Characterization and Computation of Restless Bandit Marginal Productivity Indices

José Niño-Mora
Department of Statistics
Universidad Carlos III de Madrid
C/ Madrid 126
28903 Getafe (Madrid), Spain
jnimora@alum.mit.edu

ABSTRACT

The restless bandit problem furnishes a powerful modeling paradigm for settings involving the optimal dynamic priority allocation to multiple stochatic projects, given as binary-action (active/passive) Markov decision processes (MDPs). Though generally intractable, Whittle (1988) introduced a tractable priority-index policy, which has been developed in recent work by the author in an extended framework based on the unifying concept of marginal productivity index (MPI). A growing body of empirical evidence shows MPI policies to be nearly optimal in diverse applications, which motivates the interest to compute efficiently the MPI. For such a purpose, we extend to restless bandits the parametric linear programming (LP) approach deployed in [J. Niño-Mora. A $(2/3)n^3$ fast-pivoting algorithm for the Gittins index and optimal stopping of a Markov chain, INFORMS J. Comp., in press] for classic (nonrestless) bandits. Yet the extension is not straightforward, as the MPI is only defined for the restricted range of so-called indexable bandits, which motivates the quest for methods to establish indexability. This paper furnishes algorithmic and analytical tools to realize the potential of MPI policies in large-scale applications, presenting the following contributions: (i) an algorithmic characterization of indexability, for which two block implementations are given; and (ii) new analytical conditions for indexability — termed LP-indexability — that leverage knowledge on the structure of optimal policies, under which the MPI is computed faster by the adaptive-greedy algorithm previously introduced by the author under more stringent (PCL-indexability) conditions, for which a new fast-pivoting block implementation is given. The paper further reports on a computational study, which measures the runtime performance of the algorithms and demonstrates the high prevalence of indexability and PCL-indexability.

Categories and Subject Descriptors

F.2.1 [**Numerical Algorithms and Problems**]: Computations on matrices; G.3 [**Probability and Statistics**]: Markov processes; G.4 [**Mathematical Software**]: Algorithm design and analysis

General Terms

Algorithms

Keywords

Markov decision processes, restless bandits, marginal productivity index, index policies, block algorithms

1. INTRODUCTION

The *multiarmed restless bandit problem* (MARBP) furnishes a powerful modeling framework for a variety of problems where a decision-maker must dynamically prioritize the allocation of effort to multiple projects. The latter are modeled as *restless bandits*, i.e., binary-action (active; passive) semi-Markov decision processes (SMDPs) that can change state while rested. For a range of applications see, e.g., [17, 16, 6, 8, 7, 13, 11, 15, 4, 12].

While the classic (nonrestless) case where passive projects do not change state and one project is to be engaged at each time is efficiently solved by the Gittins index policy (cf. [3]), which attaches to each project state its Gittins index and then engages at each time a project with largest index, the MARBP is generally intractable. Yet, Whittle introduced in [17] an index for restless bandits under the average criterion, and proposed to use the resultant index policy as a heuristic for the MARBP: give higher priority at each time to projects with larger index values. Such an index has been extended by the author in recent work, surveyed in [12], through the unifying concept of *marginal productivity index* (MPI). A growing body of empirical evidence shows that MPI policies are often nearly optimal in a wide variety of applications. Further, the MPI characterizes optimal policies for problems modeling the optimal dynamic allocation of effort to a single project, which have intrinsic interest.

The prime goal of this paper is to announce results on new algorithmic and analytical tools that will allow researchers to fully realize the potential of such index policies in large-scale applications. We will accomplish such a goal by drawing on parametric linear programming (cf. [2]), extending the approach developed in [9] to obtain a Gittins-index algorithm of improved complexity.

The extension is, however, far from straightforward, as the MPI is only defined for the limited range of so-called *in-*

SMCtools '07, October 26, 2007, Nantes, France

dexable bandits, which motivates the quest for useful methods to establish indexability. For such a purpose, we had introduced and developed in [5, 6, 8] sufficient conditions for indexability, termed *PCL-indexability* as they are based on satisfaction of *partial conservation laws* (PCsL), under which a project's MPI is computed by an *adaptive-greedy algorithm*. Yet, though such work shows that several models of interest are PCL-indexable, our more recent work has revealed limitations to such an approach. Specifically: (i) one condition to check was that the index sequence produced by the aforementioned algorithm be nonincreasing, which we have found to be hard to verify analytically in models with a multi-dimensional state; and, (ii) more importantly, we have encountered in results announced in [11] a relevant bandit model that is indexable, yet not PCL-indexable.

This paper overcomes such limitations, presenting the following contributions: (i) a complete algorithmic characterization of indexability, for which two block implementations are given, the *Complete-Pivoting Indexability* (CPI) algorithm and the *Reduced-Pivoting Indexability* (RPI) algorithm, which, after a common initialization stage involving the solution of a block linear equation system, perform $2n^3 + O(n^2)$ and $n^3 + O(n^2)$ arithmetic operations for an n-state bandit, respectively; and (ii) more importantly, new analytical sufficient conditions for indexability, termed *LP-indexability* as they are based on *linear programming* (LP) analyses, which leverage knowledge on the structure of optimal policies in particular models, under which the MPI is computed faster by the adaptive-greedy algorithm referred to above, for which a new fast-pivoting block implementation is given that performs — after the initialization stage — $(2/3)n^3 + O(n^2)$ operations; such conditions are also shown to be necessary, in that an indexable bandit is always *LP-indexable* relative to a certain family of policies; further, a more analytically tractable reformulation of the PCL-indexability conditions is presented. For an application where such an approach is successfully deployed, see [11].

The algorithms presented herein are described in *block-partitioned* form, i.e., based on operations on submatrices (*blocks*) of a base matrix, which has been advocated in the scientific-computing literature to partially overcome the exponentially widening gap between processor speed and memory-access times in contemporary computers, which often render traditional complexity measures based on operation counts poor predictors of runtime performance. See [1].

A computational study was conducted to compare the runtime performance of the proposed index algorithms, and to assess the prevalence of indexability and PCL-indexability among randomly generated restless bandit instances in a simulation study. The latter experiments reveals that such prevalences are extremely high, growing steeply as the number of states increases.

The remainder of the paper is organized as follows. Section 2 reviews the indexation theory for semi-Markov restless bandits. Section 3 elucidates the parametric simplex tableaux for the problem's LP formulation. Section 4 develops a simplex-based algorithmic characterization of indexability. Section 5 shows how to exploit special structure by introducing the new class of LP-indexable bandits, to which the adaptive-greedy index algorithm introduced in earlier work for PCL-indexable bandits is shown to extend, and revises the earlier definition of PCL-indexability; further, a new fast-pivoting implementation is given of such an algorithm. While previous sections focus on the discounted criterion, Section 6 discusses the extension to the average criterion. Section 7 reports on the computational study.

For full proofs of all results presented herein see [10].

2. RESTLESS BANDIT INDEXATION

This section reviews several key concepts of indexation theory to be used throughout the paper, as it applies to a finite-state semi-Markov restless bandit.

2.1 Discrete-Stage Reformulation

Consider the problem of operating optimally a single dynamic and stochastic project, modeled as a binary-action (1/active/engage; 0/passive/rest) SMDP, whose *natural state* $X(t)$ evolves continually over time $t \geq 0$ through the finite state space N. The controller observes the *embedded state* $X_k \triangleq X(t_k)$ at an increasing sequence of *decision epochs* t_k, with $t_0 = 0$ and $\lim_{k\to\infty} t_k \nearrow +\infty$, and takes an action $a_k \triangleq a(t_k) \in \{0, 1\}$ that prevails during the ensuing *stage* $[t_k, t_{k+1})$. Processes $X(t)$ and $a(t)$ are thus piecewise constant, right-continuous with left limits. Actions are prescribed through adoption of a *policy* π, drawn from the class Π of *admissible* policies, which base decisions on the history of embedded states and actions up to the present decision epoch, and on the state observed at the latter. While the project occupies state i and action a prevails, *rewards* accrue and *work* is expended at rates R_i^a and $Q_i^a \geq 0$, respectively, with $Q_i^1 > 0$ and $Q_i^1 \geq Q_i^0 \geq 0$.

We complete next the model's description, by specifying its dynamics, and discuss its discrete-stage reformulation along the lines in [14, Ch. 11]), which will be used in the sequel. If at decision epoch t_k the project occupies state $X_k = i$ and action $a_k = a$ is taken, the joint distribution of the duration $t_{k+1} - t_k$ of the ensuing (i, a)-*stage* and the next embedded state X_{k+1} is given by the transition distribution

$$F_{ij}^a(t) \triangleq \mathbb{P}\left\{t_{k+1} - t_k \leq t, X_{k+1} = j \mid X_k = i, a_k = a\right\},$$

having Laplace-Stieltjes transform (LST)

$$\phi_{ij}^a(\alpha) \triangleq \int_0^\infty e^{-\alpha t}\, dF_{ij}^a(t),$$

for $\alpha > 0$. The corresponding one-stage transition probabilities of the embedded process are

$$p_{ij}^a \triangleq \mathbb{P}\{X_{n+1} = j \mid X_k = i, a_k = a\} = \lim_{t\to\infty} F_{ij}^a(t) = \lim_{\alpha \searrow 0} \phi_{ij}^a(\alpha).$$

From $F_{ij}^a(t)$ we obtain the distribution of the duration of an (i, a)-stage,

$$F_i^a(t) \triangleq \mathbb{P}\left\{t_{k+1} - t_k \leq t \mid X_k = i, a_k = a\right\} = \sum_{j\in N} F_{ij}^a(t),$$

having LST

$$\phi_i^a(\alpha) \triangleq \mathbb{E}\left[e^{-\alpha(t_{k+1} - t_k)} \mid X_k = i, a_k = a\right] = \sum_{j\in N} \phi_{ij}^a(\alpha), \tag{1}$$

and mean

$$m_i^a \triangleq \mathbb{E}\left[t_{k+1} - t_k \mid X_k = i, a_k = a\right] = \int_0^\infty t\, dF_i^a(t).$$

We can thus represent the expected total discounted work expended and the reward earned during an (i, a)-stage, respectively, as

$$q_i^a \triangleq \mathbb{E}\left[\int_{t_k}^{t_{k+1}} Q_{X(t)}^{a_n} e^{-\alpha(t-t_k)}\, dt \mid X_k = i, a_k = a\right] \quad (2)$$

and

$$r_i^a \triangleq \mathbb{E}\left[\int_{t_k}^{t_{k+1}} R_{X(t)}^{a_k} e^{-\alpha(t-t_k)}\, dt \mid X_k = i, a_k = a\right]. \quad (3)$$

In our studies of several applications, we have found that it is often useful to partition the state space N into the set of *uncontrollable states*

$$N^{\{0\}} \triangleq \left\{i \in N \colon q_i^0 = q_i^1,\, F_{ij}^0(t) \equiv F_{ij}^1(t), j \in N\right\},$$

where both actions result in identical resource consumption and dynamics, and the remaining set $N^{\{0,1\}} \triangleq N \setminus N^{\{0\}}$ of *controllable states*. The notation $N^{\{0\}}$ reflects the convention adopted herein whereby the passive action $a = 0$ is taken at uncontrollable states. We will denote by $n \triangleq |N^{\{0,1\}}|$ and $m \triangleq |N^{\{0\}}|$ the numbers of controllable and of uncontrollable states, respectively, and assume that $n \geq 1$. As we will see, the indices of concern in this paper, which are functions of the project's state, are only defined for controllable states.

In the sequel we will focus on the discounted criterion based on measures (4)–(5), deferring to Section 6 discussion of the long-run average criterion.

2.2 Indexation and the MPI

We consider two measures to evaluate a policy π, relative to an initial state i and a discount rate $\alpha > 0$: the *reward measure*

$$f_i^\pi \triangleq \mathbb{E}_i^\pi\left[\int_0^\infty R_{X(t)}^{a(t)} e^{-\alpha t}\, dt\right] = \mathbb{E}_i^\pi\left[\sum_{k=0}^\infty r_{X_k}^{a_k} e^{-\alpha t_k}\right], \quad (4)$$

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

$$g_i^\pi \triangleq \mathbb{E}_i^\pi\left[\int_0^\infty Q_{X(t)}^{a(t)} e^{-\alpha t}\, dt\right] = \mathbb{E}_i^\pi\left[\sum_{k=0}^\infty q_{X_k}^{a_k} e^{-\alpha t_k}\right], \quad (5)$$

giving the expected total discounted amount of work expended. Notice that the right-hand side identities in (4)–(5) draw on the discrete-stage reformulation discussed above.

We will find it convenient to use the corresponding averaged measures obtained when the initial state i is drawn from an arbitrary distribution with positive probability mass $p_i > 0$ for $i \in N$:

$$f^\pi \triangleq \sum_{i \in N} p_i f_i^\pi \quad \text{and} \quad g^\pi \triangleq \sum_{i \in N} p_i g_i^\pi.$$

Introducing a wage rate ν at which work is paid for, leads us to consider the *ν-wage problem*

$$\max_{\pi \in \Pi} f^\pi - \nu g^\pi, \quad (6)$$

which is to find an admissible policy maximizing the value of rewards earned minus labor costs incurred.

The theory of finite-state and -action SMDPs ensures existence of an optimal policy for (6) that is: (i) deterministic stationary; and (ii) independent of the initial state. We represent each such policy by its *active set* $S \subseteq N^{\{0,1\}}$, or subset of controllable states where the policy prescribes to engage the project at a decision epoch, and will refer to it as the *S-active policy.*

It appears reasonable to expect that, in some models, optimal active sets should expand monotonically from $\emptyset$ to $N^{\{0,1\}}$ as the wage ν decreases from $+\infty$ to $-\infty$. Such an intuitive *indexability* property was introduced by Whittle in [17] for Markovian restless bandits with state-independent work rates $q_i^a \equiv a$ under the average criterion.

In *dynamic programming* (DP) terms, we may formulate the indexability property as follows. Letting $\vartheta_i^*(\nu)$ be the optimal value function starting at i for SMDP (6), the Bellman equations are

$$\vartheta_i^*(\nu) = \max_{a \in \{0,1\}} r_i^a - \nu q_i^a + \sum_{j \in N} \phi_{ij}^a \vartheta_j^*(\nu), \quad i \in N, \quad (7)$$

where we write $\phi_{ij}^a = \phi_{ij}^a(\alpha)$. In words, the project is indexable if, for each controllable state i, it is optimal to engage the project at i iff ν is small enough; namely, if there exists an *index* ν_i^*, for $i \in N^{\{0,1\}}$, such that it is optimal to engage the project in state i iff $\nu \leq \nu_i^*$; or, in formulas,

$$\vartheta_i^*(\nu) = r_i^1 - \nu q_i^1 + \sum_{j \in N} \phi_{ij}^1 \vartheta_j^*(\nu) \Longleftrightarrow \nu \leq \nu_i^* \quad (8)$$

Yet, in [8] we have formulated the indexability property in an alternative — though equivalent — form yielding complementary insights, as reviewed next. Let $i_1, \ldots, i_n \in N^{\{0,1\}}$ be an ordering of the n controllable states, such that the *nested active-set family*

$$\mathscr{F}_0 \triangleq \{S_0, S_1, \ldots, S_n\}, \quad (9)$$

where $S_0 \triangleq \emptyset$ and $S_k \triangleq \{i_1, \ldots, i_k\}$ for $1 \leq k \leq n$, satisfies the work-regularity condition

$$g^{S_{k-1}} < g^{S_k}, \quad 1 \leq k \leq n. \quad (10)$$

Consider the index ν_i^*, for $i \in N^{\{0,1\}}$, defined by

$$\nu_{i_k}^* \triangleq \frac{f^{S_k} - f^{S_{k-1}}}{g^{S_k} - g^{S_{k-1}}}, \quad 1 \leq k \leq n. \quad (11)$$

Definition 2.1 (Indexability; MPI). We say that the project is *indexable* if, for some $\mathscr{F}_0$ as above:

(i) index $\nu_{i_k}^*$ is nonincreasing in k, i.e., $\nu_{i_{k+1}}^* \geq \nu_{i_k}^*$; and

(ii) for ν-wage problem (6), the $\emptyset$-active policy is optimal iff $\nu \leq \nu_{i_1}^*$, the $N^{\{0,1\}}$-active policy is optimal iff $\nu \geq \nu_{i_n}^*$, and the S_k-active policy is optimal for ν-wage problem (6) iff $\nu \in [\nu_{i_{k+1}}^*, \nu_{i_k}^*]$, for $1 \leq k < n$.

We then say that the project is *$\mathscr{F}_0$-indexable*, and that ν_i^* is its *marginal productivity index* (MPI).

As noted in [6], the optimal value function of an indexable project has the representation

$$\vartheta_i^*(\nu) = \max_{S \in \mathscr{F}_0} f_i^S - \nu g_i^S = \max_{0 \leq k \leq n} f_i^{S_k} - \nu g_i^{S_k}.$$

2.3 Marginal Measures

The analyses and algorithms below will use the *marginal measures* discussed next. For an action $a \in \{0, 1\}$ and an active set $S \subseteq N^{\{0,1\}}$, denote by $\langle a, S\rangle$ the policy that takes action a in the initial stage and adopts the *S-active policy* (having active set S) thereafter. Now, for a state i and an active set S, define the *marginal work measure*

$$w_i^S \triangleq g_i^{\langle 1,S\rangle} - g_i^{\langle 0,S\rangle}, \tag{12}$$

giving the marginal increase in work expended that results from taking initially the active instead of the passive action at i, given that the S-active policy is adopted thereafter.

Define also the *marginal reward measure*

$$d_i^S \triangleq f_i^{\langle 1,S\rangle} - f_i^{\langle 0,S\rangle}, \tag{13}$$

giving the corresponding marginal increase in the value of rewards earned. Notice that marginal work measures vanish at uncontrollable states:

$$w_i^S = 0, \quad i \in N^{\{0\}}. \tag{14}$$

Finally, for $w_i^S \neq 0$, define the *marginal productivity measure*

$$\nu_i^S \triangleq \frac{d_i^S}{w_i^S}. \tag{15}$$

2.4 Reduction to No Uncontrollable States Case

While we have found the distinction between controllable and uncontrollable states to be relevant in some applications, it would considerably complicate the notation in the sequel. In the full version of this paper [10] it is shown that it suffices to restrict attention to bandits with no uncontrollable states, as these can be eliminated through suitable transformations. We will hence assume henceforth that such transformations have been carried out, if required, focusing attention on the normalized case where all states are controllable.

3. PARAMETRIC SIMPLEX TABLEAUX

We set out in this section to formulate the ν-wage problem (6) as a parametric LP problem, and to elucidate the structure of its simplex tableaux.

3.1 Parametric LP Formulation

The LP formulation of concern is well-known in SMDP theory (cf. [14]) to be

$$\begin{aligned} \vartheta^*(\nu) = &\max\,(\mathbf{r}^0 - \nu\mathbf{q}^0)\mathbf{x}^0 + (\mathbf{r}^1 - \nu\mathbf{q}^1)\mathbf{x}^1 \\ &\text{subject to} \\ &\begin{bmatrix}(\mathbf{I}-\mathbf{\Phi}^0)^\mathsf{T} & (\mathbf{I}-\mathbf{\Phi}^1)^\mathsf{T}\end{bmatrix}\begin{bmatrix}\mathbf{x}^0 \\ \mathbf{x}^1\end{bmatrix} = \mathbf{p} \\ &\mathbf{x}^0, \mathbf{x}^1 \geq \mathbf{0}, \end{aligned} \tag{16}$$

where $\mathbf{x}^a = (x_j^a)$ is a column vector, $\mathbf{r}^a = (r_j^a)$ and $\mathbf{q}^a = (q_j^a)$ are row vectors, and $^\mathsf{T}$ is the transposition operator .

Dual variables x_j^a correspond to the project's *discounted state-action occupancy measures.* For an admissible policy π, initial state i, action a and state j, let

$$x_{ij}^{a,\pi} \triangleq \mathbb{E}_i^\pi\left[\sum_{k=0}^\infty 1_{\{a(t_k)=a, X(t_k)=j\}} e^{-\alpha t_k}\right]$$

be the expected total discounted number of (j,a)-stages under policy π, starting at i. Thus, under initial state distribution $\mathbf{p}$, dual variable x_j^a corresponds to occupancy measure $x_j^{a,\pi} \triangleq \sum_i p_i x_{ij}^{a,\pi}$. Notice that reward and work measures are linear functions of occupancies: writing $\mathbf{x}^{a,\pi} = (x_j^{a,\pi})$,

$$\begin{aligned} f^\pi &= \sum_{(j,a)\in\{0,1\}\times N} r_j^a x_j^{a,\pi} = \mathbf{r}^0\mathbf{x}^{0,\pi} + \mathbf{r}^1\mathbf{x}^{1,\pi} \\ g^\pi &= \sum_{(j,a)\in\{0,1\}\times N} q_j^a x_j^{a,\pi} = \mathbf{q}^0\mathbf{x}^{0,\pi} + \mathbf{q}^1\mathbf{x}^{1,\pi}. \end{aligned} \tag{17}$$

Table 1: Simplex Tableau for S-Active BFS.

	$(\mathbf{x}_S^0)^\mathsf{T}$	x_j^1	$(\mathbf{x}_{S^c\setminus\{j\}}^1)^\mathsf{T}$
$\mathbf{x}_S^1$	$\mathbf{A}_{SS}^S$	$\mathbf{A}_{Sj}^S$	$\mathbf{A}_{S,S^c\setminus\{j\}}^S$
x_j^0	$\mathbf{A}_{jS}^S$	$\boxed{a_{jj}^S}$	$\mathbf{A}_{j,S^c\setminus\{j\}}$
$\mathbf{x}_{S^c\setminus\{j\}}^0$	$\mathbf{A}_{S^c\setminus\{j\},S}^S$	$\mathbf{A}_{S^c\setminus\{j\},j}^S$	$\mathbf{A}_{S^c\setminus\{j\},S^c\setminus\{j\}}^S$
	$\mathbf{w}_S^S$	$-w_j^S$	$-\mathbf{w}_{S^c\setminus\{j\}}^S$
	$\mathbf{d}_S^S$	$-d_j^S$	$-\mathbf{r}_{S^c\setminus\{j\}}^S$

3.2 Basic Feasible Solutions and Reduced Costs

We set out next to analyze parametric LP (16), starting with an elucidation of its *basic feasible solutions* (BFS). Clearly, these correspond to active sets $S \subseteq N^{\{0,1\}}$, and hence we will refer to the *S-active BFS*. For each such S, we decompose the above vectors and matrices as

$$\mathbf{x}^a = \begin{bmatrix}\mathbf{x}_S^a \\ \mathbf{x}_{S^c}^a\end{bmatrix}, \mathbf{p} = \begin{bmatrix}\mathbf{p}_S \\ \mathbf{p}_{S^c}\end{bmatrix}, \mathbf{\Phi}^a = \begin{bmatrix}\mathbf{\Phi}_{SS}^a & \mathbf{\Phi}_{SS^c}^a \\ \mathbf{\Phi}_{S^cS}^a & \mathbf{\Phi}_{S^cS^c}^a\end{bmatrix},$$

where we write $S^c \triangleq N \setminus S$, and introduce the matrices

$$\begin{aligned} &\mathbf{\Phi}^S \triangleq \begin{bmatrix}\mathbf{\Phi}_{SS}^1 & \mathbf{\Phi}_{S,N\setminus S}^1 \\ \mathbf{\Phi}_{S^cS}^0 & \mathbf{\Phi}_{S^cS^c}^0\end{bmatrix}, \mathbf{\Phi}^{S^c} \triangleq \begin{bmatrix}\mathbf{\Phi}_{SS}^0 & \mathbf{\Phi}_{SS^c}^0 \\ \mathbf{\Phi}_{S^cS}^1 & \mathbf{\Phi}_{S^cS^c}^1\end{bmatrix}, \\ &\mathbf{B}^S \triangleq \left(\mathbf{I}-\mathbf{\Phi}^S\right)^\mathsf{T}, \mathbf{N}^S \triangleq \left(\mathbf{I}-\mathbf{\Phi}^{S^c}\right)^\mathsf{T}, \mathbf{H}^S \triangleq \left(\mathbf{B}^S\right)^{-1}, \end{aligned} \tag{18}$$

and $\mathbf{A}^S \triangleq \mathbf{H}^S\mathbf{N}^S$. Notice that $\mathbf{\Phi}^S$ is the transition transform matrix under the S-active policy. Further, $\mathbf{B}^S$ is the *basis matrix* in LP (16) for the S-active BFS, whose *basic variables* are

$$\begin{bmatrix}\mathbf{x}_S^1 \\ \mathbf{x}_{S^c}^0\end{bmatrix};$$

and $\mathbf{N}^S$ is the matrix of non-basic columns in LP (16), whose associated *non-basic variables* are

$$\begin{bmatrix}\mathbf{x}_S^0 \\ \mathbf{x}_{S^c}^1\end{bmatrix}.$$

The next result gives the reduced costs of LP (16).

LEMMA 3.1. *The reduced costs for non-basic variables in the S-active BFS for LP* (16) *are given by*

$$\begin{bmatrix}\mathbf{d}_S^S - \nu\mathbf{w}_S^S & -\mathbf{d}_{S^c}^S + \nu\mathbf{w}_{S^c}^S\end{bmatrix}. \tag{19}$$

3.3 Parametric Simplex Tableau and Pivoting

We can now formulate the parametric simplex tableau under the S-active BFS, as shown in Table 1. The tableau is indexed by basic variables $\mathbf{x}_S^1$ and $\mathbf{x}_{S^c}^0$ in rows, and by non-basic variables $\mathbf{x}_S^0$ and $\mathbf{x}_{S^c}^1$ in columns, and includes two rows of reduced costs for non-basic variables. It includes neither the conventional right-hand side nor the objective value, as they are not needed for our purposes.

The tableau is shown in a form that highlights its structure as it is ready for *pivoting* on element a_{jj}^S, with $j \in S^c$.

Namely, for taking variable x_j^0 *out* of the basis, and putting x_j^1 *into* the basis, which corresponds to moving from the S-active to the $S \cup \{j\}$-active BFS. After such a pivot step is carried out, one obtains the updated tableau in Table 2.

3.4 Computing the Initial Tableau

In [10], we discuss how to compute the initial tableau, corresponding to the $\emptyset$-active BFS, in a numerically-stable form that applies both to the discounted criterion of concern heretofore, and to the long-run average criterion to be addressed in Section 6 below. Such a result is incorporated in the initialization stage of the algorithms presented below.

4. INDEXABILITY CHARACTERIZATION

This section draws on the above results, and on the classic parametric-objective LP algorithm in [2], adapted to the present setting, to develop a characterization of indexability. In what follows, S denotes an arbitrary active set.

4.1 Characteristic Interval of a BFS

We start by addressing the following question: For which range of values of the wage ν is the S-active BFS optimal for parametric LP (16)? Parametric LP theory furnishes the answer in the form of the so-called *characteristic interval* of such a BFS, which we next elucidate in the present context.

In the following result we assume that $\mathbf{p} > \mathbf{0}$ in (16). Note that its part (a) gives the characteristic interval for the S-active BFS, having lower and upper breakpoints

$$\underline{\nu}^S \triangleq \max_{j \in S^c, w_j^S > 0 \text{ or } j \in S, w_j^S < 0} \nu_j^S, \ \overline{\nu}^S \triangleq \min_{j \in S, w_j^S > 0 \text{ or } j \in S^c, w_j^S < 0} \nu_j^S, \tag{20}$$

respectively, while part (b) refers to concepts discussed at the end of Section 2.2. We further write

$$\underline{d}^S \triangleq \max_{j \in S^c, w_j^S = 0} d_j^S \quad \text{and} \quad \overline{d}^S \triangleq \min_{j \in S, w_j^S = 0} d_j^S. \tag{21}$$

We adopt the convention that the maximum (resp. minimum) over an empty set has the value $-\infty$ (resp. $+\infty$).

LEMMA 4.1. *The S-active BFS is optimal for* (16) *iff*

$$\underline{\nu}^S \leq \nu \leq \overline{\nu}^S, \tag{22}$$

and

$$\underline{d}^S \leq 0 \leq \overline{d}^S. \tag{23}$$

Further, it is the unique optimal solution iff the inequalities in (22)–(23) *hold strictly.*

4.2 The CPI Algorithm

We next proceed to put together the above elements to give a complete characterization of indexability, both in combinatorial and algorithmic terms. We will use the *Complete-Pivoting Indexability* (CPI) algorithm described in Table 3, where we have adopted a less unwieldy notation, replacing superscript sets by numeric superscripts, e.g., writing $a_{ij}^{(k)}$ instead of $a_{ij}^{S_k}$. The algorithm seeks to construct a state ordering $i_1, \ldots, i_n$ relative to which the project is indexable (cf. Definition 2.1), with MPI values $\nu_{i_k}^*$ and active sets S_k as in Section 2.2, in which case the Boolean variable INDEXABLE returns the value `true`. It adapts to the present setting the *parametric-objective simplex algorithm* in [2], letting the wage ν decrease from $+\infty$ to $-\infty$, and draws on Lemma 4.1 to test for the structure of successive optimal bases that ensures indexability. For moving from one basis to the next, the algorithm updates the tableau performing a complete simplex pivot step (cf. Table 2), hence its name.

Table 4: Reduced Tableau for S-Active BFS.

	$(\mathbf{x}_S^0)^\top$	x_j^1	$(\mathbf{x}_{S^c \setminus \{j\}}^1)^\top$
x_j^0	$\mathbf{A}_{jS}^S$	a_{jj}^S	$\mathbf{A}_{j,S^c \setminus \{j\}}^S$
$\mathbf{x}_{S^c \setminus \{j\}}^0$	$\mathbf{A}_{S^c \setminus \{j\},S}^S$	$\mathbf{A}_{S^c \setminus \{j\},j}^S$	$\mathbf{A}_{S^c \setminus \{j\},S^c \setminus \{j\}}^S$
	$\mathbf{w}_S^S$	$-w_j^S$	$-\mathbf{w}_{S^c \setminus \{j\}}^S$
	$\mathbf{d}_S^S$	$-d_j^S$	$-\mathbf{d}_{S^c \setminus \{j\}}^S$

The following result gives a complete characterization of indexability in terms of properties of active sets S.

THEOREM 4.2. *The project is indexable iff $\underline{d}^\emptyset \leq 0 \leq \overline{d}^N$ and, for any active set $S \subseteq N$ satisfying* (23) *and $\underline{\nu}^S \leq \overline{\nu}^S$,*

$$\begin{aligned} &\underline{\nu}^N = -\infty, \quad \overline{\nu}^\emptyset = +\infty \\ &\underline{\nu}^S = \max_{j \in S^c \colon w_j^S > 0} \nu_j^S > -\infty, \quad \textit{if } S \neq N \\ &\overline{\nu}^S = \min_{j \in S \colon w_j^S > 0} \nu_j^S < +\infty, \quad \textit{if } S \neq \emptyset. \end{aligned} \tag{24}$$

Theorem 4.2 immediately yields the following algorithmic characterization of indexability.

PROPOSITION 4.3. *The project is indexable iff algorithm CPI terminates in n steps, in which case the computed index ν_j^* is the MPI.*

We next assess the computational complexity of the CPI algorithm's loop, i.e., excluding the initialization stage.

PROPOSITION 4.4. *The CPI algorithm's loop performs at most $2n^3 + O(n^2)$ arithmetic operations.*

4.3 Reduced Tableaux and the RPI Algorithm

We seek next to eliminate unnecessary operations from the CPI algorithm. The key observation is that the tableau's rows corresponding to basic variables $\mathbf{x}_S^1$ are not used to update reduced costs in the CPI algorithm. Hence, it suffices to store and update only *reduced tableaux*, such as that shown in Table 4, which is set up for pivoting on element a_{jj}^S, for $j \in S^c$. Observation of Table 2 shows that a reduced tableau can be updated without using the deleted rows. Simplifying the CPI algorithm accordingly yields the *Reduced-Pivoting Indexability* (RPI) algorithm. For a description see [10].

As shown next, the RPI improves the operation count of the CPI algorithm by a factor of two.

PROPOSITION 4.5. *The RPI algorithm's loop performs at most $n^3 + O(n^2)$ arithmetic operations.*

5. EXPLOITING SPECIAL STRUCTURE

We proceed to discuss how one can leverage structural knowledge on a particular model to obtain substantially simpler indexability conditions and a faster index algorithm. While we had addressed such an issue in [5, 6, 8], by introducing and deploying the PCL-indexability conditions, the approach and results herein are both new, as they draw on the above simplex-based analyses, and of wider applicability. We were motivated to develop them by the difficulties

Table 2: Tableau for $S \cup \{j\}$-Active BFS, Obtained by Pivoting on a^S_{jj}.

	$(\mathbf{x}^0_S)^\mathsf{T}$	x^0_j	$(\mathbf{x}^1_{S^c\setminus\{j\}})^\mathsf{T}$
$\mathbf{x}^1_S$	$\mathbf{A}^S_{SS} - \frac{\mathbf{A}^S_{Sj}\mathbf{A}^S_{jS}}{a^S_{jj}}$	$-\frac{\mathbf{A}^S_{Sj}}{a^S_{jj}}$	$\mathbf{A}^S_{S,S^c\setminus\{j\}} - \frac{\mathbf{A}^S_{Sj}\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a^S_{jj}}$
x^1_j	$\frac{\mathbf{A}^S_{jS}}{a^S_{jj}}$	$\frac{1}{a^S_{jj}}$	$\frac{\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a^S_{jj}}$
$\mathbf{x}^0_{S^c\setminus\{j\}}$	$\mathbf{A}^S_{S^c\setminus\{j\},S} - \frac{\mathbf{A}^S_{S^c\setminus\{j\},j}\mathbf{A}^S_{jS}}{a^S_{jj}}$	$-\frac{\mathbf{A}^S_{S^c\setminus\{j\},j}}{a^S_{jj}}$	$\mathbf{A}^S_{S^c\setminus\{j\},S^c\setminus\{j\}} - \frac{\mathbf{A}^S_{S^c\setminus\{j\},j}\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a^S_{jj}}$
	$\mathbf{w}^S_S + \frac{w^S_j}{a^S_{jj}}\mathbf{A}^S_{jS}$	$\frac{w^S_j}{a^S_{jj}}$	$-\mathbf{w}^S_{S^c\setminus\{j\}} + \frac{w^S_j}{a^S_{jj}}\mathbf{A}^S_{j,S^c\setminus\{j\}}$
	$\mathbf{d}^S_S + \frac{d^S_j}{a^S_{jj}}\mathbf{A}^S_{jS}$	$\frac{d^S_j}{a^S_{jj}}$	$-\mathbf{d}^S_{S^c\setminus\{j\}} + \frac{d^S_j}{a^S_{jj}}\mathbf{A}^S_{j,S^c\setminus\{j\}}$

encountered when trying to deploy the PCL-indexability approach in the analysis of several complex models. The new approach below was useful in such cases. See, e.g., [11].

5.1 LP-Indexability and a Fast-Pivoting Index Algorithm

When investigating a restless bandit model, one is concerned with identifying analytically a range of model parameters for which the model is indexable. Similarly as in the earlier work mentioned, our proposed approach is based on identifying the structure of optimal active sets for (6), in the form of an *active-set family* $\mathscr{F} \subseteq 2^N$ that *contains* an optimal active set $S \in \mathscr{F}$ for every wage value $\nu \in \mathbb{R}$. While such an $\mathscr{F}$ need not be a nested family, it must contain the nested families $\mathscr{F}_0$ discussed in Section 2.2 that can arise as the model's parameters are varied over the range of concern.

Hence, $(N, \mathscr{F})$ is a *set system* on *ground set* N having $\mathscr{F}$ as its family of *feasible sets*. Algorithmic considerations lead us to impose strong structural properties on $(N, \mathscr{F})$, which refer to the *inner* and *outer boundaries* of an active set $S \in \mathscr{F}$, defined respectively by

$$\begin{aligned} \partial^{\text{in}}_{\mathscr{F}} S &\triangleq \{j \in S \colon S \setminus \{j\} \in \mathscr{F}\}, \\ \partial^{\text{out}}_{\mathscr{F}} S &\triangleq \{j \in S^c \colon S \cup \{j\} \in \mathscr{F}\}. \end{aligned} \tag{25}$$

We will further say that two active sets that differ by onen state, i.e., of the form S and $S \cup \{j\}$ with $j \in S^c$, are *adjacent*.

Definition 5.1. We say that $(N, \mathscr{F})$ is a *monotonically connected set system* if:

(i) $\emptyset, N \in \mathscr{F}$;

(ii) for $S, S' \in \mathscr{F}$ with $S \subset S'$, $S' \cap \partial^{\text{out}}_{\mathscr{F}} S \neq \emptyset$ and $S^c \cap \partial^{\text{in}}_{\mathscr{F}} S' \neq \emptyset$; and

(iii) for $S, S' \in \mathscr{F}$, $S \cap S' \in \mathscr{F}$ and $S \cup S' \in \mathscr{F}$.

Note that property (iii) above says that $\mathscr{F}$ is a *lattice*. While various types of set system have been investigated, e.g., matroids or greedoids, to the best of our knowledge the concept of monotonically connected set system in Definition 5.1 is first introduced herein. Such a term is motivated by the fact that, in such a set system, one can always connect two feasible sets $S \subset S'$ by a monotone increasing sequence $S_1 \subset \cdots \subset S_m$ of adjacent sets in $\mathscr{F}$, with $S_1 = S$, $S_m = S'$. One can also connect two distinct feasible sets $S \neq S'$ through two successive monotone sequences of adjacent sets in $\mathscr{F}$, the first of which is monotone increasing and connects S to $S \cup S'$, while the second is monotone decreasing and connects $S \cup S'$ to S'.

Table 5: Minimal Tableau for S-Active BFS.

	$(\mathbf{x}^1_{S^c})^\mathsf{T}$
$\mathbf{x}^0_{S^c}$	$\mathbf{A}^S_{S^cS^c}$
	$\mathbf{w}^S_{S^c}$
	$\mathbf{d}^S_{S^c}$

Table 6: Minimal Tableau for $S \cup \{j\}$-Active BFS.

	$(\mathbf{x}^1_{S^c\setminus\{j\}})^\mathsf{T}$
$\mathbf{x}^0_{S^c\setminus\{j\}}$	$\mathbf{A}^S_{S^c\setminus\{j\},S^c\setminus\{j\}} - \frac{\mathbf{A}^S_{S^c\setminus\{j\},j}\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a^S_{jj}}$
	$\mathbf{w}^S_{S^c\setminus\{j\}} - \frac{w^S_j}{a^S_{jj}}\mathbf{A}^S_{S^c\setminus\{j\},j}$
	$\mathbf{d}^S_{S^c\setminus\{j\}} - \frac{d^S_j}{a^S_{jj}}\mathbf{A}^S_{S^c\setminus\{j\},j}$

Assumption 5.2. $(N, \mathscr{F})$ is monotonically connected.

We will further refer to the *Fast-Pivoting Adaptive-Greedy* index algorithm FPAG($\mathscr{F}$) described in Table 7. This is a simplex-based implementation of the adaptive-greedy index algorithm for PCL-indexable bandits introduced in [5, 6], whose scope we extend herein to the present broader setting. The FPAG($\mathscr{F}$) algorithm is obtained by simplifying the CPI and RPI algorithms above by (i) storing and updating only *minimal tableaux* as shown in Table 5; and (ii) eliminating the indexability test at each step. Note that the minimal tableau for the $S \cup \{j\}$-active BFS is readily computed from that for the S-active BFS in Table 5, as shown in Table 6.

The results in Section 4 motivate us to introduce the following class of projects, which we term *LP($\mathscr{F}$)-indexable* as their are based on LP analyses.

Definition 5.3 (LP($\mathscr{F}$)-indexability). *We say that a project is* LP($\mathscr{F}$)-indexable *if:*

(i) $w^{\emptyset}_i, w^N_i \geq 0$ *for* $i \in N$, *and* $\underline{d}^{\emptyset} \leq 0 \leq \overline{d}^N$;

Table 3: The Complete-Pivoting Indexability (CPI) Algorithm.

```
solve $\begin{bmatrix}\mathbf{I}_{N,N\setminus\{j^*\}} - \mathbf{\Phi}^0_{N,N\setminus\{j^*\}} & \widetilde{\mathbf{m}}^0_N\end{bmatrix}^{\mathsf{T}} \mathbf{A}^{(0)} = \begin{bmatrix}\mathbf{I}_{N,N\setminus\{j^*\}} - \mathbf{\Phi}^1_{N,N\setminus\{j^*\}} & \widetilde{\mathbf{m}}^1\end{bmatrix}^{\mathsf{T}}$

$\begin{bmatrix}\mathbf{w}^{(0)} \\ \mathbf{d}^{(0)}\end{bmatrix} := \begin{bmatrix}\mathbf{q}^1 \\ \mathbf{r}^1\end{bmatrix} - \begin{bmatrix}\mathbf{q}^0 \\ \mathbf{r}^0\end{bmatrix} \mathbf{A}^{(0)}$;  $S_0 := \emptyset$;  $k := 1$;  INDEXABLE := true

if $\max_{j \in N} w_j^{(0)} \le 0$ or $\min_{j \in N} w_j^{(0)} < 0$ or $\max_{j \in N\colon w_j^{(0)} = 0} d_j^{(0)} > 0$, INDEXABLE := false

while INDEXABLE and $k \le n$ do
        $\nu_j^{(k-1)} := d_j^{(k-1)} / w_j^{(k-1)}$,  for $j \in S^c_{k-1}, w_j^{(k-1)} > 0$ and $j \in S_{k-1}, w_j^{(k-1)} < 0$

        pick $i_k \in \arg\max_{j \in S^c_{k-1}, w_j^{(k-1)} > 0} \nu_j^{(k-1)}$;  $\nu^*_{i_k} := \nu^{(k-1)}_{i_k}$;  $S_k := S_{k-1} \cup \{i_k\}$

        if $\max_{j \in S_{k-1}, w_j^{(k-1)} < 0} \nu_j^{(k-1)} > \nu^*_{i_k}$, INDEXABLE := false
        else if  $k < n$
          $p^{(k-1)} = 1/a^{(k-1)}_{i_k i_k}$;  $\mathbf{y}^{(k-1)} := p^{(k-1)} \mathbf{A}^{(k-1)}_{N i_k}$;  $\mathbf{z}^{(k-1)} := \mathbf{A}^{(k-1)}_{i_k N}$

          $\begin{bmatrix}\mathbf{w}^{(k)}_{S_k} & -\mathbf{w}^{(k)}_{S^c_k} \\ \mathbf{d}^{(k)}_{S_k} & -\mathbf{d}^{(k)}_{S^c_k}\end{bmatrix} := \begin{bmatrix}\mathbf{w}^{(k-1)}_{S_{k-1}} & -\mathbf{w}^{(k-1)}_{S^c_{k-1}} \\ \mathbf{d}^{(k-1)}_{S_{k-1}} & -\mathbf{d}^{(k-1)}_{S^c_{k-1}}\end{bmatrix} + p^{(k-1)} \begin{bmatrix}w^{(k-1)}_{i_k} \\ d^{(k-1)}_{i_k}\end{bmatrix} \left\{\mathbf{A}^{(k-1)}_{i_k N} + \mathbf{e}^{\mathsf{T}}_{i_k}\right\}$

          $\mathbf{A}^{(k)} := \mathbf{A}^{(k-1)} - \mathbf{y}^{(k-1)} \mathbf{z}^{(k-1)}$
          $\mathbf{A}^{(k)}_{N i_k} := -\mathbf{y}^{(k-1)}$;  $\mathbf{A}^{(k)}_{i_k N} := p^{(k-1)} \mathbf{z}^{(k-1)}$;  $a^{(k)}_{i_k i_k} := p^{(k-1)}$
          $\mathbf{A}^{(k)} := \mathbf{A}^{(k-1)} - p^{(k-1)} \left\{\mathbf{A}^{(k-1)}_{N i_k} \mathbf{A}^{(k-1)}_{i_k N} + \mathbf{A}^{(k-1)}_{N i_k} \mathbf{e}^{\mathsf{T}}_{i_k} - \mathbf{e}_{i_k} \mathbf{A}^{(k-1)}_{i_k N} - \mathbf{e}_{i_k} \mathbf{e}^{\mathsf{T}}_{i_k}\right\}$

          if $\max_{j \in S^c_k} w_j^{(k)} \le 0$, INDEXABLE := false
        end { if }
        $k := k + 1$
end { while }
if $k = n + 1$ and $\{\max_{j \in N} w_j^{(n)} \le 0$ or $\min_{j \in N} w_j^{(n)} < 0\}$, INDEXABLE := false
```

(ii) *for each $S \in \mathscr{F}$, $w_i^S > 0$ for $i \in \partial^{\text{in}}_{\mathscr{F}} S \cup \partial^{\text{out}}_{\mathscr{F}} S$; and*

(iii) *for every wage $\nu \in \mathbb{R}$ there exists an optimal active set $S \in \mathscr{F}$ for* (6).

Conditions (i, ii) are meant to be established through an ad hoc *work-reward analysis*, while (iii) will be typically established by DP arguments. See [11].

We are now ready to present what we consider the main result of this paper. While its part (a) says that LP($\mathscr{F}$)-indexability is a sufficient condition for indexability, with the MPI being computed by algorithm FPAG($\mathscr{F}$), its part (b) says that such a condition is also necessary, in that an indexable project is always LP-indexable, relative to some nested active-set family.

Theorem 5.4. *The following holds*:

(a) *An LP($\mathscr{F}$)-indexable project is indexable, and its MPI is computed nondecreasingly by algorithm FPAG($\mathscr{F}$).*

(b) *An indexable project is LP($\mathscr{F}_0$)-indexable relative to some nested active-set family $\mathscr{F}_0$.*

The following result assesses the computational complexity of algorithm FPAG($\mathscr{F}$), showing that it improves significantly upon that of algorithm RPI. In particular, the complexity of its "for" loop matches that of solving an $n \times n$ linear equation system by Gaussian elimination.

Proposition 5.5. *The FPAG($\mathscr{F}$) algorithm's loop performs $(2/3)n^3 + O(n^2)$ operations.*

In the special case of nonrestless semi-Markov bandits, using algorithm FPAG($\mathscr{F}$) with $\mathscr{F} = 2^N$ yields a $(2/3)n^3 + O(n^2)$ method to compute the Gittins index, as the initialization step becomes trivial, thus matching the complexity result in [9] for classic Markov bandits.

5.2 PCL-Indexability Revised

We next revise the concept of PCL($\mathscr{F}$)-indexability, introduced and developed in [5, 6, 8], in light of the above developments.

Definition 5.6. *A project is* PCL($\mathscr{F}$)-indexable *if:*

(i) *for each active set $S \in \mathscr{F}$, $w_i^S > 0$ for $i \in N$; and*

Table 7: The Fast-Pivoting Adaptive-Greedy Index Algorithm FPAG($\mathscr{F}$).

```
solve $\mathbf{A}^{(0)} \left[\mathbf{I}_{N,N\setminus\{j^*\}} - \mathbf{\Phi}^0_{N,N\setminus\{j^*\}} \quad \widetilde{\mathbf{m}}^0\right] = \left[\mathbf{I}_{N,N\setminus\{j^*\}} - \mathbf{\Phi}^1_{N,N\setminus\{j^*\}} \quad \widetilde{\mathbf{m}}^1\right]$
$\begin{bmatrix}\mathbf{w}^{(0)} \\ \mathbf{d}^{(0)}\end{bmatrix} := \begin{bmatrix}\mathbf{q}^1 \\ \mathbf{r}^1\end{bmatrix} - \begin{bmatrix}\mathbf{q}^0 \\ \mathbf{r}^0\end{bmatrix} \mathbf{A}^{(0)};\quad S_0 := \emptyset$

for $k := 1$ to $n$ do
    $\nu_i^{(k-1)} := d_i^{(k-1)}/w_i^{(k-1)}, \quad i \in \partial^{\text{out}}_{\mathscr{F}} S_{k-1}$
    pick $i_k \in \arg\max\left\{\nu_i^{(k-1)} : i \in \partial^{\text{out}}_{\mathscr{F}} S_{k-1}\right\};\quad \nu^*_{i_k} := \nu^{(k-1)}_{i_k};\quad S_k := S_{k-1} \cup \{i_k\}$
    if $k < n$ then
        $\mathbf{A}^{(k)}_{S_k^c i_k} := \mathbf{A}^{(k-1)}_{S_k^c i_k}/a^{(k-1)}_{i_k i_k};\quad \mathbf{A}^{(k)}_{S_k^c S_k^c} := \mathbf{A}^{(k-1)}_{S_k^c S_k^c} - \mathbf{A}^{(k)}_{S_k^c i_k}\mathbf{A}^{(k-1)}_{i_k S_k^c}$
    end { if }
    $\mathbf{w}^{(k)}_{S_k^c} := \mathbf{w}^{(k-1)}_{S_k^c} - w^{(k-1)}_{i_k}\mathbf{A}^{(k)}_{S_k^c i_k};\quad \mathbf{d}^{(k)}_{S_k^c} := \mathbf{d}^{(k-1)}_{S_k^c} - d^{(k-1)}_{i_k}\mathbf{A}^{(k)}_{S_k^c i_k}$
end { for }
```

(ii) *for every wage $\nu \in \mathbb{R}$ there exists an optimal active set $S \in \mathscr{F}$ for (6); or*

(ii') *algorithm FPAG($\mathscr{F}$) produces a nonincreasing index sequence: $\nu^*_{i_1} \geq \nu^*_{i_2} \geq \cdots \geq \nu^*_{i_n}$.*

Thus, a PCL($\mathscr{F}$)-indexable project is one that is LP($\mathscr{F}$)-indexable an further has positive marginal work for active sets $S \in \mathscr{F}$. Definition 5.6 differs slightly from those given in the earlier work mentioned, which only required satisfaction of conditions (i, ii'), and imposed less stringent requirements on set system $(N, \mathscr{F})$. Our motivation for introducing the above revision is applied: we have found that, in the analysis of bandit models with complex state spaces, condition (ii') can be much harder to establish than condition (ii).

PROPOSITION 5.7. *In Definition* 5.6*, conditions* (i, ii) *and* (i, ii') *are equivalent.*

6. THE AVERAGE CRITERION

In applications of indexation to problems under the (long-run) average criterion, one must address the version of ν-wage problem (6) based on reward and work measures

$$f_i^\pi \triangleq \liminf_{T \nearrow \infty} \frac{1}{T} \mathbb{E}_i^\pi \left[\int_0^T R_{X(t)}^{a(t)}\, dt\right], \tag{26}$$

and

$$g_i^\pi \triangleq \limsup_{T \nearrow \infty} \frac{1}{T} \mathbb{E}_i^\pi \left[\int_0^T Q_{X(t)}^{a(t)} e^{-\alpha t}\, dt\right]. \tag{27}$$

As in [6, Sec. 6.5], we now assume that the embedded process X_n is *communicating*, so that every state can be reached from every other state under some stationary policy. This ensures that the above measures do not depend on the initial state under a stationary deterministic policy, and hence one can write f^S and g^S for active sets $S \subseteq N$. Hence, the corresponding ν-wage problem (6) can be solved by a stationary deterministic policy independent of i, which allows one to readily extend the indexability theory above to the average criterion.

Regarding the above algorithms, they apply without modification to the average criterion, as the results in Section 3.4 show that the required tableaux emerge as limits of their discounted counterparts as the discount rate vanishes, and also shows how to compute the initial tableau. To extend the results in Section 5 one must further assume that the active-set family $\mathscr{F}$ of concern has the property that, for every $S \in \mathscr{F}$, the S-active policy is *unichain*, i.e., it induces on the embedded process X_n a single recurrent class plus a (possibly empty) set of transient states.

7. COMPUTATIONAL EXPERIMENTS

This section reports the results of several computational experiments, based on the author's MATLAB implementations of the algorithms discussed above.

7.1 Prevalence of (PCL-)Indexability

We start by assessing the prevalence of the indexability and PCL-indexability properties, in a class of randomly generated restless bandit instances.

We considered discrete-time projects, and conducted a simulation study based on generating a random i.i.d. sample of 10^7 instances with $q_i^a = a$ and dense transition probability matrices — obtained by appropriately scaling a matrix with Uniform$[0, 1]$ entries — for each of the state-space sizes $n = 3, \ldots, 7$. For each instance, we used the above algorithms to test for indexability and PCL-indexability (relative to any $\mathscr{F}$), as the discount factor β varies. Note that the value $\beta = 1$ refers to the average criterion discussed in Section 6.

Table 8 reports the results, which show that the prevalences of indexable and of PCL-indexable projects grow steeply both as the discount factor gets smaller and as the state space gets larger.

Such results suggest that, for projects with dense transition probability matrices, both indexability and PCL-indexability are highly prevalent properties. Figure 1 shows a modified version of the classification of restless bandits introduced in [5], updated to better reflect relative class sizes. Note that the figure refers to the class of *GCL-indexable* bandits, named after their satisfaction of *generalized conservation laws* (GCL), which are PCL-indexable relative to $\mathscr{F} = 2^N$.

7.2 Runtime Comparison of Index Algorithms

In contemporary computers, the runtime performance of an algorithm depends both on its arithmetic operation count and on its memory-access patterns, the latter being often the dominant factor. To compare the performance of the algorithms discussed in this paper, we have thus conducted a

Table 8: Counts on Samples of 10^7 Project Instances.

	Nonindexable					Indexable non-PCL				
	number of states					number of states				
β	3	4	5	6	7	3	4	5	6	7
0.1	0	0	0	0	0	0	0	0	0	0
0.2	0	0	0	0	0	0	0	0	0	0
0.3	0	0	0	0	0	0	0	0	0	0
0.4	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0
0.6	0	0	0	0	0	0	0	0	0	0
0.7	0	0	0	0	0	30	0	0	0	0
0.8	16	1	0	0	0	574	32	1	0	0
0.9	135	7	0	0	0	4460	509	36	5	0
1.0	818	66	4	0	0	18631	3640	425	50	3

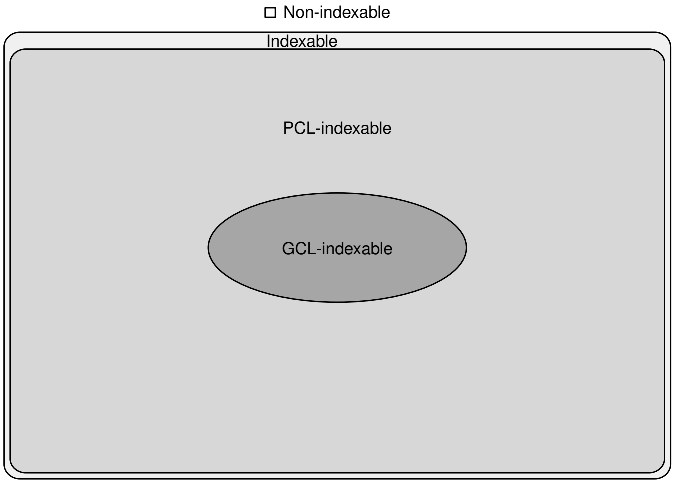

Figure 1: Classification of Restless Bandits.

computational study, using MATLAB implementations developed by the author. The experiments were performed on an HP xw9300 254 (2.8 GHz) Opteron workstation running MATLAB 2006b under Windows XP x64. For each of the state space sizes $n = 1000$ to 6000 (in 500 increments) a random bandit instance was generated. Transition matrices were obtained by scaling random matrices with Uniform$[0, 1]$ entries, dividing each row by its sum. Active rewards were generated with Uniform$[0, 1]$ entries, while passive rewards were zero. The discrete-time discount factor used was $\beta = 0.8$.

For each instance, the CPI algorithm was used to test for indexability and for PCL-indexability (by checking the signs of marginal work measures for the generated nested active-active set family). Since such tests turned out positive in each case, the MPI values were computed using the CPI, RPI and FPAG($\mathscr{F}$) algorithms, with $\mathscr{F} = 2^N$.

Figure 2 displays the recorded runtimes for each algorithm, where where the lines shown are obtained by cubic least-squares fits. The results show that the FPAG algorithm, having an operation count of $(2/3)n^3$, is indeed the fastest of the three, consistently achieving speedup factors of about 1.3 over the CPI and RPI algorithms, which exhibit similar runtimes, though the RPI algorithm was the slowest. Recall that the operation counts are $2n^3$ and n^3 for the CPI and the RPI algorithms, respectively. Such discrepancies between theoretical and actual speedup factors are accounted for by noticing the algorithm's memory-access patterns. Thus, algorithm CPI, being based on complete pivoting steps, has efficient memory-access patterns, as the coefficient matrix $\mathbf{A}$ is always updated as a contiguous memory block. In contrast, both the RPI and the FPAG algorithms reduce the operation count at the expense of using and updating submatrices of $\mathbf{A}$, which results in costly strided memory-access patterns. Yet, the large reduction in arithmetic operations in the FPAG algorithm compensates such inefficiencies, rendering it the fastest algorithm.

Acknowledgments

The author's work has been supported in part by the Spanish Ministry of Education & Science under project MTM2004-02334 and an I3 faculty endowment grant, by the European Union's Networks of Excellence EuroNGI and EuroFGI, and by the Autonomous Community of Madrid under grants UC3M-MTM-05-075 and CCG06-UC3M/ESP-0767.

8. REFERENCES

[1] J. J. Dongarra and V. Eijkhout. Numerical linear algebra algorithms and software. *J. Comput. Appl. Math.*, 123:489–514, 2000.

[2] S. Gass and T. Saaty. The computational algorithm for the parametric objective function. *Naval Res. Log. Quart.*, 2:39–46, 1955.

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

[4] M. Goyal, A. Kumar, and V. Sharma. A stochastic control approach for scheduling multimedia transmissions over a polled multiaccess fading channel. *Wireless Netw.*, 12:605–621, 2006.

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

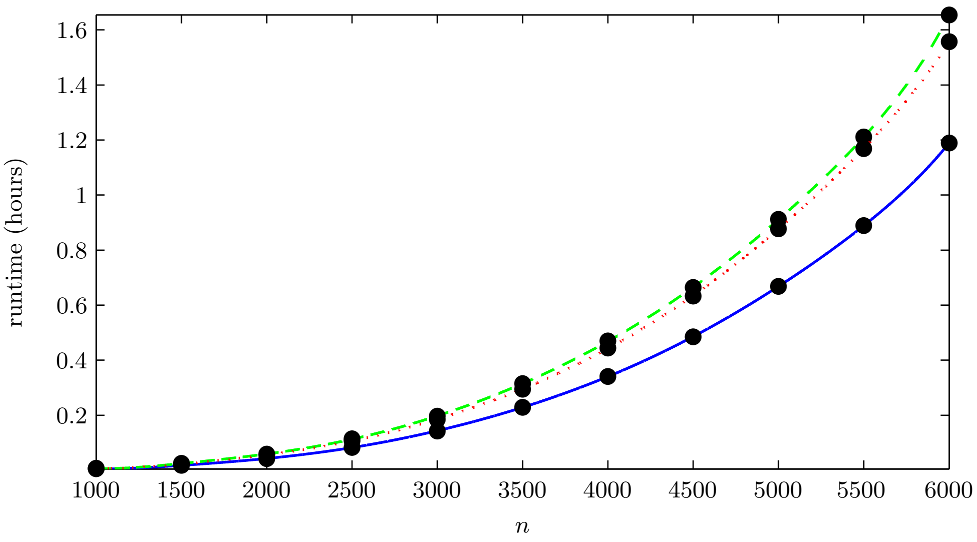

Figure 2: Runtimes with Cubic Least-Squares Fit: FPAG (solid), CPI (dotted) and RPI (dashed).

2001.

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

[7] J. Niño-Mora. Marginal productivity index policies for scheduling a multiclass delay-/loss-sensitive queue. *Queueing Syst.*, 54:281–312, 2006.

[8] 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.

[9] J. Niño-Mora. A $(2/3)n^3$ fast-pivoting algorithm for the Gittins index and optimal stopping of a Markov chain. *INFORMS J. Comp.*, 2007. In press.

[10] J. Niño-Mora. Characterization and computation of restless bandit marginal productivity indices. Working Paper 07-43, Statistics and Econometrics Series 11, Dept. of Statistics, Univ. Carlos III de Madrid, Spain, 2007.

[11] J. Niño-Mora. Computing an index policy for bandits with switching penalties. In *SMCtools '07: Proceedings from the 2007 Workshop on Tools for Solving Structured Markov Chains*. ACM, New York, NY, 2007.

[12] J. Niño-Mora. Dynamic priority allocation via restless bandit marginal productivity indices (with discussion). *Top*, 15, 2007. In press.

[13] J. Niño-Mora. Faster index computation and a computational study for bandits with switching costs. *INFORMS J. Comput.*, 2008. In press.

[14] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.

[15] M. Raissi-Dehkordi and J. S. Baras. Broadcast scheduling in information delivery systems. In *Proceedings of Global Telecommunications Conference GLOBECOM '02*, pages 2935–2939. IEEE, Piscataway, NJ, 2002.

[16] M. H. Veatch and L. M. Wein. Scheduling a multiclass make-to-stock queue: Index policies and hedging points. *Oper. Res.*, 44:634–647, 1996.

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