining $\leftarrow$ remaining $\setminus\{n'\}$;
    end
end
ordering $\leftarrow$ forward order concatenated with backward order;
```

3.1 A Birth and Death Test Case

Our restless multi-armed bandit is adapted from the Mentoring Instructions example presented by Fu *et al.* [12] and consists of $K = 100$ students (projects) and $M = 10$ mentors which can each mentor a student at each time step. Each student's state sits in one of 5 ordered study levels $\mathcal{N}_k = \{1,2,3,4,5\}$, and can increase or decrease randomly each time step according to the following birth and death transition probabilities

$$\boldsymbol{P}_k^0 = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 \\ 0.7 & 0 & 0.3 & 0 & 0 \\ 0 & 0.7 & 0 & 0.3 & 0 \\ 0 & 0 & 0.7 & 0 & 0.3 \\ 0 & 0 & 0 & 0.7 & 0.3 \end{bmatrix}, \qquad \boldsymbol{P}_k^1 = \begin{bmatrix} 0.3 & 0.7 & 0 & 0 & 0 \\ 0.3 & 0 & 0.7 & 0 & 0 \\ 0 & 0.3 & 0 & 0.7 & 0 \\ 0 & 0 & 0.3 & 0 & 0.7 \\ 0 & 0 & 0 & 0.3 & 0.7 \end{bmatrix}, \tag{17}$$

where the active action corresponds to the student receiving mentorship. Students in higher study levels give higher rewards each time step following $R_{k,n}^a = \sqrt{\frac{n}{5}}$. With these transition probablilities and expected rewards the Whittle indices can be computed to be

$$(\lambda_{k,n})_{n\in\mathcal{N}_k} = (0.3166\ldots, 0.5032\ldots, 0.5510\ldots, 0.5512\ldots, 0.1037\ldots). \tag{18}$$

The Whittle index policy for controlling this system is to prioritise allocating mentors to students in the 4th study level, then 3rd, 2nd, 1st and then leaving students in the 5th study level with the lowest priority.

We compare our method to Avrachenkov and Borkar's method (denoted AB) [3] for controlling this system. In each case we use a fixed exploration rate of 10%. This means at each time step there is a 10% probability of exploring, where all mentors are allocated randomly to the students with equal probability. As outlined in algorithm 1, when not exploring the controller assigns the mentors to the M students with highest estimated Whittle indices. Additionally we fix the learning rate in both learning algorithms to be the same at $\alpha = 0.01$, and the Whittle index learning rate in AB to 0.001. Note that the AB method requires a learning rate parameter for the Whittle index which is significantly smaller than the Q learning rate.

We ran 32 simulations for each controller for 10,000 time steps each using these parameters. The initial state of each simulation was initialised by first controlling with the Whittle index policy, including exploration, for at least 5000 time steps.

3.2 Test Case Results

We first compare how the two methods learn the Whittle index values. Figure 2 shows a typical example of how these controllers estimate the index values at each time step. There are two key differences in how the methods estimate them. Firstly, the AB method estimates of the Whittle index values vary much more smoothly than ours. However, our method appears to converge much faster before fluctuating about the correct values. The size of these fluctuations depends on the learning rate, and we anticipate that employing a learning rate schedule that tends towards zero rather than a fixed rate would allow the noise to also tend towards zero.

Despite the fluctuations, our method learns the Whittle index policy earlier than AB and consistently employs it at a higher rate. The left part of figure

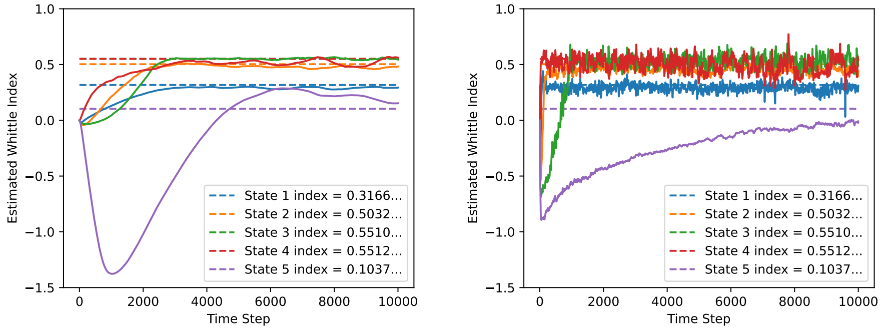

Fig. 2: Estimated Whittle index values when using AB (left) and our controller (right) for a single simulation each. Solid lines represent the estimated indices for each state at each time step and dashed lines represent the true Whittle index values. Note that the third (green) and fourth (red) Whittle index values are very close and appear as a single red dashed line at the top. AB estimates the indices more smoothly but also learns them more slowly. Our method is more noisy but could be smoothed by using a learning rate schedule that decays towards zero.

3 plots the proportion of the 32 simulations in which the controller has learnt the correct Whittle index policy. Our method fluctuates just below 50%, about double that of AB. The reason it does not converge to 100% is again due to the fixed learning rate, and also the very small difference between state 3 and state 4 Whittle indices. Learning the Whittle index policy faster and employing it more frequently is probably the reason our method achieved lower cumulative regret on average, plotted on the right part of figure 3.

Examining the rewards and time average rewards, shown in figure 4, shows that both methods appear to eventually produce rewards consistent with the Whittle index policy with exploration, and both outperform a random policy. However, our method again outperforms AB except around $t = 150$ and after around $t > 3000$ where both methods produce similar rewards.

4 Conclusion

We have developed a method controlling problems of dynamic resource allocation and presented a novel implementation of Q-learning index rules. Unlike previous Q-learning schemes which include estimates of the indices in the Q updates, our method updates based on policy estimates which are constrained to those available in indexable systems. In a single test system our method outperforms previous methods in learning the Whittle index policy as well as maximising expected rewards. These results warrant a broader empirical study that compares

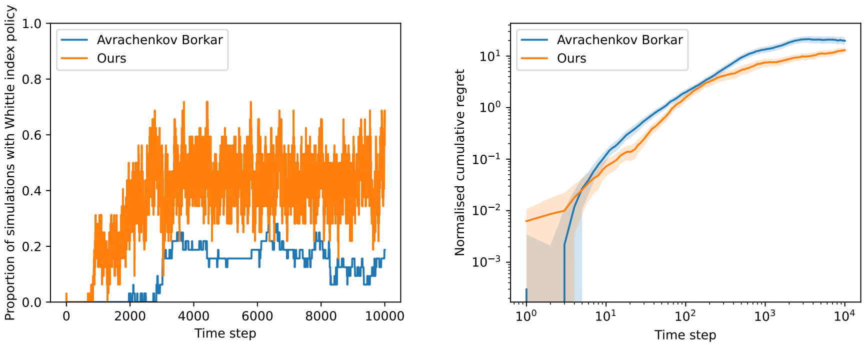

Fig. 3: Proportion of simulations that learnt the correct Whittle index state ordering (left) and normalised cumulative regret (right) averaged across 32 simulations. The cumulative regret is normalised so that the expected reward from the Whittle index policy with exploration is 1. Shaded regions represent standard errors about the sample mean. The cumulative regret when using our method is consistently lower than when using AB, probably because our method learns the Whittle index policy sooner and employs it more frequently.

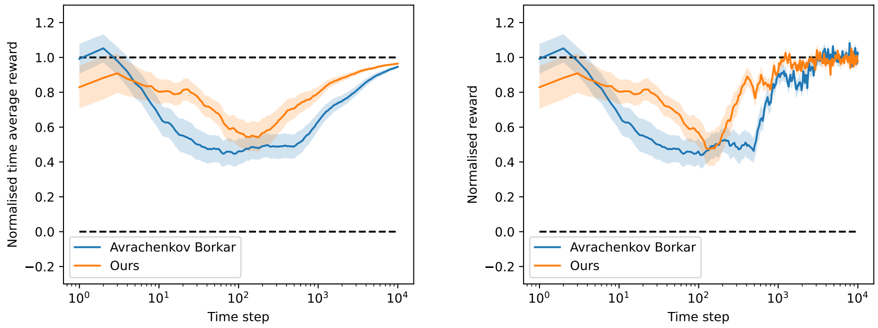

Fig. 4: Normalised time average reward (left) and normalised rewards (right) averaged across 32 simulations. Normalised so that 1 is the expected reward when following the Whittle index policy with the same exploration rate, and 0 is the expected reward following a random policy. The right is smoothed using a moving average of 100 time steps. Shaded regions represent standard errors about the sample mean.

methods across a broader range of project types and control parameters, such as learning rate schedules.

Acknowledgements L. J. Gibson's and Y. Nazarathy's research is supported by ARC grant DP180101602. L. J. Gibson is also supported by the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS).

References

[1] Abounadi, J., Bertsekas, D., and Borkar, V. S. (2001). Learning algorithms for Markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–18.

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

[3] Avrachenkov, K. E. and Borkar, V. S. (2021). Whittle index based Q-learning for restless bandits with average reward. arXiv:2004.14427.

[4] Ayesta, U., Erausquin, M., and Jacko, P. (2010). A modeling framework for optimizing the flow-level scheduling with time-varying channels. *Performance Evaluation*, 67:1014–1029.

[5] Ayesta, U., Jacko, P., and Novak, V. (2017). Scheduling of multi-class multi-server queueing systems with abandonments. *Journal of Scheduling*, 20(2):129–145.

[6] Berry, D. A. and Fristedt, B. (1985). *Bandit Problems: Sequential Allocation of Experiments*. Springer Netherlands.

[7] Burnetas, A. N. and Katehakis, M. N. (1996). Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17:122–142.

[8] Burnetas, A. N. and Katehakis, M. N. (1997). Optimal adaptive policies for Markov decision processes. *Mathematics of Operations Research*, 22(1):222–255.

[9] Cowan, W., Honda, J., and Katehakis, M. N. (2017). Normal bandits of unknown means and variances. *J. Mach. Learn. Res.*, 18:154–1.

[10] Cowan, W. and Katehakis, M. N. (2015). Minimal-exploration allocation policies: Asymptotic, almost sure, arbitrarily slow growing regret. arXiv:1505.02865v2.

[11] Duff, M. O. (1995). Q-Learning for Bandit Problems. In *Proceedings of the Twelfth International Conference on Machine Learning*. 32 pages. CMPSCI Technical Report 95-26.

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

[13] Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B*, 41(2):148–177.

[14] Gittins, J. C. (1989). *Multi-Armed Bandit Allocation Indices*. J. Wiley & Sons, New York.

[15] Gittins, J. C., Glazebrook, K., and Weber, R. (2011). *Multi-Armed Bandit Allocation Indices*. Wiley-Blackwell.

[16] Gittins, J. C. and Jones, D. M. (1974). A dynamic allocation index for the sequential design of experiments. In Gani, J., editor, *Progress in Statistics*, pages 241–266. North-Holland, Amsterdam.

[17] Glazebrook, K. and Minty, R. J. (2009). A generalized Gittins index for a class of multiarmed bandits with general resource requirements. *Mathematics of Operations Research*, 34(1):26–44.

[18] Glazebrook, K. D., Hodge, D. J., and Kirkbride, C. (2011). General notions of indexability for queueing control and asset management. *Annals of Applied Probability*, 21:876–907.

[19] Glazebrook, K. D., Hodge, D. J., Kirkbride, C., and Minty, R. J. (2014). Stochastic scheduling: A short history of index policies and new approaches to index generation for dynamic resource allocation. *Journal of Scheduling*, 17(5):407–425.

[20] Glazebrook, K. D., Kirkbride, C., Mitchell, H. M., Gaver, D. P., and Jacobs, P. A. (2007). Index policies for shooting problems. *Operations Research*, 55(4):769–781.

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

[22] Graczová, D. and Jacko, P. (2014). Generalized restless bandits and the knapsack problem for perishable inventories. *INFORMS Operations Research*, 62(3):696–711.

[23] Hauser, J. R., Urban, G. L., Liberali, G., and Braun, M. (2009). Website morphing. *Marketing Science*, 28(2):202–223.

[24] Hodge, D. J. and Glazebrook, K. D. (2015). On the asymptotic optimality of greedy index heuristics for multi-action restless bandits. *Advances in Applied Probability*, 47(3):652–667.

[25] Jacko, P. (2010). *Dynamic Priority Allocation in Restless Bandit Models*. Lambert Academic Publishing.

[26] Jacko, P. (2016). Resource capacity allocation to stochastic dynamic competitors: Knapsack problem for perishable items and index-knapsack heuristic. *Annals of Operations Research*, 241(1):83–107.

[27] Kaufmann, E. (2018). On Bayesian index policies for sequential resource allocation. *The Annals of Statistics*, 46(2):842 – 865.

[28] Larrañaga, M., Ayesta, U., and Verloop, I. M. (2016). Dynamic control of birth-and-death restless bandits: Application to resource-allocation problems. *IEEE/ACM Transactions on Networking*, 24(6):3812–3825.

[29] Lattimore, T. (2016). Regret analysis of the finite-horizon Gittins index strategy for multi-armed bandits. *Journal of Machine Learning Research*, 49:1–32.

[30] Niño-Mora, J. (2007). Dynamic priority allocation via restless bandit marginal productivity indices. *TOP*, 15(2):161–198.

[31] Ortner, R., Ryabko, D., Auer, P., and Munos, R. (2014). Regret bounds for restless Markov bandits. *Theoretical Computer Science*, 558:62–76.

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

[33] Russo, D. and van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

[34] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA.

[35] Verloop, I. M. (2016). Asymptotically optimal priority policies for indexable and nonindexable restless bandits. *The Annals of Applied Probability*, 26(4):1947–1995.

[36] Villar, S. S., Bowden, J., and Wason, J. (2015a). Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statistical Science*, 30(2):199–215.

[37] Villar, S. S., Wason, J., and Bowden, J. (2015b). Response-adaptive randomization for multi-arm clinical trials using the forward looking Gittins index rule. *Biometrics*, 71:969–978.

[38] Watkins, C. J. C. H. (1989). *Learning From Delayed Rewards.* PhD thesis, Cambridge University, UK.

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

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