MANAGEMENT SCIENCE
Vol. 26, No. 12, December 1980
Printed in U.S.A.

A NEW SPECIFICATION OF THE MULTICHAIN POLICY ITERATION ALGORITHM IN UNDISCOUNTED MARKOV RENEWAL PROGRAMS*

A. FEDERGRUEN† AND D. SPREEN‡

We consider the Policy Iteration Algorithm for undiscounted Markov Renewal Programs. Previous specifications of the policy evaluation part of this algorithm all required the analysis of the chain structure for each policy generated. The purpose of this paper is to provide a unique specification of the value vectors as well as an anticycling rule which avoids parsing the transition probability matrices into their subchains.

(DYNAMIC PROGRAMMING–MARKOV, FINITE STATE; DYNAMIC PROGRAMMING–SEMI-MARKOV)

1. Introduction and Summary

We consider a Markov Renewal Program (MRP) with $\Omega = \{1, \ldots, N\}$ as state space and $K(i)$ as the finite set of alternatives in state i $(1 \leqslant i \leqslant N)$. Let q_i^k denote the one-step expected reward and $P_{ij}^k \geqslant 0$ the transition probability to state j, when alternative $k \in K(i)$ is used in state i $(\sum_j P_{ij}^k = 1)$. Finally, $\tau_{ij}^k > 0$ denotes the expected conditional holding time, when alternative k is used in state i, given j is the next state of the system. For two N-component vectors x, y, $x > y$ implies $x_i \geqslant y_i$ and $x \neq y$. For any subset $S \subseteq \Omega$, let $\bar{S} = \Omega \backslash S$, and let $|S|$ denote the cardinality of S.

A *stationary policy* δ assigns to each state $i \in \Omega$, a single alternative $\delta(i) \in K(i)$ to be used whenever the system is in state i. Associated with each stationary policy are the N-component reward vector $q(\delta)$, and two $N \times N$ matrices $P(\delta)$ and $H(\delta)$:

$$q(\delta)_i = q_i^{\delta(i)}; \quad P(\delta)_{ij} = P_{ij}^{\delta(i)}; \quad H(\delta)_{ij} = H_{ij}^{\delta(i)} = P_{ij}^{\delta(i)}\tau_{ij}^{\delta(i)} \qquad (1 \leqslant i, j \leqslant N)$$

The nth power of $P(\delta)$, $P^n(\delta)$, is defined recursively by $P^n(\delta) = P^{n-1}(\delta)$. $P(\delta)$ for $n \geqslant 1$ and $P^0(\delta) = I$. Let $\Pi(\delta)$ denote the Cesáro limit of the sequence $\{P^n(\delta)\}_{n=1}^{\infty}$. $R(\delta) = \{j \mid \Pi(\delta)_{jj} > 0\}$ represents the set of recurrent states under $P(\delta)$ and $n(\delta)$ denotes the number of *subchains* (closed, irreducible sets of states) of $P(\delta)$ (cf. Denardo and Fox [5]). For a policy η, the system of $2N$ equations:

$$g = P(\eta)g, \tag{1}$$

$$v = q(\eta) - H(\eta)g + P(\eta)v \tag{2}$$

has the vector $g = g(\eta)$ uniquely determined, whereas the vector v is determined up to $n(\eta)$ additive constants, *one* per *subchain*, by (cf. lemma 1 in [5])

$$v_i = Z(\eta)\left[q(\eta) - H(\eta)g\right]_i + \sum_{m=1}^{n(\eta)} a_m \Phi_i^m(\eta), \qquad i \in \Omega, \tag{3}$$

* Accepted by Bennett L. Fox, former Departmental Editor; received February 6, 1980. This paper has been with the authors 2 months for 1 revision.

† Columbia University.

‡ Rheinisch-Westfalische Technische Hochschule, Aachen, West Germany.

0025-1909/80/2612/1211$01.25

with $a_1, \ldots, a_{n(\eta)}$ arbitrary scalars. The Policy Iteration Algorithm (PIA) for multichain MRPs is given by (cf. [5], [8], [9]):

Step 0: Fix a stationary policy δ.

Step 1: (*Policy evaluation*) Specify a particular solution (g, v) to the system (1), (2) with $\eta = \delta$.

Step 2: (*Policy improvement*) Define:

$$b(g,v)_i^k = q_i^k - \sum_j H_{ij}^k g_j + \sum_j P_{ij}^k v_j - v_i,$$

$$\Gamma_i(g) = \Big\{ k \in K(i) \,|\, k \quad \text{maximizes } \sum_j P_{ij}^k g_j \quad \text{over } k \in K(i) \Big\},$$

$$D_i(g,v) = \big\{ k \in \Gamma_i(g) \,|\, k \quad \text{maximizes } b(g,v)_i^k \quad \text{over } k \in \Gamma_i(g) \big\}.$$

Determine a new policy π as follows: for $i = 1, \ldots, N$, set $\pi(i) = \delta(i)$ if $\delta(i) \in D_i(g,v)$ and otherwise set $\pi(i)$ equal to some arbitrary element of $D_i(g,v)$. If $\pi = \delta$, terminate; otherwise, replace δ by π and go to step 1.

Examples are known where "unwise" choices of the additive constants cause the PIA to cycle (cf. e.g. Example 1 in [10]). To prevent cycling, the following rules have been proposed in the literature:

Rule A.1: (cf. Howard [8], Denardo and Fox [5], Jewell [9]). Set $v(\delta)_i = 0$ for the smallest (largest) i within each subchain of $P(\delta)$.

Rule A.2: (cf. Blackwell [2]). Choose $v(\delta)$ such that $\Pi(\delta)v(\delta) = 0$.

Note that Rule A.2 could implicitly be imposed by adding a third vector-equation

$$w = -v + P(\eta)w \tag{4}$$

to (1) and (2) as can be verified by multiplying both sides of (4) with $\Pi(\eta)$. The system (1), (2), (4) is in fact used in the policy evaluation step of the PIA which determines the more selective *bias-optimal* policies in discrete-time Markov decision processes (cf. Veinott [12]; an equation similar to (4) is used in the corresponding algorithm for MRP's, see Denardo [4]).

Both rules A.1 and A.2 are special cases of a general anti-cycling rule exhibited in [10]:

Rule A: for any two policies δ and π that have a common subchain C and select identical alternatives in all states belonging to C, the relative vectors $v(\delta)$ and $v(\pi)$ are chosen such that $v(\delta)_i = v(\pi)_i$ for all states i belonging to C.

Rule A (including the two special cases A.1 and A.2) gives the appearance of requiring the determination of the subchains for each policy generated by the PIA. Efficient algorithms for parsing a stochastic matrix P into its subchains can be found in Fox and Landi [7] and Denardo [3], and require $0|\{(i,j) \,|\, P_{ij} > 0\}|$ operations (cf. [3, Chapter 11, Theorem 13]). Moreover, the classical treatment of the multichain PIA suggests a need for determining the chain structure at each policy evaluation step as well.

The purpose of this paper is to *specify* the policy *evaluation step* in PIA, and in particular to reformulate the anticycling rule A, so as to avoid parsing the matrices $P(\delta)$, $\delta \in \Delta$, into their subchains.

We first need the following notation. For any policy δ and subset $S \subseteq \Omega$, the *reaching set of S under* $P(\delta)$ consists of the states i for which $\sum_{j \in S} P^n(\delta)_{ij} > 0$ for some $n \geqslant 0$. For any pair of policies δ, π and a relative value vector $v(\delta)$, let $Y(\delta,\pi) = q(\pi) - H(\pi)g(\delta) + P(\pi)v(\delta) - v(\delta)$. Let $S(\delta,\pi) = \{i \mid Y(\delta,\pi)_i > 0\}$ and let $T(\delta,\pi)$ be the reaching set of $S(\delta,\pi)$ under $P(\pi)$.

We conclude this section by stating a modified formulation of the policy evaluation step:

(a) Enter with a policy π. If π is the first policy generated, go to step (b). Otherwise, let δ be the previous policy. If $P(\pi)g(\delta) = g(\delta)$ go to step (c); otherwise, proceed.

(b) Solve system (1), (2) with $\eta = \pi$ by using the modified Gaussian elimination procedure (§3). Exit to the policy improvement step.

(c) Apply the Reaching Set Finder (§2) to compute the set $T(\delta,\pi)$ as well as to verify whether it has a closed set under $P(\pi)$. If $T(\delta,\pi)$ has a closed set under $P(\pi)$ go to step (b). Otherwise, proceed.

(d) Solve a subsystem of (2) with $\eta = \pi$, g replaced by $g(\delta)$ and v_i replaced by $v(\delta)_i$ for $i \in \overline{T(\delta,\pi)}$, to obtain $[v(\pi)_i \mid i \in T(\delta,\pi)]$. Set $v(\pi)_i = v(\delta)_i$ for $i \in \overline{T(\delta,\pi)}$ and $g(\pi) = g(\delta)$.

The above specification of the policy evaluation step avoids parsing the matrices $P(\delta)$, $\delta \in \Delta$, into their subchains. In iterations which require an execution of step (c), two tests need to be performed requiring at most $[|\overline{S(\delta,\pi)}| + 1]^2$ Boolean operations. Finally, iterations in which step (d) is executed, require the solution of a system of $|T(\delta,\pi)|$ equations and unknowns, as opposed to the traditional version of the policy evaluation step in which (1) and (2), a system of $2N$ equations, must be solved.

2. The Modified Policy Evaluation Step

Step (c) of the modified policy evaluation step requires two tests, both of which involve the determination of the reaching set of some subset S of Ω. This set can be found by considering the network (G, V) with the set of nodes $G = \{i \mid i \in \bar{S}\} U \{0\}$ and the set of arcs V defined by:

$$(i,j) \in V \Leftrightarrow \begin{cases} P(\delta)_{ij} > 0 & \text{for } j \neq 0; i \in \bar{S}; \\ \sum_{l \in S} P(\delta)_{il} > 0 & \text{for } j = 0; i \in \bar{S}. \end{cases}$$

Finally, let all arc lengths be equal to 0.

The Reaching Set Finder

The reaching set of S adds to S those nodes in $\bar{S}$ for which a path to node 0 exists, i.e., for which the shortest path to node 0 has length 0. Consequently, this set can be obtained by applying Dijkstra's algorithm (cf. [6]) to the above network, the number of required operations being bounded by $|V|$ or $1/2[|\bar{S}| + 1]^2$. Moreover, since all arc lengths equal 0, Dijkstra's algorithm requires Boolean operations only.

Let π be the successor policy of δ in the improvement part of the PIA. We distinguish between the following two cases:

C1. (i) $P(\pi)g(\delta) > g(\delta)$ or (ii) $P(\pi)g(\delta) = g(\delta)$ and $T(\delta,\pi)$ contains a closed set under $P(\pi)$.

C2. $P(\pi)g(\delta) = g(\delta)$ and $T(\delta,\pi)$ does not contain a closed set under $P(\pi)$.

Before stating the following theorem we observe

LEMMA 1. *Let $\delta, \pi \in \Delta$.*

(a) *$T(\delta,\pi)$ has a closed set under $P(\pi)$ if and only if the reaching set of $\overline{T(\delta,\pi)}$ under $P(\pi)$ is a strict subset of Ω.*

(b) *Any closed set of $T(\delta,\pi)$ under $P(\pi)$ intersects $S(\delta,\pi)$.*

PROOF. (a) To verify the if part (the reverse implication being immediate) note that the states in $T(\delta,\pi)$ that do not reach $\overline{T(\delta,\pi)}$ constitute a closed set under $P(\pi)$.

(b) Immediate from the fact that every state in $T(\delta,\pi)$ has access to $S(\delta,\pi)$. Q.E.D.

We conclude from part (a) that verification of C_1 or C_2, as required in step (c) of our algorithm, can be achieved via the Reaching Set Finder as well, requiring $1/2(|T(\delta,\pi)| + 1)^2$ Boolean operations at the most. Part (b) of the above lemma indicates that under C_1(ii), as opposed to C_2, the second test quantity vector $Y(\delta,\pi)$ has a strictly positive component for some state(s) $\in R(\pi)$. The following theorem proves that the distinction between C_1 and C_2 thus discriminates between the cases $g(\pi) > g(\delta)$ and $g(\pi) = g(\delta)$.

THEOREM 1. *Let π be the successor policy of δ in the improvement step of* PIA.

(a) *Suppose C_1 holds. Then $g(\pi) > g(\delta)$.*

(b) *Suppose C_2 holds. Then $g(\pi) = g(\delta)$. Moreover, under C_2, there exists a (unique) relative value vector $v(\pi)$ with $v(\pi)_i = v(\delta)_i$ for $i \in \overline{T(\delta,\pi)}$. This vector can be found by solving a subsystem of* (2) *with $\delta = \pi$, g replaced by $g(\delta)$ and v_i replaced by $v(\delta)_i$ for $i \in \overline{T(\delta,\pi)}$. Finally, $v(\pi)_i > v(\delta)_i$ for $i \in T(\delta,\pi)$.*

PROOF. (a) Consider first the case $P(\pi)g(\delta) > g(\delta)$. Assuming to the contrary that $g(\pi) = g(\delta)$, leads to the contradiction $g(\delta) < P(\pi)g(\delta) = P(\pi)g(\pi) = g(\pi)$. Assume next that (ii) holds. Note first that $Y(\delta,\pi) \geqslant 0$. Assume to the contrary that $g(\pi) = g(\delta)$. In view of (2) and $\Pi(\pi)P(\pi) = \Pi(\pi)$ we have:

$$0 \leqslant \Pi(\pi)Y(\delta,\pi) = \Pi(\pi)\left[q(\pi) - H(\pi)g(\delta) + (P(\pi) - I)v(\pi)\right]$$
$$- \Pi(\pi)\left[P(\pi) - I\right]\left[v(\pi) - v(\delta)\right] = 0 - 0 = 0$$

so $Y(\delta,\pi)_i = 0$ for $i \in R(\pi)$.

In view of part (b) of Lemma 1, $Y(\delta,\pi)_i = 0$ for some $i \in S(\delta,\pi)$ thus contradicting the definition of $S(\delta,\pi)$.

(b) Under C_2, $T(\delta,\pi) \subseteq \overline{R(\pi)}$ and hence $[g(\delta)_i, v(\delta)_i \,|\, i \in R(\pi)]$ satisfy:

$$g_i = P(\pi)g_i, \qquad i \in R(\pi),$$
$$v_i = q(\pi)_i - H(\pi)g_i + P(\pi)v_i, \qquad i \in R(\pi).$$

The g-part of this system being uniquely determined as $g_i = g(\pi)_i, i \in R(\pi)$, it follows that

$$g(\delta)_i = g(\pi)_i, \qquad i \in R(\pi). \tag{5}$$

Since $g(\delta) = P(\pi)g(\delta)$, we have $g(\delta) = P^n(\pi)g(\delta)$ by repeated substitutions, and hence, taking the Cesáro limit, $g(\delta) = \Pi(\pi)g(\delta) = \Pi(\pi)g(\pi) = g(\pi)$, the middle equality following from (5), and the proof of the last equality being identical to the proof of the first equality. We thus proved the first assertion. Next, consider the system (1), (2) for $\eta = \pi$. From Lemma 1(a), $\overline{T(\delta,\pi)}$ is closed under $P(\pi)$, so that the equations corresponding to the states in $\overline{T(\delta,\pi)}$ form a subsystem of (1), (2). Moreover, since $\pi(i) = \delta(i)$ for $i \in \overline{T(\delta,\pi)}$, $\{ g(\delta)_i, v(\delta)_i \,|\, i \in \overline{T(\delta,\pi)} \}$ satisfies this subsystem. Inserting $v_i = v(\delta)_i$ for $i \in \overline{T(\delta,\pi)}$ and $g = g(\delta) = g(\pi)$ in the remaining equations of (2), we observe that the remaining values $[v_i\ i \in T(\delta,\pi)]$ are uniquely determined in view of $Q(\pi) = [P(\pi)_{ij}]_{i,j \in T(\delta,\pi)}$ being transient, i.e., $I - Q(\pi)$ being invertible. Finally equation (3.3) in [10] shows $v(\pi) - v(\delta) = \sum_{n=0}^{\infty} P^n(\pi) Y(\delta,\pi)$ thereby proving the last assertion. Q.E.D.

EXAMPLE. Let $\Omega = \{1,2\}, K(1) = \{1\}, K(2) = \{1,2,3\}$. The following table specifies the parameters:

Let δ^k $(k = 1,2,3)$ be the policy which selects alternative k in state 2. Note that $T(\delta^1,\delta^2) = T(\delta^1,\delta^3) = \{2\}$ contains a closed set under $P(\delta^3)$ but not under $P(\delta^2)$. Verify that $g(\delta^2) = g(\delta^1)$ and $g(\delta^3) > g(\delta^1)$.

TABLE

i	k	P^k_{i1}	P^k_{i2}	q^k_i
1	1	1	0	0
2	1	1	0	0
2	2	1	0	1
2	3	0	1	1

We next discuss the algorithmic implications of this theorem. Under C1, an *arbitrary* solution to the system (1), (2) suffices, since $g(\pi) > g(\delta)$ excludes the possibility of cycling back to δ. Traditionally, the literature suggested specifying a *particular* solution to this system, by adding an extra linear constraint on the v-components in each of the subchains of $P(\pi)$ (cf. rules A.1 and A.2). This procedure, once again, has the disad\ intage of having to determine the chain structure of $P(\pi)$. Instead, however, since we are merely interested in an arbitrary solution, a modified Gaussian elimination procedure is presented in section 3 (this corresponds to the determination of a $\{1\}$–pseudo inverse matrix, cf. [1, Chapter 1, §2]).

Under C2, part (b) of the above theorem shows that $g(\pi)$ and a relative value vector $v(\pi)$ can be obtained by solving a transient subsystem of (2) with $|T(\delta,\pi)|$ equations. Moreover, this relative value vector $v(\pi) > v(\delta)$, thereby excluding the possibility of cycling back to δ under C2 as well. Note finally, that the choice of the relative value vector under C2 corresponds to rule A, whereas under C1 a potential deviation from rule A is "unharmful," in view of the strict inequality $g(\pi) > g(\delta)$.

3. A Modified Gaussian Elimination Procedure

This section describes how a solution to the system (1), (2) can be obtained without an a priori investigation of the chain structure of $P(\eta)$. Our procedure is a modification of the Gaussian elimination procedure for solving systems of linear equations. Recall that the Gaussian elimination procedure applied to the matrix $(I - P(\eta))$,

$\eta \in \Delta$, generates a lower triangular matrix $L(\eta)$ and a permutation matrix $F(\eta)$ such that $L(\eta)[I - P(\eta)]F(\eta)$ is an upper triangular matrix, i.e.,

$$L(\eta)(I - P(\eta))F(\eta) = \left[\begin{array}{c|c} Q_1 & Q_2 \\ \hline 0 & 0 \end{array}\right]$$

where Q_1 is an $(N - n(\eta)) \times (N - n(\eta))$-upper triangular matrix and Q_2 is an $(N - n(\eta)) \times n(\eta)$-matrix. We use the shorthand notation $F(\eta)[I] = J$ for $I, J \subseteq \Omega$, if $F(\eta)e(I) = e(J)$ where $e(I), e(J)$ represent the characteristic vectors of I, J, i.e.,

$$e(I)_i = \begin{bmatrix} 1 & \text{if } i \in I, \\ 0 & \text{otherwise.} \end{bmatrix}$$

We next present our modified Gaussian elimination procedure for solving the system (1), (2).

Modified Gaussian elimination procedure

Step 1. Apply the Gaussian elimination procedure to the matrix $[I - P(\eta)]$, and determine the lower triangle matrix $L(\eta)$. Store the set of indices $\theta = F(\eta)[\{N - n(\eta) + 1, \ldots, N\}]$ representing the components appearing in the last $n(\eta)$ columns at the completion of the Gaussian elimination procedure.

Step 2. Compute $[\frac{H_1}{H_2}] = L(\eta)H(\eta)$ and $[\frac{q^{(1)}}{q^{(2)}}] = L(\eta)q(\eta)$, $H^{(1)}$ and $q^{(1)}$ representing the first $(N - n(\eta))$-rows and $H^{(2)}$ and $q^{(2)}$ representing the last $n(\eta)$ rows. Define $\bar{g} = F(\eta)^{-1}g$ and $\bar{v} = F(\eta)^{-1}v$ and partition $\bar{v} = [\frac{\bar{v}^{(1)}}{\bar{v}^{(2)}}]$, $\bar{v}^{(1)}$ representing the first $N - n(\eta)$ components.

Step 3. Solve the $N \times N$ system

$$\left[\begin{array}{c|c} Q_1 & Q_2 \\ \hline \multicolumn{2}{c}{H^{(2)}} \end{array}\right] \bar{g} = \left[\begin{array}{c} 0 \\ \hline q^{(2)} \end{array}\right].$$

Step 4. Assign arbitrary values for the components of $\bar{v}^{(2)}$ and solve the *triangular* $(N - n(\eta)) \times (N - n(\eta))$ system $Q_1\bar{v}^{(1)} = q^{(1)} - H^{(1)}\bar{g} - Q_2\bar{v}^{(2)}$ to determine the remaining components of v.

THEOREM 2. *Let* $\eta \in \Delta$.

(a) *For any choice of values* $\{v_i, i \in \theta\}$, *i.e., for any choice of the components of* $\bar{v}^{(2)}$ *the system* (1), (2) *is uniquely determined.*

(b) *Determination of the set* θ *and decomposition of the system* (1), (2) *into two subsystems of* N *and* $N - n(\eta)$ *equations respectively can be achieved via the modified Gaussian procedure.*

PROOF. Rewrite (1), (2) as

$$\begin{bmatrix} I - P(\eta) & 0 \\ H(\eta) & I - P(\eta) \end{bmatrix} \begin{bmatrix} g \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ q(\eta) \end{bmatrix};$$

premultiply with

$$\begin{bmatrix} L(\eta) & 0 \\ 0 & L(\eta) \end{bmatrix}$$

and insert

$$\begin{bmatrix} F(\eta) & 0 \\ 0 & F(\eta) \end{bmatrix}\begin{bmatrix} F(\eta)^{-1} & 0 \\ 0 & F(\eta)^{-1} \end{bmatrix}$$

to obtain

$$\left[\begin{array}{c:c|c:c} Q_1 & Q_2 & \multicolumn{2}{c}{0} \\ \hdashline 0 & 0 & & \\ \hline \multicolumn{2}{c|}{H^{(1)}} & Q_1 & Q_2 \\ \hdashline \multicolumn{2}{c|}{H^{(2)}} & 0 & 0 \end{array}\right]\left[\begin{array}{c} \bar{g} \\ \hline \bar{v} \end{array}\right] = \left[\frac{0}{L(\eta)q(\eta)}\right]. \qquad (6\ )$$

We first verify that the system

$$\left[\frac{Q_1 \;\vdots\; Q_2}{H^{(2)}}\right]\bar{g} = \left[\frac{0}{q^{(2)}}\right]$$

determines g uniquely. Assume to the contrary that two solutions $\bar{g}$ and $\bar{g}'$ exist. (6) then shows that both for $\bar{g}$ and $\bar{g}'$, a vector $\bar{v}$ exists satisfying (6) and hence (2). This contradicts the fact that every solution pair (g, v) to (1), (2) has $g = g(\eta)$. The validity of the remaining steps in the modified elimination procedure follow immediately from (6). In particular part (a) of this theorem follows by writing the second part of the system (6) as $Q_1\bar{v}^{(1)} = -H^{(1)}\bar{g} - Q_2\bar{v}^{(2)} + q^{(1)}$ with Q_1 an upper triangular and nonsingular matrix.[1]

[1] We wish to thank Eric Denardo as well as the Associate Editor, Martin Puterman, for many useful comments and suggestions.

References

1. Ben-Israel, A. and Greville, T., *Generalized inverses; theory and applications*, Wiley, New York, 1974.
2. Blackwell, D., "Discrete Dynamic Programming," *Ann. Math. Statist.*, Vol. 33 (1962), pp. 719–726.
3. Denardo, E., *Dynamic Programming: Theory and Application*, Prentice-Hall, Englewood Cliffs, N.J. (forthcoming).
4. ———, "Markov Renewal Programs with Small Interest Rates," *Amer. Math. Statist.*, Vol. 42 (1971), pp. 477–496.
5. ——— and Fox, B., "Multichain Markov Renewal Programs," *SIAM J. Appl. Math.*, Vol. 16 (1968), pp. 468–487.
6. Dijkstra, E., "A Note on Two Problems in Connection with Graphs," *Numer. Math.*, Vol. 1 (1959), pp. 269–271.
7. Fox, B. and Landi, D., "An algorithm for Identifying the Ergodic Subchains and Transient States of a Stochastic Matrix," *Comm. ACM*, Vol. 11 (1968), pp. 619–621.
8. Howard, R., *Dynamic Programming and Markov Processes*, Wiley, New York, 1960.
9. Jewell, W., "Markov Renewal Programming," *Operations Res.*, Vol. 11 (1963), pp. 938–972.
10. Schweitzer, P. J. and Federgruen, A., "Foolproof Convergence in Multichain Policy Iteration," *J. Math. Anal. Appl.*, Vol. 64 (1978), pp. 360–368.
11. ——— and ———, "Functional Equations of Undiscounted Markov Renewal Programming." *Math. Operations Res.*, Vol. 3 (1978), pp. 308–322.
12. Veinott, A. F., Jr., "Discrete Dynamic Programming with Sensitive Discount Optimality Criteria," *Ann. Math. Statist.*, Vol. 40 (1969), pp. 1635–1660.