Computing an Index Policy for Multiarmed Bandits with Deadlines

José Niño-Mora
Department of Statistics
Universidad Carlos III de Madrid
Avda. Universidad 30
28911 Leganés (Madrid), Spain
jnimora@alum.mit.edu

ABSTRACT

This paper introduces the multiarmed bandit problem with deadlines, which concerns the dynamic selection of a live project to engage out of a portfolio of Markovian bandit projects expiring after given deadlines, to maximize the expected total discounted or undiscounted reward earned. Although the problem is computationally intractable, a natural heuristic policy is obtained by attaching to each project the finite-horizon counterpart of its Gittins index, and then engaging at each time a live project of highest index. Remarkably, while such a finite-horizon index was introduced in [R. N. Bradt, S. M. Johnson, and S. Karlin (1956). On sequential designs to maximize the sum of n observations. *Ann. Math. Statist.* 27 1060–1074], an exact polynomial-time algorithm using arithmetic operations does not seem to have been proposed until [J. Niño-Mora (2005). A marginal productivity index policy for the finite-horizon multiarmed bandit problem. In *Proceedings of CDC-ECC '05*, pp. 1718–1722, IEEE]. Yet, such an adaptive-greedy index algorithm, which draws on methods introduced by the author for restless bandit indexation, has a complexity of $O(T^3n^3)$ operations for a T-horizon n-state project, rendering it impractical for all but small instances. This paper significantly improves on the complexity of such an algorithm, decoupling it into a recursive T-stage method that performs $O(T^2n^3)$ arithmetic operations. Moreover, in an insightful special model the complexity is further reduced to $O(T^2)$ operations, and closed-form index formulae are given. Computational experiments are reported demonstrating the algorithm's runtime performance, and showing that the proposed index policy is near optimal and can substantially outperform the benchmark greedy and Gittins index policies.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Markov processes;
F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms, Performance, Design, Theory

Keywords

stochastic scheduling, deadlines, multiarmed bandits, index policies, restless bandits, marginal productivity index, adaptive-greedy algorithm

SMCTools 2008 October 20, 2008, Athens, GREECE

1. INTRODUCTION

1.1 Motivation and Background

The reader will surely have extensive personal experience on the problem addressed in this paper, as we all must figure out how to dynamically set priorities among multiple randomly-evolving projects subject to hard deadlines vying for our attention. Thus, students prioritize work on multiple assignments, exams, and exam questions. Researchers prioritize proposals and conference papers to prepare for submission before deadlines run out. Managers are continuously prioritizing multifarious time sensitive tasks or projects. Think, e.g., of selecting new-product development projects to pursue before patents expire, or, in a seasonal industry, before the season ends or starts.

People set such priorities guided by rough, intuitive rules of thumb. Thus, most people would agree that the priority of a project should increase with its "importance", and decrease with its "difficulty." However, people differ markedly into how they factor in the issue of "urgency." While procrastinators increase a project's priority as its deadline looms nearer, nonprocrastinators do the opposite. Although procrastinating is a widespread behavior, we are all taught that we should not procrastinate. Yet, such advice does not appear to have been backed up in previous work by analytical results or computational evidence.

We will address such issues in the setting of a general stochastic model where one aims to extract the maximum expected gain out of a portfolio of M time-sensitive projects labeled by $m = 1, \ldots, M$, of which at most one can be engaged at a time. Project m is modeled as a discrete-time *bandit* moving on the finite state space $\mathbb{X}_m$. If engaged before deadline T_m when it occupies state i_m, the project yields an active reward $R^1_m(i_m) = R_m(i_m)$, incurs a *continuation charge* ν, and changes state to j_m with probability $p_m(i_m, j_m)$. Otherwise, it neither yields reward (i.e., the

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

passive reward is $R_m^0(i_m) \equiv 0$) nor incurs charges, and its state remains frozen. Rewards are discounted over time with factor $0 < \beta \leq 1$, where we abuse the term "discounted" to include also the case $\beta = 1$.

We will further specialize the results to an insightful yet deceptively simple model, where the state

$$i_m \in \mathbb{X}_m \triangleq \{0, 1, \ldots, n_m\}$$

of project m is the number of remaining stages that need to be completed. Engaging the project in state $i_m \geq 1$ completes the current stage with probability p_m, so $p_m(i_m, i_m - 1) = p_m$ and $p_m(i_m, i_m) = 1 - p_m$. State 0 is absorbing. A final reward R_m is received when and if the project is completed before its deadline T_m runs out, so $R_m(i_m) = 0$ for $i_m \neq 1$, and $R_m(1) = R_m p_m$.

Decisions as to which *live* project (one whose deadline has not yet expired) to engage, if any, at each time are based on adoption of a scheduling policy $\boldsymbol{\pi}$, to be chosen from the class $\boldsymbol{\Pi}$ of *admissible policies* which are nonanticipative randomized and engage at most one live project at a time before all deadlines run out. The *multiarmed bandit problem with deadlines* (MABPD) which is the main concern of this paper is to find an admissible policy that maximizes the expected total discounted net reward earned. Denoting by $X_m(t)$ and $a_m(t)$ the prevailing state and action (1: active; 0: passive) for project m at time t, respectivley, and letting $\mathbb{E}_{\mathbf{i}}^{\boldsymbol{\pi}}[\cdot]$ denote expectation under policy $\boldsymbol{\pi}$ conditioned on the initial joint state being equal to $\mathbf{i} = (i_m)$, we can formulate such a problem as

$$\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathbb{E}_{\mathbf{i}}^{\boldsymbol{\pi}} \left[\sum_{m=1}^{M} \sum_{t=0}^{T_m - 1} \left\{ R_m^{a_m(t)}\big(X_m(t)\big) - \nu a_m(t) \right\} \beta^t \right], \quad (1)$$

While the classic *multiarmed bandit problem* (MABP) was introduced in a finite-horizon setting with all deadlines equal $T_m \equiv T$ (cf. [2] and the references therein), the above MABPD with possibly asymmetric deadlines does not appear to have been previously investigated, although a variation with stochastic deadlines is the subject of [15].

Although problem (1) is readily formulated in the *dynamic programming* (DP) framework, the computational solution of the resultant DP equations is hindered by the curse of dimensionality. We will thus pursue the more practical goals of designing and computing a well-grounded tractable heuristic policy based on priority indices that performs well.

A (priority-) *index policy* for (1) is obtained by attaching a numeric *index* $\nu_m(t_m, i_m)$ to each project m as a function of its state i_m and the time remaining to expiry, or *time to go*, $1 \leq t_m \leq T_m$, and then engaging at each time $0 \leq t < \max_m T_m$ a live project m (with $t < T_m$) of highest index $\nu_m\big(T_m - t, X_m(t)\big)$, among those for which the latter exceeds ν, if any, breaking ties arbitrarily. Intuition suggests that optimal policies should have the *nonprocrastinating property* that, in comparing two otherwise identical projects, one should give higher priority to the one whose deadline is further away. Hence, to be consistent with such a property, a priority index must be monotone nondecreasing in the time to go:

$$\nu_m(t_m, i_m) \geq \nu_m(t_m - 1, i_m). \quad (2)$$

Index policies are intuitively appealing and well-suited for practical implementation, which raises the issues of how to define and efficiently compute an index yielding policies that perform well. For such a purpose we will draw on the classical calibration approach to bandit indexation introduced in [2]. For each project m, consider the *parametric optimal-stopping problem*

$$\max_{0 \leq \tau_m \leq T_m} \mathbb{E}_{i_m}^{\tau_m} \left[\sum_{t=0}^{\tau_m - 1} \big\{ R_m\big(X_m(t)\big) - \nu \big\} \beta^t \right], \quad (3)$$

which is to find an optimal stopping-time rule τ_m —where τ_m denotes both a stopping time and the corresponding stopping rule— for stopping the project starting in state i_m by its deadline T_m, which maximizes the expected total discounted value of rewards earned minus continuation charges incurred, where the latter accrue at the parametric rate ν per period. Note that taking $\tau_m = 0$ means to never engage the project, whereas taking $\tau_m = T_m$ means to engage the project until it expires.

Now, we can reformulate (3) as the equivalent problem

$$\max_{0 \leq \tau_m \leq T_m} \mathbb{E}_{i_m}^{\tau_m} \left[\sum_{t=0}^{\tau_m - 1} R_m\big(X_m(t)\big) \beta^t + \sum_{t=\tau_m}^{T_m - 1} \nu \beta^t \right], \quad (4)$$

since the objectives of (3) and (4) differ by a constant. The latter turns out to formulate the *finite-horizon two-armed bandit problem with one arm known*, where one must decide which arm to pull at each time: the known arm, yielding the constant reward ν, or the unknown arm which, when pulled, yields a state-dependent reward and changes state in a Markovian fashion. Such a problem was first solved in [2, Sec. 4] for a Bayesian Bernoulli bandit with $\beta = 1$, the objective being to maximize the expected number of successes. Note that formulation (4) incorporates the intuitive result in [2, Lemma 4.1] on *optimality of write-off policies*, whereby if the known arm is optimal at any time then it is also optimal thereafter. Such a result, which reduces the finite-horizon two-armed bandit problem with one arm known to an optimal stopping problem, extends to the present setting.

In [2, Lemma 4.2] the idea was introduced of using the parametric continuation charge ν as a calibrating device to characterize optimal policies via *break-even* (BE) values $\nu_m^*(t_m, i_m)$: at a time $0 \leq t < T_m$ where the project occupies state $X_m(t) = i_m$ and $t_m = T_m - t$ periods remain to go, one should pull the known arm in (4), and hence stop in (3), iff $\nu_m^*(t_m, i_m) \leq \nu$. If one takes $\nu_m^*(t_m, i_m)$ as the priority index to be used for the MABPD, Lemma 4.3 in [2] ensures that such an index satisfies the nonprocrastinating property (2). Further, their Theorem 4.1 gives an economically intuitive index representation, which extends to the present setting (cf. [4, p. 155]) as

$$\nu_m^*(t_m, i_m) = \max_{0 < \tau_m \leq t_m} \frac{\mathbb{E}_{i_m}^{\tau_m} \left[\sum_{t=0}^{\tau_m - 1} R_m\big(X_m(t)\big) \beta^t \right]}{\mathbb{E}_{i_m}^{\tau_m} \left[\sum_{t=0}^{\tau_m - 1} \beta^t \right]}. \quad (5)$$

Namely, $\nu_m^*(t_m, i_m)$ is the maximum rate of expected discounted reward which can be earned on the project per unit of expected discounted time expended starting at i_m with t_m periods to go.

Such an approach was extended to an infinite-horizon Bayesian Bernoulli bandit model in [1], where optimal policies are characterized by an index $\nu_m^*(i_m)$: the limit of $\nu_m^*(t_m, i_m)$ as $t_m \nearrow \infty$. In a landmark result, [6] first showed that the

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

resultant index —nowadays known as the *Gittins index*— policy is optimal for the infinite-horizon (strictly) discounted MABP in a general Markovian setting.

Although the index policy for the MABPD based on index $\nu_m^*(t_m, i_m)$ is generally suboptimal, the above discussion suggests it is a natural heuristic which is worth investigating. Yet, for such an index policy to be actually used in practice, a key issue that must be addressed is the design of an efficient index algorithm. While computation of the Gittins index has attracted substantial research attention (cf. [11] and the references therein), such has not been the case with its finite-horizon counterpart in (5), which might account for its relatively limited practical impact. Thus, e.g., in the application of the finite-horizon MABP discussed in [3], an ad hoc index policy is investigated, based on an index meant to approximate another index defined via calibration which is said to be hard to compute: the latter is indeed the classic index in (5), which is overlooked in that paper.

While [2, pp. 14–15] outlines two approaches to evaluate their break-even index, they are overly expensive computationally. The scant subsequent work on computation of index $\nu_m^*(t_m, i_m)$ has viewed it in a subordinate role as a means to approximate the Gittins index. Thus, [16] shows that, if $\beta < 1$, the rate of convergence of $\nu_m^*(t_m, i_m)$ to $\nu_m^*(i_m)$ as $t_m \nearrow \infty$ is linear. An exact iterative method for computing $\nu_m^*(t_m, i_m)$ in Bayesian Bernoulli bandits with beta priors is discussed in [4, Sec. 7]. Yet, it involves nonelementary operations: computing the suprema in formula (11) of [4]. An approximate method is proposed in [5, Ch. 1]: the DP equations are solved by backwards recursion for a grid of ν values, and then interpolation is used to approximate the index.

More recently, the author has introduced in [9] an approach to the finite-horizon MABP (with all deadlines equal) based on reformulating a finite-horizon classic bandit project (i.e., one whose state does not change while passive) as an infinite-horizon *restless bandit* project (i.e., one whose state can change while passive). Such a reformulation allows one to deploy the powerful theoretical and algorithmic results available for *restless bandit indexation*. This was introduced by in [17], presenting an extension of the Gittins index that applies to a restricted class of restless bandits termed *indexable*. In recent work including [7, 8, 10], which is reviewed in [13], the author has developed such an approach to obtain a body of theory and algorithms of wide applicability, leading to the unifying concept of *marginal productivity index* (MPI) and to an *adaptive-greedy algorithm* for its computation. Further, such an algorithm can be ued to compute the above finite-horizon index, as this also emerges as the bandit's MPI in its restless reformulation. Yet, having a cubic complexity in the number of states of the restless project, using the fast-pivoting algorithm introduced in [12], direct application of such a one-pass algorithm to a T-horizon n-state project gives an MPI algorithm that performs $O(T^3n^3)$ arithmetic operations. Such a high complexity renders index computation intractable for all but small instances.

1.2 Goals and Contributions

Motivated by the above issues, the main goal of this paper is to significantly improve on the $O(T^3n^3)$ complexity count by introducing a relatively tractable algorithm based on arithmetic operations to compute the finite-horizon index in (5). We accomplish such a goal by drawing on and extending the restless bandit indexation approach introduced in [9], realizing that the classic finite-horizon index is indeed the project's MPI in its restless reformulation. We then exploit special structure to simplify the general one-pass adaptive-greedy index algorithm, decoupling it into a more efficient recursive method. For a given horizon T, the new algorithm presented herein proceeds recursively in T stages, computing at stage $t = 2, \ldots, T$ the index $\{\nu^*(t, i) \colon i \in \mathbb{X}\}$ based on the previous index $\{\nu^*(t-1, i) \colon i \in \mathbb{X}\}$, where we have dropped the project label m and $\mathbb{X}$ is the state space. The recursion starts with the greedy index $\nu^*(1, i) = R(i)$. Using the new algorithm, all such index values are computed by performing only $O(T^2n^3)$ arithmetic operations, thus achieving a T-fold decrease in complexity relative to the previous $O(T^3n^3)$ algorithm, which dramatically increases the size and horizon for which the index can be obtained in practice. Yet, note that computing the finite-horizon index to approximate the Gittins index, as has been proposed elsewhere, is extremely inefficient, as the latter can be computed much faster in $(2/3)n^3 + O(n^2)$ arithmetic operations as shown in [11].

Further, in the special model outlined above of an n-stage project with deadline T, we exploit special structure to obtain a much faster index algorithm, which computes all index values $\nu^*(t, i)$ for $1 \le i \le t \le T$ performing only $O(T^2)$ arithmetic operations, and even give closed-form expressions for the index in the case $\beta = 1$.

The second goal of this paper is to test experimentally the quality of the proposed index policy for the MABPD discussed above. We present experimental evidence of the usefulness of such an index, as it yields a tractable index policy for the MABPD that is nearly optimal, and that significantly outperforms the benchmark greedy and Gittins index policies, across an extensive range of randomly-generated two-project instances.

1.3 Organization of the Paper

The remainder of the paper is organized as follows. Section 2 reviews the relevant results on restless bandit indexation. Section 3 summarizes the key result of a work-reward analysis that exploits special structure, which yields a decoupling of the one-pass adaptive-greedy index algorithm for restless bandits into the new recursive method introduced herein. Section 4 presents the new $O(T^2n^3)$ recursive index algorithm. Section 5 gives the improved $O(T^2)$ index algorithm for the special model referred to above. Finally, Section 6 reports the results of the computational study.

No proofs are included due to space constraints. Detailed proofs will be included in the full version of the paper, currently under preparation. We remark that a similar approach to that in this paper has been deployed by the author in [14] to obtain a fast index algorithm for bandits with switching costs.

2. RESTLESS BANDIT INDEXATION

2.1 Infinite-Horizon Restless Bandit Reformulation

Consider the MABPD discussed above. Proceeding as in [9], consider for each project m the *augmented state*

$$Y_m(t) \triangleq \begin{cases} \big(T_m - t, X_m(t)\big) & \text{if } 0 \le t \le T_m - 1 \\ \big(0, X_m(T_m)\big) & \text{if } t \ge T_m, \end{cases}$$

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

which yields a reformulation of a finite-horizon classic bandit moving over the original state space $\mathbb{X}_m$ as an infinite-horizon restless bandit moving over the *augmented state space* $\mathbb{Y}_{m,T_m} \triangleq \{0,1,\ldots,T_m\} \times \mathbb{X}_m$. The active and passive transition probabilities are given, respectively, by

$$p_m^1\big((t_m,i_m),(t_m-1,j_m)\big) \triangleq p_m\,(i_m,j_m)$$

and

$$p_m^0\big((t_m,i_m),(t_m-1,i_m)\big) \equiv 1$$

for $1 \leq t_m \leq T_m$, while, for $a_m \in \{0,1\}$,

$$p_m^{a_m}\big((0,i_m),(0,i_m)\big) \equiv 1$$

All other transition probabilities are zero. The one-period rewards are $R_m^{a_m}(t_m,i_m) \triangleq R_m^{a_m}(i_m)$ for $1 \leq t \leq T_m$, and $R_m^{a_m}(0,i_m) \equiv 0$. We will find it convenient to partition the restless project's state space as $\mathbb{Y}_{m,T_m} = \overline{\mathbb{Y}}_{m,T_m} \cup \mathbb{Y}_m^{\{0\}}$, where $\overline{\mathbb{Y}}_{m,T_m} \triangleq \{1,\ldots,T_m\} \times \mathbb{X}_m$ is the *controllable state space*, where both actions are available and differ (i.e., before the deadline), and $\mathbb{Y}_m^{\{0\}} \triangleq \{0\} \times \mathbb{X}_m$ is the *uncontrollable state space*, where both actions are identical (i.e., at or after the deadline). The notation $\mathbb{Y}_m^{\{0\}}$ reflects the convention we adopt herein whereby the passive action is taken at uncontrollable states.

We thus reformulate the MABPD (1) as the infinite-horizon *multiarmed restless bandit problem* (MARBP)

$$\max_{\boldsymbol{\pi}\in\boldsymbol{\Pi}} \mathbb{E}_{\mathbf{Y}(0)}^{\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \sum_{m=1}^{M} R_m^{a_m(t)}\big(Y_m(t)\big)\beta^t \right], \tag{6}$$

where $\mathbf{Y}(0)$ is the initial joint augmented state.

2.2 Indexability and the MPI

We discuss below restless bandit indexation (cf. Section 1), as it applies to a single finite-horizon project as above, in its infinite-horizon restless reformulation. We hence drop the project label m henceforth so that, e.g., $\mathbb{X}$ and $\mathbb{Y}_T \triangleq \{0,\ldots,T\} \times N$ denote the project's original and augmented state spaces, respectively, and Π is the class of admissible (history-dependent randomized) project-operating policies π.

We use two criteria to evaluate a policy π, relative to an initial state $(t_0,i_0) \in \mathbb{Y}_T$ —i.e., when $t_0 \leq T$ periods remain to go starting at i_0: the *reward measure*

$$f_{(t_0,i_0)}^{\pi} \triangleq \mathbb{E}_{(t_0,i_0)}^{\pi} \left[\sum_{t=0}^{\infty} R^{a(t)}\big(Y(t)\big)\beta^t \right],$$

giving the expected total discounted value of rewards earned; and the *work measure*

$$g_{(t_0,i_0)}^{\pi} \triangleq \mathbb{E}_{(t_0,i_0)}^{\pi} \left[\sum_{t=0}^{\infty} a(t)\beta^t \right],$$

giving the expected total discounted amount of *work* expended. Note that $f_{(0,i_0)}^{\pi} \equiv g_{(0,i_0)}^{\pi} \equiv 0$.

Imagining that work is paid for at the *wage* rate ν leads us to consider the *ν-wage problem*

$$\max_{\pi\in\Pi} f_{(t_0,i_0)}^{\pi} - \nu g_{(t_0,i_0)}^{\pi}, \tag{7}$$

which is to find an admissible project-operating policy maximizing the value of rewards earned minus labor costs incurred. We will use (7) to *calibrate* the marginal value of work at each state, by analyzing the structure of optimal policies as ν varies.

DP theory ensures that for every wage $\nu \in \mathbb{R}$ there exists an optimal policy that is Markov deterministic and independent of the initial state. We represent each such policy by its *active set*, consisting of those states where the policy prescribes to engage the project. For given original-state subsets $S_1,\ldots,S_T \subseteq \mathbb{X}$, we denote by

$$S_1 \oplus \cdots \oplus S_T \triangleq \{1\} \times S_1 \cup \cdots \cup \{T\} \times S_T \subseteq \overline{\mathbb{Y}}_T$$

the active set for the policy that engages the project when t periods remain to expiry if $X(T-t) \in S_t$, for $t = 1,\ldots,T$.

Further, since the intersection of optimal active sets is optimal (this being an immediate consequence of the Bellman equations), it follows that for any wage ν there exists a unique minimal optimal active set $S_0^*(\nu) \oplus \cdots \oplus S_T^*(\nu) \subseteq \overline{\mathbb{Y}}_T$. We say that the project is *indexable* if there is an *index* $\nu_{(t,i)}^*$ for $(t,i) \in \overline{\mathbb{Y}}_T$ such that, for any wage $\nu \in \mathbb{R}$,

$$S_t^*(\nu) = \big\{i \in \mathbb{X} \colon \nu_{(t,i)}^* > \nu\big\}, \quad t = 1,\ldots,T. \tag{8}$$

We then say that $\nu_{(t,i)}^*$ is the project's *marginal productivity index* (MPI).

Equivalently, the project is indexable with MPI $\nu_{(t,i)}^*$ if it is optimal in (7) to engage (resp. idle) the project when it occupies state (t,i) iff $\nu_{(t,i)}^* \geq \nu$ (resp. $\nu_{(t,i)}^* \leq \nu$).

To establish indexability and analyze the MPI we will deploy the approach developed in [7, 8, 10], based on guessing —and then establishing— the structure of optimal active sets, by identifying an *active-set family* $\mathscr{F}_T \subseteq 2^{\overline{\mathbb{Y}}_T}$ that contains minimal optimal active sets $S_0^*(\nu) \oplus \cdots \oplus S_T^*(\nu)$ as ν varies over $\mathbb{R}$. The result that, if it is optimal to engage (resp. idle) a project in a given state when t periods remain to go, then it is also optimal to do so when more (resp. less) periods remain, leads us to postulate that the right choice of $\mathscr{F}_T$ must be

$$\mathscr{F}_T \triangleq \big\{S_1 \oplus \cdots \oplus S_T \colon S_1 \subseteq \cdots \subseteq S_T \subseteq \mathbb{X}\big\}. \tag{9}$$

Note that $\mathscr{F}_T$ is the family of Markov deterministic *write-off* or *nonprocrastinating* policies which is consistent with (2), i.e., which stop the project when it is close enough to expiry. We will thus aim to establish the project's indexability relative to such a family, or *$\mathscr{F}_T$-indexability*, meaning that the project is indexable and $S_1^*(\nu) \oplus \cdots \oplus S_T^*(\nu) \in \mathscr{F}_T$ for $\nu \in \mathbb{R}$.

2.3 PCL-Indexability and Adaptive-Greedy Index Algorithm.

We next outline the approach we will deploy to establish $\mathscr{F}_T$-indexability and compute the MPI of the restless projects of concern herein, based on showing that they are PCL-indexable relative to $\mathscr{F}_T$, and using the adaptive-greedy index algorithm that computes the MPI of such projects.

Given an action $a \in \{0,1\}$ and an active set $S_1 \oplus \cdots \oplus S_T \in \mathscr{F}_T$, denote by $\langle a, S_1 \oplus \cdots \oplus S_T\rangle$ the policy that takes action a in the initial period and adopts the $S_1 \oplus \cdots \oplus S_T$*-active policy* thereafter. Now, for a controllable state $(t,i) \in \overline{\mathbb{Y}}_T$, define the *marginal work measure*

$$w_{(t,i)}^{S_1\oplus\cdots\oplus S_T} \triangleq g_{(t,i)}^{\langle 1,S_1\oplus\cdots\oplus S_T\rangle} - g_{(t,i)}^{\langle 0,S_1\oplus\cdots\oplus S_T\rangle}, \tag{10}$$

along with the *marginal reward measure*

$$r_{(t,i)}^{S_1\oplus\cdots\oplus S_T} \triangleq f_{(t,i)}^{\langle 1,S_1\oplus\cdots\oplus S_T\rangle} - f_{(t,i)}^{\langle 0,S_1\oplus\cdots\oplus S_T\rangle} \tag{11}$$

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

and the *marginal productivity measure*

$$\nu_{(t,i)}^{S_1\oplus\cdots\oplus S_T} \triangleq \frac{r_{(t,i)}^{S_1\oplus\cdots\oplus S_T}}{w_{(t,i)}^{S_1\oplus\cdots\oplus S_T}}. \tag{12}$$

As we will see (cf. Proposition 3.2), the latter measure is well defined, as its denominator is positive.

We will deploy the PCL-indexability approach to indexation introduced and developed in [7, 8, 10]. For an active set $\mathbf{S} = S_1 \oplus \cdots \oplus S_T \in \mathscr{F}_T$, define the *outer boundary of* $\mathbf{S}$ *relative to* $\mathscr{F}_T$ by

$$\begin{aligned}\partial^{\text{out}}_{\mathscr{F}_T}\mathbf{S} &\triangleq \big\{(t,i) \in \mathbf{S}^c \colon \mathbf{S} \cup \{(t,i)\} \in \mathscr{F}_T\big\} \\ &= \{(T,i) \colon i \in S_T^c\} \cup \bigcup_{t=1}^{T-1} \big\{(t,i) \colon i \in S_{t+1} \setminus S_t\big\},\end{aligned} \tag{13}$$

where we write $S_t^c \triangleq \mathbb{X} \setminus S_t$ and $\mathbf{S}^c \triangleq \overline{\mathbb{Y}}_T \setminus \mathbf{S}$.

We will further refer to the *adaptive-greedy algorithmic scheme* $\text{AG}_{\mathscr{F}_T}$ shown in Table 1, where $n \triangleq |\mathbb{X}|$ denotes the number of project states in the original (nonrestless) formulation. The algorithm's output consists of a string $\{(t_k, i_k)\}_{k=1}^{Tn}$ of distinct augmented states spanning $\overline{\mathbb{Y}}_T$, with $\mathbf{S}^k \triangleq \{(t_1,i_1),\ldots,(t_k,i_k)\} \in \mathscr{F}_T$ for $1 \le k \le Tn$, along with corresponding index values $\{\nu^*_{(t_k,i_k)}\}_{k=1}^{Tn}$. Ties for picking the (t_k, i_k) are broken arbitrarily. We use the term *algorithmic scheme* as it is not yet specified how to compute required marginal productivity rates.

Table 1: Adaptive-Greedy Algorithmic Scheme $\text{AG}_{\mathscr{F}_T}$.

ALGORITHM $\text{AG}_{\mathscr{F}_T}$
Output: $\big\{(t_k,i_k),\nu^*_{(t_k,i_k)}\big\}_{k=1}^{Tn}$

$\mathbf{S}^0 := \emptyset \oplus \cdots \oplus \emptyset$
for $k := 1$ **to** Tn **do**
  pick $(t_k,i_k) \in \arg\max\big\{\nu^{\mathbf{S}^{k-1}}_{(t,i)} : (t,i) \in \partial^{\text{out}}_{\mathscr{F}_T}\mathbf{S}^{k-1}\big\}$
  $\nu^*_{(t_k,i_k)} := \nu^{\mathbf{S}^{k-1}}_{(t_k,i_k)}$; $\mathbf{S}^k := \mathbf{S}^{k-1} \cup \{(t_k,i_k)\}$
end { for }

Now, we say that the project is *PCL-indexable* relative to $\mathscr{F}_T$, or *PCL($\mathscr{F}_T$)-indexable*, if:

(i) for each active set $\mathbf{S} \in \mathscr{F}_T$, $w^{\mathbf{S}}_{(t,i)} > 0$ for $(t,i) \in \overline{\mathbb{Y}}_T$; and

(ii) the index value sequence produced by algorithm $\text{AG}_{\mathscr{F}_T}$ is monotone nonincreasing.

We will later invoke the following key result introduced and developed in [7, 8, 10], which refers to a generic restless project and active-set family $\mathscr{F}$.

THEOREM 2.1. *A PCL($\mathscr{F}$)-indexable project is indexable and algorithm* $\text{AG}_{\mathscr{F}}$ *computes its MPI.*

3. WORK-REWARD ANALYSIS

We summarize in this section the results of an analysis of marginal work and reward measures, which exploits the structure of the model of concern and leads to the new recursive index algorithm.

Regarding marginal reward measure $r_{(t,i)}^{S_1\oplus\cdots\oplus S_T}$, it is readily seen that it does not depend on $S_t,\ldots,S_T$, and hence we will write it as $r_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-1}}$ for $t \ge 2$, and as $r_i \triangleq R_i$ for $t = 1$. The following result gives useful recursive relations on marginal rewards, where $1_{\{i\notin S_{t-1}\}}$ denotes the corresponding indicator and δ_{ij} is Kronecker's delta.

PROPOSITION 3.1. *For $(t,i) \in \overline{\mathbb{Y}}_T, S_1 \oplus \cdots \oplus S_T \in \mathscr{F}_T$:*

(a) $r_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-1}}$ *equals*

$$\begin{cases}(1-\beta)R_i + \beta\displaystyle\sum_{j\in S_{t-1}} p_{ij} r_{(t-1,j)}^{S_1\oplus\cdots\oplus S_{t-2}}, & i \in S_{t-1}, t \ge 2\\ (1-\beta)R_i + \beta r_{(t-1,i)}^{S_1\oplus\cdots\oplus S_{t-2}} & \\ \qquad +\beta\displaystyle\sum_{j\in S_{t-1}} p_{ij} r_{(t-1,j)}^{S_1\oplus\cdots\oplus S_{t-2}}, & i \in S_{t-1}^c, t \ge 2\\ R_i, & t = 1;\end{cases}$$

(b) *for $i^* \in S_{t-1}^c$,* $r_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-1}\cup\{i^*\}}$ *equals*

$$r_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-1}} - \beta(\delta_{ii^*} - p_{ii^*}) r_{(t-1,i^*)}^{S_1\oplus\cdots\oplus S_{t-2}};$$

(c) *for $i^* \in S_{t-1} \setminus S_{t-2}$,* $r_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-2}\cup\{i^*\}\oplus S_{t-1}}$ *equals*

$$\begin{aligned}(1-\beta)R_i + 1_{\{i\notin S_{t-1}\}}\beta r_{(t-1,i)}^{S_1\oplus\cdots\oplus S_{t-2}\cup\{i^*\}} \\ +\beta\sum_{j\in S_{t-1}} p_{ij} r_{(t-1,j)}^{S_1\oplus\cdots\oplus S_{t-2}\cup\{i^*\}};\end{aligned}$$

(d) *for $i^* \in S_{t-k+1} \setminus S_{t-k}$, with $3 \le k \le t-1$,* $r_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-k}\cup\{i^*\}\oplus\cdots\oplus S_{t-1}}$ *equals*

$$\begin{aligned}(1-\beta)R_i + 1_{\{i\notin S_{t-1}\}}\beta r_{(t-1,i)}^{S_1\oplus\cdots\oplus S_{t-k}\cup\{i^*\}\oplus\cdots\oplus S_{t-2}} \\ +\beta\sum_{j\in S_{t-1}} p_{ij} r_{(t-1,j)}^{S_1\oplus\cdots\oplus S_{t-k}\cup\{i^*\}\oplus\cdots\oplus S_{t-2}}.\end{aligned}$$

For marginal work measures $w_{(t,i)}^{S_1\oplus\cdots\oplus S_{t-1}}$, a counterpart to Proposition 3.1 is obtained by setting $R_i \equiv 1$. From such a result, we obtain the required positivity of marginal workloads.

PROPOSITION 3.2. *For $(t,i) \in \mathbb{Y}_T^{\{0,1\}}, \mathbf{S} \in \mathscr{F}_T$, $w^{\mathbf{S}}_{(t,i)} > 0$.*

While Proposition 3.2 shows that the restless projects of concern satisfy condition (i) in the above definition of PCL($\mathscr{F}_T$)-indexability, it remains to establish condition (ii), namely that the index sequence produced by the adaptive-greedy algorithm is monotone nonincreasing. The next result states that such is indeed the case, which ensures the validity of MPI algorithm $\text{AG}_{\mathscr{F}_T}$ via Theorem 2.1.

THEOREM 3.3. *The restless bandit reformulation of a T-horizon classic bandit project is PCL($\mathscr{F}_T$)-indexable.*

We further have the following result.

PROPOSITION 3.4. *The restless project's MPI $\nu^*(t,i)$ is precisely the classic finite-horizon break-even index in* (5).

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

Table 2: Stage T of the recursive index algorithm RI.

ALGORITHM RI_T

Input: $\{i_t^{k_t}, \nu^*_{(t,i_t^{k_t})}\}_{1\le k_t\le n, 1\le t\le T-1}, \{w^{(l)}_{(T-1,i)}, r^{(l)}_{(T-1,i)}\}_{1\le l\le (T-1)n, i\in\mathbb{X}}$

Output: $\{i_T^{k_T}, \nu^*_{(T,i_T^{k_T})}\}_{1\le k_T\le n}, \{w^{(k)}_{(T,i)}, r^{(k)}_{(T,i)}\}_{1\le k\le Tn, i\in\mathbb{X}}$

{ Initialization: } $\beta_{ij} := \beta p_{ij}$, $i,j\in\mathbb{X}$; $b := 1-\beta$; $\widehat{R}_i := bR_i$, $i\in\mathbb{R}$

$S_1^0 := \cdots := S_T^0 := \emptyset$; $k := 1$; $l := 1$; $k_1 := \cdots := k_T := 0$; $\nu^{\max} := -\infty$

$\begin{bmatrix} w^{(0)}_{(T,i)} \\ r^{(0)}_{(T,i)} \end{bmatrix} := \begin{bmatrix} 1 \\ R_i \end{bmatrix}$, $i\in\mathbb{X}$

for $k := 1$ to Tn **do** {note: $S^{k-1} = S_1^{k_1}\oplus\cdots\oplus S_T^{k_T}$, $\widehat{S}^{l-1} = S_1^{k_1}\oplus\cdots\oplus S_{T-1}^{k_{T-1}}$ }

  if $k_T < n$ **then**

    pick $i^*\in\arg\max_{i\in\mathbb{X}\setminus S_T^{k_T}} \frac{r^{(T,k-1)}_{(T,i)}}{w^{(T,k-1)}_{(T,i)}}$; $t^* := T$; $\nu^{\max} := \frac{r^{(T,k-1)}_{(T,i^*)}}{w^{(T,k-1)}_{(T,i^*)}}$

  end { if }

  for $t = T-1$ **down to** 1 **do**

    if $k_t < k_{t+1}$ **and** $i_t^{k_t+1}\in S_{t+1}^{k_{t+1}}$ **and** $\nu^*_{(t,i_t^{k_t+1})} > \nu^{\max}$ **then**

      $(t^*, i^*) := (t, i_t^{k_t+1})$; $\nu^{\max} := \nu^*_{(t,i_t^{k_t+1})}$

    end { if }

  end { for }

  if $t^* = T$ **then**

    $i_T^{k_T+1} := i^*$; $\nu^*_{(T,i^*)} := \nu^{\max}$; $\begin{bmatrix} w^{(k)}_{(T,i)} \\ r^{(k)}_{(T,i)} \end{bmatrix} := \begin{bmatrix} w^{(k-1)}_{(T,i)} \\ r^{(k-1)}_{(T,i)} \end{bmatrix}$, $i\in\mathbb{X}\setminus S_{T-1}^{k_{T-1}}$

  else if $t^* = T-1$

    $\begin{bmatrix} w^{(k)}_{(T,i)} \\ r^{(k)}_{(T,i)} \end{bmatrix} := \begin{bmatrix} w^{(k-1)}_{(T,i)} \\ r^{(k-1)}_{(T,i)} \end{bmatrix} - (\beta\delta_{ii^*} - \beta_{ii^*}) \begin{bmatrix} w^{(l)}_{(T-1,i)} \\ r^{(l)}_{(T-1,i)} \end{bmatrix}$, $i\in\mathbb{X}\setminus S_{T-1}^{k_{T-1}}$

  else { $t^* < T-1$ }

    $\begin{bmatrix} w^{(k)}_{(T,i)} \\ r^{(k)}_{(T,i)} \end{bmatrix} := \begin{bmatrix} b \\ \widehat{R}_i \end{bmatrix} + \beta \begin{bmatrix} w^{(l)}_{(T-1,i)} \\ r^{(l)}_{(T-1,i)} \end{bmatrix} + \sum_{j\in S_{T-1}^{k_{T-1}}} \beta_{ij} \begin{bmatrix} w^{(l)}_{(T-1,j)} \\ r^{(l)}_{(T-1,j)} \end{bmatrix}$, $i\in\mathbb{X}\setminus S_{T-1}^{k_{T-1}}$

  end { if }

  $S_{t^*}^{k_{t^*}+1} := S_{t^*}^{k_{t^*}}\cup\{i^*\}$; $k_{t^*} := k_{t^*}+1$; $k := k+1$; **if** $t^* < T$ **then** $l := l+1$

end { for }

4. RECURSIVE INDEX COMPUTATION

We next draw on the above results to design an efficient implementation of the single-stage adaptive-greedy index algorithm $\mathrm{AG}_{\mathscr{F}_T}$, which decouples it into a T-stage *recursive index* (RI) algorithm.

In order to explain the dynamics of such an RI algorithm, let us focus on its Tth stage RI_T for a given time to go $T\ge 2$, which is described in Table 2. The *input* of algorithm RI_T consists of two blocks: (i) a first block comprised of the original-state strings and MPI values $\{i_t^{k_t}, \nu^*_{(t,i_t^{k_t})}\}_{1\le k_t\le n}$ for smaller times to go $1\le t\le T-1$; and (ii) a second block comprised of certain values of marginal work and marginal reward measures for the previous stage, $T-1$, which are denoted by $\{w^{(l)}_{(T-1,i)}, r^{(l)}_{(T-1,i)}\}_{1\le l\le (T-1)n, i\in\mathbb{X}}$. As for the *output* of algorithm RI_T, it gives the required information to recursively construct the input for the next stage, so it likewise consists of two blocks: (i) a first block comprised of the original-state strings and MPI values $\{i_T^{k_T}, \nu^*_{(T,i_T^{k_T})}\}_{1\le k_T\le n}$ for time to go T; and (ii) a second block comprised of corresponding marginal work and marginal reward measure values $\{w^{(k)}_{(T,i)}, r^{(k)}_{(T,i)}\}_{1\le k\le Tn, i\in\mathbb{X}}$.

Algorithm RI_T performs Tn steps labeled by $k = 1,\ldots,Tn$ to build up the successive active sets $S^{k-1} = S_1^{k_1}\oplus\cdots\oplus S_T^{k_T}$ (with $k = 1+\sum_{t=1}^T k_t$) generated in algorithm $\mathrm{AG}_{\mathscr{F}_T}$, using the information given by its input to avoid redundant computations.

Let us first focus on a given step k of algorithm RI_T, which corresponds to step k of algorithm $\mathrm{AG}_{\mathscr{F}_T}$. Such a step identifies the augmented state (t^*, i^*) that is to be added to the current active set $S^{k-1} = S_1^{k_1}\oplus\cdots\oplus S_T^{k_T}$ to obtain the next one, given by $S^k = S^{k-1}\cup\{(t^*,i^*)\}$. The MPI of augmented state (t^*,i^*) is then given by $\nu^*_{(t^*,i^*)} = \nu^{S^{k-1}}_{(t^*,i^*)} = r^{(k-1)}_{(t^*,i^*)}/w^{(k-1)}_{(t^*,i^*)}$. Now, while algorithm $\mathrm{AG}_{\mathscr{F}_T}$ looks for such a (t^*,i^*) by maximizing from scratch marginal productivity rates $\nu^{S^{k-1}}_{(t,i)}$ for each t such that $k_t < k_{t+1}$, thus obtaining i_t^* and then picking (t^*,i^*) as the best (t,i_t^*) combination, algorithm RI_T finds (t^*,i^*) in a more economic fashion that exploits available information. Thus, the maximization of

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

$\nu^{S^{k-1}}_{(t,i)}$ needs only be carried out for $t = T$ (provided that $k_T < n$). For $t < T$ such that $k_t < k_{t+1}$ and $i_t^{k_t+1} \in S^{k_{t+1}}_{t+1}$ a maximizing i^*_t is already available from the algorithm's input: one can use $i^*_t = i_t^{k_t+1}$ (note that $i^*_t \in S^{k_{t+1}}_{t+1} \setminus S^{k_t}_t$). Moreover, there is no need to evaluate the corresponding marginal productivity rate $\nu^{S^{k-1}}_{(t,i^*_t)}$, since this is precisely the MPI value $\nu^*_{(t,i^*_t)}$, which is again available from the input.

As the step counter k advances from 1 to Tn, algorithm RI_T further spells out how to recursively construct the required marginal work and reward measures $w^{(k-1)}_{(T,i)}$ and $r^{(k-1)}_{(T,i)}$ that are needed to evaluate marginal productivity rates $\nu^{S^{k-1}}_{(T,i)} = r^{(k-1)}_{(T,i)}/w^{(k-1)}_{(T,i)}$. To do so it draws on the recursive relations developed above. Such recursions start by setting $w^{(1)}_{(T,i)} \equiv 1$ and $r^{(1)}_{(T,i)} \equiv R_i$.

The above discussion applies to a stage $T \geq 2$. As for the *greedy* case $T = 1$ corresponding to the initial stage, the required quantities are immediately computed by ordering the states consistently with the greedy index $\nu^*_{(1,i)} = R_i$, i.e., $R_{i^1_1} \geq \cdots \geq R_{i^n_1}$, and setting $w^{(l)}_{(1,i)} \equiv 1$, $r^{(l)}_{(1,i)} \equiv R_i$.

The following is the main result of this paper. It states the validity of such a T-stage index algorithm, and gives its complexity both in terms of arithmetic operations and computer memory requirements.

THEOREM 4.1. *The RI algorithm computes index $\nu^*_{(t,i)}$, for $(t,i) \in \overline{\mathbb{Y}}_T$, performing $O(T^2n^3)$ arithmetic operations, and using $O(Tn^2)$ floating-point storage locations.*

5. FASTER INDEX COMPUTATION FOR A SPECIAL MODEL

While the above RI algorithm applies to an arbitrary bandit model, we can obtain substantially faster index computation schemes in special models, such as that outlined in the Introduction. In such a model, the project state i represents the number of stages remaining to completion, so $\mathbb{X} = \{0, 1, \ldots, n\}$. Working in a period completes the current stage with probability $0 < p < 1$, and does not give a reward except when the last stage is completed before the deadline $T \geq n$, in which a reward R is obtained. We write $q = 1 - p$, $\tilde{p} = \beta p$, and $\tilde{q} = \beta q$, where $0 < \beta \leq 1$ is the discount factor. Note that the index values of interest are $\nu^*_{(t,i)}$ for $t \geq i$, as other index values are 0.

For convenience we use the notation

$$
\begin{aligned}
r_{(t,i)} &\triangleq r^{\{1\}\oplus\cdots\oplus\{1,\ldots,i-1\}\oplus\cdots\oplus\{1,\ldots,i-1\}}_{(t,i)} \\
w_{(t,i)} &\triangleq w^{\{1\}\oplus\cdots\oplus\{1,\ldots,i-1\}\oplus\cdots\oplus\{1,\ldots,i-1\}}_{(t,i)} \\
a_{(t,i)} &\triangleq r^{\{1\}\oplus\cdots\oplus\{1,\ldots,i\}\oplus\cdots\oplus\{1,\ldots,i\}}_{(t,i)} \\
b_{(t,i)} &\triangleq w^{\{1\}\oplus\cdots\oplus\{1,\ldots,i\}\oplus\cdots\oplus\{1,\ldots,i\}}_{(t,i)}.
\end{aligned}
$$

We can show that the index $\nu^*_{(t,i)}$ can be evaluated as $\nu^*_{(t,i)} = r_{(t,i)}/w_{(t,i)}$. Now, to compute the required $r_{(t,i)}$ and $w_{(t,i)}$ one needs to compute the auxiliary quantities $a_{(t,i)}$ and $b_{(t,i)}$. The recursions relating all such quantities are given next:

$$
\begin{aligned}
a_{(t,i)} &= \tilde{p}a_{(t-1,i-1)} + \tilde{q}a_{(t-1,i)}, \quad 2 \leq i \leq t-1 \\
a_{(t,1)} &= \frac{1-\beta+(\beta-\tilde{q})\tilde{q}^{t-1}}{1-\tilde{q}}Rp, \quad t \geq 1 \\
a_{(t,t)} &= r_{(t,t)} = \tilde{p}^{t-1}Rp, \quad t \geq 1,
\end{aligned}
$$

$$
\begin{aligned}
r_{(t,i)} &= \beta r_{(t-1,i)} + \tilde{p}a_{(t-1,i-1)}, \quad 2 \leq i \leq t-1 \\
r_{(t,t)} &= \tilde{p}^{t-1}Rp \\
r_{(t,1)} &= Rp,
\end{aligned}
$$

$$
\begin{aligned}
b_{(t,i)} &= 1-\beta+\tilde{p}b_{(t-1,i-1)} + \tilde{q}b_{(t-1,i)}, \quad 2 \leq i \leq t-1 \\
b_{(t,1)} &= \frac{1-\beta+(\beta-\tilde{q})\tilde{q}^{t-1}}{1-\tilde{q}}, \quad t \geq 1 \\
b_{(t,t)} &= w_{(t,t)}, \quad t \geq 1.
\end{aligned}
$$

and

$$
\begin{aligned}
w_{(t,i)} &= 1-\beta+\beta w_{(t-1,i)} + \tilde{p}b_{(t-1,i-1)}, \quad 2 \leq i \leq t-1 \\
w_{(t,t)} &= 1+\tilde{p}w_{(t-1,t-1)} = \frac{1-\tilde{p}^t}{1-\tilde{p}}, \quad t \geq 1 \\
w_{(t,1)} &= 1
\end{aligned}
$$

It is readily seen that, for a given deadline T, all index values $\nu^*_{(t,i)}$ for $1 \leq i \leq t \leq T$ are computed using the above recursions by performing only $O(T^2)$ arithmetic operations, which represents a dramatic improvement in complexity relative to the general RI algorithm in the previous section.

Further, in the undiscounted case $\beta = 1$ we can actually solve in closed-form such recursions, obtaining

$$
b_{(t,i)} = q^{t-i}\sum_{s=0}^{i-1}\binom{t-i+s}{s}p^s,
$$

$$
w_{(t,i)} = \frac{1-p^i}{1-p} + p\sum_{s=0}^{i-2}\sum_{k=i}^{t-1}\binom{k-i+s+1}{s}q^{k-i+1}p^s,
$$

$$
a_{(t,i)} = \binom{t-1}{i-1}p^{i-1}q^{t-i}Rp,
$$

and

$$
r_{(t,i)} = p^{i-1}Rp\sum_{s=0}^{t-i}\binom{s+i-2}{s}q^s.
$$

From the above recursions one can readily obtain some properties of the index $\nu^*_{(t,i)}$, which translate into insightful properties on the resultant priority-index rule for a multi-project setting. Thus, e.g., the index satisfies the inequalities

$$
\nu^*_{(t-1,i-1)} \geq \nu^*_{(t,i)},
$$

which imply that the MPI policy prescribes sticking to a project as long as stages are successfully completed, consistently with the classical and intuitive "stay on a winner" principle.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

6. COMPUTATIONAL EXPERIMENTS

This section reports the results of a computational study, based on the author's MATLAB implementations of the algorithms described herein. The experiments were performed running MATLAB R2007a under Windows XP x64 in an HP xw9300 dual AMD Opteron 254 (2.8 GHz) workstation with 16 GB of RAM.

The first experiment was set up to assess the runtime and arithmetic operation count (AOC) performance of the RI algorithm. For each state space size $n = 500, 600, \ldots, 1000$ and horizon 50, the algorithm was run on a randomly generated project instance with Uniform[0, 1] rewards and transition probability matrices, the latter being then scaled by dividing each row by its sum. The discount factor was set to $\beta = 1$. Figure 1 plots the runtime (in hours) and the AOC vs. n for each intermediate horizon $T = 1, \ldots, 50$, along with curves obtained by cubic least-squares fit. The results are consistent with the cubic complexity growth in n. Figure 2 displays the same data, but now plotting the runtime and the AOC for each size n vs. T, along with curves obtained by quadratic least-squares fit. The results are consistent with the quadratic complexity growth in T.

The second experiment aimed to assess the relative performance of the proposed finite-horizon/MPI policy for the MABD discussed in Section 1 in two-project instances with asymmetric deadlines, both against the optimal performance and against the two conventional benchmarks: the greedy and the Gittins index policies. Due to computer memory limitations, the experiment was based on two-project instances with $n = 8$ states each, using a maximum deadline of $T = 16$, and a range of values for β. A uniform random sample of 100 such instances was generated. For each instance, the optimal performance was computed solving with CPLEX —interfaced with MATLAB via TOMLAB— the corresponding linear programming (LP) formulation of the Bellman equations. The three alternative index policies were evaluated solving with MATLAB the appropriate linear equation systems.

A sample of 100 two-project instances (where each project has state space $\mathbb{X} = \{1, \ldots, n\}$ with $n = 10$) was randomly generated. For each instance, parameter values for each project were independently generated: transition probabilities (obtained by scaling a matrix with Uniform[0, 1] entries dividing each row by its sum) and active rewards (Uniform[0, 1]). For each instance $k = 1, \ldots, 100$ and deadline-pair $(T_1, T_2) \in \{1, \ldots, T\}^2$ the optimal objective value $v^{(k),\mathrm{opt}}_{(T_1,T_2)}$ and the objective values of the MPI ($v^{(k),\mathrm{MPI}}_{(T_1,T_2)}$) and the benchmark ($v^{(k),\mathrm{bench}}_{(T_1,T_2)}$) policies were evaluated as follows. First, the corresponding *value functions* $v^{(k),\boldsymbol{\pi}}_{((T_1,i_1),(T_2,i_2)}$ were computed for $\boldsymbol{\pi} \in \{\mathrm{opt}, \mathrm{MPI}, \mathrm{bench}\}$, as vectors indexed by the initial joint augmented state $((T_1, i_1), (T_2, i_2))$. Then, the objective values were evaluated by averaging over the initial original states:

$$v^{(k),\boldsymbol{\pi}}_{(T_1,T_2)} \triangleq \frac{1}{n^2} \sum_{i_1,i_2 \in \mathbb{X}} v^{(k),\boldsymbol{\pi}}_{((T_1,i_1),(T_2,i_2))}, \; \boldsymbol{\pi} \in \{\mathrm{opt}, \mathrm{MPI}, \mathrm{bench}\}. \tag{14}$$

Further, the corresponding relative (%) suboptimality gap of the MPI policy $\Delta^{(k),\mathrm{MPI}}_{(T_1,T_2)} \triangleq 100(v^{(k),\mathrm{opt}}_{(T_1,T_2)} - v^{(k),\mathrm{MPI}}_{(T_1,T_2)})/v^{(k),\mathrm{opt}}$, and the relative percentage gain of the MPI over the benchmark policies $\Delta^{(k),\mathrm{MPI,bench}}_{(T_1,T_2)} \triangleq 100(v^{(k),\mathrm{MPI}}_{(T_1,T_2)} - v^{(k),\mathrm{bench}}_{(T_1,T_2)})/v^{(k),\mathrm{bench}}_{(T_1,T_2)}$ were computed. The latter were then averaged over the 100 instances for each (T_1, T_2) deadline-pair to obtain the averaged values $\Delta^{\mathrm{av,MPI}}_{(T_1,T_2)}$ and $\Delta^{\mathrm{av,MPI,bench}}_{(T_1,T_2)}$. To further assess the performance, the maxima of the above quantities over the 100 instances were also computed for each (T_1, T_2), to obtain the values $\Delta^{\mathrm{max,MPI}}_{(T_1,T_2)}$ and $\Delta^{\mathrm{max,MPI,bench}}_{(T_1,T_2)}$.

Figure 3 shows the relative suboptimality gap of the MPI policy, both in the worst case over the 100 instances (left pane) and in average (right pane), as a function of the deadline-pair (T_1, T_2). While the worst such relative gap can be as high as 6%, the average relative gap remains rather small, about a third of the worst case value. The suboptimality is minimal in the symmetric horizon case. The results correspond to the undiscounted case $\beta = 1$.

Figure 4 shows corresponding plots for the relative performance gain of the MPI over the Gittins index policy. The left pane shows that such gains can be as high as 11%, while on average they can reach up to 2.5% for near-term deadlines, as the Gittins-index policy is farsighted. Figure 5 is a counterpart of Figure 4 for the greedy index policy. The left pane shows that such gains can be as high as 35%, while on average they can reach up to 6% for long-term deadlines, as the greedy-index policy is nearsighted.

Acknowledgments

The author's research has been supported in part by the Spanish Ministry of Education and Science under projects MTM2004-02334, MTM2007-63140, and an I3 faculty endowment grant, by the European Union's Network of Excellence Euro-FGI, and by the Autonomous Community of Madrid under grants UC3M-CCG06-UC3M/ESP-0767 and CCG07-UC3M/ESP-3389.

7. REFERENCES

[1] R. Bellman. A problem in the sequential design of experiments. *Sankhyā*, 16:221–229, 1956.

[2] R. N. Bradt, S. M. Johnson, and S. Karlin. On sequential designs for maximizing the sum of n observations. *Ann. Math. Statist.*, 27:1060–1074, 1956.

[3] F. Caro and J. Gallien. Dynamic assortment with demand learning for seasonal consumer goods. *Management Sci.*, 53:276–292, 2007.

[4] J. C. Gittins. Bandit processes and dynamic allocation indices. *J. Roy. Statist. Soc. Ser. B*, 41:148–177, 1979.

[5] J. C. Gittins. *Multi-armed Bandit Allocation Indices*. Wiley, Chichester, UK, 1989.

[6] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In J. Gani, K. Sarkadi, and I. Vincze, editors, *Progress in Statistics (European Meeting of Statisticians, Budapest, 1972)*, pages 241–266. North-Holland, Amsterdam, The Netherlands, 1974.

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

[8] J. Niño-Mora. Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach. *Math. Program.*, 93:361–413, 2002.

[9] J. Niño-Mora. A marginal productivity index policy for the finite-horizon multiarmed bandit problem. In *CDC-ECC'05: Proceedings of the 44th IEEE*

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

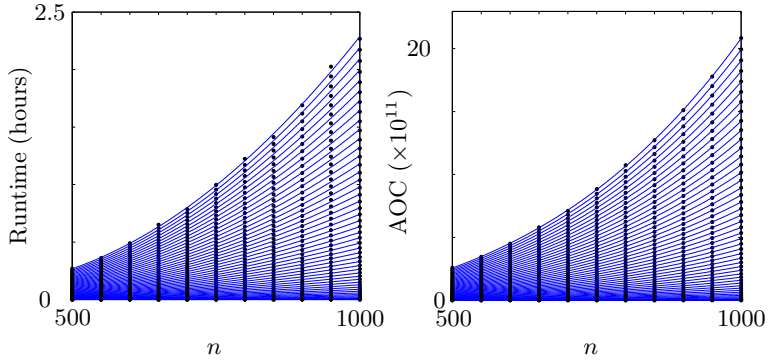

Figure 1: Runtime and AOC vs. n for $T = 1, \ldots, 50$, with Cubic Least-squares Fit.

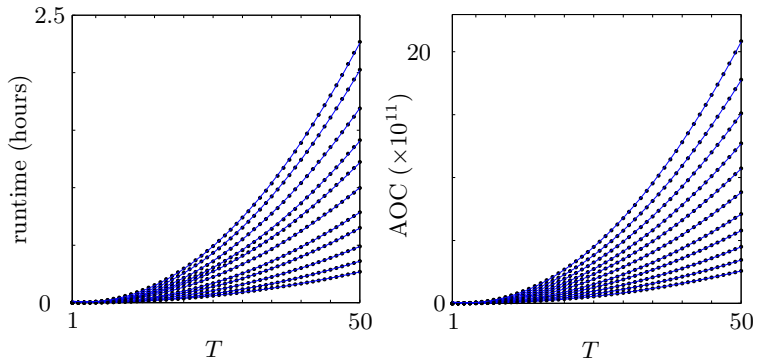

Figure 2: Runtime and AOC vs. T for $n = 500, 550, \ldots, 1000$, with Quadratic Least-squares Fit.

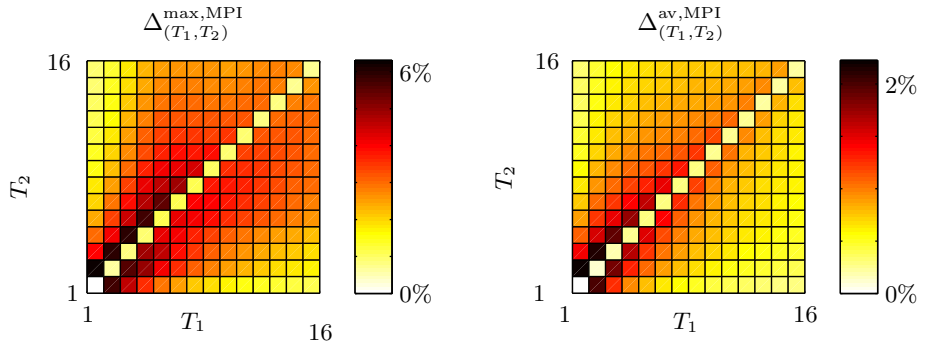

Figure 3: Maximum and Average Relative Suboptimality Gap of MPI Policy vs. (T_1, T_2) for $\beta = 1$.

Conference on Decision and Control and European Control Conference ECC 2005 (Sevilla, Spain), pp. 1718–1722. IEEE, 2005.

[10] J. Niño-Mora. Restless bandit marginal productivity indices, diminishing returns and optimal control of make-to-order/make-to-stock $M/G/1$ queues. *Math. Oper. Res.*, 31:50–84, 2006.

[11] J. Niño-Mora. A $(2/3)n^3$ fast-pivoting algorithm for the Gittins index and optimal stopping of a Markov chain. *INFORMS J. Comput.*, 19:596–606, 2007.

[12] J. Niño-Mora. Characterization and computation of restless bandit marginal productivity indices. In *ValueTools '07: Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools (Nantes, France)*, ACM International Conference Proceedings Series, vol. 321. ICST, Brussels, Belgium, 2007. Published online in the ACM Digital Library.

[13] J. Niño-Mora. Dynamic priority allocation via restless bandit marginal productivity indices. *TOP*, 15:161–198, 2007. Followed by six discussions by I. J. B. F. Adan and O. J. Boxma, E. Altman, O. Hernández-Lerma, R. Weber, P. Whittle, and D. D. Yao.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406

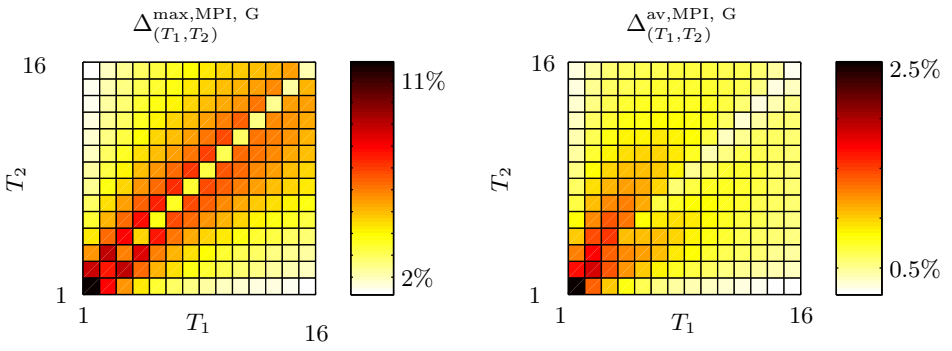

Figure 4: Maximum and Average Relative Gain of MPI over Gittins Index Policy vs. (T_1, T_2) for $\beta = 1$.

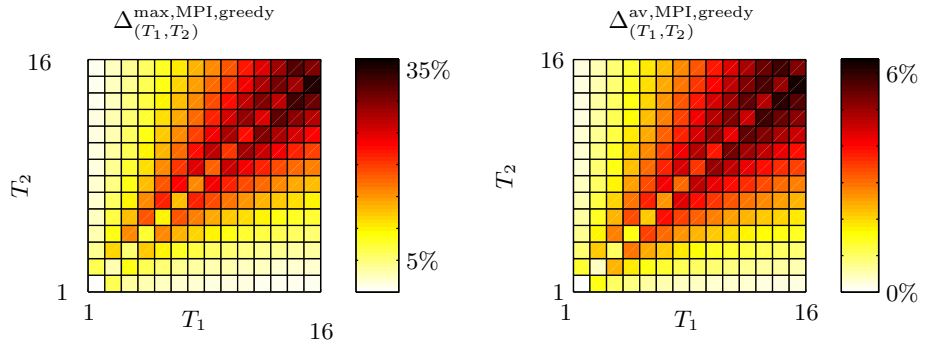

Figure 5: Maximum and Average Relative Gain of MPI over Greedy Index Policy vs. (T_1, T_2) for $\beta = 1$.

[14] J. Niño-Mora. A faster index algorithm and a computational study for bandits with switching costs. *INFORMS J. Comput.*, 20:255–269, 2008.

[15] R. Righter and J. G. Shanthikumar. Independently expiring multiarmed bandits. *Probab. Engrg. Inform. Sci.*, 12:453–468, 1998.

[16] Y.-G. Wang. Error bounds for calculation of the Gittins indices. *Austral. J. Statist.*, 39:225–233, 1997.

[17] P. Whittle. Restless bandits: Activity allocation in a changing world. In J. Gani, editor, *A Celebration of Applied Probability*, spec. vol. 25A of *J. Appl. Probab.*, pp. 287–298. Applied Probability Trust, Sheffield, UK, 1988.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2008.4406
http://dx.doi.org/10.4108/ICST.VALUETOOLS2008.4406