November, 1993

LIDS- P 2212

Research Supported By:

NSF CCR-9103804

Generic Rank-One Corrections for Value Iteration in Markovian Decision Problems

Bertsekas, D.P.

November 1993 LIDS-P-2212

GENERIC RANK-ONE CORRECTIONS FOR VALUE ITERATION[1] IN MARKOVIAN DECISION PROBLEMS

by

Dimitri P. Bertsekas[2]

Abstract

Given a linear iteration of the form $x := F(x)$, we consider modified versions of the form $x := F(x+\gamma d)$, where d is a fixed direction, and γ is chosen to minimize the norm of the residual $\|x + \gamma d - F(x + \gamma d)\|$. We propose ways to choose d so that the convergence rate of the modified iteration is governed by the subdominant eigenvalue of the original. In the special case where F relates to a Markovian decision problem, we obtain a new extrapolation method for value iteration. In particular, our method accelerates the Gauss-Seidel version of the value iteration method for discounted problems in the same way that McQueen's error bounds accelerate the standard version. Furthermore, our method applies equally well to undiscounted problems.

[1] Research supported by NSF under Grant CCR-9103804.

[2] Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, Mass., 02139.

1. INTRODUCTION

Consider a linear iteration of the form $x := F(x)$, where

$$F(x) = h + Qx, \tag{1}$$

Q is a given $n \times n$ matrix with eigenvalues strictly within the unit circle, and h is a given vector in $\Re^n$. Let x^* be the unique fixed point of F. We focus on modified iterations of the form

$$x := F(x + \tilde{\gamma}d) = F(x) + \tilde{\gamma}z,$$

where

$$z = Qd, \tag{2}$$

and $\tilde{\gamma}$ is obtained by minimizing over γ

$$\big\|x + \gamma d - (F(x) + \gamma z)\big\|.$$

(In our notation, $\|\cdot\|$ is the standard norm in the n-dimensional Euclidean space $\Re^n$. Furthermore, all vectors in this paper are viewed as column vectors, and prime denotes transposition.) It is straightforward to show that

$$\tilde{\gamma} = \frac{(d-z)'\big(F(x)-x\big)}{\|d-z\|^2}. \tag{3}$$

We write the iteration $x := F(x + \tilde{\gamma}d)$ as

$$x := M(x),$$

where

$$M(x) = F(x) + \tilde{\gamma}z, \tag{4}$$

and we note that it requires only slightly more computation than the regular iteration $x := F(x)$, since the vector z is computed once and the computation of $\tilde{\gamma}$ is simple. However, the iteration $x := M(x)$ need not converge to x^* when the direction d is chosen arbitrarily.

Extrapolation methods of the form $x := M(x)$ have been considered in the context of Markovian decision problems starting with the work of McQueen [McQ66] for discounted problems, and followed by many others; see the surveys [Por81a], [Put90], and the textbook presentation [Ber87]. (A Markovian decision problem is referred to as discounted in this paper if all the row sums of Q are strictly less than one; otherwise it is referred to as undiscounted.) In particular, when $Q = \alpha P$ where $\alpha \in (0,1)$ is a discount factor, P is a stochastic matrix, and d is the unit vector $e = (1, 1, \ldots, 1)$,

it is known [Mor71] that the iteration $x_{k+1} = M(x_k)$ converges geometrically at a rate governed by the subdominant eigenvalue of Q. (By this we mean that for every s that is larger than the second largest eigenvalue modulus of Q, there is a $c > 0$ such that $\|x_k - x^*\| \leq cs^k$ for all k.) This method is often much more effective than the ordinary value iteration method $x_{k+1} = F(x_k)$ that converges geometrically at a rate governed by α, the dominant eigenvalue of Q.

Additional rank-one and higher-rank extrapolation methods have been considered by Porteus and by Totten [Por75], [Por81b], [PoT78], [Tot71], in connection with other types of value iteration methods for problems involving a matrix $Q \neq \alpha P$ (such as Gauss-Seidel with and without row reordering). Of the methods in these works, the ones that are closest to ours are based on L_2 norm extrapolation [PoT78], and use a correction of $F(x)$ along the unit vector e, or along the subspace spanned by e and $F(x) - x$ (every two iterations), or along the subspace spanned by e, and $\big(F^2(x) - F(x)\big) - \big(F(x) - x\big)$ (every three iterations), supplemented with an overrelaxation factor. No theoretical convergence result was provided, but in tests with some randomly generated problems these methods required relatively few iterations [PoT78].

The purpose of this note is to recommend a new and simple method for choosing d, which guarantees convergence, and achieves comparable acceleration to that provided by McQueen's bounds for discounted problems. Our method applies to a broad class of problems, including undiscounted problems for which no effective rank-one acceleration method with guaranteed convergence is currently available.

Our main observation is that if d is chosen to be an eigenvector of Q, then extrapolation along d nulifies the effect of the corresponding eigenvalue in the convergence rate of the iteration $x := M(x)$ (Prop. 1 in the next section). In particular, if d is an eigenvector corresponding to a dominant simple eigenvalue of Q, then this iteration converges at a rate governed by the subdominant eigenvalue. This result holds for any matrix Q that has a real eigenvector corresponding to a dominant eigenvalue with modulus less than one. We thus propose using such an eigenvector as the vector d in the extrapolation scheme $x := F(x) + \tilde{\gamma} Qd$ [cf. Eqs. (1)-(4)].

A first difficulty with our approach is that it assumes the existence of a real eigenvector that corresponds to a maximal modulus eigenvalue. For Markovian decision problems where the matrix Q has nonnegative elements, this is not an issue in view of the Perron-Frobenius theorem. A second difficulty with our approach is that it requires finding the eigenvector d. This can be done approximately, however, by using the power method, that is, by applying F a sufficiently large number of times k to some vector x to obtain $F^k(x)$, and estimating d as the normalized residual

$$d \approx \frac{F^k(x) - F^{k-1}(x)}{\|F^k(x) - F^{k-1}(x)\|}. \tag{5}$$

In particular, let $\lambda_1, \ldots, \lambda_m$ be the eigenvalues of Q. Suppose that

$$|\lambda_j| < |\lambda_1| < 1, \qquad \forall\, j = 2, \ldots, m$$

and that the eigenspace corresponding to λ_1 has dimension equal to the multiplicity of λ_1. The initial error $x - x^*$ can then be decomposed as

$$x - x^* = \sum_{j=1}^{m} \xi_j e_j,$$

where each e_j is a vector in the eigenspace of the corresponding eigenvalue λ_j, and $\xi_1, \ldots, \xi_m$ are some scalars. The residual $F^k(x) - F^{k-1}(x)$ can be written as

$$F^k(x) - F^{k-1}(x) = Q^{k-1}\big(F(x) - x\big) = Q^{k-1}\big(F(x) - F(x^*) - (x - x^*)\big) = Q^{k-1}(Q - I)(x - x^*),$$

so it will be nearly equal to $\xi_1 \lambda_1^{k-1}(\lambda_1 - 1)e_1$ for large k, implying that the vector $d = e_1/\|e_1\|$ can be obtained approximately from Eq. (5). In order to decide whether k has been chosen large enough, one can test to see if the successive residuals $F^k(x) - F^{k-1}(x)$ and $F^{k-1}(x) - F^{k-2}(x)$ are very close to being aligned; if this is so, the components of $F^k(x) - F^{k-1}(x)$ along the eigenspace elements $e_2, \ldots, e_m$ must also be very small.

We thus suggest a two-phase approach: in the first phase, we apply several times the regular iteration $x := F(x)$ both to improve our estimate of x and also to obtain an estimate d of an eigenvector corresponding to a dominant eigenvalue; in the second phase we use the modified iteration $x := M(x)$ that involves extrapolation along d. It can be shown that the two-phase method converges to x^* provided the error in the estimation of d is small enough, that is, the absolute value of the cosine of the angle between d and Qd as measured by the ratio

$$\frac{\big|\big(F^k(x) - F^{k-1}(x)\big)'\big(F^{k-1}(x) - F^{k-2}(x)\big)\big|}{\big\|F^k(x) - F^{k-1}(x)\big\| \cdot \big\|F^{k-1}(x) - F^{k-2}(x)\big\|} \tag{6}$$

is sufficiently close to one. This approach turned out to be practically feasible and often surprisingly effective in our computational experiments, as reported in Section 3.

Note that the computation of the first phase is not wasted since it uses the regular iteration $x := F(x)$ that we are trying to accelerate. Furthermore, since the second phase involves the calculation of $F(x)$ at the current iterate x, any error bounds or termination criteria based on $F(x)$ can be used to terminate the algorithm. As a result, the same finite termination mechanism can be used for both iterations $x := F(x)$ and $x := M(x)$. Thus our approach can be considered successful as long as by passing onto the second phase, we end up doing fewer iterations up to termination than if we were to continue exclusively with the first phase.

We mention, however, that our method is ineffective if there is little or no separation between the dominant and the subdominant eigenvalue modulii, both because the convergence rate of the power method for obtaining d is slow, and also because the convergence rate of the modified iteration $x := M(x)$ is not much faster than the one of the regular iteration $x := F(x)$. Such problems are not suitable for rank-one correction methods that use a fixed direction d, but it is possible that they can be dealt with effectively through the use of adaptive low-rank aggregation methods, such as those proposed in [BeC89].

Another shortcoming of the two-phase method outlined above when applied to Markovian decision problems is that it assumes a fixed policy. In the case of optimization over several policies, the mapping F has the form

$$F_i(x) = \min_{u \in U(i)} \left\{ h_i(u) + \sum_{j=1}^{n} q_{ij}(u) x_j \right\}, \qquad i = 1, \ldots, n, \tag{7}$$

where $U(i)$ is a finite set of control actions for each state i. One can then use our approach in two different ways:

(1) Compute iteratively the cost vectors of the policies generated by a policy iteration scheme (see e.g. [Ber87]).

(2) Guess at an optimal policy within the first phase, switch to the second phase, and then return to the first phase if the policy changes "substantially" during the second phase. In particular, in the first phase, the ordinary value iteration $x := F(x)$ is used, where F is the nonlinear mapping (7), and a switch to the second phase occurs, when the ratio (6) gets sufficiently close to one. The vector z is taken to be equal to Q^*d, where d is obtained from Eq. (5), and Q^* is the matrix whose ith row corresponds to the minimizing control in Eq. (7) at the time of the switch. The second phase consists of the iteration $x := F(x) + \tilde{\gamma} z$, where $\tilde{\gamma}$ is given by Eq. (3). To guard against subsequent changes in policy, which induce corresponding changes in the matrix Q^*, one should ensure that the method is working properly, for example, by recomputing d if the policy changes and/or the error $\|F(x) - x\|$ is not reduced at a satisfactory rate. Based on our computational experiments, this method seems to be workable (and can lead to significant savings) because the value iteration method typically finds an optimal policy much before it finds the optimal cost vector.

2. MAIN RESULT

The following proposition gives our main result and provides the basis for the two-phase method described in the preceding section.

Proposition 1: Consider the iteration $x := M(x)$ defined by Eqs. (1)-(4). Assume that d is a real eigenvector of Q and let λ_1 be the corresponding eigenvalue.

(a) $M(x)$ can be written as

$$M(x) = g + Rx,$$

where g is some vector in $\Re^n$ and

$$R = Q + \frac{\lambda_1}{(1-\lambda_1)\|d\|^2} dd'(Q - I). \tag{8}$$

Furthermore, $Rd = 0$ and for all k we have

$$R^k = RQ^{k-1},$$

so the iteration $x := M(x)$ converges to x^*.

(b) Let $\lambda_2, \ldots, \lambda_m$ be the remaining eigenvalues of Q, and assume that

$$|\lambda_j| < |\lambda_1| < 1, \qquad \forall\, j = 2, \ldots, m.$$

Suppose that a vector x can be written as

$$x = x^* + \xi_1 d + \sum_{j=2}^{n} \xi_j e_j, \tag{9}$$

where each e_j is a vector in the eigenspace of the corresponding eigenvalue λ_j, and $\xi_1, \ldots, \xi_m$ are some scalars. Then for all $k > 1$

$$M^k(x) = x^* + \sum_{j=2}^{n} \xi_j RQ^{k-1} e_j, \tag{10}$$

so $M^k(x)$ converges to x^* at a geometric rate governed by the subdominant eigenvalue of Q.

Proof: (a) By straightforward calculation using Eqs. (1)-(4), we have for any d with $d \neq z$,

$$\begin{aligned} M(x) &= F(x) + \tilde{\gamma} z \\ &= h + Qx + \frac{(d-z)'(h + Qx - x)}{\|d-z\|^2} z \\ &= h + \frac{z(d-z)'}{\|d-z\|^2} h + Qx + \frac{z(d-z)'(Q-I)}{\|d-z\|^2} x. \end{aligned}$$

For the particular choice of d assumed here, we have $z = Qd = \lambda_1 d$, so we obtain

$$M(x) = h + \frac{\lambda_1 dd'}{(1-\lambda_1)\|d\|^2}h + Qx + \frac{\lambda_1 dd'}{(1-\lambda_1)\|d\|^2}(Q-I)x.$$

By letting

$$g = \left(I + \frac{\lambda_1 dd'}{(1-\lambda_1)\|d\|^2}\right)h,$$

and R as given by Eq. (8), we obtain the form $M(x) = g + Rx$.

The relation $Rd = 0$ is easily verified using Eq. (8) and the fact $Qd = \lambda_1 d$. Finally, to show the relation $R^k = RQ^{k-1}$, we first show it for $k = 2$ by using Eq. (8), the fact $Rd = 0$, and the calculation

$$R^2 = R\left(Q + \frac{\lambda_1}{(1-\lambda_1)\|d\|^2}dd'(Q-I)\right) = RQ,$$

and we then show it for all k by using $R^2 = RQ$ and the calculation

$$R^k = R^{k-2}R^2 = R^{k-2}RQ = R^{k-3}R^2Q = R^{k-3}RQ^2 = \cdots = RQ^{k-1}.$$

(b) Equation (10) follows from Eq. (9), and the fact $Rd = 0$ and $M(x^*) = x^*$, which has been proved in part (a). Furthermore, by the properties of eigenspace vectors [LaT85], each sequence $\{\|RQ^{k-1}e_j\|\}$ converges geometrically at a rate governed by $|\lambda_j|$. **Q.E.D.**

We note that there is a multidimensional version of the above proposition. In particular, let D be a full-rank $n \times m$ matrix, and consider the iteration

$$x := M_D(x) = F(x + D\tilde{\gamma}),$$

where $\tilde{\gamma}$ is the vector in $\Re^m$ that minimizes the residual norm

$$\|x + D\gamma - F(x + D\gamma)\|$$

over all vectors $\gamma \in \Re^m$. It is easily verified that

$$\tilde{\gamma} = \big((D-Z)'(D-Z)\big)^{-1}(D-Z)'\big(F(x)-x\big),$$

where

$$Z = QD.$$

Furthermore, a straightforward calculation shows that $M_D(x)$ has the form

$$M_D(x) = g_D + R_D x,$$

where g_D is some vector and the $n \times n$ matrix R_D is given by

$$R_D = Q + Z\big((D-Z)'(D-Z)\big)^{-1}(D-Z)'(Q-I). \tag{11}$$

From this formula and the definition $Z = QD$, it is seen that

$$R_D D = 0.$$

Suppose now that the range space of D is invariant under multiplication with Q, that is, for every column d of Q, the vector Qd is a linear combination of columns of Q; this is true for example if the columns of D are eigenvectors of Q or, more generally, if the range space of D is the direct sum of eigenspaces of Q. Then the columns of Z are linear combinations of the columns of D, which combined with $R_D D = 0$ implies that

$$R_D Z = 0.$$

It follows from Eq. (11) that $R_D^2 = R_D Q$, and more generally that

$$R_D^k = R_D Q^{k-1},$$

generalizing part (a) of Prop. 1. Similar to part (b) of Prop. 1, it follows that the iteration $x := M_D(x)$ converges to x^* and the convergence rate is governed by the eigenvalues of Q other the ones corresponding to the range space of D. This result may be useful when Q has multiple dominant eigenvalues if a suitable matrix D can be identified. One possibility is to use as the columns of D a sufficient number of successive residuals $F^k(x) - F^{k-1}(x)$, after a number of iterations k that is sufficiently large. However, we have not investigated this possibility further.

3. COMPUTATIONAL RESULTS FOR STOCHASTIC SHORTEST PATHS

To assess the potential of our two-phase method, we have tested it with a variety of Markovian decision problems. In this section we will present some computational results for stochastic shortest path problems (also known as *first passage problems*). These are undiscounted problems, originally introduced in [EaZ62], and investigated in several subsequent works [Ber87], [BeT89], [BeT91], [Der70], [Kus71], [Pal67]. For these problems, there has been no proposal to date of a simple and effective method to accelerate the convergence of value iteration. We have also obtained similar results for discounted problems, but for such problems we have found that our method is not much better than the regular value iteration method, supplemented with McQueen-like error bounds.

In summary, we have verified that for stochastic shortest path problems the acceleration potential of the method depends on the problem' s structure, and particularly on the separation between dominant and subdominant eigenvalues. When this separation is substantial, and we will see that this happens in some fairly "normal" randomly generated problems, the resulting acceleration is spectacular.

Let us denote by q_{ij}, $i,j = 1,\ldots,n$ the elements of Q. In the context of the stochastic shortest path problem, the elements q_{ij} are nonnegative and all the row sums $\sum_{j=1}^{n} q_{ij}$ are less or equal to one. We may view q_{ij} as the probability of a system moving from state i to state j, and we may view $1-\sum_{j=1}^{n} q_{ij}$ as the probability of the system moving from i to a cost-free and absorbing termination state. If the ith component of the vector h is the expected cost when moving from state i, then the components of x^* are the expected costs starting from the corresponding states up to reaching the termination state.

We have tested two versions of the two-phase method, called *Jacobi* and *Gauss-Seidel*. The Jabobi version corresponds to the mapping F with components

$$F_i(x) = h_i + \sum_{j=1}^{n} q_{ij}x_j, \qquad i = 1,\ldots,n. \tag{12}$$

The Gauss-Seidel version corresponds to the mapping F with components

$$F_i(x) = h_i + \sum_{j=1}^{i-1} q_{ij}F_j(x) + \sum_{j=i}^{n} q_{ij}x_j, \qquad i = 1,\ldots,n. \tag{13}$$

In all tests the switch to phase two (the rank-one correction iteration) was made when the cosine of the angle betwen successive residuals, as measured by the ratio (6), was within 10^{-4} of unity. The iterations were terminated when the residual norm $||F(x)-x||$ became less than 10^{-7}.

In all our problems the components of the cost vector h were chosen according to a uniform distribution from the interval $[0,100]$. We used three types of randomly generated problems, the first two of which involve a fixed policy:

(1) **Random Transition Graphs**: Here each transition probability q_{ij} is specified to be 0 or positive according to a given probability r, called the *sparsity factor*. Each of the *escape probabilitities*, that is, the probabilities $1-\sum_{j=1}^{n} q_{ij}$ of transition from i to the termination state is selected to be either a fixed positive number $p < 1$, or 0 with probabilities r and $1-r$, respectively. The positive q_{ij} are then selected according to a uniform distribution, and they are appropriately normalized, taking into account the escape probabilities specified earlier.

(2) **Linear Transition Graphs**: Here for each state $i \neq 1,n$ there are two possible transitions, the *left* transition to a fixed state randomly chosen from the set $\{1,\ldots,i-1\}$, and the *right*

transition to a fixed state randomly chosen from the set $\{i+1,\ldots,n\}$. The left and the right transition probabilities are randomly chosen from the interval $[0,1]$ and then are normalized to add to one. From the state 1, there is a fixed probability p, called the *escape probability*, of moving to the termination state, and a probability $1-p$ of moving to state 2. Similarly, from the state n, there is a given probability p, called the *escape probability*, of moving to the termination state, and a probability $1-p$ of moving to state $n-1$.

(3) **Two-Action Linear Transition Graphs**: Here the states and the possible transitions at each state are as in the preceding class of problems. However, at each state there are two possible actions: when the first action is chosen the state evolves probabilistically as in the preceding class of problems; when the second action is chosen at a state $i \neq 1,n$, the left and the right transitions occur with equal probability $1/2$. We implemented a heuristic mechanism whereby a switch from the first to the second phase and reversely can be done, depending on the progress of the algorithm. In particular, a switch from the second to the first phase was done when the second phase could not maintain a "substantial" reduction factor in the normed residual $\|F(x)-x\|$. Furthermore, a switch to the first phase was also done after the first five iterations of the second phase. The motivation for this latter switch was that frequently, following the initial switch to the second phase, the policy produced by value iteration changed significantly, in which case it is sensible to recalculate the vector d by switching back to the first phase.

In Tables 1-3, we give the number of iterations required by four methods. The first two are called *Jacobi-Acc* and *Jacobi*, and are based on the Jacobi iteration [cf. Eq. (12)]; the former uses the rank-one correction in the two-phase scheme described above, while the latter uses no corrections, that is, it consists of just phase one. The Gauss-Seidel versions [cf. Eq. (13)] of these two Jacobi methods are called *Gauss-Seidel-Acc* and *Gauss-Seidel*, respectively. Some of the larger problems were not solved with the regular Jacobi and Gauss-Seidel methods in view of the excessive number of iterations required.

The results of these tables show that the two-phase scheme is extremely effective, dramatically reducing the number of iterations of the regular Jacobi and Gauss-Seidel value iteration methods. This is not surprising, since similarly dramatic savings are known to be possible for discounted problems under comparable circumstances.

We also solved some of the problems of Tables 1-3 with the rank-one correction method that uses the unit vector $e=(1,1,\ldots,1)$ as the direction d, instead of using a dominant eigenvector. This method does not offer convergence guarantees, but nonetheless it accelerated considerably the regular value iteration method for the problems of Tables 1 and 2. However, the number of iterations

required was much larger than the number of iterations for our method, frequently by a factor of three or four. For the two-action-per-state problems of Table 3, we were not able to implement a properly working rank-one correction method with $d = e$, because of difficulties due to nonmonotonic changes in $||F(x) - x||$.

Finally, it is worth repeating our earlier warning that the two-phase scheme is not effective when there is little or no separation between the dominant and the subdominant eigenvalue modulii. As an example consider the linear transition graph problem with two states. The matrix Q is given by

$$Q = \begin{pmatrix} 0 & 1-p \\ 1-p & 0 \end{pmatrix}$$

and its two eigenvalues are $(1-p)$ and $-(1-p)$. When the two-phase Jacobi method is applied to this problem, the switch to phase two typically never occurs because the power method cannot identify a dominant eigenvector.

n	**Sparsity**	**Esc. Prob.**	**Jac.-Acc**	**G.-Seidel-Acc**	**Jac.**	**G.-Seidel**
75	1.0	0.01	12	14	2339	1221
150	1.0	0.01	11	15	2450	1245
225	1.0	0.01	11	16	2503	1274
300	1.0	0.01	10	16	2545	1314
75	0.1	0.01	395	52	22209	11631
150	0.1	0.01	129	21	21565	14318
225	0.1	0.01	146	17		
300	0.1	0.01	90	18		

Table 1: Experiments with random transition graph problems. Each entry gives the number of iterations averaged over 5 randomly generated problems. For such problems the subdominant eigenvalue modulus is small, particularly for dense problems. This explains the dramatic savings achieved by our rank-one correction method.

n	Esc. Prob.	Jac.-Acc	G.-Seidel-Acc	Jac.	G.-Seidel
100	0.1	109	57	3954	2024
200	0.1	173	97	5235	2767
300	0.1	210	86	6765	3545
400	0.1	131	67	7036	3617
500	0.1	238	82	8311	4185

Table 2: Experiments with linear transition graph problems. Each entry gives the number of iterations averaged over 5 randomly generated problems.

n	Esc. Prob.	Jac.-Acc	G.-Seidel-Acc	Jac.	G.-Seidel
100	0.1	105	59	2691	1308
200	0.1	124	72	2687	1296
300	0.1	125	71	3148	1565
400	0.1	117	69	4704	2278
500	0.1	129	73	4443	2126

Table 3: Experiments with two-action linear transition graph problems. Each entry gives the number of iterations averaged over 5 randomly generated problems.

REFERENCES

[Ber87] Bertsekas, D. P., Dynamic Programming: Deterministic and Stochastic Models, Prentice-Hall, Englewood Cliffs, NJ.

[BeC89] Bertsekas, D. P., and Castañon, D. A., "Adaptive Aggregation Methods for Infinite Horizon Dynamic Programmng," IEEE Trans. on Aut. Control, Vol. 34, pp. 589-598.

[BeT89] Bertsekas, D. P., and Tsitsiklis, J. N., Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ.

[BeT91] Bertsekas, D. P., and Tsitsiklis, J. N., "An Analysis of Stochastic Shortest Path Problems," Mathematics of Operations Research, Vol. 16, pp. 580-595.

[Der70] Derman, C., Finite State Markovian Decision Processes, Academic Press, N.Y., 1970.

[EaZ62] Eaton, J. H., and Zadeh, L. A., "Optimal Pursuit Strategies in Discrete State Probabilistic Systems," Trans. ASME Ser. D, J. Basic Eng., Vol. 84, pp. 23-29.

[Kus71] Kushner, H., Introduction to Stochastic Control, Holt, Rinehart, and Winston, N.Y., 1971.

[LaT85] Lancaster, P., and Tismenetsky, M., The Theory of Matrices, Academic Press, New York, NY.

[McQ66] McQueen, J., "A Modified Dynamic Programming Method for Markovian Decision Problems," J. Math. Anal. and Appl., Vol. 14, pp. 38-43.

[Mor71] Morton, T., "On the Asymptotic Convergence Rate of Cost Differences for Markovian Decision Processes," Operations Res., Vol. 19, pp. 244-248.

[Pal67] Pallu de la Barriere, R., Optimal Control Theory, Saunders, Phila.

[Por75] Porteus, E., "Bounds and Transformations for Finite Markovian Decision Chains," Operations Res., Vol. 23, pp. 761-784.

[Por81a] Porteus, E., "Overview of Iterative Methods for Discounted Finite Markov and Semi-Markov Decision Chains," Rec. Developments in Markov Decision Processes, R. Hartley, L. C. Thomas, and D. J. White (eds.), Academic Press, London.

[Por81b] Porteus, E., "Computing the Discounted Return in Markov and Semi-Markov Chains," Naval Research Logistics Quarterly, Vol. 28, pp. 567-577.

[PoT78] Porteus, E., and Totten, J., "Accelerated Computation of the Expected Discounted Return in a Markov Chain," Operations Res., Vol. 26, pp. 350-358.

[Put90] Puterman, M. L., "Markov Decision Processes," in Handbooks in OR and MS, Vol. 2, D. P. Heyman, and M. J. Sobel, (eds.), Elsevier Science Publishers, Amsterdam.

[Tot71] Totten, J., "Computational Methods for Finite State Finite Valued Markovian Decision Problems," ORC 71-9, Operations Research Center, Univ. of Calif., Berkeley.