Reprinted from JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS

Vol. 110, No. 2, September 1985
Printed in Belgium

Generalized Polynomial Approximations in Markovian Decision Processes

PAUL J. SCHWEITZER

*The Graduate School of Management,
The University of Rochester, Rochester, New York 14627*

AND

ABRAHAM SEIDMANN

*Department of Industrial Engineering, Tel Aviv University,
Ramat Aviv, Israel*

Submitted by E. Stanley Lee

Fitting the value function in a Markovian decision process by a linear superposition of M basis functions reduces the problem dimensionality from the number of states down to M, with good accuracy retained if the value function is a smooth function of its argument, the state vector. This paper provides, for both the discounted and undiscounted cases, three algorithms for computing the coefficients in the linear superposition: linear programming, policy iteration, and least squares.

1. INTRODUCTION

Consider a (stationary, infinite horizon) semi-Markovian decision process with finite state-space Ω [11, 12]. Solving Bellman's functional equations supplies the value function $\{V(i)^*, i \in \Omega\}$, the gain rate in the undiscounted case, and an optimal policy. The principal computational algorithms are linear programming [4, 14], successive substitutions (value-iteration) [6, 17, 20] and successive approximation in policy space (policy iteration) [11, 12].

These algorithms are practical for medium-size problems but become extremely costly or infeasible when $|\Omega|$ exceeds a few thousand. Furthermore, exact solution is not of central interest for such very large problems, even if available, because storing and table-lookup of the optimal policy is unwieldy. Instead, we seek a good but simpler policy that is easier to implement. This motivates approximation techniques for large problems,

0022-247X/85 $3.00

where one seeks an approximation to the value-function and gain rate, and a good but suboptimal policy.

This paper analyzes approximation of the value-function $\{V(i)^*, i \in \Omega\}$ by a polynomial [1, 3] or more generally by a linear superposition of (say) M fitting functions

$$V(i)^* \simeq \sum_{m=1}^{M} a_m f_m(i) \equiv w(i, \mathbf{a}), \qquad i \in \Omega, \tag{1.1}$$

where the $\{f_m, 1 \leqslant m \leqslant M\}$ are given and the $\{a_m, 1 \leqslant m \leqslant M\}$ must be chosen to give a good fit. This is a non-routine problem in numerical analysis because the functional equations defined V^* are non-linear rather than linear. Nevertheless, some classical fitting techniques [2] (least squares, Galerkin, etc.) may be adapted for our purposes as well as the variational characterizations of V^* (linear programming, etc.).

The incentive for attempting a prior (rather than posterior) fit via (1.1) is the reduction in problem dimensionality from $|\Omega|$ to M, with significant savings in both computer time and storage requirements if M is much smaller than the number of states. A successful fit occurs if one can find a *small* set of fitting functions $\{f_m, 1 \leqslant m \leqslant M\}$ such that one obtains both a good approximation (1.1) to the value function and gain rate, and a good suboptimal policy. The fitting functions can be chosen in essentially any appropriate manner: polynomials, splines [3], etc.

The authors have observed that good fits are possible for several types of queueing networks, where the state is an R-component vector $i = (n_1, n_2,..., n_R)$ with n_j = number of customers at server j. The functional equations were solved *exactly*, and the value function $V(n_1, n_2,..., n_R)^*$ was then fitted with a polynomial of degree 2 or 3, i.e., the $\{f_m(n_1, n_2,..., n_R)\}$ were taken to be 1; $\{n_j, 1 \leqslant j \leqslant R\}$; $(n_j n_k, \ 1 \leqslant j \leqslant k \leqslant R\}$; $\{n_j n_k n_l, 1 \leqslant j \leqslant k \leqslant l \leqslant R\}$. In all cases examined, these posterior fits were accurate within a few percent, over a wide range of server utilizations and other parameter variations (the a_m's varying accordingly). Problems with hundreds of states required only a few dozen f_m's. The quality of these parsimonious posterior fits motivated the present investigation into prior fits.

Section 2 treats the *discounted* case, and describes both the exact functional equations and fitting techniques based upon *linear programming* (LP), *policy iteration* (PIA), and *least squares* (LS). Section 3 does the same for the *undiscounted* case, where one must also estimate the gain rate. A subsequent paper [19] gives our computational experience for a specific example, optimal scheduling in a manufacturing process.

2. Discounted Markovian Decision Processes

2.1. *Exact Functional Equations*

The exact functional equations to be solved for $\{V(i)^*,\ i \in \Omega\}$ are [11, 12]

$$V(i)^* = \max_{k \in A(i)} \{q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, V(j)^*\}, \qquad i \in \Omega, \tag{2.1}$$

where Ω = finite set of states, $A(i)$ = finite set of actions in state i, and $q(i, k)$ and $H(j \mid i, k)$ are respectively, the expected one-step discounted reward and discounted one-step transition probability to state j if action k is chosen while in state i. These satisfy

$$H(j \mid i, k) \geqslant 0, \qquad H(\text{sum} \mid i, k) \equiv \sum_{j \in \Omega} H(j \mid i, k) < 1. \tag{2.2}$$

Equation (2.1) fixes V^* uniquely as the fixed point of a contraction operator.

The following terminology will be employed. A *policy* $d = \{d(i),\ i \in \Omega\}$ consists of specification of an action $d(i) \in A(i)$ for every state i. An *optimal policy* is a policy such that for every state $i \varepsilon \Omega$, $d(i)$ is one of the maximizing actions on the right-hand side of (2.1). The value-function $\{V(d, i),\ i \in \Omega\}$ associated with policy d is the unique solution to the $|\Omega|$ linear equations

$$V(d, i) = q(i, d(i)) + \sum_{j \in \Omega} H(j \mid i, d(i))\, V(d, j), \qquad i \in \Omega. \tag{2.3}$$

2.2. *Assessing the Quality of an Approximation*

Given any approximation $\{V(i),\ i \in \Omega\}$ to $\{V(i)^*,\ i \in \Omega\}$, e.g. from (1.1), the quality of the approximation may be estimated from the bounds [6, 7, 13, 15]

$$V(i) + \min_{j \in \Omega} \Delta(j) \leqslant V(i)^* \leqslant V(i) + \max_{j \in \Omega} \Delta(j), \qquad i \in \Omega, \tag{2.4}$$

where

$$\Delta(i) \equiv \max_{k \in A(i)} \frac{q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, V(j) - V(i)}{1 - H(\text{sum} \mid i, k)}, \qquad i \in \Omega. \tag{2.5}$$

The suggested suboptimal policy $\hat{d}$ is one which achieves all the maxima on the right-hand side of (2.5). The quality of the value-function associated with this policy is given by [7, 15]

$$V(i) + \min_{j \in \Omega} \Delta(j) \leqslant V(\hat{d}, i) \leqslant V(i)^*, \qquad i \in \Omega. \tag{2.6}$$

Note that if $\{V(i)\}$ are given by $\{w(i, \mathbf{a})\}$, it is possible to implement the policy $\hat{\partial}$ without storing and looking up either $\{V(i)\}$ or $\{\hat{\partial}(i)\}$. One stores only the fitting coefficients $\{a_m, 1 \leqslant m \leqslant M\}$. The $\{V(i)\}$ are generated as needed from (1.1) and $\hat{\partial}(i)$ is generated, when needed, by performing the maximization in (2.5).

2.3. *Linear Programming Approximation*

The exact LP for the discounted Markov decision process is [5, 14]

$$\min \sum_{i \in \Omega} c(i)\, V(i)$$
$$V(i) - \sum_{j \in \Omega} H(j \mid i, k)\, V(j) \geqslant q(i, k), \qquad i \in \Omega, k \in A(i) \tag{2.7}$$
$$\{V(i)\} \text{ unconstrained in sign,}$$

where the c's are strictly positive but otherwise arbitrary constraints. This LP has a unique optimal solution $V = V^*$, and the optimal policy uses actions where the dual variables are non-vanishing.

The approximation method inserts (1.1) into this LP and uses $\{a_m, 1 \leqslant m \leqslant M\}$ rather than $\{V(i), i \in \Omega\}$ as decision variables. The resulting LP for the a's is

$$\min \sum_{m=1}^{M} a_m \left[\sum_{i \in \Omega} c(i)\, f_m(i)\right]$$
$$\sum_{m=1}^{M} a_m \left[f_m(i) - \sum_{j \in \Omega} H(j \mid i, k)\, f_m(j)\right] \geqslant q(i, k), \qquad i \in \Omega, k \in A(i) \tag{2.8}$$
$$\{a_m\} \text{ unconstrained in sign.}$$

This LP is *feasible* provided one of the fitting functions is a constant, say $f_1(i) \equiv 1$, because a feasible solution is

$$a_1 \geqslant \max\{q(i, k)/(1 - H(\text{sum} \mid i, k)), i \in \Omega, k \in A(i)\}$$
$$a_m = 0, \qquad 2 \leqslant m \leqslant M.$$

The objective function in (2.8) is *bounded below* for any feasible $\{a_m\}$ because then $w(i, \mathbf{a}) = \sum_{m=1}^{M} a_m f_m(i) \geqslant V(i)^*$ for all $i \in \Omega$.

Consequently, the LP (2.8), and its dual, both have finite optima. For both (2.7) and (2.8), the *dual* LP's are computationally preferable, having fewer constraints. In particular, the dual LP to (2.8) has only M constraints and is likely to be manageable even when the action-spaces are large.

2.4. *Policy Iteration Approximation*

The PIA for computing the a's differs from the exact PIA [11, 12] only in that a least squares fit is employed in the policy evaluation step, i.e., $\{V(d, i), i \in \Omega\}$ is approximated by $\{w(i, \mathbf{a}), i \in \Omega\}$ where the a's are chosen to minimize the sum of the squares of the residual errors. A limit, say LMAX, must be imposed on the number of iterations to prevent cycling because, unlike the exact PIA, the sequence of value-functions is not necessarily monotone.

The algorithm is:

Initialization. Enter with LMAX. Set $L = 0$. Enter with initial policy d.

Policy Evaluation Step. Enter with policy d. Compute

$$\mathbf{a} = \{a_m, 1 \leqslant m \leqslant M\} \text{ to minimize}$$

$$h(\mathbf{a}) = \sum_{i \in \Omega} c(i) \left[q(i, d(i)) + \sum_{j \in \Omega} H(j \mid i, d(i))\, w(j, \mathbf{a}) - w(i, \mathbf{a}) \right]^2, \tag{2.9}$$

where the c's are strictly positive but otherwise arbitrary weights. (See Eqs. (2.10)–(2.12) for performing the minimization.)

Policy Improvement Step. Enter with d and $\mathbf{a}$. Compute a successor policy d^{new}, where $d^{\text{new}}(i)$ achieves the maximum in

$$\max_{k \in A(i)} \left[q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, w(j, \mathbf{a}) \right], \qquad i \in \Omega.$$

(Break ties arbitrarily except to retain $d^{\text{new}}(i) = d(i)$ if possible.)

Termination Test. Exit successfully with $\{w(i, \mathbf{a}), \ i \in \Omega\}$ as an approximation to $\{V(i)^*, t \in \Omega\}$ if $d^{\text{new}} = d$ or if (2.4) shows $\{w(i, \mathbf{a}), i \in \Omega\}$ is sufficiently close to $\{V(i)^*, i \in \Omega\}$. If not but $L \geqslant \text{LMAX}$, exit unsec-cessfully. Otherwise increase L by unity, replace d by d^{new}, and return to the policy evaluation step.

The M Simultaneous Linear Equations

The minimization of the quadratic form $h(\mathbf{a})$ involves solution of M simultaneous linear equations (rather than $|\Omega|$ equations for the original policy evaluation step), hence is practical for M not exceeding a few hundred. Specifically,

$$h(\mathbf{a}) = E(d) + 2 \sum_{m=1}^{M} F(d)_m a_m + \textstyle\sum_{m,n=1}^{M} G(d)_{mn} a_m a_n, \tag{2.10}$$

where $E(d)$, $F(d)_m$ and $G(d)_{mn} = G(d)_{nm}$ have straightforward expressions. Minimizing $h(\mathbf{a})$ by zeroing its gradient leads to M linear equations

$$G(d)\mathbf{a} = -\mathbf{F}(d) \tag{2.11}$$

with solution

$$\mathbf{a} = -G(d)^{-1}\mathbf{F}(d) \equiv \mathbf{x}(d). \tag{2.12}$$

The solvability of these equations is guaranteed by

LEMMA 1. *Assume that the fitting functions $\{f_m, 1 \leqslant m \leqslant M\}$ are linearly independent over Ω. Then, for any policy d, $G(d)$ is non-singular and strictly positive definite.*

Proof by Contradiction. Assume $G(d)$ is singular or not strictly positive definite. Then there exists non-vanishing $\mathbf{y} = \{y_m, 1 \leqslant m \leqslant M\}$ such that $G(d)\mathbf{y} = \mathbf{0}$, or $\mathbf{y}^T G(d)\mathbf{y} = 0$. This implies

$$\sum_{m=1}^{M} y_m \left[f_m(i) - \sum_{j \in \Omega} f_m(j)\, H(j \mid i, d(i)) \right] = 0, \qquad i \in \Omega$$

or

$$w(i, \mathbf{y}) = \sum_{j \in \Omega} w(j, \mathbf{y})\, H(j \mid i, d(i)), \qquad i \in \Omega.$$

Using (2.2), this implies $w(i, \mathbf{y}) = 0$ for all $i \in \Omega$. Since the f's are linearly independent, $\mathbf{y}$ must vanish, a contradiction.

The form of $G(d)$ shows immediately that it is positive semi-definite: $\mathbf{y}^T G(d)\mathbf{y} \geqslant 0$ for all $\mathbf{y}$. If this vanishes for non-zero $\mathbf{y}$, we would get $G(d)\mathbf{y} = \mathbf{0}$, and the contradiction $\mathbf{y} = G(d)^{-1}\mathbf{0} = \mathbf{0}$. Q.E.D.

Remark. If the f's are not linearly independent, $h(\mathbf{a})$ can still be minimized but the minimizing a's are no longer unique; if $\mathbf{a}^*$ achieves the minimum, then so does $\mathbf{a}^* + \mathbf{y}$ whenever $\mathbf{y}$ satisfies $w(i, \mathbf{y}) = 0$ for all $i \in \Omega$.

Galerkin Procedure. Equation (2.11) can be replaced by an alternate set of M simultaneous equations for estimating $\{a_m, 1 \leqslant m \leqslant M\}$, having an $M \times M$ coefficient matrix that is *simpler to compute* but *non-symmetric*.

To derive these, anticipate that

$$w(i, \mathbf{a}) \simeq q(i, d(i)) + \sum_{j \in \Omega} H(j \mid i, d(i))\, w(j, \mathbf{a}), \qquad i \in \Omega, \tag{2.13}$$

when a good choice of $\mathbf{a}$ is employed. Demand that the scalar product of both sides with $\{b(i)\, f_m(i), i \in \Omega\}$ be equal for $1 \leqslant m \leqslant M$, where the b's are an arbitrary positive set of weights. This gives M linear equations for the a's.

2.5. *Global Least Squares Fit*

Here $\mathbf{a}$ is chosen such that (1.1) provides a minimum least squares fit between the left- and right-hand sides of (2.1), i.e., to achieve

$$\min_{\mathbf{a}} u(\mathbf{a}), \tag{2.14}$$

where

$$u(\mathbf{a}) \equiv \sum_{i \in \Omega} c(i) \left[\max_{k \in A(i)} q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, w(j, \mathbf{a}) - w(i, \mathbf{a}) \right]^2, \tag{2.15}$$

where the $\{c(i)\}$ are positive but otherwise arbitrary weights. This involves minimization of a *piecewise-quadratic* function of the a's: if $\mathbf{a}^0$ is such that the maximizing policy d^0 in (2.15) is unique, then $u(\mathbf{a})$ has the form (2.10) in a neighborhood of this $\mathbf{a}^0$, where d^0 remains a maximizing policy.

The following projected gradient algorithm is proposed for minimizing $u(\mathbf{a})$, for the case where d^0 is unique for each $\mathbf{a}^0$ tested by the algorithm.

Initialization. Initial guess $\mathbf{a}^0$.

Loop Step. Enter with $\mathbf{a}^0$. Compute d^0 from (2.15) and $\mathbf{x}(d^0)$ from (2.12).

Exact Termination Test. If $\mathbf{x}(d_0) = \mathbf{a}^0$, exit with $\mathbf{a}^0$ as local minimum for $u(\mathbf{a})$.

Step Size Determination. Compute λ^* achieving

$$\min_{\lambda > 0} u((1-\lambda)\mathbf{a}^0 + \lambda \mathbf{x}(a^0)); \tag{2.16}$$

replace $\mathbf{a}^0$ by $(1-\lambda^*)\mathbf{a}^0 + \lambda^* \mathbf{x}(a^0)$.

Convergence Test. Exit if $u(\mathbf{a}^0)$ is sufficiently small; or if (2.4) shows $\{w(i, \mathbf{a}^0),\ i \in \Omega\}$ is sufficiently close to $\{V(i)^*, i \in \Omega\}$; or if $\mathbf{a}^0$ or $u(\mathbf{a}^0)$ or $w(i, \mathbf{a}^0)$ are no longer changing appreciably. Otherwise return to loop step.

We will show that $x(\mathbf{a}^0) - \mathbf{a}^0$ is a downhill direction, hence the minimization (2.16) produces a *strict reduction* in u. To see this, note that

$$\begin{aligned} \operatorname{grad} u(a^0) &= 2[F(d^0) + G(d^0)\mathbf{a}^0] \\ &= 2G(d^0)[\mathbf{a}^0 - \mathbf{x}(d^0)]. \end{aligned}$$

If $\mathbf{a}^0 = \mathbf{x}(d^0)$, $\mathbf{a}^0$ is a local minimum of u; if not, $[\mathbf{x}(d^0) - \mathbf{a}^0]^T$ grad

$u(a^0) < 0$ because $G(d^0)$ is strictly positive definite. Hence $\mathbf{x}(d^0) - \mathbf{a}^0$ is a downhill direction.

The line search in (2.16) need not be carried out exactly; any substantive value of λ which reduces u is acceptable. For simplicity, it is worth checking if $\lambda = 1$ achieves a reduction in u and employing $\lambda = 1$ if so; this mimics the PIA in Section 2.4. The present algorithm is more flexible, however, in that it can employ $\lambda < 1$ in order to force reduction in u.

Note that the approximate PIA in Section 2.4 allows only parameters $\mathbf{a}$ of the simple form $\mathbf{a} = \mathbf{x}(d)$ for some policy d, while the present algorithm allows more general choices. This extra generality is unnecessary if there exists a *unique* policy d^* achieving

$$\max_{k \in A(i)} \left[q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, w(j, \mathbf{a}^*) \right], \qquad i \in \Omega,$$

where $\mathbf{a}^*$ achieves the minimum in (2.14), because the vanishing of grad $u(\mathbf{a}^*)$ implies $\mathbf{a}^* = \mathbf{x}(d^*)$, i.e., the simple form suffices.

The Galerkin procedure again provides an alternative framework. Anticipate that

$$w(i, \mathbf{a}) \simeq \max_{k \in A(i)} \left[q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, w(j, \mathbf{a}) \right], \qquad i \in \Omega$$

when a good choice of $\mathbf{a}$ is employed. Demand that the scalar product of both sides with $\{b(i) f_m(i),\ 1 \leqslant m \leqslant M\}$ be equal for $1 \leqslant m \leqslant M$, where the b's are arbitrary positive weights. This leads to M non-linear equations for $\{a_m,\ 1 \leqslant m \leqslant M\}$. These equations are of *fixed point type* if the f's are orthonormal:

$$\sum_{i \in \Omega} f_m(i) f_n(i)\, b(i) = \delta_{nm}, \qquad 1 \leqslant n, m \leqslant M,$$

namely

$$a_m = \sum_{i \in \Omega} b(i) f_m(i) \max_{k \in A(i)} \left[q(i, k) + \sum_{j \in \Omega} H(j \mid i, k)\, w(j, \mathbf{a}) \right], \qquad 1 \leqslant m \leqslant M.$$

They may be solvable, in some cases, by successive substitution or policy iteration.

3. Undiscounted Markovian Decision Processes

3.1. *Exact Functional Equations*

The exact functional equations to be solved for the maximal gain rate g^* (expected reward per unit time) and relative value vector $\{V(i)^*, i \in \Omega\}$ are [11, 12]

$$V(i)^* = \max_{k \in A(i)} \left\{ q(i, k) - g^* T(i, k) + \sum_{j \in \Omega} P(j \mid i, k)\, V(j)^* \right\}, \qquad i \in \Omega \tag{3.1a}$$

$$V(r)^* = 0, \tag{3.1b}$$

where Ω and $A(i)$ are as before, and where $q(i, k)$, $T(i, k)$, and $P(j \mid i, k)$ are respectively the undiscounted one-transition expected reward, expected holding time in state i, and probability that the next state is j, conditioned on choosing action k while in state i. These satisfy

$$T(i, k) > 0, \qquad P(j \mid i, k) \geqslant 0, \qquad \sum_{j \in \Omega} P(j \mid i, k) = 1. \tag{3.2}$$

Constraint (3.1b), where $r \in \Omega$ is arbitrary, fixes an otherwise-arbitrary additive constant in the $\{V(i)^*\}$.

We make the following:

UNICHAIN ASSUMPTION. For every policy d, the transition probability matrix $P(d) = [P(j \mid i, d(i))]_{i,j \in \Omega}$ has a *single* closed irreducible set of states (one subchain) hence a *unique* equilibrium distribution $\pi(d) = \pi(d)\, P(d)$, $\sum_{i \in \Omega} \pi(d)_i = 1$.

This assumption assures [18] that the $|\Omega| + 1$ equations in (3.1) *uniquely* determine the $|\Omega| + 1$ unknowns $\{g^*; V(i)^*, i \in \Omega\}$. Note transient states are allowed.

The following terminology is employed. As before, an *optimal policy* is one achieving all maxima on the right-hand side of (3.1a). For any policy d, the gain rate $g(d)$ and relative value vector $\{V(d, i),\ i \in \Omega\}$ are the solution (unique under the unichain assumption) to the $|\Omega| + 1$ linear equations

$$V(d, i) = q(i, d(i)) - g(d)\, T(i, d(i)) + \sum_{j \in \Omega} P(j \mid i, d(i))\, V(d, j), \qquad i \in \Omega \tag{3.3a}$$

$$V(d, r) = 0. \tag{3.3b}$$

3.2. *Assessing the Quality of an Approximation*

Given any approximation $\{V(i), i \in \Omega\}$ to $\{V(i)^*, i \in \Omega\}$, e.g., from (1.1), bounds on the maximal gain rate g^* are given by [9, 16]

$$\min_{j \in \Omega} \Delta(j) \leqslant g^* \leqslant \max_{j \in \Omega} \Delta(j), \tag{3.4}$$

where

$$\Delta(i) \equiv \max_{k \in A(i)} \frac{q(i, k) + \sum_{j \in \Omega} P(j \mid i, k)\, V(j) - V(i)}{T(i, k)}, \qquad i \in \Omega. \tag{3.5}$$

The suggested estimate of the gain rate is

$$g^* \simeq \tfrac{1}{2}[\min_{j \in \Omega} \Delta(j) + \max_{j \in \Omega} \Delta(j)]. \tag{3.6}$$

The suggested suboptimal policy $\hat{d}$ is any policy achieving all the maxima on the right-hand side of (3.5). The bounds on the quality of the gain rate of this policy are given by [9]

$$\min_{j \in \Omega} \Delta(j) \leqslant g(\hat{d}) \leqslant g^*. \tag{3.7}$$

The bounds on $\{V(i)^*, i \in \Omega\}$ are given by

$$\max_{i \in \Omega} |V(i)^* - V(i)| \leqslant b[\max_{i \in \Omega} \Delta(i) - \min_{i \in \Omega} \Delta(i)], \tag{3.8}$$

where the constant b is given in [8], and it is assumed $V(r) = V(r)^* = 0$.

3.3. *Linear Programming Approximation*

The exact LP for finding g^* is [4, 5, 10, 14]

$$\begin{aligned} &\min g \\ &V(i) - \sum_{j \in \Omega} P(j \mid i, k)\, V(j) + gT(i, k) \geqslant q(i, k), \qquad i \in \Omega, k \in A(i) \\ &g \text{ and } \{V(i)\} \text{ unconstrained in sign.} \end{aligned} \tag{3.9}$$

This LP is always feasible and, assuming the solvability of (3.1), has an optimal $g = g^*$. Under the unichain assumption, the optimal $\{V(i)\}$ for the LP differ from $\{V(i)^*\}$ only by an additive constant, for all i which are recurrent for $P(d^*)$ with d^* an optimal policy. A separate parsing procedure [10] gives the $\{V(i)^*\}$ for the transient states.

As in Section 2.3, the LP approximation is obtained by inserting (1.1) into the exact LP and using $\{g; a_m, 1 \leqslant m \leqslant M\}$ instead of $\{g; V(i), i \in \Omega\}$ as decision variables. The resulting LP for g and the a's is

$$\begin{aligned} &\min g \\ &\sum_{m=1}^{M} a_m[f_m(i) - \sum_{j \in \Omega} P(j \mid i, k) f_m(j)] + gT(i, k) \geqslant q(i, k), \qquad i \in \Omega, k \in A(i) \\ &g \text{ and } \{a_m\} \text{ unconstrained in sign.} \end{aligned} \tag{3.10}$$

This LP is always feasible (take $g = \max\{q(i,k)/T(i,k)|$ all $i, k\}$, $a_m = 0$ for $1 \leqslant m \leqslant M\}$), and any feasible g satisfies $g \geqslant g^*$. Hence the LP has a finite optimum. As in Section 2.3, the *dual* LP's to (3.9) and (3.10) are computationally more attractive.

The constraint $V(r)^* = 0$ is unnecessary in these LP's (since a constant may be added to every $V(i)$) and has been omitted. The constraint can be restored, if desired, by adding $V(r) = 0$ to (3.9) and adding

$$0 = \sum_{m=1}^{M} a_m f_m(r) \qquad [= w(r, \mathbf{a})] \tag{3.11}$$

to (3.10). More simply, one can satisfy (3.11) by having fitting functions satisfy

$$f_m(r) = 0, \qquad 1 \leqslant m \leqslant M. \tag{3.12}$$

3.4. *Policy Iteration Approximation*

The approximate PIA is obtained from the exact one [11, 12] by the same approach as in Section 2.4, using a least squares fit with $\{a_0; w(i, \mathbf{a})\}$ for $\{g(d); V(d, i)\}$. The algorithm is:

Initialization. Enter with LMAX. Set $L = 0$. Enter with initial policy d.

Policy Evaluation Step. Enter with policy d. Compute $\{a_0, \mathbf{a}\}$ to minimize

$$\begin{aligned} h(a_0, \mathbf{a}) = \sum_{i \in \Omega} c(i)[q(i, d(i)) - a_0 T(i, d(i)) \\ + \sum_{j \in \Omega} P(j \mid i, d(i))\, w(j, \mathbf{a}) - w(i, \mathbf{a})]^2, \end{aligned} \tag{3.13}$$

where the c's are strictly positive but otherwise arbitrary weights. (See Eq. (3.14) for performing the minimization.)

Policy Improvement Step. Enter with d, a_0 and $\mathbf{a}$. Compute a successor policy d^{new} where $d^{\text{new}}(i)$ achieves the maximum in

$$\max_{k \in A(i)} \frac{q(i, k) - a_0 T(i, k) + \sum_{j \in \Omega} P(j \mid i, k)\, w(j, \mathbf{a}) - w(i, \mathbf{a})}{1 \text{ or } T(i, k)}.$$

(Break ties arbitrarily except to retain $d^{\text{new}}(i) = d(i)$ if possible.)

Termination Test. Exit successfully with $\{a_0; w(i, \mathbf{a}),\ i \in \Omega\}$ as approximations to $\{g^*; V(i)^*, i \in \Omega\}$ if $d^{\text{new}} = d$ or if the upper and lower bounds in (3.4), with $\{V(i)\} = \{w(i, \mathbf{a})\}$, are sufficiently close. If not but $L \geqslant$ LMAX, exit unsuccessfully. Otherwise increase L by unity, replace d by d^{new}, and return to the policy evaluation step.

The $M+1$ Simultaneous Linear Equations

The minimization of the quadratic form $h(a_0, \mathbf{a})$ involves solution of $M+1$ simultaneous linear equations, rather than $|\Omega|+1$ equations for the exact PIA. Specifically,

$$h(a_0, \mathbf{a}) = E(d) + 2 \sum_{m=0}^{M} F(d)_m a_m + \sum_{m,n=0}^{M} G(d)_{mn} a_m a_n,$$

where $E(d)$, $F(d)_m$ and $G(d)_{mn} = G(d)_{nm}$ have straightforward expressions. Minimizing $h(a_0, \mathbf{a})$ by zeroing its gradient leads to $M+1$ linear equations

$$G(d)[a_0; \mathbf{a}] = -\mathbf{F}(d) \tag{3.14}$$

with solution

$$[a_0; \mathbf{a}] = -G(d)^{-1}\mathbf{F}(d).$$

The solvability of (3.14) is assured by

LEMMA 2. *If the unichain assumption holds, and if the only linear combination of the $\{f_m\}$ which forms a vector with all components equal is the trivial (zeroweighting) case, then, for any policy d, $G(d)$ is non-singular and strictly positive definite.*

Proof by Contradiction. Assume $G(d)$ is singular. Then there exists a set of numbers $(y_0, y_1, ..., y_M) = [y_0; \mathbf{y}]$, not all vanishing, such that $\sum_{n=0}^{M} G(d)_{mn} y_n = 0$ for $0 \leqslant m \leqslant N$, hence $\sum_{m=0}^{M} \sum_{n=0}^{M} G(d)_{mn} y_m y_n = 0$ and $G(d)$ is positive semi-definite but not strictly positive-definite. The vanishing of the double sum implies

$$0 = y_0 T(i, d(i)) + w(i, \mathbf{y}) - \sum_{j \in \Omega} w(j, \mathbf{y}) P(j \mid i, d(i)), \qquad i \in \Omega. \tag{3.15}$$

Multiply (3.15) by the equilibrium distribution $\pi(d)_i$ of $P(d)$, and sum over $i \in \Omega$ to obtain, using $\pi(d) P(d) - \pi(d) = \mathbf{0}$, $0 = y_0 \sum_{i \in \Omega} \pi(d)_i T(i, d(i))$. Conclude that $y_0 = 0$ and, from (3.15), that $w(i, \mathbf{y})$ is a right eigenvector of $P(d)$ with eigenvalue unity. Under the unichain assumption, any such eigenvector must have all components equal. This violates the second assumption of the lemma since $\mathbf{y} \neq \mathbf{0}$. Q.E.D.

Remark. If the second assumption in Lemma 2 does not hold, then $h(a_0, \mathbf{a})$ can still be minimized but the minimum is no longer unique: if $[a_0^*, \mathbf{a}^*]$ achieves the minimum, then so does $[a_0^*, \mathbf{a}^* + \mathbf{y}]$, where $\mathbf{y}$ is any vector such that $\{w(i, \mathbf{y}), i \in \Omega\}$ has all components equal.

Remark. If (3.12) holds, then the second assumption of Lemma 2 reduces to: $\{f_m, 1 \leqslant m \leqslant M\}$ are linearly independent over Ω.

Constrained Minimization

The unconstrained minimization of $h(a_0, \mathbf{a})$ can be replaced by minimization subject to the constraint (3.11) with only a minor increase in the computations. Assume at least one $\{f_m(r), 1 \leqslant m \leqslant M\}$ is non-vanishing, lest (3.11) be met trivially. Introducing a Lagrange multiplier λ to dualize the constraint, we obtain an unconstrained minimization of the same form, but with $F(d)_m$ replaced by

$$\begin{aligned} F(d)_m^{\text{new}} &\equiv F(d)_m, && m = 0 \\ &\equiv F(d)_m + \lambda f_m(r), && 1 \leqslant m \leqslant M. \end{aligned}$$

Using the non-singularity of $G(d)$, given by Lemma 2, the minimum now occurs at

$$[a_0, \mathbf{a}]^{\text{new}} = -G(d)^{-1} F(d)^{\text{new}} = [a_0, \mathbf{a}]^{\text{old}} + \lambda [b_0, \mathbf{b}],$$

where

$$[b_0, \mathbf{b}] = -G(d)^{-1} [0, f_1(r), f_2(r), ..., f_M(r)]^T.$$

Then $w(i, \mathbf{a}^{\text{new}}) = w(i, \mathbf{a}^{\text{old}}) + \lambda w(i, \mathbf{b})$. The constraint (3.11) is met by choosing

$$\lambda = -w(r, \mathbf{a}^{\text{old}})/w(r, \mathbf{b}).$$

(The denominator $w(r, \mathbf{b})$ is non-vanishing because

$$w(r, \mathbf{b}) = -\sum_{m=1}^{N} \sum_{n=1}^{N} f_m(r) [G(d)^{-1}]_{mn} f_n(r)$$

and $G(d)$, hence $G(d)^{-1}$, is strictly positive definite.)

Galerkin Procedure

As in the discounted case, (3.14) can be replaced by a different set of $M + 1$ linear equations, with simpler but non-symmetric matrix elements.

3.5. *Global Least Squares Fit*

The procedure parallels the one in Section 2.5. The parameters $[a_0, \mathbf{a}] = (a_0, a_1, a_2, ..., a_m)$ are chosen to minimize

$$u(a_0, \mathbf{a}) = \sum_{i \in \Omega} c(i) \Big[\max_{k \in A(i)} q(i, k) - a_0 T(i, k) + \sum_{j \in \Omega} P(j \mid i, k)\, w(j, \mathbf{a}) - w(i, \mathbf{a}) \Big]^2,$$

where the c's are arbitrary strictly positive weights, with constraint (3.11) possibly included. The minimization of this piecewise-quadratic objective function can be undertaken by the projected-gradient algorithm given in the discounted case. If $[a_0^*, \mathbf{a}^*]$ achieves the minimum, then $\{V(i)^*\}$ is estimated by $\{w(i, \mathbf{a}^*)\}$ and g^* is estimated either by a_0^* or by (3.6) evaluated at $\{V(i)\} = \{w(i, \mathbf{a}^*)\}$.

References

1. R. Bellman, R. Kalaba, and B. Kotkin, Polynomial approximation—A new computational technique in dynamic programming, *Math. Comp.* **17**, No. 8 (1963), 155–161.
2. E. W. Cheney, "Introduction to Approximation Theory," McGraw–Hill, New York, 1966.
3. J. W. Daniel, Splines and efficiency in dynamic programming, *J. Math. Anal. Appl.* **54** (1976), 402–407.
4. E. V. Denardo and B. Fox, Multichain Markov renewal programs, *SIAM J. Appl. Math.* **16** (1968), 468–487.
5. C. Derman, "Finite State Markovian Decision Processes," Academic Press, New York, 1970.
6. A. Federgruen and P. J. Schweitzer, A survey of asymptotic value-iteration for undiscounted Markovian decision processes, *in* "Recent Developments in Markov Decision Processes" (R. Hartley, L. C. Thomas, and D. White, Ed.), Academic Press, New York, 1980.
7. A. Federgruen and P. J. Schweitzer, Data transformations for Markovian decision processes, in preparation.
8. A. Federgruen and P. J. Schweitzer, Lyapunov functions for Markovian decision process, in preparation.
9. N. A. J. Hastings, Bounds on the gain rate of a Markov decision process, *Oper. Res.* **19** (1971), 240–244.
10. A. Hordijk and L. C. M. Kallenberg, Linear programming and Markov decision chains, Manage. *Sci.* **25** (1979), 352–362.
11. R. A. Howard, Semi-Markovian decision processes, *Bull. Int. Statist. Inst.* **40**, Part 2 (1963), 625–652.
12. W. S. Jewell, Markov renewal programming I and II, *Oper. Res.* **11** (1963), 938–971.
13. J. B. MacQueen, A modified dynamic programming method for Markovian decision problems, *J. Math. Anal. Appl.* **14** (1966), 38–43.
14. S. Osaki and H. Mine, Linear programming algorithms for semi-Markovian decision processes, *J. Math. Anal. Appl.* **22** (1968), 256–381.

15. E. L. Porteus, Bounds and transformations for discounted finite Markov decision chains, *Oper. Res.* **33** (1975), 761–784.
16. P. J. Schweitzer, Multiple policy improvements in undiscounted Markov renewal programming, *Oper. Res.* **19** (1971), 784–793.
17. P. J. Schweitzer, Iterative solution of the functional equations of undiscounted Markov renewal programming, *J. Math. Anal. Appl.* **34** (1971), 495–501.
18. P. J. Schweitzer and A. Federgruen, The functional equations of undiscounted Markov renewal programs, *Math. Oper. Res.* **3** (1978), 308–322.
19. A. Seidmann and P. J. Schweitzer, Approximation methods for optimal control of a flexible manufacturing cell, in preparation.
20. D. J. White, Dynamic programming, Markov chains, and the method of successive approximation, *J. Math. Anal. Appl.* **6** (1963), 373–376.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium