INFORMS Journal on Computing
Vol. 20, No. 2, Spring 2008, pp. 255–269
ISSN 1091-9856 | EISSN 1526-5528 | 08 | 2002 | 0255

informs®

DOI 10.1287/ijoc.1070.0238

A Faster Index Algorithm and a Computational Study for Bandits with Switching Costs

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

We address the intractable multi-armed bandit problem with switching costs, for which an index that partially characterizes optimal policies was introduced (Asawa, M., D. Teneketzis. 1996. Multi-armed bandits with switching penalties. *IEEE Trans. Automatic Control* **41** 328–348), attaching to each project state a "continuation index" (its Gittins index) and a "switching index." Asawa and Teneketzis proposed to jointly compute both as the Gittins index of a project with $2n$ states—when the original project has n states—resulting in an eightfold increase in $O(n^3)$ arithmetic operations relative to those to compute the continuation index. We present a faster decoupled computation method, which in a first stage computes the continuation index and then, in a second stage, computes the switching index an order of magnitude faster in at most $n^2 + O(n)$ arithmetic operations, achieving overall a fourfold reduction in arithmetic operations and substantially reduced memory operations. The analysis exploits the fact that the Asawa and Teneketzis index is the marginal productivity index of the project in its restless reformulation, using methods introduced by the author. Extensive computational experiments are reported, which demonstrate the dramatic runtime speedups achieved by the new algorithm, as well as the near optimality of the resultant index policy and its substantial gains against the benchmark Gittins index policy across a wide range of randomly generated two- and three-project instances.

Key words: dynamic programming; Markov; finite state; bandits, restless; switching costs; index policies; marginal productivity index; analysis of algorithms; computational complexity
History: Received March 8, 2006; revised September 2006, May 2007, June 2007. Published online in *Articles in Advance* January 25, 2008. Updated March 12, 2008.

1. Introduction

The *Gittins index* furnishes a tractable solution to the *multi-armed bandit problem* (MABP). This involves finding an optimal sequential effort allocation policy for a finite collection of stochastic projects, one of which must be engaged at each discrete time period over an infinite horizon. Projects are modeled as *bandits*, i.e., binary-action (1: active; 0: passive) *Markov decision processes* (MDPs) that can only change state when active. The optimal policy engages at each time a project of largest index, where the index is defined separately for each project as a function of its state; see Gittins and Jones (1974) and Gittins (1979).

A critical assumption underlying such a result is that switching projects is costless. Yet, as noted in Banks and Sundaram (1994, p. 687), "it is difficult to imagine a relevant economic decision problem in which the decision-maker may costlessly move between alternatives." Incorporation of such costs yields the *multi-armed bandit problem with switching costs* (MABPSC); see, e.g., Agrawal et al. (1988), Van Oyen and Teneketzis (1994), and the survey of Jun (2004).

In light of the Gittins and Jones (1974) result for the MABP, it is appealing to try to design good index policies for the MABPSC. Any such a policy attaches an index $\nu_m(a_m^-, i_m)$ to each project m, which is a function of its previous action a_m^- and current state i_m, thus decoupling into a "continuation index" $\nu_m(1, i_m)$ and a "switching index" $\nu_m(0, i_m)$. The index policy prescribes to engage at each time a project of currently largest index.

Although index policies are generally suboptimal for the MABPSC (cf. Banks and Sundaram 1994), Asawa and Teneketzis (1996) introduced an index that partially characterizes optimal policies. Their continuation index is the Gittins index, while their switching index is the maximum rate of expected discounted reward minus switching cost per unit of expected discounted time achievable by stopping rules that engage an initially passive project.

Asawa and Teneketzis (1996) proposed to jointly compute both indices by (i) formulating a modified project *without* switching costs, yet having *twice* the number of states—the (a_m^-, i_m); and (ii) computing the Gittins index of the latter. While several algorithms are available to compute the Gittins index, at present the best complexity for a general n-state project is achieved by the *fast-pivoting algorithm* in Niño-Mora (2007a), which performs $(2/3)n^3 + O(n^2)$ arithmetic operations. Yet using it to compute the Asawa and Teneketzis (AT) indices via their scheme yields an operation count of $(2/3)(2n)^3 + O(n^2)$: an eightfold increase relative to the effort to compute the continuation index alone, which severely hinders deployment to large-scale models. This raises the need to develop a significantly faster computation method, which is the prime goal of this paper.

We accomplish such a goal via a seemingly indirect route, which exploits the reformulation of a classic bandit with switching costs as a *restless bandit*—one that can change state when passive—*without* switching costs. This allows us to deploy the approach to restless bandit indexation introduced by Whittle (1988) and developed by the author (cf. Niño-Mora 2007b), which defines a *marginal productivity index* (MPI) for a restricted class of bandits termed *indexable*. This forms the basis for a heuristic index policy for the intractable *multi-armed restless bandit problem* (MARBP).

We prove and exploit the fact that, for bandits with switching costs, their MPI and their AT index are the same. This provides an economically insightful justification for use of such an index policy as a heuristic for the MABPSC, based on the economic characterization given in Niño-Mora (2002, 2006) of the MPI as a state-dependent measure of the marginal productivity of work on a project. MPI-based priority policies dynamically allocate effort where it appears to be currently more productive, using the MPI of each project as a proxy—as it ignores interactions—marginal productivity measure.

While Asawa and Teneketzis (1996) focused on projects with constant start-up costs, we allow state-dependent start-up and shutdown costs, provided that their sum is nonnegative at each state. This ensures that the continuation index is at least as large as the switching index, consistently with the intuitive *hysteresis* property of optimal policies noticed in Banks and Sundaram (1994, p. 691): "it is obvious that in comparing two otherwise identical arms, one of which was used in the previous period, the one which was in use must necessarily be more attractive than the one which was idle." See also Dusonchet and Hongler (2003).

We deploy the methods introduced in Niño-Mora (2001, 2002, 2006), showing that the restless bandits of concern are *PCL-indexable*—after their satisfaction of *partial conservation laws* (PCLs), which allows us to compute the MPI using the *adaptive-greedy algorithm* introduced in such work. We further decouple the latter into a faster two-stage method: a first stage that computes the continuation index along with additional quantities; and a second stage that uses the first stage's output to compute the switching index.

To implement such a scheme, one can use for the first stage several algorithms in Niño-Mora (2007a), such as the *fast-pivoting algorithm with extended output* FP(1), which performs $(4/3)n^3 + O(n^2)$ arithmetic operations. For the second stage, we present a switching-index algorithm that performs *at most* $n^2 + O(n)$ arithmetic operations, rendering negligible the marginal effort to compute the switching index. Relative to the AT scheme, our two-stage method achieves overall a fourfold reduction in arithmetic operations along with substantially reduced memory operations. Such an algorithm is the main contribution of this paper.

A computational study demonstrates that such an improved complexity yields dramatic runtime speedups. The study is complemented by a set of experiments that demonstrate the near optimality of the index policy and its substantial gains against the benchmark Gittins-index policy across an extensive range of two- and three-project instances.

Section 2 discusses the model and its restless-bandit reformulation. Section 3 reviews restless-bandit indexation, gives an indexability proof based on dynamic programming (DP), shows that the MPI is precisely the AT index, and outlines the approach to be deployed. Section 4 carries out a PCL-indexability analysis of a project with switching costs, in its restless reformulation. Section 5 draws on such an analysis to develop the new decoupled MPI algorithm. Section 7 reports the computational study's results. Section 8 concludes.

2. MABPSC and Restless-Bandit Reformulation

2.1. Model Formulation

We next describe the MABPSC outlined in §1. Consider a collection of M finite-state projects, one of which must be engaged (*active*) at each discrete time period $t = 0, 1, 2, \ldots$ over an infinite horizon, while the others are rested (*passive*). When project m occupies state i_m—belonging in its state space N_m—and is engaged, it yields an *active reward* $R_m^1(i_m) = R_m(i_m)$ and its state moves to j_m with probability $p_m(i_m, j_m)$. If the project is instead rested, it yields a zero *passive reward* $R_m^0(i_m) = 0$ and its state remains frozen.

Switching projects is costly. When project m occupies state i_m and is freshly engaged (respectively, rested), a *startup cost* $c_m(i_m)$ (respectively, *shutdown cost* $d_m(i_m)$) is incurred. We assume that $c_m(i_m) + d_m(i_m) \geq 0$. Rewards and costs are time discounted with factor $0 < \beta < 1$.

Actions are chosen by adoption of a *scheduling policy* $\boldsymbol{\pi}$, drawn from the class $\boldsymbol{\Pi}$ of *admissible policies*, which are nonanticipative relative to the history of states and actions and engage one project at a time.

The MABPSC is to find an admissible policy maximizing the expected total discounted value of rewards earned minus switching costs incurred.

Denote by $X_m(t) \in N_m$ and $a_m(t) \in \{0, 1\}$ the state and action for project m at time t, respectively. We will use the notation

$$a_m^-(t) \triangleq a_m(t-1), \qquad \bar{a}_m(t) \triangleq 1 - a_m(t), \qquad \text{and} \quad \bar{a}_m^-(t) \triangleq \bar{a}_m(t-1). \tag{1}$$

Because initial conditions must specify whether each project m is initially set up, we denote such a status by $a_m^-(0)$. We further define the project's *augmented state* $\widehat{X}_m(t) \triangleq (a_m^-(t), X_m(t))$, which moves over the *augmented state space* $\widehat{N}_m \triangleq \{0, 1\} \times N_m$. The system's *joint augmented state* is thus $\widehat{\mathbf{X}}(t) \triangleq (\widehat{X}_m(t))_{m=1}^M$ and its *joint action process* is $\mathbf{a}(t) \triangleq (a_m(t))_{m=1}^M$.

Given a joint initial state $\widehat{\mathbf{X}}(0) = \hat{\boldsymbol{\imath}} = ((a_m^-, i_m))_{m=1}^M$, we can formulate the MABPSC as

$$\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathbb{E}_{\hat{\boldsymbol{\imath}}}^{\boldsymbol{\pi}} \left[\sum_{m=1}^M \sum_{t=0}^\infty \{ R_m^{a_m(t)}(X_m(t)) - c_m(X_m(t)) \bar{a}_m^-(t) a_m(t) - d_m(X_m(t)) a_m^-(t) \bar{a}_m(t) \} \beta^t \right], \tag{2}$$

where $\mathbb{E}_{\hat{\boldsymbol{\imath}}}^{\boldsymbol{\pi}}[\cdot]$ is expectation under $\boldsymbol{\pi}$ conditioned on the initial joint state being equal to $\hat{\boldsymbol{\imath}}$.

2.2. Restless-Bandit Reformulation

Taking $\widehat{X}_m(t)$ as the state of project m yields a reformulation of (2) as an MARBP (cf. §1) *without* switching costs. The rewards and transition probabilities in such a reformulation are as follows. If the project occupies augmented state (a_m^-, i_m) and is engaged, the active reward $\widehat{R}_m^1(a_m^-, i_m) \triangleq R_m^1(i_m) - c_m(i_m)\bar{a}_m^-$ accrues, and its augmented state moves to $(1, j_m)$ with active transition probability $\hat{p}_m^1((a_m^-, i_m), (1, j_m)) \triangleq p_m(i_m, j_m)$. If it is rested, the one-period passive reward $\widehat{R}_m^0(a_m^-, i_m) \triangleq R_m^0(i_m) - d_m(i_m) a_m^-$ accrues, and its augmented state moves to $(0, i_m)$ with a unity passive transition probability: $\hat{p}_m^0((a_m^-, i_m), (0, i_m)) \equiv 1$.

We can thus reformulate (2) as the MARBP

$$\max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathbb{E}_{\hat{\boldsymbol{\imath}}}^{\boldsymbol{\pi}} \left[\sum_{m=1}^M \sum_{t=0}^\infty \widehat{R}_m^{a_m(t)}(\widehat{X}_m(t)) \beta^t \right]. \tag{3}$$

This allows us to deploy the approach to restless-bandit indexation introduced by Whittle (1988) and developed by the author in recent work (cf. Niño-Mora 2007b). It must be emphasized, though, that the resultant index policies are generally suboptimal.

3. Restless-Bandit Indexation

In this section, we review restless-bandit indexation, give an indexability proof based on DP, show that the MPI is the AT index, and outline the approach to be deployed. We draw on Whittle (1988) and Niño-Mora (2001, 2002, 2006) for the definition, analysis, and computation of the MPI, as it applies to a single project as above—in its restless reformulation. We thus drop below the project label m so that now, e.g., $\widehat{N} \triangleq \{0, 1\} \times N$ denotes the project's augmented state space. We denote by π and Π the policies and policy space for operating the project, where the notation distinguishes them from their boldface counterparts $\boldsymbol{\pi}$ and $\boldsymbol{\Pi}$ used above in the multiproject setting.

We assume henceforth that switching costs satisfy the following key condition.

Assumption 3.1. *$c_i + d_i \geq 0$ for $i \in N$.*

3.1. Indexability, MPI, and Hysteresis

We evaluate the value of *net rewards* (i.e., accounting for switching costs) earned on the project under a policy $\pi \in \Pi$ starting at $(a_0^-, i_0) \in \widehat{N}$ by the *reward measure*

$$f_{(a_0^-, i_0)}^\pi \triangleq \mathbb{E}_{(a_0^-, i_0)}^\pi \left[\sum_{t=0}^\infty \widehat{R}^{a(t)}(\widehat{X}(t)) \beta^t \right],$$

and further evaluate the corresponding amount of work expended by the *work measure*

$$g_{(a_0^-, i_0)}^\pi \triangleq \mathbb{E}_{(a_0^-, i_0)}^\pi \left[\sum_{t=0}^\infty a(t) \beta^t \right].$$

Imagining that work on the project is paid for at the *wage rate* ν per unit work performed leads us to consider the *ν-wage problem*

$$\max_{\pi \in \Pi} f_{(a_0^-, i_0)}^\pi - \nu g_{(a_0^-, i_0)}^\pi, \tag{4}$$

which is to find an admissible policy achieving the maximum value $v_{(a_0^-, i_0)}^*(\nu)$ of net rewards earned minus labor costs incurred. We will use (4) to *calibrate the marginal value of work* at each state, by analyzing the structure of its optimal policies as the wage ν varies.

Because (4) is a finite-state and finite-action MDP, we know that for any wage $\nu \in \mathbb{R}$, there exists an optimal policy that is stationary deterministic and independent of the initial state. Each such policy is characterized by its *active-state set*, or *active set*, which is the subset of augmented states where it prescribes to engage the project. We write active sets as

$$S_0 \oplus S_1 \triangleq \{0\} \times S_0 \cup \{1\} \times S_1, \quad S_0, S_1 \subseteq N.$$

Thus, the policy that we denote by $S_0 \oplus S_1$ engages the project when it was previously rested (respectively, engaged) if its original state $X(t)$ lies in S_0 (respectively, in S_1).

Hence, for any ν, there is a unique *minimal optimal active set* $S_0^*(\nu) \oplus S_1^*(\nu) \subseteq \widehat{N}$, consisting of all augmented states where engaging the project is the only optimal action. The dependence of such sets on ν is used next to define the *indexability property*.

Definition 3.2 (Indexability; MPI). We say that the project is *indexable* if there exists an *index* $\nu^*_{(a^-,i)}$ for $(a^-, i) \in \widehat{N}$ such that

$$S_0^*(\nu) = \{(0, i)\colon \nu^*_{(0,i)} > \nu\} \quad \text{and} \\ S_1^*(\nu) = \{(1, i)\colon \nu^*_{(1,i)} > \nu\}, \quad \nu \in \mathbb{R}. \tag{5}$$

We say that $\nu^*_{(a^-,i)}$ is the project's *marginal productivity index* (MPI), terming $\nu^*_{(1,i)}$ the *continuation MPI* and $\nu^*_{(0,i)}$ the *switching MPI*.

Thus, the project is indexable with MPI $\nu^*_{(a^-,i)}$ if it is optimal to engage (respectively, rest) the project when it occupies (a^-, i) iff $\nu^*_{(a^-,i)} \geq \nu$ (respectively, $\nu^*_{(a^-,i)} \leq \nu$). Whittle (1988) introduced the concept of indexability for restless bandits under the average criterion. We introduced the term MPI in Niño-Mora (2006), motivated by the characterization of the index as a measure of the state-dependent marginal productivity of work on a project.

To establish indexability and compute the MPI, Niño-Mora (2001, 2002, 2006) develop an approach based on guessing the structure of optimal active sets, as an *active-set family* $\widehat{\mathscr{F}} \subseteq 2^{\widehat{N}}$ that contains sets $S_0^*(\nu) \oplus S_1^*(\nu)$ as ν varies. The intuition that, under Assumption 3.1, optimal policies should have the hysteretic property (cf. §1) that, if it is optimal to engage a project when it was previously rested, then it should also be optimal to engage it when it was previously active, leads us to posit that the right choice of $\widehat{\mathscr{F}}$ must be

$$\widehat{\mathscr{F}} \triangleq \{S_0 \oplus S_1\colon S_0 \subseteq S_1 \subseteq N\}. \tag{6}$$

Definition 3.3 ($\widehat{\mathscr{F}}$-Indexability). We say that the project is *$\widehat{\mathscr{F}}$-indexable* if (i) it is indexable, and (ii) $S_0^*(\nu) \oplus S_1^*(\nu) \in \widehat{\mathscr{F}}$ for $\nu \in \mathbb{R}$.

Note that $\widehat{\mathscr{F}}$ represents a family of policies. We will establish in Theorem 3.7 that, with $\widehat{\mathscr{F}}$ defined as in (6), the restless projects of concern in this paper are indeed $\widehat{\mathscr{F}}$-indexable.

3.2. Reduction to the Normalized Zero Shutdown Costs Case

This section shows that it suffices to consider the zero shutdown costs case, extending a result in Banks and Sundaram (1994, §3) for the constant switching-costs case $c_i \equiv c$, $d_i \equiv d$. Suppose that at a certain time, which we take to be $t = 0$, a project is freshly engaged for a random duration given by a stopping-time rule τ. Denoting by $\mathbf{R} = (R_j)$, $\mathbf{c} = (c_j)$, and $\mathbf{d} = (d_j)$ its state-dependent active reward, start-up, and shutdown cost vectors, we can write the expected discounted net earnings over such a time span starting at $X(0) = i$ as

$$f_i^\tau(\mathbf{R}, \mathbf{c}, \mathbf{d}) \triangleq \mathbb{E}_i^\tau\left[-c_i + \sum_{t=0}^{\tau-1} R_{X(t)}\beta^t - d_{X(\tau)}\beta^\tau\right]. \tag{7}$$

We have the following result, where $\mathbf{I}$ is the identity matrix indexed by the state space N, $\mathbf{P} = (p_{ij})_{i,j\in N}$ is the transition probability matrix, and $\mathbf{0}$ is a vector of zeros.

Lemma 3.4. $f_i^\tau(\mathbf{R}, \mathbf{c}, \mathbf{d}) = f_i^\tau(\mathbf{R} + (\mathbf{I} - \beta\mathbf{P})\mathbf{d}, \mathbf{c} + \mathbf{d}, \mathbf{0})$.

Proof. Use the elementary identity

$$d_{X(\tau)}\beta^\tau = d_i - \sum_{t=0}^{\tau-1}\{d_{X(t)} - \beta d_{X(t+1)}\}\beta^t$$

to obtain

$$\begin{aligned} f_i^\tau(\mathbf{R}, \mathbf{c}, \mathbf{d}) &\triangleq -c_i + \mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1} R_{X(t)}\beta^t - d_{X(\tau)}\beta^\tau\right] \\ &= -c_i - d_i + \mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1}\{R_{X(t)} + d_{X(t)} - \beta d_{X(t+1)}\}\beta^t\right] \\ &= f_i^\tau(\mathbf{R} + (\mathbf{I} - \beta\mathbf{P})\mathbf{d}, \mathbf{c} + \mathbf{d}, \mathbf{0}). \quad \square \end{aligned}$$

Lemma 3.4 shows how to eliminate shutdown costs: one simply incorporates them into modified start-up costs and active rewards given by the transformations

$$\tilde{\mathbf{c}} \triangleq \mathbf{c} + \mathbf{d} \quad \text{and} \quad \tilde{\mathbf{R}} \triangleq \mathbf{R} + (\mathbf{I} - \beta\mathbf{P})\mathbf{d}. \tag{8}$$

Note that in the case $c_j \equiv c$ and $d_j \equiv d$ discussed in Banks and Sundaram (1994), such transformations reduce to $\tilde{c}_j \equiv c + d$ and $\tilde{R}_j = R_j + (1 - \beta)d$, in agreement with their results.

For simplicity, we will hence focus the discussion henceforth in the *normalized case* $c_i \geq 0$, $d_i \equiv 0$, assuming that the transformations in (8) have been carried out if required.

3.3. AT Index, MPI, and Indexability Proof

We proceed to give representations of the continuation and switching AT indices in the normalized case, to present a DP-based proof of $\widehat{\mathscr{F}}$-indexability, and to show that the AT indices match their MPI counterparts. The representations are given in terms of work and reward measures g_i^S and f_i^S for the underlying project without switching costs starting at $i \in N$, under the policy that engages the project when its original state $X(t)$ lies in $S \subseteq N$.

Define the continuation and the switching AT index by

$$\begin{aligned} \nu^{\mathrm{AT}}_{(a^-,i)} &\triangleq \max_{\tau > 0} \frac{\mathbb{E}_i^\tau\left[-(1 - a^-)c_i + \sum_{t=0}^{\tau-1} R_{X(t)}\beta^t\right]}{\mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1}\beta^t\right]} \\ &= \max_{S\subseteq N\colon i\in S} \frac{f_i^S - (1 - a^-)c_i}{g_i^S}, \end{aligned} \tag{9}$$

where τ denotes a stopping-time rule. Hence, $\nu^{\mathrm{AT}}_{(1,i)}$ is precisely the project's Gittins index. Note that the right-most identity in (9) follows because it suffices to restrict attention to stationary deterministic stopping rules, which are represented by their continuation sets.

The next result shows that such indices are consistent with the policy family $\widehat{\mathscr{F}}$ in (6).

Lemma 3.5. *$\nu^{\mathrm{AT}}_{(1,i)} \geq \nu^{\mathrm{AT}}_{(0,i)}$ for $i \in N$.*

Proof. Fix $i \in N$. For each set $S \subseteq N$, Assumption 3.1, i.e., $c_i \geq 0$, implies

$$\frac{f_i^S}{g_i^S} \geq \frac{f_i^S - c_i}{g_i^S}.$$

The result follows by maximizing over $S \subseteq N$ in each side of latter inequality, which preserves it, and using the definitions of $\nu^{\mathrm{AT}}_{(1,i)}$ and $\nu^{\mathrm{AT}}_{(0,i)}$ in (9). □

We next set out to use DP to show that the indices defined in (9) are indeed the project's continuation and switching MPIs in Definition 3.2. We will use the *Bellman equations* on the optimal value function $v^*_{(a^-,i)}(\nu)$ for ν-wage problem (4): for $i \in N$,

$$v^*_{(a^-,i)}(\nu) = \max\left\{\beta v^*_{(0,i)}(\nu),\, R_i - (1-a^-)c_i - \nu + \beta \sum_{j\in N} p_{ij} v^*_{(1,j)}(\nu)\right\}. \quad (10)$$

We start by establishing that it suffices to restrict attention to active-set family $\widehat{\mathscr{F}}$.

Lemma 3.6. *If it is optimal to rest the project in $(1,i)$, then it is optimal to rest it in $(0,i)$.*

Proof. The result follows from the implication

$$\beta v^*_{(0,i)}(\nu) \geq R_i - \nu + \beta \sum_{j\in N} p_{ij} v^*_{(1,j)}(\nu)$$
$$\Longrightarrow \quad \beta v^*_{(0,i)}(\nu) \geq -c_i + R_i - \nu + \beta \sum_{j\in N} p_{ij} v^*_{(1,j)}(\nu),$$

which holds because $c_i \geq 0$ by Assumption 3.1 (recall that $d_i \equiv 0$). □

We next give the main result of this section (its proof uses results from §3.5).

Theorem 3.7. *The reformulated restless project is $\widehat{\mathscr{F}}$-indexable, with MPI given by* (9).

Proof. We first show that it is optimal to rest the project at $(1,i)$ iff $\nu \geq \nu^{\mathrm{AT}}_{(1,i)}$. We have:

it is optimal to rest at $(1,i)$

$$\Longleftrightarrow 0 \geq \max_{S_0\subseteq S_1\subseteq N:\, i\in S_1} f^{S_0\oplus S_1}_{(1,i)} - \nu g^{S_0\oplus S_1}_{(1,i)}$$
$$\Longleftrightarrow \nu \geq \max_{S_0\subseteq S_1\subseteq N:\, i\in S_1} \frac{f^{S_0\oplus S_1}_{(1,i)}}{g^{S_0\oplus S_1}_{(1,i)}}$$
$$\Longleftrightarrow \nu \geq \max_{S_1\subseteq N:\, i\in S_1} \frac{f_i^{S_1}}{g_i^{S_1}} = \nu^{\mathrm{AT}}_{(1,i)},$$

where we have used Lemma 3.6 and the following relations in Lemmas 4.2(b) and 4.6(b) below: $g^{S_0\oplus S_1}_{(1,i)} = g_i^{S_1}$ and $f^{S_0\oplus S_1}_{(1,i)} = f_i^{S_1}$ for $S_0 \subseteq S_1 \subseteq N$ and $i \in S_1$.

It remains to show that it is optimal to rest the project at $(0,i)$ iff $\nu \geq \nu^{\mathrm{AT}}_{(0,i)}$. We have:

it is optimal to rest at $(0,i)$

$$\Longleftrightarrow 0 \geq \max_{S_0\subseteq S_1\subseteq N:\, i\in S_0} f^{S_0\oplus S_1}_{(0,i)} - \nu g^{S_0\oplus S_1}_{(0,i)}$$
$$\Longleftrightarrow \nu \geq \max_{S_0\subseteq S_1\subseteq N:\, i\in S_0} \frac{f^{S_0\oplus S_1}_{(0,i)}}{g^{S_0\oplus S_1}_{(0,i)}}$$
$$\Longleftrightarrow \nu \geq \max_{S_1\subseteq N:\, i\in S_0} \frac{f_i^{S_1} - c_i}{g_i^{S_1}} = \nu^{\mathrm{AT}}_{(0,i)},$$

where we have used Lemma 3.6 and the following relations in Lemmas 4.2(c) and 4.6(c) below: $g^{S_0\oplus S_1}_{(0,i)} = g_i^{S_1}$ and $f^{S_0\oplus S_1}_{(0,i)} = f_i^{S_1} - c_i$ for $i \in S_0 \subseteq S_1 \subseteq N$.

The above establishes that the project is indexable, with continuation and switching MPIs given by (9). The fact that it is $\widehat{\mathscr{F}}$-indexable then follows from Lemma 3.5. □

3.4. On a Property of Optimal Policies

The main result of Asawa and Teneketzis (1996) establishes that their continuation and switching indices partially characterize optimal policies for the MABPSC.

Theorem 3.8 (Asawa and Teneketzis 1996). *An optimal policy for the MABPSC has the property that decisions about which project to engage need to be made only at stopping times that achieve the appropriate* (*continuation or switching*) *index.*

In Asawa and Teneketzis (1996, §3.C), it is specified how to construct the continuation (stopping) sets achieving the stopping times in Theorem 3.8. The next result revisits such an issue in light of the above MPI characterization of the indices in (9). It follows immediately from the previous section's results, and hence we omit the proof.

Proposition 3.9. (a) *The minimal continuation set achieving the continuation index $\nu^*_{(1,i)}$ in* (9) *is*

$$S_1^*(\nu^*_{(1,i)}) = \{j \in N\colon \nu^*_{(1,j)} > \nu^*_{(1,i)}\}.$$

(b) *The minimal continuation set achieving the switching index $\nu^*_{(0,i)}$ in* (9) *is*

$$S_1^*(\nu^*_{(0,i)}) = \{j \in N\colon \nu^*_{(1,j)} > \nu^*_{(0,i)}\}.$$

Note that Proposition 3.9 shows that the MPI-based policy for the MABPSC, although generally not optimal, satisfies the property of optimal policies in Theorem 3.8.

3.5. PCL-Indexability and the MPI Algorithm

We next outline the approach we will deploy to compute the MPI, based on the *adaptive-greedy* MPI algorithm introduced in Niño-Mora (2001, 2002). Such an algorithm computes the MPI of restless bandits that satisfy a strong form of indexability termed *PCL-indexability*.

Given $a \in \{0,1\}$ and $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$, denote by $\langle a, S_0 \oplus S_1\rangle$ the policy that takes action a in the initial period and adopts the $S_0 \oplus S_1$-*active policy* thereafter. For each augmented state (a^-, i) and active set $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$, define the *marginal work measure*

$$w_{(a^-,i)}^{S_0\oplus S_1} \triangleq g_{(a^-,i)}^{\langle 1, S_0\oplus S_1\rangle} - g_{(a^-,i)}^{\langle 0, S_0\oplus S_1\rangle}, \tag{11}$$

along with the *marginal reward measure*

$$r_{(a^-,i)}^{S_0\oplus S_1} \triangleq f_{(a^-,i)}^{\langle 1, S_0\oplus S_1\rangle} - f_{(a^-,i)}^{\langle 0, S_0\oplus S_1\rangle} \tag{12}$$

and the *marginal productivity measure*

$$\nu_{(a^-,i)}^{S_0\oplus S_1} \triangleq \frac{r_{(a^-,i)}^{S_0\oplus S_1}}{w_{(a^-,i)}^{S_0\oplus S_1}}. \tag{13}$$

We will see later (cf. Proposition 4.4) that $w_{(a^-,i)}^{S_0\oplus S_1} > 0$, ensuring that $\nu_{(a^-,i)}^{S_0\oplus S_1}$ is well defined.

We will use the adaptive-greedy MPI algorithm $\mathrm{AG}_{\widehat{\mathcal{F}}}$ in Table 1, where $n \triangleq |N|$ is the number of project states in the original formulation. The algorithm's input consists of the project's parameters (which are not explicitly stated), along with the given active-set family $\widehat{\mathcal{F}}$. It returns as output a string $\{(a_k^-, i_k)\}_{k=1}^{2n}$ of distinct augmented states spanning $\widehat{N}$, which satisfy $\widehat{S}^k \triangleq \{(a_1^-, i_1), \ldots, (a_k^-, i_k)\} \in \widehat{\mathcal{F}}$ for $1 \le k \le 2n$, along with corresponding index values $\{\nu^*_{(a_k^-, i_k)}\}_{k=1}^{2n}$. Note that each active set $\widehat{S}^k$ is of the form $S_0 \oplus S_1$ for certain $S_0 \subseteq S_1 \subseteq N$. We use in this algorithm the compact form $\widehat{S}^k$ for notational convenience.

Definition 3.10 (PCL($\widehat{\mathcal{F}}$)-Indexability). We say the project is *PCL($\widehat{\mathcal{F}}$)-indexable* if:

(i) Positive marginal workloads: $w_{(a^-,i)}^{S_0\oplus S_1} > 0$ for $(a^-, \ i) \in \widehat{N}$ and $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$.

(ii) Monotone nonincreasing index: algorithm $\mathrm{AG}_{\widehat{\mathcal{F}}}$'s output satisfies

$$\nu^*_{(a_1^-, i_1)} \ge \nu^*_{(a_2^-, i_2)} \ge \cdots \ge \nu^*_{(a_{2n}^-, i_{2n})}.$$

Table 1 **Version 1 of MPI Algorithm** $\mathrm{AG}_{\widehat{\mathcal{F}}}$

ALGORITHM $\mathrm{AG}_{\widehat{\mathcal{F}}}$:
Output: $\{(a_k^-, i_k), \nu^*_{(a_k^-, i_k)}\}_{k=1}^{2n}$
$\widehat{S}^0 := \varnothing$
for $k := 1$ **to** $2n$ **do**
  pick $(a_k^-, i_k) \in \arg\max\{\nu_{(a^-,i)}^{\widehat{S}^{k-1}} : (a^-, i) \in \widehat{N}\setminus \widehat{S}^{k-1}, \widehat{S}^{k-1} \cup \{(a^-, i)\} \in \widehat{\mathcal{F}}\}$
  $\nu^*_{(a_k^-, i_k)} := \nu^{\widehat{S}^{k-1}}_{(a_k^-, i_k)}$; $\widehat{S}^k := \widehat{S}^{k-1} \cup \{(a_k^-, i_k)\}$
end {for}

The following result is established in Niño-Mora (2001, Corollary 2) and, in a more general setting whose formulation we adopt herein, in Niño-Mora (2002, Theorem 6.3).

Theorem 3.11. *A PCL($\widehat{\mathcal{F}}$)-indexable restless project is $\widehat{\mathcal{F}}$-indexable, and the index $\nu^*_{(a^-, i)}$ computed by algorithm $\mathrm{AG}_{\widehat{\mathcal{F}}}$ is its MPI.*

While $\widehat{\mathcal{F}}$-indexability was already established in Theorem 3.7 above, we will invoke Theorem 3.11 in §4.4 to ensure the validity of algorithm $\mathrm{AG}_{\widehat{\mathcal{F}}}$ to compute the MPI. Note that ties for picking the (a_k^-, i_k) in algorithm $\mathrm{AG}_{\widehat{\mathcal{F}}}$ can be broken arbitrarily.

3.6. Version 2 of the MPI Algorithm

The MPI algorithm in Table 1 (Version 1) is readily reformulated into the more explicit Version 2 in Table 2, which uses the definition of $\widehat{\mathcal{F}}$ in (6). We use in this and later versions a more convenient algorithm-like notation, writing, e.g., $\nu_{(0,j)}^{S_0^{k_0-1}\oplus S_1^{k_1-1}}$ as $\nu_{(0,j)}^{(k_0-1,k_1-1)}$. Note that the active sets constructed in both versions along with respective counters k and k_0, k_1 are related by $\widehat{S}^{k-1} = S_0^{k_0-1} \oplus S_1^{k_1-1}$ with $k = k_0 + k_1 - 1$ and $k_0 \le k_1$.

Such a Version 2 decouples its output into the two augmented-state strings $\{(0, i_0^{k_0})\}_{k_0=1}^n$ and $\{(1, i_1^{k_1})\}_{k_1=1}^n$, with $N = \{i_0^1, \ldots, i_0^n\} = \{i_1^1, \ldots, i_1^n\}$, along with corresponding switching and continuation index values. The active sets $S_0^{k_0}$ and $S_1^{k_1}$ in the algorithm are given by $S_0^{k_0} = \{i_0^1, \ldots, i_0^{k_0}\}$ and $S_1^{k_1} = \{i_1^1, \ldots, i_1^{k_1}\}$, and satisfy $S_0^{k_0} \subset S_1^{k_1}$ for $1 \le k_0 < k_1 \le n$.

We will later show that Version 2 can still be substantially simplified, decoupling the computation of the continuation and switching MPIs into two algorithms. Yet to achieve such simplifications, we will need the PCL-indexability analysis carried out in the next section.

Table 2 **Version 2 of MPI Algorithm** $\mathrm{AG}_{\widehat{\mathcal{F}}}$

ALGORITHM $\mathrm{AG}_{\widehat{\mathcal{F}}}$:
Output: $\left\{(0, i_0^{k_0}), \nu^*_{(0, i_0^{k_0})}\right\}_{k_0=1}^n$, $\left\{(1, i_1^{k_1}), \nu^*_{(1, i_1^{k_1})}\right\}_{k_1=1}^n$
$S_0^0 := \varnothing$; $S_1^0 := \varnothing$; $k_0 := 1$; $k_1 := 1$
while $k_0 + k_1 \le 2n + 1$ **do**
  if $k_1 \le n$ **pick** $j_1^{\max} \in \arg\max\{\nu_{(1,j)}^{(k_0-1,k_1-1)} : j \in N \setminus S_1^{k_1-1}\}$
  if $k_0 < k_1$ **pick** $j_0^{\max} \in \arg\max\{\nu_{(0,j)}^{(k_0-1,k_1-1)} : j \in S_1^{k_1-1} \setminus S_0^{k_0-1}\}$
  if $k_1 = n+1$ **or** $\left\{k_0 < k_1 \le n \text{ and } \nu_{(1, j_1^{\max})}^{(k_0-1,k_1-1)} < \nu_{(0, j_0^{\max})}^{(k_0-1,k_1-1)}\right\}$
    $i_0^{k_0} := j_0^{\max}$; $\nu^*_{(0, i_0^{k_0})} := \nu^{(k_0-1,k_1-1)}_{(0, i_0^{k_0})}$; $S_0^{k_0} := S_0^{k_0-1} \cup \{i_0^{k_0}\}$; $k_0 := k_0 + 1$
  else
    $i_1^{k_1} := j_1^{\max}$; $\nu^*_{(1, i_1^{k_1})} := \nu^{(k_0-1,k_1-1)}_{(1, i_1^{k_1})}$; $S_1^{k_1} := S_1^{k_1-1} \cup \{i_1^{k_1}\}$; $k_1 := k_1 + 1$
  end {if}
end {while}

4. PCL-Indexability Analysis

We set out in this section to carry out a PCL-indexability analysis of a single project with switching costs as above, in its restless reformulation.

4.1. Work and Marginal Work Measures

We start by addressing calculation of work and marginal work measures $g^{S_0 \oplus S_1}_{(a^-,i)}$ and $w^{S_0 \oplus S_1}_{(a^-,i)}$ for $(a^-, i) \in \widehat{N}$ and $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$. We will show that they are closely related to corresponding measures g_i^S and w_i^S for the underlying nonrestless project, where stationary deterministic policies are represented by their active sets $S \subseteq N$.

For each $S \subseteq N$, work measures g_i^S are characterized by the evaluation equations

$$g_i^S = \begin{cases} 1 + \beta \sum_{j \in S} p_{ij} g_j^S & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Note that (14) can be reduced to a linear system with coefficient matrix $\mathbf{I}_S - \beta \mathbf{P}_{SS}$, where $\mathbf{I}_S$ is the identity matrix indexed by S and $\mathbf{P}_{SS} \triangleq (p_{ij})_{i,j \in S}$. Its solution is unique because matrix theory ensures that a matrix of the form $\mathbf{I}_S - \beta \mathbf{P}_{SS}$ is invertible if $\mathbf{P}_{SS}$ is substochastic (i.e., it has nonnegative elements and its rows add up to at most unity) and $0 < \beta < 1$.

Further, the marginal work measure w_i^S is evaluated by

$$w_i^S \triangleq g_i^{\langle 1, S \rangle} - g_i^{\langle 0, S \rangle} = 1 + \beta \sum_{j \in N} p_{ij} g_j^S - \beta g_i^S$$
$$= \begin{cases} (1-\beta) g_i^S & \text{if } i \in S, \\ 1 + \beta \sum_{j \in S} p_{ij} g_j^S & \text{otherwise.} \end{cases} \tag{15}$$

Note that (15) implies

$$w_i^S > 0, \quad i \in N. \tag{16}$$

We now return to the project's restless reformulation. The following result gives the evaluation equations for work measure $g^{S_0 \oplus S_1}_{(a^-,i)}$, for a given active set $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$.

Lemma 4.1.

$$g^{S_0 \oplus S_1}_{(a^-,i)} = \begin{cases} 1 + \beta \sum_{j \in N} p_{ij} g^{S_0 \oplus S_1}_{(1,j)} & \textit{if } i \in S_{a^-}, \\ \beta g^{S_0 \oplus S_1}_{(0,i)} & \textit{otherwise.} \end{cases}$$

The next result represents work measure $g^{S_0 \oplus S_1}_{(a^-,i)}$ in terms of the g_i^S.

Lemma 4.2. *For* $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$:
(a) $g^{S_0 \oplus S_1}_{(a^-,i)} = g_i^{S_1} = 0$ *for* $a^- \in \{0, 1\}, i \in N \setminus S_1$.
(b) $g^{S_0 \oplus S_1}_{(1,i)} = g_i^{S_1}$ *for* $i \in S_1$.
(c) $g^{S_0 \oplus S_1}_{(0,i)} = g_i^{S_1}$ *for* $i \in S_0$.
(d) $g^{S_0 \oplus S_1}_{(0,i)} = 0$ *for* $i \in S_1 \setminus S_0$.

Proof. (a) This part follows immediately from the definition of policy $S_0 \oplus S_1$.

(b) For $i \in S_1$, we can write

$$g^{S_0 \oplus S_1}_{(1,i)} = 1 + \beta \sum_{j \in S_1} p_{ij} g^{S_0 \oplus S_1}_{(1,j)} + \beta \sum_{j \in N \setminus S_1} p_{ij} g^{S_0 \oplus S_1}_{(1,j)}$$
$$= 1 + \beta \sum_{j \in S_1} p_{ij} g^{S_0 \oplus S_1}_{(1,j)},$$

where we have used Lemma 4.1 and part (a). Hence, the $g^{S_0 \oplus S_1}_{(1,i)}$ satisfy the evaluation equations in (14) for the $g_i^{S_1}$, for $i \in S_1$, which yields the result.

(c) For $i \in S_0$, we have

$$g^{S_0 \oplus S_1}_{(0,i)} = 1 + \beta \sum_{j \in S_1} p_{ij} g^{S_0 \oplus S_1}_{(1,j)} + \beta \sum_{j \in N \setminus S_1} p_{ij} g^{S_0 \oplus S_1}_{(1,j)}$$
$$= g^{S_0 \oplus S_1}_{(1,i)} = g_i^{S_1},$$

where we have used Lemma 4.1, part (b) and the relation $S_0 \subseteq S_1$.

(d) This part follows immediately from the definition of policy $S_0 \oplus S_1$. □

Regarding $w^{S_0 \oplus S_1}_{(a^-,i)}$, we draw on (11) and Lemma 4.1 to obtain

$$w^{S_0 \oplus S_1}_{(0,i)} = w^{S_0 \oplus S_1}_{(1,i)} = 1 + \beta \sum_{j \in N} p_{ij} g^{S_0 \oplus S_1}_{(1,j)} - \beta g^{S_0 \oplus S_1}_{(0,i)}. \tag{17}$$

The following result represents $w^{S_0 \oplus S_1}_{(a^-,i)}$ in terms of the w_i^S in (15).

Lemma 4.3. *For* $a^- \in \{0, 1\}$, $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$:
(a) $w^{S_0 \oplus S_1}_{(a^-,i)} = w_i^{S_1}$ *for* $i \in S_0 \cup N \setminus S_1$.
(b) $w^{S_0 \oplus S_1}_{(a^-,i)} = w_i^{S_1}/(1-\beta)$ *for* $i \in S_1 \setminus S_0$.

Proof. (a) We can write, for $i \in S_0 \cup N \setminus S_1$,

$$w^{S_0 \oplus S_1}_{(a^-,i)} = 1 + \beta \sum_{j \in N} p_{ij} g^{S_0 \oplus S_1}_{(1,j)} - \beta g^{S_0 \oplus S_1}_{(0,i)}$$
$$= 1 + \beta \sum_{j \in S_1} p_{ij} g_j^{S_1} - \beta g_i^{S_1} = w_i^{S_1},$$

where we have used (17), Lemma 4.2(a, b, c) and (15).

(b) We can write, for $i \in S_1 \setminus S_0$,

$$w^{S_0 \oplus S_1}_{(a^-,i)} = 1 + \beta \sum_{j \in N} p_{ij} g^{S_0 \oplus S_1}_{(1,j)} - \beta g^{S_0 \oplus S_1}_{(0,i)}$$
$$= 1 + \beta \sum_{j \in S_1} p_{ij} g_j^{S_1} = w_i^{S_1} + \beta g_i^{S_1} = w_i^{S_1}/(1-\beta),$$

where we have used (17), Lemma 4.2(a, b, d) and (15). □

From the above, we obtain the following result (cf. Definition 3.10(i)).

Proposition 4.4. $w^{S_0 \oplus S_1}_{(a^-,i)} > 0$, *for* $(a^-, i) \in \widehat{N}$, $S_0 \oplus S_1 \in \widehat{\mathcal{F}}$.

Proof. The result follows immediately from (16) via Lemma 4.3. □

4.2. Reward and Marginal Reward Measures

We next address the calculation of measures $f^{S_0 \oplus S_1}_{(a^-,\, i)}$ and $r^{S_0 \oplus S_1}_{(a^-,\, i)}$, relating them to their counterparts f^S_i and r^S_i for the underlying nonrestless project with no start-up costs.

For each active set $S \subseteq N$, the reward measure f^S_i is characterized by the equations

$$f^S_i = \begin{cases} R_i + \beta \sum_{j \in S} p_{ij} f^S_j & \text{if } i \in S \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

whose solution is unique, while the marginal reward measure r^S_i is given by

$$\begin{aligned} r^S_i &\triangleq f^{\langle 1,\, S\rangle}_i - f^{\langle 0,\, S\rangle}_i = R_i + \beta \sum_{j \in S} p_{ij} f^S_j - \beta f^S_i \\ &= \begin{cases} (1-\beta) f^S_i & \text{if } i \in S, \\ R_i + \beta \sum_{j \in S} p_{ij} f^S_j & \text{otherwise.} \end{cases} \end{aligned} \tag{19}$$

Returning to the restless formulation, the next result gives the evaluation equations for reward measures $f^{S_0 \oplus S_1}_{(a^-,\, i)}$, for a given active set $S_0 \oplus S_1 \in \widehat{\mathscr{F}}$. Recall the notation in (1).

Lemma 4.5.

$$f^{S_0 \oplus S_1}_{(a^-,\, i)} = \begin{cases} R_i - (1-a^-) c_i + \beta \sum_{j \in N} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} & \textit{if } i \in S_{a^-}, \\ \beta f^{S_0 \oplus S_1}_{(0,\, i)} & \textit{otherwise.} \end{cases}$$

The next result represents reward measure $f^{S_0 \oplus S_1}_{(a^-,\, i)}$ in terms of the f^S_i.

Lemma 4.6. *For* $S_0 \oplus S_1 \in \widehat{\mathscr{F}}$:
(a) $f^{S_0 \oplus S_1}_{(a^-,\, i)} = 0 = f^{S_1}_i$ *for* $a^- \in \{0, 1\}$, $i \in N \setminus S_1$.
(b) $f^{S_0 \oplus S_1}_{(1,\, i)} = f^{S_1}_i$ *for* $i \in S_1$.
(c) $f^{S_0 \oplus S_1}_{(0,\, i)} = f^{S_1}_i - c_i$ *for* $i \in S_0$.
(d) $f^{S_0 \oplus S_1}_{(0,\, i)} = 0 = f^{S_0}_i$ *for* $i \in S_1 \setminus S_0$.

Proof. (a) This part follows immediately from the definition of policy $S_0 \oplus S_1$.
(b) We can write, for $i \in S_1$,

$$\begin{aligned} f^{S_0 \oplus S_1}_{(1,\, i)} &= R_i + \beta \sum_{j \in S_1} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} + \beta \sum_{j \in N \setminus S_1} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} \\ &= R_i + \beta \sum_{j \in S_1} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)}, \end{aligned}$$

where we have used Lemma 4.5 and part (a). Hence, the $f^{S_0 \oplus S_1}_{(1,\, i)}$, for $i \in S_1$, satisfy the evaluation equations in (18) for corresponding terms $f^{S_1}_i$, which yields the result.

(c) We can write, for $i \in S_0$,

$$\begin{aligned} f^{S_0 \oplus S_1}_{(0,\, i)} &= R_i - c_i + \beta \sum_{j \in S_1} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} + \beta \sum_{j \in N \setminus S_1} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} \\ &= f^{S_0 \oplus S_1}_{(1,\, i)} - c_i = f^{S_1}_i - c_i, \end{aligned}$$

where we have used $S_0 \subseteq S_1$ along with parts (a, b).
(d) This part follows immediately from the definition of policy $S_0 \oplus S_1$. □

Regarding marginal reward measure $r^{S_0 \oplus S_1}_{(a^-,\, i)}$, we draw on (12) and Lemma 4.5 to obtain

$$r^{S_0 \oplus S_1}_{(a^-,\, i)} = R_i - (1-a^-) c_i + \beta \sum_{j \in N} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} - \beta f^{S_0 \oplus S_1}_{(0,\, i)}. \tag{20}$$

The following result represents $r^{S_0 \oplus S_1}_{(a^-,\, i)}$ in terms of the r^S_i.

Lemma 4.7. *For* $S_0 \oplus S_1 \in \widehat{\mathscr{F}}$:
(a) $r^{S_0 \oplus S_1}_{(0,\, i)} = r^{S_0 \oplus S_1}_{(1,\, i)} - c_i$ *for* $i \in N$.
(b) $r^{S_0 \oplus S_1}_{(1,\, i)} = r^{S_1}_i$ *for* $i \in N \setminus S_1$.
(c) $r^{S_0 \oplus S_1}_{(1,\, i)} = r^{S_1}_i + \beta c_i$ *for* $i \in S_0$.
(d) $r^{S_0 \oplus S_1}_{(1,\, i)} = r^{S_1}_i / (1-\beta)$ *for* $i \in S_1 \setminus S_0$.

Proof. (a) This part follows immediately from (20).
(b) We can write, for $i \in N \setminus S_1$,

$$\begin{aligned} r^{S_0 \oplus S_1}_{(1,\, i)} &= R_i + \beta \sum_{j \in N} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} - f^{S_0 \oplus S_1}_{(1,\, i)} \\ &= R_i + \beta \sum_{j \in S_1} p_{ij} f^{S_1}_j - f^{S_1}_i = r^{S_1}_i, \end{aligned}$$

where we have used (20), Lemma 4.6(a, b), and (19).
(c) We can write, for $i \in S_0$,

$$\begin{aligned} r^{S_0 \oplus S_1}_{(1,\, i)} &= R_i + \beta \sum_{j \in N} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} - \beta f^{S_0 \oplus S_1}_{(0,\, i)} \\ &= R_i + \beta \sum_{j \in S_1} p_{ij} f^{S_1}_j - \beta (f^{S_1}_i - c_i) = r^{S_1}_i + \beta c_i, \end{aligned}$$

where we have used $S_0 \subseteq S_1$, (20), Lemma 4.6(a, b, c), and (19).
(d) We can write, for $i \in S_1 \setminus S_0$,

$$r^{S_0 \oplus S_1}_{(1,\, i)} = R_i + \beta \sum_{j \in N} p_{ij} f^{S_0 \oplus S_1}_{(1,\, j)} - \beta f^{S_0 \oplus S_1}_{(0,\, i)} = f^{S_1}_i = \frac{r^{S_1}_i}{1-\beta},$$

where we have used (20), Lemma 4.6(a, b, d), (18), and (19). This completes the proof. □

4.3. Marginal Productivity Measures

We continue to address calculation of marginal productivity measures $\nu^{S_0 \oplus S_1}_{(a^-,\, i)}$ in (13), relating them to their counterparts for the underlying nonrestless project, given by

$$\nu^S_i \triangleq \frac{r^S_i}{w^S_i}, \quad i \in N,\ S \subseteq N. \tag{21}$$

The next result represents the required $\nu^{S_0 \oplus S_1}_{(a^-,\, i)}$ in terms of the ν^S_i.

Table 3 **Version 3 of Algorithm** $\mathrm{AG}_{\widehat{\mathscr{F}}}$

ALGORITHM $\mathrm{AG}_{\widehat{\mathscr{F}}}$:
Output: $\Big\{(0, i_0^{k_0}), \nu^*_{(0, i_0^{k_0})}\Big\}_{k_0=1}^{n}, \quad \Big\{i_1^{k_1}, \nu^*_{i_1^{k_1}}\Big\}_{k_1=1}^{n}$

$S_0^0 := \varnothing; \quad S_1^0 := \varnothing; \quad k_0 := 1; \quad k_1 := 1$
while $k_0 + k_1 \le 2n+1$ **do**
  if $k_1 \le n$ **pick** $j_1^{\max} \in \arg\max\{\nu_j^{(k_1-1)} : j \in N \setminus S_1^{k_1-1}\}$
  $\nu_{(0,j)}^{(0,k_1-1)} := \nu_j^{(k_1-1)} - (1-\beta)c_j/w_j^{(k_1-1)}, \quad j \in S_1^{k_1-1} \setminus S_0^{k_0-1}$
  if $k_0 < k_1$ **pick** $j_0^{\max} \in \arg\max\{\nu_{(0,j)}^{(0,k_1-1)} : j \in S_1^{k_1-1} \setminus S_0^{k_0-1}\}$
  if $k_1 = n+1$ **or** $\Big\{k_0 < k_1 \le n$ **and** $\nu_{j_1^{\max}}^{(k_1-1)} < \nu_{(0, j_0^{\max})}^{(0,k_1-1)}\Big\}$
    $i_0^{k_0} := j_0^{\max}; \quad \nu^*_{(0, i_0^{k_0})} := \nu^{(0,k_1-1)}_{(0, i_0^{k_0})}; \quad S_0^{k_0} := S_0^{k_0-1} \cup \{i_0^{k_0}\}; \quad k_0 := k_0 + 1$
  else
    $i_1^{k_1} := j_1^{\max}; \quad \nu^*_{i_1^{k_1}} := \nu^{(k_1-1)}_{i_1^{k_1}}; \quad S_1^{k_1} := S_1^{k_1-1} \cup \{i_1^{k_1}\}; \quad k_1 := k_1 + 1$
  end {if}
end {while}

Lemma 4.8. *For* $S_0 \oplus S_1 \in \widehat{\mathscr{F}}$:
(a) $\nu_{(0,i)}^{S_0 \oplus S_1} = \nu_{(1,i)}^{S_0 \oplus S_1} - c_i/w_{(1,i)}^{S_0 \oplus S_1}$ *for* $i \in N$.
(b) $\nu_{(1,i)}^{S_0 \oplus S_1} = \nu_i^{S_1} = \nu_{(1,i)}^{\varnothing \oplus S_1}$ *for* $i \in N \setminus S_0$.
(c) $\nu_{(1,i)}^{S_0 \oplus S_1} = \nu_i^{S_1} + \beta c_i / w_i^{S_1}$ *for* $i \in S_0$.
(d) $\nu_{(0,i)}^{S_0 \oplus S_1} = \nu_i^{S_1} - (1-\beta)c_i/w_i^{S_1} = \nu_{(0,i)}^{\varnothing \oplus S_1}$, $i \in S_1 \setminus S_0$.

Proof. All parts follow immediately from (13), (21), Lemma 4.3, and Lemma 4.7. □

The above results allow us to reformulate Version 2 of the MPI algorithm into the even more explicit Version 3 shown in Table 3. As before, we write, e.g., $\nu_{(0,j)}^{S_{k_0}^0 \oplus S_{k_1}^1}$ as $\nu_{(0,j)}^{(k_0,k_1)}$, and $\nu_j^{S_{k_1}}$ as $\nu_j^{(k_1)}$. Note that in Version 3, we use $\nu_{(0,j)}^{(0,k_1-1)}$ (which denotes $\nu_{(0,j)}^{S_0 \oplus S_{k_1-1}}$) in place of $\nu_{(0,j)}^{(k_0-1,k_1-1)}$, drawing on Lemma 4.8(d). We do so for computational reasons because storage of quantities $\nu_{(0,j)}^{(0,k_1-1)}$ requires one less dimension than storage of the $\nu_{(0,j)}^{(k_0-1,k_1-1)}$.

4.4. Proving $PCL(\widehat{\mathscr{F}})$-Indexability

We are now ready to establish $PCL(\widehat{\mathscr{F}})$-indexability (cf. Theorem 3.11).

Theorem 4.9. *The reformulated restless project is* $PCL(\widehat{\mathscr{F}})$*-indexable.*

Proof. Definition 3.10(i) is established in Proposition 4.4. It remains to prove condition (ii), stating that the successive index values computed by algorithm $\mathrm{AG}_{\widehat{\mathscr{F}}}$ are nonincreasing. Referring to Version 1 of the algorithm, we will thus show that the kth and $(k+1)$th computed index values satisfy $\nu^*_{(a_k^-, i_k)} \ge \nu^*_{(a_{k+1}^-, i_{k+1})}$ for $1 \le k < 2n$. Now, letting $\widehat{S}^{k-1}$ and $\widehat{S}^k = \widehat{S}^{k-1} \cup \{(a_k^-, i_k)\}$ be as in Table 1, we use Niño-Mora (2002, Proposition 6.4(b, c)) to write

$$r_{(a^-,i)}^{\widehat{S}^k} - r_{(a^-,i)}^{\widehat{S}^{k-1}} = \frac{r_{(a_k^-,i_k)}^{\widehat{S}^{k-1}}}{w_{(a_k^-,i_k)}^{\widehat{S}^{k-1}}} \Big(w_{(a^-,i)}^{\widehat{S}^k} - w_{(a^-,i)}^{\widehat{S}^{k-1}}\Big), \quad (a^-, i) \in \widehat{N},$$

which is immediately reformulated using (13) and $\nu^*_{(a_k^-,i_k)} = \nu_{(a_k^-,i_k)}^{\widehat{S}^{k-1}}$ as

$$\nu_{(a^-,i)}^{\widehat{S}^k} = \nu^*_{(a_k^-,i_k)} - \frac{w_{(a^-,i)}^{\widehat{S}^{k-1}}}{w_{(a^-,i)}^{\widehat{S}^k}} \Big(\nu^*_{(a_k^-,i_k)} - \nu_{(a^-,i)}^{\widehat{S}^{k-1}}\Big), \quad (a^-, i) \in \widehat{N}.$$

Now, taking $(a^-, i) = (a_{k+1}^-, i_{k+1})$ in the latter identity and using Proposition 4.4, we obtain

$$\nu^*_{(a_k^-,i_k)} \ge \nu^*_{(a_{k+1}^-,i_{k+1})} \iff \nu^*_{(a_k^-,i_k)} \ge \nu_{(a_{k+1}^-,i_{k+1})}^{\widehat{S}^{k-1}},$$

which shows that it suffices to prove that $\nu^*_{(a_k^-,i_k)} \ge \nu_{(a_{k+1}^-,i_{k+1})}^{\widehat{S}^{k-1}}$.

For such a purpose, recall that the algorithm picks (a_k^-, i_k) and (a_{k+1}^-, i_{k+1}) so that

$$(a_k^-, i_k) \in \arg\max\big\{\nu_{(a^-,i)}^{\widehat{S}^{k-1}} : (a^-, i) \in N \setminus \widehat{S}^{k-1}, \; \widehat{S}^{k-1} \cup \{(a^-, i)\} \in \widehat{\mathscr{F}}\big\},$$
$$(a_{k+1}^-, i_{k+1}) \in \arg\max\big\{\nu_{(a^-,i)}^{\widehat{S}^k} : (a^-, i) \in N \setminus \widehat{S}^k, \; \widehat{S}^k \cup \{(a^-, i)\} \in \widehat{\mathscr{F}}\big\},$$

and hence there are two cases to consider. If $\widehat{S}^{k-1} \cup \{(a_{k+1}^-, i_{k+1})\} \in \widehat{\mathscr{F}}$, the above relations yield $\nu^*_{(a_k^-,i_k)} = \nu_{(a_k^-,i_k)}^{\widehat{S}^{k-1}} \ge \nu_{(a_{k+1}^-,i_{k+1})}^{\widehat{S}^{k-1}}$, as required. If $\widehat{S}^{k-1} \cup \{(a_{k+1}^-, i_{k+1})\} \notin \widehat{\mathscr{F}}$, because $\widehat{S}^k \cup \{(a_{k+1}^-, i_{k+1})\} \in \widehat{\mathscr{F}}$, the structure of $\widehat{\mathscr{F}}$ as defined in (6) implies that it must be $a_k^- = 1$ and $(a_{k+1}^-, i_{k+1}) = (0, i_k)$. We now use Lemma 4.8(a) along with the latter identities to write

$$\nu^*_{(a_k^-,i_k)} = \nu^*_{(1,i_k)} = \nu_{(1,i_k)}^{\widehat{S}^{k-1}} = \nu_{(0,i_k)}^{\widehat{S}^{k-1}} + c_i/w_{(1,i_k)}^{\widehat{S}^{k-1}} \ge \nu_{(0,i_k)}^{\widehat{S}^{k-1}} = \nu_{(a_{k+1}^-,i_{k+1})}^{\widehat{S}^{k-1}},$$

where the inequality follows from Assumption 3.1 and Proposition 4.4. □

5. Decoupled Computation of the MPI

We set out in this section to further simplify Version 3 of the MPI algorithm $\mathrm{AG}_{\widehat{\mathscr{F}}}$ by decoupling computation of the continuation and the switching MPIs in a two-stage scheme.

5.1. First Stage: Computing the Continuation Index

We start with the continuation index $\nu^*_{(1,i)}$, which is (cf. §3.3) the Gittins index ν_i^* of the project without switching costs. We will need additional quantities to feed the second-stage switching-index algorithm discussed in §5.2.

To compute the Gittins index and additional quantities, we use the algorithmic scheme AG^1 in Table 4. This is a variant of the algorithm of Varaiya et al. (1985), reformulated as in Niño-Mora (2007a). See

Table 4 **Gittins-Index Algorithmic Scheme** AG[1]

```
ALGORITHM AG^1:
Output: { i_1^{k_1}, ν*_{i_1^{k_1}}, (w_j^{(k_1)}, ν_j^{(k_1)}), j ∈ S_1^{k_1} }_{k_1=1}^n
set S_1^0 := ∅; compute {(w_i^{(0)}, ν_i^{(0)}): i ∈ N}
for k_1 := 1 to n do
   pick i_1^{k_1} ∈ arg max{ν_i^{(k_1−1)}: i ∈ N\S_1^{k_1−1}}
   ν*_{i_1^{k_1}} := ν_{i_1^{k_1}}^{(k_1−1)};   S_1^{k_1} := S_1^{k_1−1} ∪ {i_1^{k_1}}
   compute {(w_i^{(k_1)}, ν_i^{(k_1)}): i ∈ N}
end
```

the latter paper for implementations such as algorithm FP(1), which performs $(4/3)n^3 + O(n^2)$ arithmetic operations, or the *complete-pivoting algorithm* CP—which we have found to be faster in practice despite its worse operation count.

5.2. Second Stage: Computing the Switching Index

We next address computation of the switching index. Consider algorithm AG^0 in Table 5, which is fed the output of AG^1 and produces a sequence of states $i_0^{k_0}$ spanning N, along with nonincreasing index values $\nu^*_{(0,\, i_0^{k_0})}$. We can now give the main result of this paper.

Theorem 5.1. *Algorithm* AG^0 *computes the switching index* $\nu^*_{(0,\, i)}$.

Proof. The result follows by noticing that algorithm AG^0 is obtained from Version 3 of MPI algorithm $\mathrm{AG}_{\widehat{\mathcal{F}}}$ in Table 3 by decoupling the computation of the $\nu^*_{(0,\, i)}$ and the ν^*_i. □

Table 5 **Switching-Index Algorithm** AG^0

```
ALGORITHM AG^0:
Input: { i_1^{k_1}, ν*_{i_1^{k_1}}, (w_j^{(k_1)}, ν_j^{(k_1)}), j ∈ S_1^{k_1} }_{k_1=1}^n
Output: { i_0^{k_0}, ν*_{(0, i_0^{k_0})} }_{k_0=1}^n
ĉ_j := (1 − β)c_j, j ∈ N;   S_0^0 := ∅;   S_1^0 := ∅;   k_0 := 0
for k_1 := 1 to n do
  S_1^{k_1} := S_1^{k_1−1} ∪ {i_1^{k_1}};   AUGMENT_1 := false
  ν_{(0,j)}^{(0,k_1)} := ν_j^{(k_1)} − ĉ_j/w_j^{(k_1)},   j ∈ S_1^{k_1}\S_0^{k_0}
  while k_0 < k_1 and not(AUGMENT_1) do
    pick j_0^max ∈ arg max{ν_{(0,j)}^{(0,k_1)}: j ∈ S_1^{k_1}\S_0^{k_0}}
    if k_1 = n or ν*_{i_1^{k_1}} < ν_{(0, j_0^max)}^{(0,k_1)}
      i_0^{k_0+1} := j_0^max;   ν*_{(0, i_0^{k_0+1})} := ν_{(0, i_0^{k_0+1})}^{(0,k_1)}
      S_0^{k_0+1} := S_0^{k_0} ∪ {i_0^{k_0+1}};   k_0 := k_0 + 1
    else
      AUGMENT_1 := true
    end {if}
  end {while}
end {for}
```

We next assess the computational complexity of the switching-index algorithm.

Proposition 5.2. *Algorithm* AG^0 *performs at most* $n^2 + O(n)$ *arithmetic operations.*

Proof. The operation count is dominated by the statements

$$\nu^{(0,\, k_1)}_{(0,\, j)} := \nu^{(k_1)}_j - \hat{c}_j / w^{(k_1)}_j, \quad j \in S^{k_1}_1 \setminus S^{k_0}_0,$$

performing at most $2k_1$ operations. Adding up over $k_1 = 1, \ldots, n$ yields the result. □

We thus see that the switching index can be efficiently computed *an order of magnitude faster* than the continuation index. Note also that algorithm AG^0 involves handling matrices of size $n \times n$, in contrast to the matrices of size $2n \times 2n$ required by the AT scheme, which further yields substantial savings in expensive memory operations.

6. Index Dependence on Switching Costs

Next, we discuss some properties on the indices' dependence on switching costs, in the case $c_i \equiv c$ and $d_i \equiv d$. We write below $\nu^*_{(1,\, i)}(d)$—because it does not depend on c—and $\nu^*_{(0,\, i)}(c, d)$, and denote by ν^*_i the Gittins index of the underlying project with no switching costs.

Proposition 6.1. (a) $\nu^*_{(1,\, i)}(d) = \nu^*_i + (1-\beta)d$.

(b) *For large enough* $c+d$, $\nu^*_{(0,\, i)}(c, d) = \nu^N_i - (1-\beta)c$.

(c) $\nu^*_{(0,\, i)}(c, d)$ *is piecewise linear convex in* (c, d), *decreasing in* c, *and nonincreasing in* d.

Proof. (a) Note that $\nu^*_{(1,\, i)}(d)$ is the Gittins index of a project with modified rewards $\tilde{R}_j = R_j + (1-\beta)d$ (cf. §3.2). The effect of such an addition of a constant term to rewards is to increment the Gittins index by the same constant, which yields the result.

(b) Using the results in §§3.2 and 3.3, we obtain

$$\begin{aligned} \nu^*_{(0,\, i)}(c, d) &= \max_{S \subseteq N:\, i \in S} \frac{f^S_i - c - \{1 - (1-\beta) g^S_i\} d}{g^S_i} \\ &= (1-\beta)d + \max_{S \subseteq N:\, i \in S} \frac{f^S_i - (c+d)}{g^S_i}, \end{aligned} \quad (22)$$

where f^S_i is the reward measure of the underlying nonrestless project with rewards R_j—note that the corresponding reward measure with modified rewards $\tilde{R}_j$ as above is $f^S_i(d) = f^S_i + (1-\beta) d g^S_i$. The second identity in (22) implies that, for $c+d$ large enough, term $(c+d)/g^S_i$ becomes dominant, and hence the maximum value of the given expression is attained by maximizing the denominator g^S_i. The latter's maximum value is achieved by $S = N$, for which $g^N_i = 1/(1-\beta)$. Because $\nu^N_i = r^N_i / w^N_i = f^N_i / g^N_i$, this yields the result.

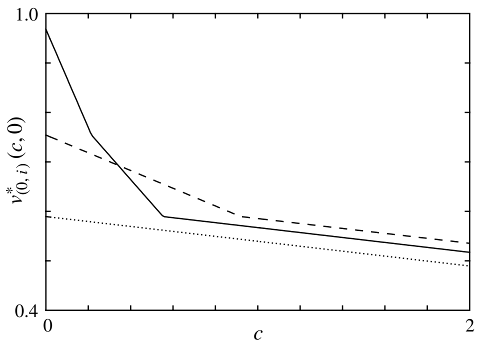

Figure 1 Dependence of Switching Index on Start-up Cost

(c) The first identity in (22) represents $\nu^*_{(0,i)}(c,d)$ as the maximum of linear functions in (c,d) that are decreasing in c and nonincreasing in d, which yields the result. □

Note that Proposition 6.1(a) shows that the incentive to stay on an active project increases linearly with its shutdown cost. Also, as $\beta \nearrow 1$, the dependence on switching costs vanishes because the switching index converges to the undiscounted Gittins index.

We next give two examples to illustrate the above results. The first concerns the project instance with state space $N = \{1,2,3\}$, start-up cost c,

$$d = 0, \quad \beta = 0.95, \quad \mathbf{R}^0 = \mathbf{0}, \quad \mathbf{R}^1 = \begin{bmatrix} 0.7221 \\ 0.9685 \\ 0.1557 \end{bmatrix}, \quad \text{and}$$

$$\mathbf{P} = \begin{bmatrix} 0.8061 & 0.1574 & 0.0365 \\ 0.1957 & 0.0067 & 0.7976 \\ 0.1378 & 0.5959 & 0.2663 \end{bmatrix}.$$

Figure 1 plots the switching index for each state versus c, consistently with Proposition 6.1(b, c), showing that the state ordering induced by the index can change as c varies.

In the second example, we let the shutdown cost d vary, taking $c = 0$. Figure 2 plots the indices for each state versus d. Again, the plots are consistent with Proposition 6.1.

7. Computational Experiments

We next report on a computational study based on the author's implementations of the results herein. The experiments were performed running MATLAB R2006b under Windows XP x64 in an HP xw9300 AMD Opteron 254 (2.8 GHz) workstation with 4 GB of memory.

The first experiment investigated the runtime performance of the decoupled index computation method. We made MATLAB generate a random project instance with start-up costs for each of the state-space sizes $n = 500, 1{,}000, \ldots, 5{,}000$. For each such n, MATLAB recorded the time to compute (i) the continuation index and additional quantities with algorithm FP(1) in Niño-Mora (2007a); (ii) the switching index by algorithm AG^0; and (iii) both indices via the AT scheme, using algorithm FP(0) in Niño-Mora (2007a).

The results are displayed in Figure 3. The left pane plots total runtimes in hours for computing both indices versus n, along with cubic least-squares (LS) fits, which are consistent with the theoretical $O(n^3)$ complexity. The dotted line corresponds to the AT scheme, while the solid line corresponds to our two-stage method. The results show that the latter consistently achieved about a fourfold speedup over the former.

The right pane plots runtimes, in *seconds*, for the switching-index algorithm versus n, along with a quadratic least-squares fit, which is consistent with the theoretical $O(n^2)$ complexity. The change of time scale demonstrates the order-of-magnitude runtime improvement.

In the following experiments, the optimal policy was computed for each instance solving the linear programming (LP) formulation of the DP equations using the CPLEX LP solver, interfaced with MATLAB via TOMLAB. The MPI and benchmark (Gittins index) policies were evaluated solving with MATLAB their evaluation equations.

The second experiment assessed how the performance of the MPI policy on two-project instances depends on a common constant start-up cost c and

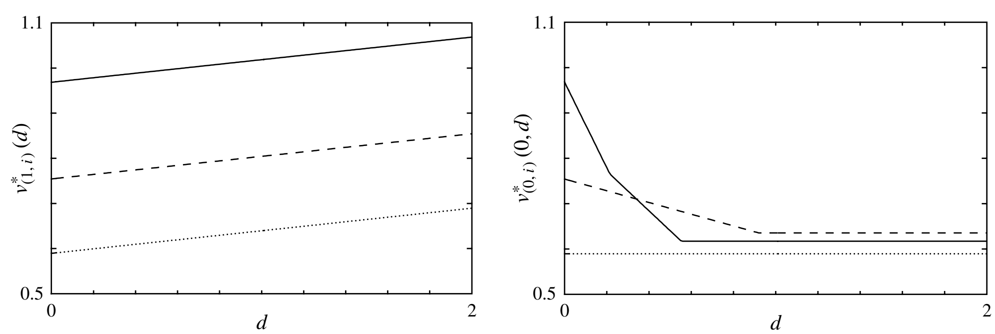

Figure 2 Dependence of Continuation and Switching Indices on Shutdown Cost

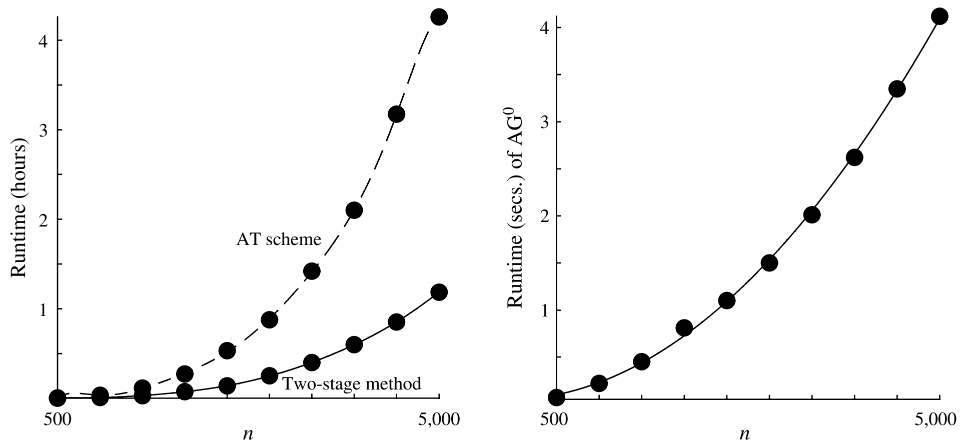

Figure 3 **Experiment 1: Runtimes of Index Algorithms**

discount factor β—shutdown costs are zero. A sample of 100 instances (where each project has state space $N = \{1, \ldots, n\}$ with $n = 10$) was randomly generated. In 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$[0,1]$). Passive rewards were set to zero. For each instance, $k = 1, \ldots, 100$ and pair $(c, \beta) \in [0,1] \times [0.2, 0.9]$—using a 0.1 grid—the optimal objective value $v^{(k),\,\mathrm{opt}}$ and the objective values of the MPI ($v^{(k),\,\mathrm{MPI}}$) and the benchmark ($v^{(k),\,\mathrm{bench}}$) policies were computed, along with the corresponding relative suboptimality gap of the MPI policy $\Delta^{(k),\,\mathrm{MPI}} = 100(v^{(k),\,\mathrm{opt}} - v^{(k),\,\mathrm{MPI}})/|v^{(k),\,\mathrm{opt}}|$, and the suboptimality-gap ratio of the MPI over the benchmark policy $\rho^{(k),\,\mathrm{MPI,\,bench}} = 100(v^{(k),\,\mathrm{opt}} - v^{(k),\,\mathrm{MPI}})/(v^{(k),\,\mathrm{opt}} - v^{(k),\,\mathrm{bench}})$. The latter were then averaged over the 100 instances for each (c, β) pair to obtain the average values Δ^{MPI} and $\rho^{\mathrm{MPI,\,bench}}$.

Objective values $v^{(k),\,\mathrm{opt}}$, $v^{(k),\,\mathrm{MPI}}$, and $v^{(k),\,\mathrm{bench}}$ were evaluated as follows. First, the corresponding *value functions* $v^{(k),\,\mathrm{opt}}_{((a_1^-, i_1), (a_2^-, i_2))}$, $v^{(k),\,\mathrm{MPI}}_{((a_1^-, i_1), (a_2^-, i_2))}$, and $v^{(k),\,\mathrm{bench}}_{((a_1^-, i_1), (a_2^-, i_2))}$ were computed as mentioned above, as vectors indexed by the initial joint augmented state $((a_1^-, i_1), (a_2^-, i_2))$. Then, the objective values were evaluated by taking both projects to be initially passive:

$$v^{(k),\,\pi} \triangleq \frac{1}{n^2} \sum_{i_1,\, i_2 \in N} v^{(k),\,\pi}_{((0,\, i_1),\, (0,\, i_2))}, \qquad \pi \in \{\mathrm{opt}, \mathrm{MPI}, \mathrm{bench}\}. \tag{23}$$

The left pane in Figure 4 plots Δ^{MPI} versus the start-up cost c for multiple β, using MATLAB's cubic interpolation. Such a gap starts at zero for $c = 0$—as the optimal policy is then recovered—then increases up to a maximum value below 0.25% at around $c \approx 0.3$, and then decreases, dropping again to zero at about $c \approx 0.9$ and staying there for larger values of c. Such a behavior is to be expected: for large enough c, the optimal policy will simply pick a project and hold to it. The plot in the right pane shows that Δ^{MPI} increases with β.

Figure 5 displays corresponding plots for the suboptimality-gap ratio $\rho^{\mathrm{MPI,\,bench}}$ of the MPI over the benchmark policy. The plot in the left pane shows that the average suboptimality gap for the MPI policy remains below 40% of the gap for the benchmark policy. Further, such a ratio takes the value zero both for $c = 0$ and for c large enough, because the MPI policy is then optimal. The plot in the right pane shows that the ratio increases with β.

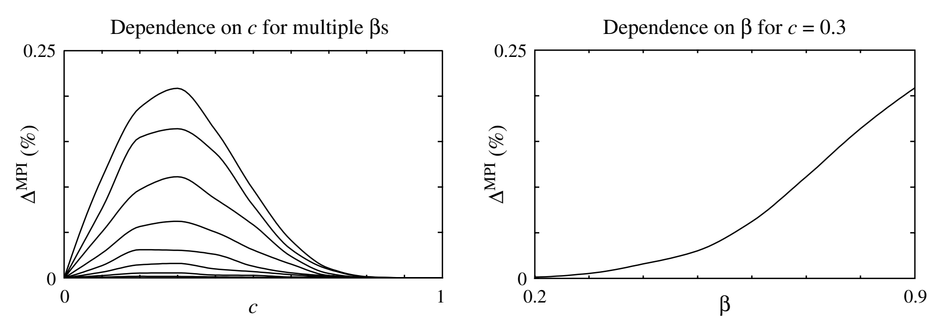

Figure 4 **Experiment 2: Average Relative Suboptimality Gap of MPI Policy**

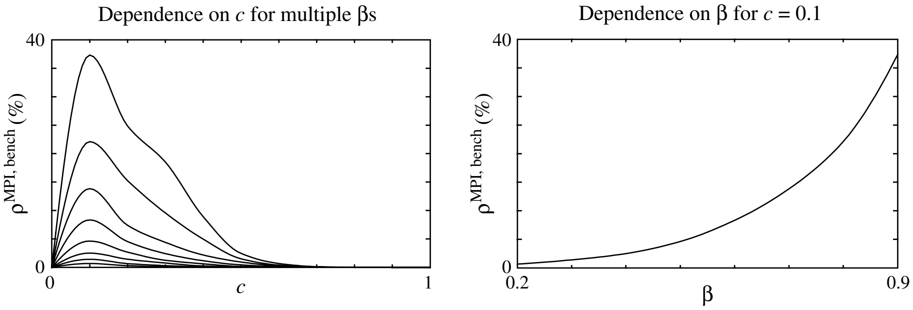

Figure 5 Experiment 2: Average Suboptimality-Gap Ratio of MPI over Benchmark Policy

The third experiment investigated the influence of a common shutdown cost d on the relative performance of the MPI policy—start-up costs were set to zero. It was identical in other respects to the second experiment. The resultant plots, shown in Figures 6 and 7, are similar to those of the previous experiment. Hence, the influence of a common shutdown cost on relative performance appears to be similar to that of a common start-up cost.

The fourth experiment assessed the effect of asymmetric constant start-up costs $(c_1, c_2) \in [0, 1]^2$ in two-project instances with no shutdown costs and $\beta = 0.9$. Figure 8 displays the resultant contour plots for Δ^{MPI} and $\rho^{\mathrm{MPI,\,bench}}$. We obtain that Δ^{MPI} reaches a maximum value of about 0.23%, vanishing as both start-up costs approach zero and as either grows large enough. As for $\rho^{\mathrm{MPI,\,bench}}$, it reaches a maximum value of about 39.4%, and vanishes as either start-up cost grows large enough and as both costs approach zero.

The fifth experiment evaluated the effect of state-dependent start-up costs in two-project instances with no shutdown costs as β varies. We had MATLAB generate independent Uniform[0, 1] state-dependent start-up costs for each instance. The left pane of Figure 9 plots Δ^{MPI} versus β, showing the former to be increasing in the latter while remaining well below 0.14%. The right pane plots $\rho^{\mathrm{MPI,bench}}$, which also increases with β, and remains below 4%.

The sixth and last experiment is a three-project counterpart of the second experiment, based on a random sample of 100 instances of three eight-state projects each. The results are shown in Figures 10 and 11, which are the counterparts of experiment 2's Figures 4 and 5. Comparison of Figures 4 and 10 reveals a slight performance degradation of the MPI policy in the latter, although Δ^{MPI} still remains quite small, below 0.3%. Comparison of Figures 5 and 11 reveals similar values for $\rho^{\mathrm{MPI,\,bench}}$.

8. Conclusions

This paper has demonstrated the tractability and usefulness of the index policy for bandits with switching costs based on the switching index introduced by Asawa and Teneketzis (1996). The approach was based on deploying the restless bandit indexation theory introduced by Whittle (1988) and developed by the author (cf. Niño-Mora 2007b). Niño-Mora (2007c) announces results on the model extension that also incorporates switching delays. The present analyses extend only in part to such a case, as the restless reformulation then yields semi-Markov projects that need not be PCL-indexable. Other extensions of interest concern restless bandits and particular models, as in Reiman and Wein (1998).

Acknowledgments

The author thanks the associate editor and two reviewers for suggestions that helped improve the paper. This work, which the author started as a faculty member at Universidad Pompeu Fabra (Barcelona), was supported

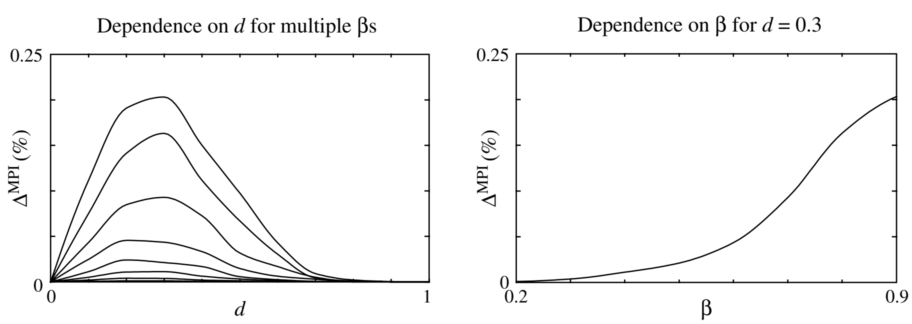

Figure 6 Experiment 3: Average Relative Suboptimality Gap of MPI Policy

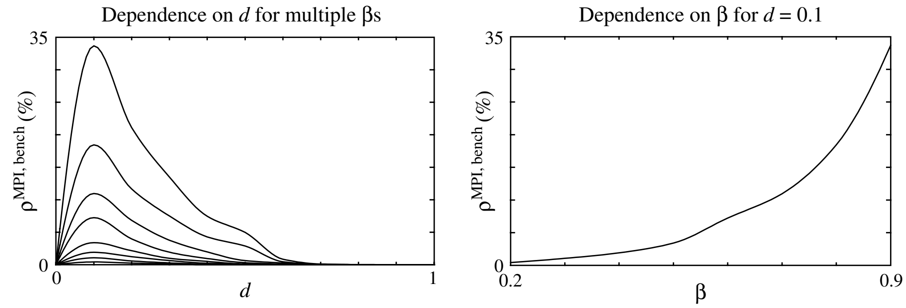

Figure 7 **Experiment 3: Average Suboptimality-Gap Ratio of MPI over Benchmark Policy**

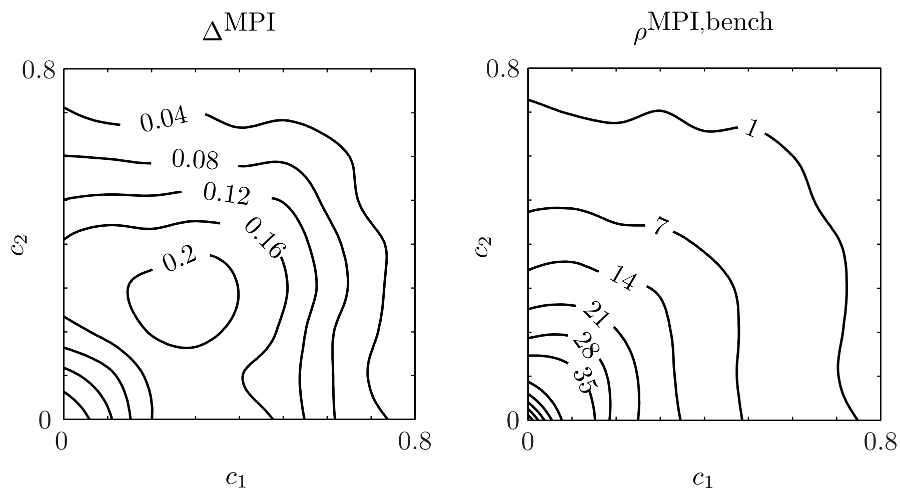

Figure 8 **Experiment 4: Average Relative (%) Performance of MPI Policy vs.** (c_1, c_2) **for** $\beta = 0.9$

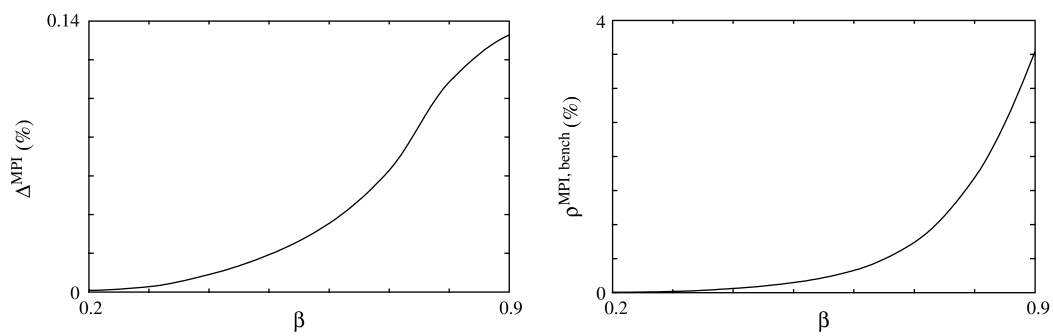

Figure 9 **Experiment 5: Average Performance of MPI Policy for State-Dependent Start-up Costs**

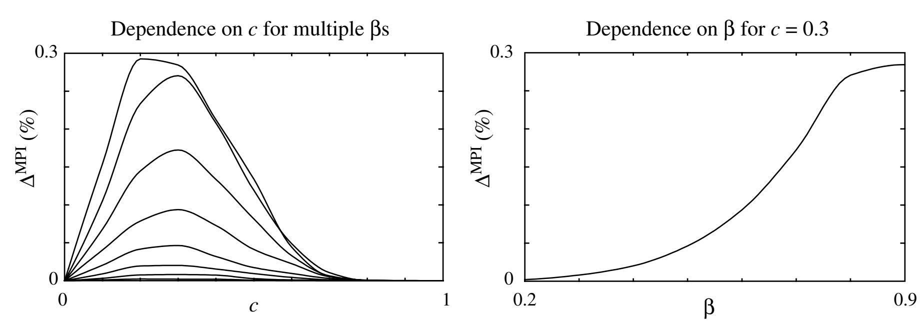

Figure 10 **Experiment 6: Counterpart of Figure 4 for Three-Project Instances**

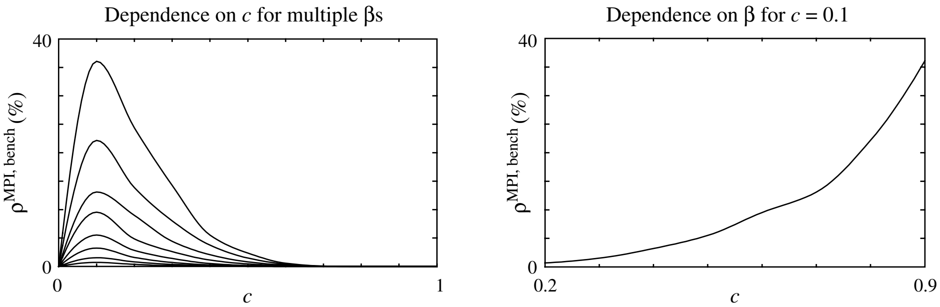

Figure 11 **Experiment 6: Counterpart of Figure 5 for Three-Project Instances**

by the Spanish Ministry of Education and Science under projects BEC2000-1027 and MTM2004-02334, a Ramón y Cajal Investigator Award and an I3 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. The author presented part of this work at the Schloss Dagstuhl Seminar on Algorithms for Optimization under Incomplete Information (Wadern, Germany, 2005).

References

Agrawal, R., M. V. Hegde, D. Teneketzis. 1988. Asymptotically efficient adaptive allocation rules for the multiarmed bandit problem with switching cost. *IEEE Trans. Automatic Control* **33** 899–906.

Asawa, M., D. Teneketzis. 1996. Multi-armed bandits with switching penalties. *IEEE Trans. Automatic Control* **41** 328–348.

Banks, J. S., R. K. Sundaram. 1994. Switching costs and the Gittins index. *Econometrica* **62** 687–694.

Dusonchet, F., M.-O. Hongler. 2003. Optimal hysteresis for a class of deterministic deteriorating two-armed bandit problem with switching costs. *Automatica* **39** 1947–1955.

Gittins, J. C. 1979. Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Statist. Soc. Ser. B* **41** 148–177.

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

Jun, T. 2004. A survey on the bandit problem with switching costs. *De Economist* **152** 513–541.

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

Niño-Mora, J. 2002. Dynamic allocation indices for restless projects and queueing admission control: A polyhedral approach. *Math. Programming* **93** 361–413.

Niño-Mora, J. 2006. 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.

Niño-Mora, J. 2007a. 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.

Niño-Mora, J. 2007b. Dynamic priority allocation via restless bandit marginal productivity indices (with discussion). *Top* **15** 161–198.

Niño-Mora, J. 2007c. Computing an index policy for bandits with switching penalties. *SMCtools '07: Proc. from the 2007 Workshop on Tools for Solving Structured Markov Chains*. ICST, Brussels.

Reiman, M. I., L. M. Wein. 1998. Dynamic scheduling of a two-class queue with setups. *Oper. Res.* **46** 532–547.

Van Oyen, M. P., D. Teneketzis. 1994. Optimal stochastic scheduling of forest networks with switching penalties. *Adv. Appl. Probab.* **26** 474–497.

Varaiya, P. P., J. C. Walrand, C. Buyukkoc. 1985. Extensions of the multiarmed bandit problem: The discounted case. *IEEE Trans. Automatic Control* **30** 426–439.

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