INFORMS Journal on Computing
Vol. 19, No. 4, Fall 2007, pp. 596–606
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1904 | 0596

DOI 10.1287/ijoc.1060.0206

A $(2/3)n^3$ Fast-Pivoting Algorithm for the Gittins Index and Optimal Stopping of a Markov Chain

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

This paper presents a new *fast-pivoting* algorithm that computes the n Gittins index values of an n-state bandit—in the discounted and undiscounted cases—by performing $(2/3)n^3 + O(n^2)$ arithmetic operations, thus attaining better complexity than previous algorithms and matching that of solving a corresponding linear-equation system by Gaussian elimination. The algorithm further applies to the problem of optimal stopping of a Markov chain, for which a novel Gittins-index solution approach is introduced. The algorithm draws on Gittins and Jones' (1974) index definition via calibration, on Kallenberg's (1986) proposal of using parametric linear programming, on Dantzig's simplex method, on the Varaiya et al. (1985) algorithm, and on the author's earlier work. This paper elucidates the structure of parametric simplex tableaux. Special structure is exploited to reduce the computational effort of pivot steps, decreasing the operation count by a factor of three relative to conventional pivoting, and by a factor of 3/2 relative to recent state-elimination algorithms. A computational study demonstrates significant time savings against alternative algorithms.

Key words: dynamic programming, Markov, finite state; Gittins index; bandits; optimal stopping; Markov chain; simplex method; analysis of algorithms; computational complexity
History: Accepted by Edward P. C. Kao, Area Editor for Computational Probability and Analysis; received November 2005; revised April 2006, August 2006; accepted August 2006. Published online in *Articles in Advance* September 21, 2007.

1. Introduction

Consider the following *optimal-stopping problem*. A Markov chain with state $X(t)$ evolves through the finite or countable state space N, according to transition probabilities p_{ij}. At each discrete time period $t \geq 0$, one must decide either to let the chain continue or to stop it, based on information so far. Continuing the chain when it occupies state i yields an immediate reward R_i, but a charge ν is incurred. If the chain is instead stopped, a terminal reward Q_i is earned. The R_i's and Q_i's are assumed to be bounded in the countable-state case. Rewards earned over time are discounted with factor $0 < \beta \leq 1$. We want to find an optimal *stopping rule* that maximizes the expected total discounted value of rewards earned minus charges incurred. This problem can be formulated as

$$\max_{\tau \geq 0} \mathbb{E}_i^\tau \left[\sum_{t=0}^{\tau-1} (R_{X(t)} - \nu)\beta^t + Q_{X(\tau)}\beta^\tau \right], \tag{1}$$

where $\mathbb{E}_i^\tau[\cdot]$ denotes expectation starting at i and, abusing notation, τ denotes both a stopping rule and the corresponding *stopping time* $0 \leq \tau \leq +\infty$.

An important application area for optimal stopping is computational finance, where several fundamental problems are formulated as special cases of (1), such as deciding the optimal time to sell a stock, or to exercise a perpetual American option. While prevailing models are continuous, there is emerging interest in more numerically tractable Markov-chain approximations, to which the results in this paper are applicable (see Duan et al. 2003).

Consider now the special zero-terminal-rewards case of problem (1), i.e., $Q_j \equiv 0$:

$$\max_{\tau \geq 0} \mathbb{E}_i^\tau \left[\sum_{t=0}^{\tau-1} (R_{X(t)} - \nu)\beta^t \right]. \tag{2}$$

Gittins and Jones (1974) showed—in the reformulation of problem (2) where ν plays the role of a subsidy per passive period—that one can attach to every state i a quantity ν_i^*, now termed the *Gittins index*, such that it is optimal to stop at state i if and only if $\nu_i^* \leq \nu$. Such an index result was first obtained by Bradt et al. (1956) in the classical setting of a Bayesian Bernoulli bandit. Whittle (1980) highlighted and exploited the alternative interpretation of the Gittins index as a critical *constant* terminal reward in an optimal-stopping problem—think of the reformulation of (2) where a terminal reward of $\nu/(1-\beta)$ is earned.

Gittins and Jones (1974) further proved that such an index furnishes an efficient solution to the *multiarmed-bandit problem*, concerning the optimal dynamic effort allocation to a collection of bandits, one of which must be engaged at a time: One should always engage a bandit with largest index. Gittins (1979) showed that

$$\nu_i^* = \max_{\tau>0} \frac{\mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1} R_{X(t)}\beta^t\right]}{\mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1}\beta^t\right]}, \tag{3}$$

so that ν_i^* represents the maximum rate of expected discounted reward per unit of expected discounted time that can be achieved starting at i. The corresponding *undiscounted Gittins index* is obtained by setting $\beta = 1$ above.

This paper presents two main contributions: (i) it is shown that problem (1) can be reduced to (2), and can thus be solved via the Gittins index; and (ii) for a bandit with a finite number n of states, an efficient algorithm is given based on the parametric version of Dantzig's simplex method that computes the Gittins index by performing $(2/3)n^3 + O(n^2)$ arithmetic operations, thus achieving better complexity than previous algorithms. It appears unlikely that such a complexity can be improved, as it matches that of solving a corresponding linear-equation system by Gaussian elimination.

Previous algorithmic approaches to (1) and (2) are reviewed next. The optimal stopping of a Markov chain is a fundamental problem that has been the subject of extensive research attention. See Sonin (1999). Conventional approaches are based on formulating the Bellman equations, which are then solved by the value-iteration, policy-iteration or linear-programming (LP) methods, where the required number of iterations is not clear a priori. In contrast, Sonin (1999) introduced a one-pass algorithm based on state-elimination ideas, which recursively solves an n-state problem in n iterations.

Regarding the index, Varaiya et al. (1985) introduced an n-step algorithm that performs $(1/3)n^4 + O(n^3)$ arithmetic operations. Both this and the Klimov (1974) index algorithm were elucidated in Bertsimas and Niño-Mora (1996) as special cases of an *adaptive-greedy algorithm* that solves the underlying polyhedral LP formulation.

Chen and Katehakis (1986) showed that the index of a *fixed state* can be computed by solving an LP problem with $n + 1$ variables and n constraints. Kallenberg (1986) proposed computing the index by solving a parametric LP problem as in Saaty and Gass (1954). Such an approach involves n pivot steps of the parametric version of Dantzig's simplex method, which requires $2n^3 + O(n^2)$ arithmetic operations.

More recently, Katta and Sethuraman (2004) and Sonin (2005) introduced index algorithms based on state elimination, which perform $n^3 + O(n^2)$ operations.

The new algorithm developed in this paper draws on Gittins (1979), Kallenberg (1986), Dantzig's simplex method, Varaiya et al. (1985), and Niño-Mora (2001, 2002, 2006). The structure of the parametric simplex tableau is elucidated. By exploiting special structure, the computational effort of pivot steps is reduced, decreasing the operation count by a factor of three relative to using conventional pivoting, and by a factor of 3/2 relative to state-elimination algorithms. Such a *fast-pivoting* algorithm applies both to the discounted and the undiscounted Gittins index. A computational study demonstrates that the algorithm achieves in practice significant time savings against conventional-pivoting and state-elimination algorithms.

A variant of the algorithm is further given that efficiently computes additional quantities that are useful for other purposes. Such a variant is used in Niño-Mora (2007a) as the first stage of a two-stage method to compute the *switching index* of Asawa and Teneketzis (1996) efficiently for bandits with switching costs.

Section 2 shows that optimal stopping problem (1) can be solved via the Gittins index. Section 3 describes the bandit model of concern, reviews a convenient definition of the Gittins index, and discusses the algorithm of Varaiya et al. (1985). Section 4 elucidates the structure of the LP parametric simplex tableau for computing the index and formulates the conventional-pivoting algorithm. Section 5 exploits special structure to reduce the computational effort of pivot steps. Section 6 describes the new fast-pivoting algorithm. Section 7 discusses relations with the state-elimination algorithm. Section 8 reviews the concept of undiscounted Gittins index, and shows how to compute it with the fast-pivoting algorithm. Section 9 reports the results of a computational study. Section 10 summarizes this paper.

2. Gittins-Index Solution to Optimal Stopping

In this section, it is assumed that the chain's state space N is either finite or countably infinite and, in the latter case, that rewards R_i and Q_i are bounded. Consider optimal-stopping problem (1), in the discounted case $0 < \beta < 1$. The discounted value of rewards earned under stopping rule τ starting at i is given by the *reward measure*

$$f_i^\tau(\mathbf{R}, \mathbf{Q}) \triangleq \mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1} R_{X(t)}\beta^t + Q_{X(\tau)}\beta^\tau\right], \tag{4}$$

where the dependence on continuation rewards $\mathbf{R} = (R_j)_{j\in N}$ and terminal rewards $\mathbf{Q} = (Q_j)_{j\in N}$ is made explicit. The corresponding discounted amount of effort expended is given by the *work measure*

$$g_i^\tau \triangleq \mathbb{E}_i^\tau\left[\sum_{t=0}^{\tau-1}\beta^t\right]. \tag{5}$$

Thus the optimal-stopping problem (1) is restated as

$$\max_{\tau\geq 0} f_i^\tau(\mathbf{R},\mathbf{Q}) - \nu g_i^\tau. \tag{6}$$

The key observation is now made that the above discounted optimal-stopping problem is readily reformulated as an *infinite-horizon Markov decision process* (MDP) with two actions available at each state and time period: $a=1$ (active) and $a=0$ (passive). In the latter, if the active action is taken in state i, reward R_i accrues and the state moves according to transition probabilities p_{ij}; if the passive action is taken instead, reward $(1-\beta)Q_i$ accrues and the state does not change. Note that the scale factor $1-\beta$ incorporated into passive rewards accounts for the fact that receiving a lump terminal reward of Q_i is equivalent to receiving a discounted pension of $(1-\beta)Q_i$ at every period over an infinite horizon.

The reader might be concerned at this point that such an infinite-horizon MDP formulation—which differs from the conventional one based on introducing a terminal state—allows policies that take the active action after the passive one has been used, which is not allowed in (1). Such an apparent discrepancy is, however, resolved through MDP theory, which, assuming bounded rewards, ensures existence of an *optimal* policy for such an infinite-horizon MDP that is Markov stationary, and hence consistent with a stopping rule (including the possibility of never stopping).

Next MDP theory is further deployed in the reformulated infinite-horizon problem. Consider the *discounted state-action occupancy measures*: for any stopping rule τ (now viewed as an MDP policy in the reformulated model), initial state i, action a, and state j, let

$$x_{ij}^{a,\tau} \triangleq \mathbb{E}_i^\tau\left[\sum_{t=0}^{\infty} 1_{\{a(t)=a,\, X(t)=j\}}\beta^t\right]$$

be the expected total discounted time expended taking action a in state j under policy τ starting at i. It is well known that, for fixed i, the measures $x_{ij}^{a,\tau}$ for $(j,a)\in N\times\{0,1\}$ satisfy a system of linear equations. Writing $\mathbf{x}_i^{a,\tau} = (x_{ij}^{a,\tau})_{j\in N}$ as a *row vector*, denoting by $\mathbf{I}$ the identity matrix, and by $\mathbf{e}_i\in\mathbb{R}^N$ the unit coordinate row vector having the one in the position of state i, such an equation system is

$$(1-\beta)\mathbf{x}_i^{0,\tau} + \mathbf{x}_i^{1,\tau}(\mathbf{I}-\beta\mathbf{P}) = \mathbf{e}_i. \tag{7}$$

Reward and work measures are formulated as linear functions of occupancies

$$\begin{aligned} f_i^\tau(\mathbf{R},\mathbf{Q}) &= \sum_{j\in N} R_j x_{ij}^{1,\tau} + \sum_{j\in N}(1-\beta)Q_j x_{ij}^{0,\tau} \\ &= \mathbf{x}_i^{1,\tau}\mathbf{R} + (1-\beta)\mathbf{x}_i^{0,\tau}\mathbf{Q}, \\ g_i^\tau &= \sum_{j\in N} x_{ij}^{1,\tau} = \mathbf{x}_i^{1,\tau}\mathbf{1}. \end{aligned} \tag{8}$$

Drawing on the above shows that terminal rewards can be eliminated from (1), both in the discounted and the undiscounted cases.

Lemma 2.1. *Under any stopping rule τ it holds that, for $0<\beta\leq 1$*

$$f_i^\tau(\mathbf{R},\mathbf{Q}) = Q_i + f_i^\tau(\mathbf{R}-(\mathbf{I}-\beta\mathbf{P})\mathbf{Q}, \mathbf{0}). \tag{9}$$

Proof. In the discounted case $\beta<1$, use the above to write

$$\begin{aligned} f_i^\tau(\mathbf{R},\mathbf{Q}) &= \mathbf{x}_i^{1,\tau}\mathbf{R} + (1-\beta)\mathbf{x}_i^{0,\tau}\mathbf{Q} \\ &= \mathbf{x}_i^{1,\tau}\mathbf{R} + \{\mathbf{e}_i - \mathbf{x}_i^{1,\tau}(\mathbf{I}-\beta\mathbf{P})\}\mathbf{Q} \\ &= Q_i + \mathbf{x}_i^{1,\tau}\{\mathbf{R}-(\mathbf{I}-\beta\mathbf{P})\}\mathbf{Q} \\ &= Q_i + f_i^\tau(\mathbf{R}-(\mathbf{I}-\beta\mathbf{P})\mathbf{Q},\mathbf{0}). \end{aligned}$$

Taking the limit as $\beta\nearrow 1$ in the latter identity yields the result for $\beta=1$. □

Consider now the bandit with zero passive rewards and modified active-reward vector

$$\hat{\mathbf{R}} \triangleq \mathbf{R} - (\mathbf{I}-\beta\mathbf{P})\mathbf{Q}, \tag{10}$$

and let $\hat{\nu}_i^*$ be its Gittins index.

The main result of this section follows.

Theorem 2.2. *It is optimal to stop at state i in problem (1) iff $\hat{\nu}_i^*\leq\nu$. The minimum optimal-stopping time is thus $\tau^*=\min\{t\geq 0\colon \hat{\nu}_{X(t)}^*\leq\nu\}$.*

Proof. The above discussion shows that (1) is equivalent to a corresponding problem without terminal rewards of the form (2), with rewards $\hat{R}_i$ given by (10). Since the latter is solved by the stated Gittins-index policy, the result follows. □

3. Gittins Index and VWB Algorithm

The remainder of this paper focuses on a bandit having a finite number n of states. This section reviews a convenient definition of the Gittins index, the concepts of marginal work, reward, and productivity measures introduced in Niño-Mora (2001, 2002, 2006), and the index algorithm of Varaiya et al. (1985), to which we will refer as VWB.

In light of Section 2, we focus on (2), formulated as

$$\max_{\tau\geq 0} f_i^\tau - \nu g_i^\tau. \tag{11}$$

MDP theory shows that, for any continuation charge ν, there exists an optimal stopping rule for (11) that is Markov-stationary deterministic and independent of the initial state. We represent such rules by their *continuation sets*. Hence, to any charge ν there corresponds a *minimal optimal continuation set* $S^*(\nu) \subseteq N$. The latter is characterized by the *Gittins index* ν_j^* that is attached to every state j, so that

$$S^*(\nu) = \{j \in N\colon \nu_j^* > \nu\}, \quad \nu \in \mathbb{R}.$$

For an action a and continuation set S, let $\langle a, S\rangle$ be the policy that takes action a in the first period and adopts the *S-active policy* (which continues over S and stops over $S^c \triangleq N \setminus S$) thereafter. Define, for a state i, the (i, S)*-marginal work measure*

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

as the marginal increase in work expended that results from continuing, instead of stopping in the first period, starting at i, provided that the S-active policy is adopted thereafter. Define further the (i, S)-*marginal reward measure*

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

as the corresponding marginal increase in rewards earned. We will see that marginal workloads are positive, which allows us to define the (i, S)*-marginal productivity rate*

$$\nu_i^S \triangleq \frac{r_i^S}{w_i^S}. \tag{14}$$

Next the above quantities—which are readily computed by solving linear equation systems, as will be shown in the next section—are used to formulate the version of the VWB algorithm in Table 1. The algorithm proceeds in n steps by generating a sequence of states i_k with nonincreasing index values and a corresponding nested sequence of continuation sets $S_0 = \varnothing$, $S_k = \{i_1, \ldots, i_k\}$, for $k = 1, \ldots, n$.

In Varaiya et al. (1985), the ratios $\nu_i^{S_{k-1}}$ in Table 1 are calculated as $\nu_i^{S_{k-1}} = a_i^{(k)}/b_i^{(k)}$, where $a_i^{(k)}$ and $b_i^{(k)}$ are obtained by solving the following linear equation systems: for $i \in N$,

$$a_i^{(k)} = \beta R_i + \beta \sum_{j \in S_{k-1}} p_{ij} a_j^{(k)}$$

$$b_i^{(k)} = \beta + \beta \sum_{j \in S_{k-1}} p_{ij} b_j^{(k)}.$$

Using the results of the next section and a bit of algebra yields that such quantities are related to our $w_i^{S_{k-1}}$ and $r_i^{S_{k-1}}$ above by

$$a_i^{(k)} = \begin{cases} \beta r_i^{S_{k-1}}/(1-\beta) & \text{if } i \in S_{k-1} \\ \beta r_i^{S_{k-1}} & \text{if } i \in S_{k-1}^c \end{cases} \quad \text{and}$$

$$b_i^{(k)} = \begin{cases} \beta w_i^{S_{k-1}}/(1-\beta) & \text{if } i \in S_{k-1} \\ \beta w_i^{S_{k-1}} & \text{if } i \in S_{k-1}^c. \end{cases}$$

The computational complexity of algorithm VWB is next assessed.

Proposition 3.1. *Algorithm VWB performs* $(1/3)n^4 + O(n^3)$ *arithmetic operations.*

Proof. The count is dominated by the solution of two linear equation systems of size k at step k, each taking $(2/3)k^3 + O(k^2)$ arithmetic operations, yielding a total of

$$2\sum_{k=2}^{n}\{(2/3)k^3 + O(k^2)\} = (1/3)n^4 + O(n^3). \quad \square$$

Table 1 Formulation of the VWB Gittins-Index Algorithm via the w_i^S's and r_i^S's

```
Algorithm VWB:
  set S_0 := ∅
  for k := 1 to n do
    compute w_i^{S_{k-1}}, r_i^{S_{k-1}}, ν_i^{S_{k-1}} = r_i^{S_{k-1}}/w_i^{S_{k-1}}, i ∈ S_{k-1}^c
    pick i_k ∈ arg max{ν_i^{S_{k-1}}: i ∈ S_{k-1}^c}
    ν_{i_k}^* := ν_{i_k}^{S_{k-1}}, S_k := S_{k-1} ∪ {i_k}
  end
```

4. LP Formulation and Parametric Simplex Tableau

This section sets out to formulate (11) as a parametric LP problem, drawing on MDP theory, and to elucidate the structure of its simplex tableaux.

Introducing *variables* x_j^a corresponding to occupancy measures $x_{ij}^{a,\tau}$ in Section 2, we use (7) to reformulate (11) as the following parametric LP problem:

$$\begin{aligned} &\max \;\; \mathbf{x}^1(\mathbf{R} - \nu\mathbf{1}) \\ &\text{subject to } \; [\mathbf{x}^0 \;\; \mathbf{x}^1]\begin{bmatrix}(1-\beta)\mathbf{I} \\ \mathbf{I} - \beta\mathbf{P}\end{bmatrix} = \mathbf{e}_i \\ &[\mathbf{x}^0 \;\; \mathbf{x}^1] \geq \mathbf{0}. \end{aligned} \tag{15}$$

Note that in LP (15) variables are written as a row instead of as a column vector, which is counter to conventional usage in LP theory. This is done for notational convenience, as in this way use of tranposed matrices such as $\mathbf{P}^{\mathsf{T}}$ is avoided.

To analyze such an LP problem, note that its *basic feasible solutions* (BFS) correspond to continuation sets

$S \subseteq N$. Refer to the *S-active BFS*. For each such an S, decompose the above vectors and matrices as

$$\mathbf{x}^a = [\mathbf{x}_S^a \quad \mathbf{x}_{S^c}^a], \qquad \mathbf{e}_i = [\mathbf{e}_{iS} \quad \mathbf{e}_{iS^c}],$$
$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{SS} & \mathbf{P}_{SS^c} \\ \mathbf{P}_{S^cS} & \mathbf{P}_{S^cS^c} \end{bmatrix}, \qquad \mathbf{I} = \begin{bmatrix} \mathbf{I}_S & \mathbf{0}_{SS^c} \\ \mathbf{0}_{S^cS} & \mathbf{I}_{S^c} \end{bmatrix},$$

where, e.g., $\mathbf{I}_S$ is the identity matrix indexed by $S \times S$, and introduce the matrices

$$\mathbf{P}^S \triangleq \begin{bmatrix} \mathbf{P}_{SS} & \mathbf{P}_{SS^c} \\ \mathbf{0}_{S^cS} & \mathbf{I}_{S^c} \end{bmatrix}, \qquad \mathbf{P}^{S^c} \triangleq \begin{bmatrix} \mathbf{I}_S & \mathbf{0}_{SS^c} \\ \mathbf{P}_{S^cS} & \mathbf{P}_{S^cS^c} \end{bmatrix},$$
$$\mathbf{B}^S \triangleq \mathbf{I} - \beta\mathbf{P}^S, \qquad \mathbf{N}^S \triangleq \mathbf{I} - \beta\mathbf{P}^{S^c}, \tag{16}$$
$$\mathbf{H}^S \triangleq \{\mathbf{B}^S\}^{-1}, \qquad \mathbf{A}^S \triangleq \mathbf{N}^S\mathbf{H}^S.$$

Note that $\mathbf{P}^S$ (respectively, $\mathbf{P}^{S^c}$) is the transition-probability matrix under the S-active (respectively, S^c-active) policy. Further, $\mathbf{B}^S$ is the *basis matrix* in (15) corresponding to the S-active BFS, whose *basic variables* (corresponding to matrix rows) are $[\mathbf{x}_S^1 \; \mathbf{x}_{S^c}^0]$ and $\mathbf{N}^S$ is the matrix of nonbasic rows in (15), with associated *nonbasic variables* $[\mathbf{x}_S^0 \; \mathbf{x}_{S^c}^1]$.

Next use such a framework to represent performance measures of concern under the S-active policy. First, reformulate the constraints in LP (15) as

$$[[\mathbf{x}_S^1 \quad \mathbf{x}_{S^c}^0] \quad [\mathbf{x}_S^0 \quad \mathbf{x}_{S^c}^1]] \begin{bmatrix} \mathbf{B}^S \\ \mathbf{N}^S \end{bmatrix} = \mathbf{e}_i. \tag{17}$$

Obtain the occupancies $x_{ij}^{a,S}$ under the S-active policy as the corresponding S-active BFS, by setting to zero the nonbasic variables, i.e., $[\mathbf{x}_{iS}^{0,S} \; \mathbf{x}_{iS^c}^{1,S}] = \mathbf{0}$, and calculating the basic variables by solving the linear-equation system

$$[\mathbf{x}_S^1 \quad \mathbf{x}_{S^c}^0]\mathbf{B}^S = \mathbf{e}_i \;\Rightarrow\; [\mathbf{x}_{iS}^{1,S} \quad \mathbf{x}_{iS^c}^{0,S}] = \mathbf{e}_i\mathbf{H}^S. \tag{18}$$

Now use (18) to represent work measure (cf. (8)) vector $\mathbf{g}^S = (g_i^S)_{i\in N}$:

$$g_i^S = \sum_{j\in S} x_{ij}^{1,S} = [\mathbf{x}_{iS}^{1,S} \quad \mathbf{x}_{iS^c}^{0,S}] \begin{bmatrix} \mathbf{1}_S \\ \mathbf{0}_{S^c} \end{bmatrix}$$
$$= \mathbf{e}_i\mathbf{H}^S \begin{bmatrix} \mathbf{1}_S \\ \mathbf{0}_{S^c} \end{bmatrix} \;\Rightarrow\; \mathbf{g}^S = \mathbf{H}^S \begin{bmatrix} \mathbf{1}_S \\ \mathbf{0}_{S^c} \end{bmatrix}. \tag{19}$$

Similarly, represent reward measure (cf. (8)) vector $\mathbf{f}^S = (f_i^S)_{i\in S}$:

$$f_i^S = \sum_{j\in S} R_j x_{ij}^{1,S} = [\mathbf{x}_{iS}^{1,S} \quad \mathbf{x}_{iS^c}^{0,S}] \begin{bmatrix} \mathbf{R}_S \\ \mathbf{0}_{S^c} \end{bmatrix}$$
$$= \mathbf{e}_i\mathbf{H}^S \begin{bmatrix} \mathbf{R}_S \\ \mathbf{0}_{S^c} \end{bmatrix} \;\Rightarrow\; \mathbf{f}^S = \mathbf{H}^S \begin{bmatrix} \mathbf{R}_S \\ \mathbf{0}_{S^c} \end{bmatrix}. \tag{20}$$

Further, represent marginal workloads $\mathbf{w}^S = (w_i^S)_{i\in N}$ using (12) and $\mathbf{g}_{S^c}^S = \mathbf{0}$:

$$\mathbf{w}_S^S = \mathbf{g}_S^S - \beta\mathbf{g}_S^S = (1-\beta)\mathbf{g}_S^S$$
$$\mathbf{w}_{S^c}^S = \mathbf{1}_{S^c} + \beta\mathbf{P}_{S^cN}\mathbf{g}^S - \mathbf{g}_{S^c}^S = \mathbf{1}_{S^c} + \beta\mathbf{P}_{S^cS}\mathbf{g}_S^S. \tag{21}$$

Note that (21) implies that $\mathbf{w}^S > \mathbf{0}$, i.e., *marginal workloads are positive*, as mentioned before. We now reformulate the first identities in (21), using (19), as

$$\begin{bmatrix} \mathbf{w}_S^S \\ -\mathbf{w}_{S^c}^S \end{bmatrix} = \mathbf{N}^S\mathbf{g}^S - \begin{bmatrix} \mathbf{0}_S \\ \mathbf{1}_{S^c} \end{bmatrix} = \mathbf{N}^S\mathbf{H}^S \begin{bmatrix} \mathbf{1}_S \\ \mathbf{0}_{S^c} \end{bmatrix} - \begin{bmatrix} \mathbf{0}_S \\ \mathbf{1}_{S^c} \end{bmatrix}$$
$$= \mathbf{A}^S \begin{bmatrix} \mathbf{1}_S \\ \mathbf{0}_{S^c} \end{bmatrix} - \begin{bmatrix} \mathbf{0}_S \\ \mathbf{1}_{S^c} \end{bmatrix}. \tag{22}$$

Finally, obtain marginal reward vector $\mathbf{r}^S = (r_i^S)_{i\in N}$ from its definition in (13):

$$\mathbf{r}_S^S = \mathbf{f}_S^S - \beta\mathbf{f}_S^S = (1-\beta)\mathbf{f}_S^S$$
$$\mathbf{r}_{S^c}^S = \mathbf{R}_{S^c} + \beta\mathbf{P}_{S^cN}\mathbf{f}^S - \mathbf{f}_{S^c}^S = \mathbf{R}_{S^c} + \beta\mathbf{P}_{S^cN}\mathbf{f}^S. \tag{23}$$

Now reformulate (23), using (20), as

$$\begin{bmatrix} \mathbf{r}_S^S \\ -\mathbf{r}_{S^c}^S \end{bmatrix} = \mathbf{N}^S\mathbf{f}^S - \begin{bmatrix} \mathbf{0}_S \\ \mathbf{R}_{S^c} \end{bmatrix} = \mathbf{N}^S\mathbf{H}^S \begin{bmatrix} \mathbf{R}_S \\ \mathbf{0}_{S^c} \end{bmatrix} - \begin{bmatrix} \mathbf{0}_S \\ \mathbf{R}_{S^c} \end{bmatrix}$$
$$= \mathbf{A}^S \begin{bmatrix} \mathbf{R}_S \\ \mathbf{0}_{S^c} \end{bmatrix} - \begin{bmatrix} \mathbf{0}_S \\ \mathbf{R}_{S^c} \end{bmatrix}. \tag{24}$$

The next result characterizes the marginal workload and marginal reward measures introduced above as reduced costs of corresponding LP problems. It further characterizes the reduced costs of parametric LP problem (15).

Proposition 4.1. *Reduced costs for nonbasic variables in the S-active BFS for LPs*

$$\max\left\{\mathbf{x}^1\mathbf{R}\colon [\mathbf{x}^0 \quad \mathbf{x}^1] \begin{bmatrix} (1-\beta)\mathbf{I} \\ \mathbf{I}-\beta\mathbf{P} \end{bmatrix} = \mathbf{e}_i,\; [\mathbf{x}^0 \quad \mathbf{x}^1] \geq \mathbf{0}\right\},$$
$$\max\left\{\mathbf{x}^1\mathbf{1}\colon [\mathbf{x}^0 \quad \mathbf{x}^1] \begin{bmatrix} (1-\beta)\mathbf{I} \\ \mathbf{I}-\beta\mathbf{P} \end{bmatrix} = \mathbf{e}_i,\; [\mathbf{x}^0 \quad \mathbf{x}^1] \geq \mathbf{0}\right\}$$

and

$$\max\left\{\mathbf{x}^1(\mathbf{R}-\nu\mathbf{1})\colon [\mathbf{x}^0 \quad \mathbf{x}^1] \begin{bmatrix} (1-\beta)\mathbf{I} \\ \mathbf{I}-\beta\mathbf{P} \end{bmatrix} = \mathbf{e}_i,\; [\mathbf{x}^0 \quad \mathbf{x}^1] \geq \mathbf{0}\right\}$$

are given, respectively, by (24), (22), *and*

$$\begin{bmatrix} \mathbf{r}_S^S - \nu\mathbf{w}_S^S \\ -\mathbf{r}_{S^c}^S + \nu\mathbf{w}_{S^c}^S \end{bmatrix}. \tag{25}$$

Therefore, such LPs' objectives can be represented, respectively, as

$$\mathbf{x}^1\mathbf{R} = f_i^S - \sum_{j\in S} r_j^S x_j^0 + \sum_{j\in S^c} r_j^S x_j^1, \tag{26}$$

$$\mathbf{x}^1\mathbf{1} = g_i^S - \sum_{j\in S} w_j^S x_j^0 + \sum_{j\in S^c} w_j^S x_j^1 \tag{27}$$

and

$$\mathbf{x}^1(\mathbf{R}-\nu\mathbf{1}) = f_i^S - \nu g_i^S - \sum_{j\in S}(r_j^S - \nu w_j^S)x_j^0 + \sum_{j\in S^c}(r_j^S-\nu w_j^S)x_j^1. \tag{28}$$

Proof. The results follow from the representation of reduced costs in LP theory, as given by (22) and (24), along with the representation of an LP's objective in terms of the current BFS value and reduced costs. Note that the result for the LP with objective $\mathbf{x}^1\mathbf{R}$ implies the results for the other LPs by appropriate choice of $\mathbf{R}$. □

The next result, which follows directly from Proposition 4.1, gives representations of measures g_i^τ, f_i^τ, and objective $f_i^\tau - \nu g_i^\tau$ relative to the S-active policy. Such identities were first obtained in Niño-Mora (2001) through algebraic arguments.

Proposition 4.2.
(a) $g_i^\tau = g_i^S - \sum_{j\in S} w_j^S x_{ij}^{0,\tau} + \sum_{j\in S^c} w_j^S x_{ij}^{1,\tau}$.
(b) $f_i^\tau = f_i^S - \sum_{j\in S} r_j^S x_{ij}^{0,\tau} + \sum_{j\in S^c} r_j^S x_{ij}^{1,\tau}$.
(c) $f_i^\tau - \nu g_i^\tau = f_i^S - \nu g_i^S - \sum_{j\in S}(r_j^S - \nu w_j^S)x_{ij}^{0,\tau} + \sum_{j\in S^c}(r_j^S - \nu w_j^S)x_{ij}^{1,\tau}$.

The characterization of reduced costs in Proposition 4.1 is used next to give a necessary and sufficient optimality test for the S-active BFS in parametric LP problem (15), and therefore for the S-active policy in ν-wage problem (11).

Proposition 4.3. *The S-active BFS is optimal for LP problem* (15)*—and hence so is the S-active policy for ν-wage problem* (11)*—for every initial state $i\in N$ iff*

$$\max\{\nu_j^S\colon j\in S^c\} \le \nu \le \min\{\nu_j^S\colon j\in S\}. \tag{29}$$

Proof. The "if" part is the sufficient optimality test in LP theory that checks nonnegativity of reduced costs for nonbasic variables. The inequalities (29) follow by reformulating such a condition, using Proposition 4.1(c), positivity of marginal workloads w_j^S, and the definition of marginal productivity rates ν_j^S.

The "only if" part follows by considering a variation on LP (15) where the right-hand side $\mathbf{e}_i$ is replaced by a positive initial-state probability row vector $\mathbf{p} = (p_j)_{j\in N} > \mathbf{0}$. The reduced-cost optimality test for such an LP is the same as for the LPs with right-hand sides $\mathbf{e}_i$. Yet such an LP is nondegenerate, and hence the optimality test is also necessary for it. This completes the proof. □

Table 2 **Augmented Tableau for S-Active BFS, Ready for Pivoting on a_{jj}^S**

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

We now have all the elements to represent the *parametric simplex tableau* for the S-active BFS, as shown in Table 2. Note that such tableau is *transposed* relative to conventional simplex tableaux, so that nonbasic variables $\mathbf{x}_S^0$ and $\mathbf{x}_{S^c}^1$ correspond to *rows*, whereas basic variables $\mathbf{x}_S^1$ and $\mathbf{x}_{S^c}^0$ correspond to *columns*. Further, it includes two columns of reduced costs for nonbasic variables, corresponding to the first two LP problems in Proposition 4.1.

Actually, Table 2 represents an *augmented tableau*, ready for *pivoting* on element a_{jj}^S, with $j\in S^c$, i.e., for taking variable x_j^0 out of the current basis, and putting x_j^1 into the basis. To prepare the ground for carrying out the required pivoting operations, the tableau includes an extra row, corresponding to basic variable x_j^0.

After such a pivoting step is performed, one obtains the tableau for the $S\cup\{j\}$-active BFS, shown in Tables 3 and 4.

The structure of the initial tableau, corresponding to the $\varnothing$-active BFS, must be elucidated. Letting $S=\varnothing$, it is readily obtained from (16) that

$$\mathbf{B}^\varnothing = (1-\beta)\mathbf{I}, \qquad \mathbf{N}^\varnothing = \mathbf{I}-\beta\mathbf{P}, \\ \mathbf{H}^\varnothing = \frac{1}{1-\beta}\mathbf{I}, \qquad \mathbf{A}^\varnothing = \frac{1}{1-\beta}(\mathbf{I}-\beta\mathbf{P}). \tag{30}$$

Further, using (30), (22), and (24), the initial reduced costs are obtained from

$$\mathbf{w}^\varnothing = \mathbf{1} \quad \text{and} \quad \mathbf{r}^\varnothing = \mathbf{R}. \tag{31}$$

Table 3 **Part $\mathbf{A}^{S\cup\{j\}}$ of Tableau for $S\cup\{j\}$-Active BFS, Obtained by Pivoting**

	$\mathbf{x}_S^1$	x_j^1	$\mathbf{x}^0_{S^c\setminus\{j\}}$
$\{\mathbf{x}_S^0\}^\mathsf{T}$	$\mathbf{A}_{SS}^S - \dfrac{\mathbf{A}_{Sj}^S\mathbf{A}_{jS}^S}{a_{jj}^S}$	$\dfrac{\mathbf{A}_{Sj}^S}{a_{jj}^S}$	$\mathbf{A}^S_{S,S^c\setminus\{j\}} - \dfrac{\mathbf{A}_{Sj}^S\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a_{jj}^S}$
x_j^0	$-\dfrac{\mathbf{A}_{jS}^S}{a_{jj}^S}$	$\dfrac{1}{a_{jj}^S}$	$-\dfrac{\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a_{jj}^S}$
$\{\mathbf{x}^1_{S^c\setminus\{j\}}\}^\mathsf{T}$	$\mathbf{A}^S_{S^c\setminus\{j\},S} - \dfrac{\mathbf{A}^S_{S^c\setminus\{j\},j}\mathbf{A}^S_{jS}}{a_{jj}^S}$	$\dfrac{\mathbf{A}^S_{S^c\setminus\{j\},j}}{a_{jj}^S}$	$\mathbf{A}^S_{S^c\setminus\{j\},S^c\setminus\{j\}} - \dfrac{\mathbf{A}^S_{S^c\setminus\{j\},j}\mathbf{A}^S_{j,S^c\setminus\{j\}}}{a_{jj}^S}$

Table 4 Parts $\mathbf{w}^{S\cup\{j\}}$, $\mathbf{r}^{S\cup\{j\}}$ of Tableau for $S\cup\{j\}$-Active BFS, Obtained by Pivoting

$\{\mathbf{x}_S^0\}^{\mathsf{T}}$	$\mathbf{w}_S^S + \dfrac{w_j^S}{a_{jj}^S}\mathbf{A}_{Sj}^S$	$\mathbf{r}_S^S + \dfrac{r_j^S}{a_{jj}^S}\mathbf{A}_{Sj}^S$
x_j^0	$\dfrac{w_j^S}{a_{jj}^S}$	$\dfrac{r_j^S}{a_{jj}^S}$
$\{\mathbf{x}_{S^c\setminus\{j\}}^1\}^{\mathsf{T}}$	$-\mathbf{w}_{S^c\setminus\{j\}}^S + \dfrac{w_j^S}{a_{jj}^S}\mathbf{A}_{S^c\setminus\{j\},j}^S$	$-\mathbf{r}_{S^c\setminus\{j\}}^S + \dfrac{r_j^S}{a_{jj}^S}\mathbf{A}_{S^c\setminus\{j\},j}^S$

We are now ready to formulate the *conventional-pivoting* (CP) index algorithm, shown in Table 5, which implements the parametric simplex approach of Kallenberg (1986). Table 5 adopts an algorithm-like notation, substituting superscript counters (k) for superscript sets S_k. Note that such an algorithm only applies to the discounted case $\beta < 1$, since the computation of $\mathbf{A}^{(0)}$ involves division by $1-\beta$.

The computational complexity of the CP algorithm is assessed next.

Proposition 4.4. *The CP algorithm performs* $2n^3 + O(n^2)$ *arithmetic operations.*

Proof. Counting shows that the algorithm performs $n^3 + O(n^2)$ multiplications and divisions and the same order of additions and subtractions, which yields the result. □

5. Exploiting Special Structure

This section sets out to exploit special structure to reduce the number of operations performed in a pivoting step.

5.1. Updating Marginal Productivity Rates

We start by showing that there is no need to update marginal rewards r_j^S in the tableaux. It suffices to update required marginal work measures w_j^S, and then use them to update required marginal productivity rates ν_j^S. While the next result is proven in Niño-Mora (2002), for self-completeness a new proof is given here, drawing on the pivot step that leads from the tableau in Table 2 to that in Tables 3 and 4.

Proposition 5.1. *For* $i \in N$ *and* $j \in S^c$, $\nu_i^{S\cup\{j\}} = \nu_j^S - (w_i^S/w_i^{S\cup\{j\}})(\nu_j^S - \nu_i^S)$.

Proof. Start with the case $i \in S$. From Table 4 and $\nu_i^{S\cup\{j\}} = r_i^{S\cup\{j\}}/w_i^{S\cup\{j\}}$, we have

$$\begin{aligned}\nu_i^{S\cup\{j\}} &= \frac{\nu_i^S w_i^S + \nu_j^S (w_j^S/a_{jj}^S)a_{ij}^S}{w_i^S + (w_j^S/a_{jj}^S)a_{ij}^S}\\ &= \nu_j^S - \frac{w_i^S}{w_i^S + (w_j^S/a_{jj}^S)a_{ij}^S}(\nu_j^S - \nu_i^S)\\ &= \nu_j^S - \frac{w_i^S}{w_i^{S\cup\{j\}}}(\nu_j^S - \nu_i^S).\end{aligned}$$

Consider now the case $i = j$. Then, again Table 4 yields

$$\nu_j^{S\cup\{j\}} = \frac{r_j^S/a_{jj}^S}{w_j^S/a_{jj}^S} = \frac{r_j^S}{w_j^S} = \nu_j^S.$$

Finally, consider the case $i \in S^c\setminus\{j\}$. Then, using again Table 4 we obtain

$$\begin{aligned}\nu_i^{S\cup\{j\}} &= \frac{\nu_i^S w_i^S - \nu_j^S (w_j^S/a_{jj}^S)a_{ij}^S}{w_i^S - (w_j^S/a_{jj}^S)a_{ij}^S}\\ &= \nu_j^S - \frac{w_i^S}{w_i^S - (w_j^S/a_{jj}^S)a_{ij}^S}(\nu_j^S - \nu_i^S)\\ &= \nu_j^S - \frac{w_i^S}{w_i^{S\cup\{j\}}}(\nu_j^S - \nu_i^S),\end{aligned}$$

as required. This completes the proof. □

Table 5 The CP Gittins-Index Algorithm

Algorithm CP:

$S_0 := \varnothing$, $\mathbf{A}^{(0)} := \dfrac{1}{1-\beta}(\mathbf{I} - \beta\mathbf{P})$, $\mathbf{w}^{(0)} := \mathbf{1}$, $\mathbf{r}^{(0)} := \mathbf{R}$

for $k := 1$ **to** n **do**

  pick $i_k \in \arg\max\{r_i^{(k-1)}/w_i^{(k-1)} : i \in S_{k-1}^c\}$; $\nu_{i_k}^* := r_{i_k}^{(k-1)}/w_{i_k}^{(k-1)}$,

    $S_k := S_{k-1} \cup \{i_k\}$

  if $k < n$ **then**

    $p^{(k)} := a_{i_k i_k}^{(k-1)}$, $a_{i_k i_k}^{(k-1)} := 1$, $\mathbf{v}^{(k)} := (1/p^{(k)})\mathbf{A}_{Ni_k}^{(k-1)}$, $\mathbf{h}^{(k)} := -\mathbf{A}_{i_k N}^{(k-1)}$

    $\mathbf{A}^{(k)} := \mathbf{A}^{(k-1)} + \mathbf{v}^{(k)}\mathbf{h}^{(k)}$, $\mathbf{A}_{Ni_k}^{(k)} := \mathbf{v}^{(k)}$, $\mathbf{A}_{i_k N}^{(k)} := (1/p^{(k)})\mathbf{h}^{(k)}$

    $\mathbf{w}_{S_k^c}^{(k)} := \mathbf{w}_{S_k^c}^{(k-1)} - w_{i_k}^{(k-1)}\mathbf{A}_{S_k^c i_k}^{(k)}$, $\mathbf{w}_{S_{k-1}}^{(k)} := \mathbf{w}_{S_{k-1}}^{(k-1)} + w_{i_k}^{(k-1)}\mathbf{A}_{S_{k-1} i_k}^{(k)}$,

      $w_{i_k}^{(k)} := w_{i_k}^{(k-1)}/p^{(k)}$

    $\mathbf{r}_{S_k^c}^{(k)} := \mathbf{r}_{S_k^c}^{(k-1)} - r_{i_k}^{(k-1)}\mathbf{A}_{S_k^c i_k}^{(k)}$, $\mathbf{r}_{S_{k-1}}^{(k)} := \mathbf{r}_{S_{k-1}}^{(k-1)} + r_{i_k}^{(k-1)}\mathbf{A}_{S_{k-1} i_k}^{(k)}$,

      $r_{i_k}^{(k)} := r_{i_k}^{(k-1)}/p^{(k)}$

  end { if }

end { for }

5.2. The Reduced Tableau

Let us proceed to elucidate which is the minimal information that needs be updated at each pivoting step, which will be stored in a *reduced tableau*. For such a purpose, first observe that basis inverse matrix $\mathbf{H}^S$ is readily partitioned and represented as

$$\begin{aligned}\mathbf{H}^S &= \begin{bmatrix}\mathbf{H}_{SS}^S & \mathbf{H}_{SS^c}^S\\ \mathbf{H}_{S^cS}^S & \mathbf{H}_{S^cS^c}^S\end{bmatrix}\\ &= \begin{bmatrix}(\mathbf{I}_S - \beta\mathbf{P}_{SS})^{-1} & \dfrac{\beta}{1-\beta}(\mathbf{I}_S - \beta\mathbf{P}_{SS})^{-1}\mathbf{P}_{SS^c}\\ \mathbf{0}_{S^cS} & \dfrac{1}{1-\beta}\mathbf{I}_{S^c}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{H}_{SS}^S & \dfrac{\beta}{1-\beta}\mathbf{H}_{SS}^S\mathbf{P}_{SS^c}\\ \mathbf{0}_{S^cS} & \dfrac{1}{1-\beta}\mathbf{I}_{S^c}\end{bmatrix}.\end{aligned} \tag{32}$$

Now use (32) to partition and represent matrix $\mathbf{A}^S$ as

$$\mathbf{A}^S = \begin{bmatrix} \mathbf{A}_{SS}^S & \mathbf{A}_{SS^c}^S \\ \mathbf{A}_{S^cS}^S & \mathbf{A}_{S^cS^c}^S \end{bmatrix} = \mathbf{N}^S \mathbf{H}^S$$

$$= \begin{bmatrix} (1-\beta)\mathbf{I}_S & \mathbf{0}_{SS^c} \\ -\beta \mathbf{P}_{S^cS} & \mathbf{I}_{S^c} - \beta \mathbf{P}_{S^cS^c} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{SS}^S & \dfrac{\beta}{1-\beta}\mathbf{H}_{SS}^S \mathbf{P}_{SS^c} \\ \mathbf{0}_{S^cS} & \dfrac{1}{1-\beta}\mathbf{I}_{S^c} \end{bmatrix}$$

$$= \begin{bmatrix} (1-\beta)\mathbf{H}_{SS}^S & \beta \mathbf{H}_{SS}^S \mathbf{P}_{SS^c} \\ -\beta \mathbf{P}_{S^cS}\mathbf{H}_{SS}^S & \dfrac{1}{1-\beta}(\mathbf{I}_{S^c} - \beta \mathbf{P}_{S^cS^c}) - \dfrac{\beta^2}{1-\beta}\mathbf{P}_{S^cS}\mathbf{H}_{SS}^S \mathbf{P}_{SS^c} \end{bmatrix}$$

$$= \begin{bmatrix} (1-\beta)\mathbf{H}_{SS}^S & \mathbf{A}_{SS^c}^S \\ \mathbf{A}_{S^cS}^S & \dfrac{1}{1-\beta}(\mathbf{I}_{S^c} - \beta \mathbf{P}_{S^cS^c} + \beta \mathbf{A}_{S^cS}^S \mathbf{P}_{SS^c}) \end{bmatrix}. \quad (33)$$

From (33) and the above discussion, it should be clear that it suffices to keep the information on the S-active BFS shown in the reduced tableau in Table 6.

To obtain the reduced tableau for the $S \cup \{j\}$-active BFS from that in Table 6, first compute pivot element a_{jj}^S by noting that, by (33),

$$a_{jj}^S = \frac{1 - \beta p_{jj} + \beta \mathbf{A}_{jS}^S \mathbf{P}_{Sj}}{1-\beta}. \quad (34)$$

Then use again (33) to compute the required vector

$$\mathbf{A}_{S^c \setminus \{j\}, j}^S = -\frac{\beta}{1-\beta}\left\{\mathbf{P}_{S^c \setminus \{j\}, j} - \mathbf{A}_{S^c \setminus \{j\}, S}^S \mathbf{P}_{Sj}\right\}. \quad (35)$$

One can then readily compute the reduced tableau for the $S \cup \{j\}$-active BFS, as shown in Tables 7 and 8.

6. The Fast-Pivoting Algorithm

Drawing on the above, one readily obtains the *fast-pivoting* algorithm FP(EO) in Table 9. The input EO is a Boolean variable that, when fed the value 1, makes the algorithm produce an *extended output*. Such a capability is useful in some settings, where one may need, besides the Gittins index ν_i^*, marginal work and marginal productivity measures $w_i^{S_k}$ and $\nu_i^{S_k}$ corresponding to the continuation sets S_k generated by the algorithm. Thus, e.g., such quantities are used in Niño-Mora (2007a) as input for a fast algorithm that computes the *switching index* of Asawa and Teneketzis (1996).

The computational complexity of the FP(EO) algorithm is assessed next.

Table 6 Reduced Tableau for S-Active BFS

	$\mathbf{x}_S^1$		
$\{\mathbf{x}_{S^c}^1\}^{\mathsf{T}}$	$\mathbf{A}_{S^cS}^S$	$\mathbf{w}_{S^c}^S$	$\boldsymbol{\nu}_{S^c}^S$

Table 7 Reduced Tableau for $S \cup \{j\}$-Active BFS: $\mathbf{A}^{S \cup \{j\}}$ Part

	$\mathbf{x}_S^1$	x_j^1
$\{\mathbf{x}_{S^c \setminus \{j\}}^1\}^{\mathsf{T}}$	$\mathbf{A}_{S^c \setminus \{j\}, S}^S - \dfrac{\mathbf{A}_{S^c \setminus \{j\}, j}^S \mathbf{A}_{jS}^S}{a_{jj}^S}$	$\dfrac{\mathbf{A}_{S^c \setminus \{j\}, j}^S}{a_{jj}^S}$

Proposition 6.1. (a) *The* FP(0) *algorithm performs* $(2/3)n^3 + O(n^2)$ *arithmetic operations*.

(b) *The* FP(1) *algorithm performs* $(4/3)n^3 + O(n^2)$ *arithmetic operations*.

Proof. (a) Step k of the algorithm performs about $(k+1)(2(n-k)+1)$ multiplications and divisions, yielding a total of

$$\sum_{k=2}^{n-1}(k+1)(2(n-k)+1) + O(n) = (1/3)n^3 + (3/2)n^2 + O(n).$$

The same count is obtained for additions and subtractions, which yields the result.

(b) Counting shows that the algorithm performs $(2/3)n^3 + (5/2)n^2 + O(n)$ multiplications and divisions, as well as additions and subtractions, which gives the result. □

Instead of computing marginal productivity rates ν_i^S as stated, one could have updated marginal rewards r_i^S, using the update formula

$$\mathbf{r}_{S_k^c}^{S_k} := \mathbf{r}_{S_k^c}^{S_{k-1}} - r_{i_k}^{S_{k-1}} \mathbf{A}_{S_k^c i_k}^{S_k},$$

and then have computed $\nu_i^{S_k} = r_i^{S_k} / w_i^{S_k}$ for $i \in S_k^c$. The complexity of such an alternative scheme is the same as that of the one proposed.

Note that the FP(0) algorithm applies both to the discounted and the undiscounted index. The latter is computed by setting $\beta = 1$ in the stated calculations.

7. Relation with the State-Elimination Algorithm

Katta and Sethuraman (2004) and Sonin (2005) introduced Gittins-index algorithms based on state-elimination ideas, discussing them as they apply to bandits with a terminal state. Yet, any bandit with a positive discount factor is readily transformed into such a case. See Sonin (2005). Table 10 formulates the *state-elimination* (SE) algorithm.

The above discussions allow us to show that such an algorithm computes the Gittins index, by elucidating its relation with the CP algorithm.

Table 8 Reduced Tableau for $S \cup \{j\}$-Active BFS: $\mathbf{w}^{S \cup \{j\}}$ and $\boldsymbol{\nu}^{S \cup \{j\}}$ Parts

$\{\mathbf{x}_{S^c \setminus \{j\}}^1\}^{\mathsf{T}}$	$\mathbf{w}_{S^c \setminus \{j\}}^S - \dfrac{w_j^S}{a_{jj}^S} \mathbf{A}_{S^c \setminus \{j\}, j}^S$	$\nu_j^S - \dfrac{w_i^S}{w_i^{S \cup \{j\}}}(\nu_j^S - \nu_i^S),$	$i \in S^c \setminus \{j\}$

Table 9 The FP(EO) Gittins-Index Algorithm

Algorithm FP(EO):

$S_0 := \varnothing$, $\mathbf{w}^{(0)} := \mathbf{1}$, $\boldsymbol{\nu}^{(0)} := \mathbf{R}$
for $k := 1$ **to** n **do**
  pick $i_k \in \arg\max\{\nu_i^{(k-1)}: i \in S_{k-1}^c\}$; $\nu_{i_k}^* := \nu_{i_k}^{(k-1)}$, $S_k := S_{k-1} \cup \{i_k\}$
  if $k = 1$ **then**
    $\alpha^{(1)} := -\beta/(1 - \beta p_{i_1 i_1})$, $\mathbf{A}_{S_1^c i_1}^{(1)} := \alpha^{(1)} \mathbf{P}_{S_1^c i_1}$
    if EO **then** $\mathbf{A}_{i_1 S_1^c}^{(1)} := -\alpha^{(1)} \mathbf{P}_{i_1 S_1^c}$ **end** { if }
  else if $k < n$ **or** {EO **and** $k = n$} **then**
    $\alpha^{(k)} := -\beta/\{1 - \beta(p_{i_k i_k} - \mathbf{A}_{i_k S_{k-1}}^{(k-1)} \mathbf{P}_{S_{k-1} i_k})\}$
    $\mathbf{A}_{S_k^c i_k}^{(k)} := \alpha^{(k)}\{\mathbf{P}_{S_k^c i_k} - \mathbf{A}_{S_k^c S_{k-1}}^{(k-1)} \mathbf{P}_{S_{k-1} i_k}\}$,
     $\mathbf{A}_{S_k^c S_{k-1}}^{(k)} := \mathbf{A}_{S_k^c S_{k-1}}^{(k-1)} - \mathbf{A}_{S_k^c i_k}^{(k)} \mathbf{A}_{i_k S_{k-1}}^{(k-1)}$
    if EO **then**
     $\mathbf{A}_{i_k S_k^c}^{(k)} = -\alpha^{(k)}\{\mathbf{P}_{i_k S_k^c} + \mathbf{P}_{i_k S_{k-1}} \mathbf{A}_{S_{k-1} S_k^c}^{(k-1)}\}$,
      $\mathbf{A}_{S_{k-1} S_k^c}^{(k)} := \mathbf{A}_{S_{k-1} S_k^c}^{(k-1)} + \mathbf{A}_{S_{k-1} i_k}^{(k-1)} \mathbf{A}_{i_k S_k^c}^{(k)}$
    end { if }
  end { if }
  $\mathbf{w}_{S_k^c}^{(k)} := \mathbf{w}_{S_k^c}^{(k-1)} - w_{i_k}^{(k-1)} \mathbf{A}_{S_k^c i_k}^{(k)}$,
   $\nu_i^{(k)} := \nu_{i_k}^* - \{w_i^{(k-1)}/w_i^{(k)}\}\{\nu_{i_k}^* - \nu_i^{(k-1)}\}$, $i \in S_k^c$
  if EO **then**
   $w_{i_k}^{(k)} := -\{\alpha^{(k)}(1-\beta)/\beta\} w_{i_k}^{(k-1)}$, $\mathbf{w}_{S_{k-1}}^{(k)} := \mathbf{w}_{S_{k-1}}^{(k-1)} + w_{i_k}^{(k)} \mathbf{A}_{S_{k-1} i_k}^{(k-1)}$
   $\nu_i^{(k)} := \nu_{i_k}^* - \{w_i^{(k-1)}/w_i^{(k)}\}\{\nu_{i_k}^* - \nu_i^{(k-1)}\}$, $i \in S_{k-1}$, $\nu_{i_k}^{(k)} := \nu_{i_k}^*$
  end { if }
end { for }

Proposition 7.1. *The SE algorithm computes the Gittins index.*

Proof. Straightforward algebra yields that the quantities computed in the SE algorithm are related to those computed in the CP algorithm by

$$\begin{aligned} \widetilde{\mathbf{P}}_{S_k^c S_k^c}^{(k)} &= \mathbf{I} - (1-\beta)\mathbf{A}_{S_k^c S_k^c}^{(k)} \\ \mathbf{r}_{S_k^c}^{(k)} &= (1-\beta)\mathbf{r}_{S_k^c}^{(k)} \\ \boldsymbol{\beta}_{S_k^c}^{(k)} &= \mathbf{1}_{S_k^c} - (1-\beta)\mathbf{w}_{S_k^c}^{(k)}, \end{aligned} \tag{36}$$

for each k. Therefore, the SE algorithm computes the same index as does the CP algorithm. This completes the proof. □

Table 10 The SE Gittins-Index Algorithm

Algorithm SE:

$S_0 := \varnothing$, $\widetilde{\mathbf{P}}^{(0)} := \beta\mathbf{P}$, $\boldsymbol{\beta}^{(0)} := \beta\mathbf{1}$, $\tilde{\mathbf{r}}^{(0)} := (1-\beta)\mathbf{R}$
for $k := 1$ **to** n **do**
  pick $i_k \in \arg\max\{\tilde{r}_i^{(k-1)}/\{1 - \beta_i^{(k-1)}\}: i \in S_{k-1}^c\}$
  $\nu_{i_k}^* := \tilde{r}_{i_k}^{(k-1)}/\{1 - \beta_{i_k}^{(k-1)}\}$, $S_k := S_{k-1} \cup \{i_k\}$
  if $k < n$ **then**
   $\widetilde{\mathbf{P}}_{S_k^c i_k}^{(k)} := \{1/(1 - \tilde{p}_{i_k i_k}^{(k-1)})\}\widetilde{\mathbf{P}}_{S_k^c i_k}^{(k-1)}$, $\widetilde{\mathbf{P}}_{S_k^c S_k^c}^{(k)} := \widetilde{\mathbf{P}}_{S_k^c S_k^c}^{(k-1)} + \widetilde{\mathbf{P}}_{S_k^c i_k}^{(k)} \widetilde{\mathbf{P}}_{i_k S_k^c}^{(k-1)}$
   $\boldsymbol{\beta}_{S_k^c}^{(k)} := \widetilde{\mathbf{P}}_{S_k^c S_k^c}^{(k)} \mathbf{1}_{S_k^c}$, $\tilde{\mathbf{r}}_{S_k^c}^{(k)} := \tilde{\mathbf{r}}_{S_k^c}^{(k-1)} + \tilde{r}_{i_k}^{(k-1)} \widetilde{\mathbf{P}}_{S_k^c i_k}^{(k)}$
  end { if }
end { for }

The computational complexity of the SE algorithm is assessed next.

Proposition 7.2. *The SE algorithm performs $n^3 + O(n^2)$ arithmetic operations.*

Proof. Step k of the algorithm performs $3(n-k) + (n-k)^2 + O(1)$ multiplications and divisions, giving a total of

$$n^2 + \sum_{k=1}^{n-1}\{3(n-k) + (n-k)^2\} + O(n) = (1/3)n^3 + 2n^2 + O(n).$$

It further performs $2(n-k)^2 + 2(n-k) + O(1)$ additions and subtractions, giving

$$2\sum_{k=1}^{n-1}\{(n-k)^2 + (n-k)\} + O(1) = (2/3)n^3 + O(n).$$

The stated total operation count follows. □

8. Computation of the Undiscounted Gittins Index

To motivate the interest of computing the undiscounted Gittins index, it is briefly discussed next. Denote by $\nu_i^{\beta,*}$ the discounted Gittins index for discount factor $0 < \beta < 1$. Kelly (1981) showed that $\nu_i^{\beta,*}$ is nondecreasing in β. From (3), it follows that the $\nu_i^{\beta,*}$'s are uniformly bounded above by any upper bound on active rewards. Hence, there exists a unique finite limiting index obtained as β approaches one: $\nu_i^* \triangleq \lim_{\beta \nearrow 1} \nu_i^{\beta,*}$.

It is thus natural to consider the corresponding Taylor-series expansion

$$\nu_i^{\beta,*} = \nu_i^* + \gamma_i^*(1-\beta) + o(1-\beta) \quad \text{as } \beta \nearrow 1,$$

whose validity was established by Katehakis and Rothblum (1996), who further showed that the limiting first- and second-order indices ν_i^* and γ_i^* yield an optimal policy for the multiarmed bandit problem under the *average criterion*: Highest priority is awarded to a bandit with largest first-order index; ties are broken using the second-order index. Kelly (1981) gave an earlier result in such vein.

It is thus of interest to compute ν_i^*. Yet, if one tries to take limits as β tends to one in the VWB, CP, or SE algorithms, they all break down, as they involve divisions or multiplications by $1-\beta$. In contrast, the FP(0) algorithm readily computes the undiscounted Gittins index: One need simply set $\beta = 1$ in the stated computations.

9. Computational Study

It is well known that, in contemporary computers, the arithmetic-operation count of an algorithm need not be the prime driver of its runtime performance.

Table 11 Runtime (Seconds) Comparison of Index Algorithms

n	$t^{FP(0)}$	$t^{FP(1)}$	t^{CP}	t^{SE}	$t^{FP(1)}/t^{FP(0)}$	$t^{CP}/t^{FP(0)}$	$t^{SE}/t^{FP(0)}$	$t^{FP(1)}/t^{CP}$
1,000	15.3	33.0	24.6	24.1	2.16	1.61	1.58	1.34
1,500	49.3	113.3	82.3	80.6	2.30	1.67	1.63	1.38
2,000	118.1	270.1	195.8	194.6	2.29	1.66	1.65	1.38
2,500	230.1	520.6	379.9	377.3	2.26	1.65	1.64	1.37
3,000	395.7	914.0	659.6	648.6	2.31	1.68	1.66	1.39
3,500	627.4	1,429.0	1,051.6	1,043.7	2.28	1.68	1.66	1.36
4,000	937.6	2,122.8	1,568.6	1,559.7	2.26	1.67	1.66	1.35
4,500	1,347.0	3,064.6	2,250.5	2,217.0	2.28	1.67	1.65	1.36
5,000	1,832.7	4,225.6	3,092.6	3,051.5	2.31	1.68	1.67	1.37
5,500	2,458.2	5,629.6	4,136.9	4,058.7	2.29	1.68	1.65	1.36
6,000	3,195.3	7,283.8	5,402.2	5,373.6	2.28	1.69	1.68	1.35

Instead, the exponentially widening gap since the 1990s between processor and memory performance makes computation bottlenecks increasingly due to memory-access times. For a relevant discussion of such issues see, e.g., Dongarra and Eijkhout (2000). It is thus necessary to test whether the improved complexity of the FP algorithm translates into improved runtimes.

A computational study measuring running times on random instances of different sizes was thus conducted, using the author's MATLAB implementations of the algorithms described herein. The experiments were run on an HP xw9300 2.8 GHz AMD Opteron workstation with 4 GB of RAM, using MATLAB R2006a 64-bit on Windows XP x64.

The results are reported in Table 11. For each of the stated state space sizes n, a random problem instance was generated, and the elapsed times $t^{FP(0)}$, $t^{FP(1)}$, t^{CP}, and t^{SE} (in seconds) expended to compute the n Gittins index values were recorded for the FP(0), FP(1), CP, and SE algorithms, respectively. The table further shows the speedup factors $t^{FP(1)}/t^{FP(0)}$, $t^{CP}/t^{FP(0)}$, $t^{SE}/t^{FP(0)}$, and $t^{FP(1)}/t^{CP}$.

The results show that the theoretical speedup factors of 2 and 3/2 of the FP(0) algorithm against the FP(1) and SE algorithms, respectively, slightly underestimate the measured factors. They further show that the theoretical speedup factor of 3 of the FP(0) algorithm relative to the CP algorithm strongly overestimates the measured factor, which lies sligthly above that against the SE algorithm. Further, despite the FP(1) algorithm's smaller arithmetic-operation count relative to the CP algorithm ($(4/3)n^3$ versus $2n^3$), it is strongly outperformed by the latter. Overall, the FP(0) algorithm is the clear winner in such an experiment, followed by the SE algorithm, which shows only slight improvements over the CP algorithm.

The observed discrepancy between theoretical and measured speedup factors is due to memory-access issues. The computation bottleneck of the CP algorithm, as revealed by profiling, is due to the update of matrix $\mathbf{A}^{(k)}$. In contrast, the FP(0), FP(1), and SE algorithms have two computation bottlenecks each, corresponding to the major matrix updates at each step. In the author's implementation, the latter are performed on indexed submatrices of preallocated matrices, involving expensive noncontiguous, strided memory-access patterns. In contrast, matrix $\mathbf{A}^{(k)}$ is efficiently accessed as a contiguous memory block.

10. Conclusions

The parametric version of Dantzig's simplex method is used here as the basis for designing a new algorithm to compute the Gittins index of an n-state bandit, having an improved operation count of $(2/3)n^3$, which matches that of solving a linear-equation system by Gaussian elimination. The algorithm has further been shown to outperform alternative methods. However, direct implementation using indexed submatrices results in expensive noncontiguous, random-stride memory-access patterns, which represent its computation bottleneck. It would be worth investigating whether the latter can be reduced through advanced approaches in numerical linear algebra, such as block-partitioned implementations that exploit advanced-architecture computers, as discussed in Dongarra and Eijkhout (2000). Corresponding simplex-based algorithms are developed in Niño-Mora (2007b) for restless bandits, which can change state when passive. This paper has further introduced a novel Gittins-index solution to the classical problem of optimal stopping of a Markov chain, which renders the new algorithm applicable to the latter. Such an approach yields new tools for the study of optimal-stopping problems that warrant further investigation.

Acknowledgments
The author thanks the anonymous associate editor and two reviewers for valuable suggestions that led to improvements in this paper. This work was supported by the Spanish Ministry of Education and Science under a Ramón y Cajal Investigator Award and Grant MTM2004-02334, by the European Commission's Network of Excellence Euro-NGI, and by the Autonomous Community of Madrid-UC3M through Grant UC3M-MTM-05-075.

References

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

Bertsimas, D., J. Niño-Mora. 1996. Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems. *Math. Oper. Res.* **21** 257–306.

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

Chen, Y.-R., M. N. Katehakis. 1986. Linear programming for finite state multi-armed bandit problems. *Math. Oper. Res.* **11** 180–183.

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

Duan, J.-C., G. Gauthier, J.-G. Simonato. 2003. A Markov chain method for pricing contingent claims. D. D. Yao, H. Zhang, X. Y. Zhou, eds. *Stochastic Modeling and Optimization*. Springer, New York, 333–362.

Gittins, J. C. 1979. Bandit processes and dynamic allocation indices. *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, The Netherlands, 241–266.

Kallenberg, L. C. M. 1986. A note on M. N. Katehakis' and Y.-R. Chen's computation of the Gittins index. *Math. Oper. Res.* **11** 184–186.

Katehakis, M. N., U. G. Rothblum. 1996. Finite state multi-armed bandit problems: Sensitive-discount, average-reward and average-overtaking optimality. *Ann. Appl. Probab.* **6** 1024–1034.

Katta, A.-K., J. Sethuraman. 2004. A note on bandits with a twist. *SIAM J. Discrete Math.* **18** 110–113.

Kelly, F. P. 1981. Multi-armed bandits with discount factor near one: The Bernoulli case. *Ann. Statist.* **9** 987–1001.

Klimov, G. P. 1974. Time-sharing service systems. Part I. *Theory Probab. Appl.* **19** 532–551.

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. Program.* **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. An faster index algorithm and a computational study for bandits with switching costs. *INFORMS J. Comput.* Forthcoming.

Niño-Mora, J. 2007b. Characterization and computation of restless bandit marginal productivity indices. *SMCtools '07, Proc. 2007 Workshop on Tools for Solving Structured Markov Chains.* ACM Press, New York. Forthcoming.

Saaty, T., S. Gass. 1954. Parametric objective function. Part I. *J. Oper. Res. Soc. Amer.* **2** 316–319.

Sonin, I. 1999. The elimination algorithm for the problem of optimal stopping. *Math. Methods Oper. Res.* **49** 111–123.

Sonin, I. 2005. A generalized Gittins index for a Markov chain and its recursive calculation. Technical report, Department of Mathematics, University of North Carolina at Charlotte, Charlotte, NC.

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

Whittle, P. 1980. Multi-armed bandits and the Gittins index. *J. Roy. Statist. Soc. Ser. B* **42** 143–149.