ACCELERATED PROCEDURES FOR THE SOLUTION OF DISCRETE MARKOV CONTROL PROBLEMS

H. J. Kushner, et al

Brown University
Providence, Rhode Island

March 1970

Distributed... 'to foster, serve and promote the nation's economic development and technological advancement.'

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

This document has been approved for public release and sale.
accelerated Procedures for the Solution of
Discrete Markov Control Problems

by

H. J. Kushner*

and

A. J. Kleinman**

This document has been approved for public
release and sale; its distribution is unlimited.

This research was supported in part by the National Science Foundation under
Grant No. 16-2760, in part by the National Aeronautics and Space Administration
under Grant No. NGL 40-008-015 and in part by the Air Force Office of Scientific
Research under Grant No. AF49(638)-07-0052A, Brown Univ., Div. of Applied Math.
Providence, Rhode Island 02912

*This research was supported by the National Science Foundation under Grant No.
CE 2/189.
**This research was supported by the National Science Foundation under Grant No.
CE 2/189.
Accelerated procedures for computing optimal controls for a Markov chain model are discussed, and numerical results are presented. For the example, one of the methods described gave a 10 fold decrease in computation time over a more usual procedure of dynamic programming.
(C1) and (C2) below, the least cost and optimal control satisfy

\[(2) \quad V_1 = \min_{u_i} \left\{ \sum_{j=1}^{N} p_{i,j}(u_i) \nu_j + k(i,u_j) \right\} = \sum_{j=1}^{N} p_{i,j}(\bar{u}_i) \nu_j + k(i,\bar{u}_i) \]

or, in vector form

\[(3) \quad V = \min_{u} [R(u)\nu + K(u)] = R(\bar{u})V + K(\bar{u}). \]

Also, for any fixed control vector \( u_0 \),

\[(4) \quad V(u) = R(u)V(u) + K(u). \]

(C1) \( k(i,a) \) and \( p_{i,j}(a) \) are continuous for each \( i,j \) and \( a \) in the compact set \( U_1 \).

(C2) There is an \( \epsilon > 0 \) so that the \( N \) step transition probabilities satisfy \( p_{i,j}^{(N)}(u) \leq \epsilon > 0 \) for all \( u \in U_1 \times \ldots \times U_N \) and all \( i \).

(C3) can be weakened, but it is quite often satisfied by problems, such as the numerical example, which are derived from diffusion models. It states that for any fixed control policy, the target can eventually be reached, since if \( p_{i,j}^{(n)}(u) > 0 \) for some \( n \), then (C2) holds for \( n = N \).

In addition to the methods discussed here, there is the 'iteration (or approximation) in policy space' method. However, at least for the type of Markov problems which are derived as discretizations of continuous time problems, it seems to be inferior to the methods compared in this paper.

2. Iterative Methods for the Solution of (2)–(3).

The backward iterative procedure of dynamic programming (henceforth referred to as the Jacobi procedure, owing to the analogy to the Jacobi procedure for solving linear equations [2]) is

\[(5) \quad V_{n+1} = \min_{u} [R(u)\nu + K(u)] = R(u^n)V + K(u^n) \]

or, in vector form

\[(6) \quad V_{n+1} = \min_{u} \left[ \sum_{j=1}^{N} p_{i,j}(u_i) \nu_j + k(i,u_j) \right] \]

where \( v^n \) is the column vector \( (v_1^n, \ldots, v_N^n) \), and converges to \( V \) for any \( v^0 \).

In [1], it was shown that the Gauss-Seidel or successive substitution procedure

\[(6a) \quad V_{n+1} = \min_{u} \left[ \sum_{j=1}^{N} p_{i,j}(u_i) \nu_j + k(i,u_j) \right] \]

converges at least as fast as (5). Experience indicates that (6) is generally far superior to (5).

According to [1] (to which the reader is referred for more details),

(6) can be written as

\[(6b) \quad V_{n+1} = \min_{u} \left[ \sum_{j=1}^{N} p_{i,j}(u_i) \nu_j + k(i,u_j) \right] \]

or, in vector form,

\[(6c) \quad v_{n+1} = \min_{u} \left[ \sum_{j=1}^{N} p_{i,j}(u_i) v_j + k(i,u_j) \right] \]

\( u^n \) is used to denote the minimizing control at the \( n \)th iteration, for all procedures. The values may, of course, be different for different procedures, and different initial values \( v^0 \).
where
\[ q_{ij}(u) = P_{ij}(u) + \sum_{k=1}^{i-1} P_{ik}(u) q_{kj}(u), \quad j > i \geq 1 \]
\[ = P_{ij}(u), \quad \text{for } i = 1 \]
\[ = q_{ij}(u), \quad \text{for } j = i \geq 1 \]
\[ k(i,u) = \sum_{j=1}^{i-1} P_{ij}(u) \bar{R}(j,u) + \bar{R}(i,u). \]

\( q_i(u) \) is the transition matrix for a Markov chain \( Y_i \) (and \( q_0(u) = 1 \), \( q_{ij}(u) = 1 - \sum_{k=1}^{i} q_{ij}(u) \)). Indeed, \( Y_i \) is absorbed at a faster than \( Y_0 \), for each fixed control: this is the clue to the preferability of (6) over (5).

In (6), the ordering of the states is important, in that a judicious ordering can greatly improve the convergence. See [1], [3], for a discussion of preferable orderings.

Most useful state orderings have the property (see Theorem 2 [1])
\[ \sum_{l=1}^{n} P_{il}(u) = 1, \quad \forall \alpha < 1 \]
\[ P_{ij}(u) > 0 \quad \text{for some } 1 \leq j(i,u) < i-1 \quad \text{(for } i > 1) \]
\[ P_{ij}(u) > 0 \quad \text{for some } j(i,u) > 1. \]

(8) guarantees (9), and states that state 1 is always connected to state 0.

state 1 is always connected to some state below 1, and to some state other than the target. (C) is used (11) to show that \( Y_i \) is absorbed at a "faster" than \( Y_0 \). Note, for future reference, that under (C), for any \( u \), (1)
\[ l = q_{i0}(u) = \sum_{j=1}^{n} q_{ij}(u) < \sum_{j=1}^{n} P_{ij}(u) = 1 - P_{i0}(u) \]
\[ \text{for } i > 1 \]

(7)
\[ q_{i0}(u) = P_{i0}(u) > 0. \]

2. Accelerated Procedures

Next, the accelerated procedures are defined. Let \( \Omega \) denote a diagonal matrix whose \( i \)th diagonal entry is \( \alpha_i, \alpha_i > 1 \). The accelerated Jacobi (AJ) procedure is
\[ v_{i}^{n+1} = \min_{u} \{ \bar{R}(u)v_{i}^{n} + \bar{R}(u) + (I-\Omega)v_{i}^{n} \}
\]
\[ = \min_{u} \{ \bar{R}(u)v_{i}^{n} + \bar{R}(u) \}
\]
\[ \bar{R}(u) = \bar{R}(u) + I-\Omega. \]

The semi-accelerated Gauss-Seidel (SAGS) procedure is
\[ v_{i}^{n+1} = \min_{u} \{ \sum_{j=1}^{n} P_{ij}(u)v_{j}^{n} + k(i,u) \}
\]
\[ v_{i}^{n+1} = \alpha_i v_{i}^{n} + (1-\alpha_i) v_{i}^{n}. \]

The accelerated Gauss-Seidel procedure (AGS) is
\[
V_{i+1}^n = u_i \min_{u_i} \left\{ \sum_{j=1}^{i-1} p_{ij}(u_j)V_{j+1}^n + \sum_{j=i}^{N} p_{ij}(u_j)V_j^n + k_i(u_i) \right\} + (1-u_i)V_i^n
\]

(9) is 'accelerated' after a complete cycle, while (10) is 'accelerated' at each step with a cycle. The term 'accelerated' is taken from analogous procedures for accelerating the convergence of iterative methods for linear equations [2], [4].

For judicious choices of parameters (see numerical results where \( R = \omega R \), for a scalar \( \omega \)), (10) is preferable to (9) which is preferable to (5), (6) and (8). Some mild experimentation yields useful values of \( \omega \). Usually, in practice the same problem is solved many times with varying costs \( k(i,u_i) \). For a fixed \( \omega \), the convergence seems independent of the costs; hence useful values of \( \omega \) on one run should be useful for other runs. Using (6c), (9) can be written as

\[
V_{i+1}^n = \min_{u_i} \left\{ (3q(u_i) + K_0(u_i))V_i^n + (1-\omega)q(u_i) \right\}
\]

Similarly, (10) can be written as

\[
V_{i+1}^n = \min_{u_i} \left\{ q_0(u_i)V_i^n + K_0(u_i) \right\}
\]

(9a)

(10a)

for some \( q_0(u) \) and \( K_0(u) \); \( V_i^n \) is not usually a transition matrix or even part of one.

Upon setting \( u^n = \bar{u}_i \), (5), (6), (9), (10) hold for \( V_{i+1}^n = V_i^n = V_0 \), the optimal cost.

5. Acceleration in the Linear Problem. As useful background, consider a linear problem. Suppose that the spectral radius \( p(A) \) of \( A \) is less than one, and we seek to solve the vector system \( X = AX + b \). The Jacobi iteration is \( X_{n+1} = AX^n + b \), the Gauss-Seidel is

\[
X_{n+1} = \sum_{j=1}^{i-1} a_{ij}X_j^n + \sum_{j=i}^{N} a_{ij}X_j^n + b_i
\]

or equivalently \( X_{n+1} = \tilde{A}X^n + \tilde{b} \), where \( \tilde{A} \) and \( \tilde{b} \) are calculated by the rules (6c). Then \( p(\tilde{A}) < p(A) \). The accelerated Jacobi process is

\[
X_{n+1} = (\omega A + (1-\omega)I)X^n + \omega b = \tilde{A}X^n + \tilde{b}
\]

Let \( \omega = \omega \). Clearly, the eigenvalues of \( \tilde{A} \) are \( \omega \lambda + (1-\omega) \), where \( \lambda \) are the eigenvalues of \( A \). For suitable \( \omega \), \( p(\tilde{A}) < p(A) \) and similarly for the Gauss-Seidel procedure (see [2], [4]).

Unfortunately, the concept of spectral radius does not apply to the non-linear iterations, and the analysis of the non-linear procedures for the control problem uses a max norm, rather than a norm akin to the spectral radius, and the \( \omega \) or \( R \) estimates are very conservative.

4. Analysis of the Nonlinear Iteration.
Theorem 1. Write \((\text{AJ, SGS, AGS, ...})\)

\[(11a) \quad V = [\mathcal{Q}(\mathbf{u}) + (I-D)]V + \mathcal{K}(\mathbf{u})\]
\[(11b) \quad V = [\mathcal{Q}(\mathbf{u}) + (I-D)]V + \mathcal{K}(\mathbf{u})\]
\[(11c) \quad V = \mathcal{Q}(\mathbf{u})V + \mathcal{K}_0(\mathbf{u}).\]

If the spectral radius of a matrix multiplying \(V\) on a right side of \((11)\) are less than unity, then

\[
\lim sup \frac{V^n}{V_0} \leq V_k
\]

for that procedure.

Proof. Only \((11b)\) will be treated. Write \(\mathcal{Q}(u) = \mathcal{Q}(u) + (I-D)\), and

\[
V = \mathcal{Q}(u)V + \mathcal{K}(u) + [\mathcal{Q}(u^n) + \mathcal{K}(u^n)]
\]

Then

\[
V^{n+1} = \mathcal{Q}(u^n)V^n + \mathcal{K}(u^n) \leq \mathcal{Q}(u)V^n + \mathcal{K}(u^n).
\]

Next convergence of \(V^n\) will be discussed via a method that is analogous to that used for a general non-linear problem in [4], Chapter 3.3. To analyze the lower bound in \((12)\), we are unfortunately forced to use a max norm, and seek conditions under which \(\max |D_j| \leq (1-\text{max } |V^n - V|)\) for some \(\delta > 0\).

Under \((C1), (C2)\) or \((C1), (C3)\), for \(n = 1\), \(\mathcal{Q}(u^1) = \mathcal{Q}(u^n)\) and \(\mathcal{Q}(u^2)\) \(\mathcal{Q}(u^n) \to 0\) as \(n \to \infty\), and similarly for the Jacobi procedure.

Proceeding (see \((9a)\)) and noting that the sums of the \(k\) th row elements of \(\mathcal{Q}(u^n) + (I-D)\) are less than \([1 - \omega_k(1-\omega_k)] + \omega_k \sum q_{ij}(u^n) \]

\[
|D_j| \leq (1 - \omega_k(1-\omega_k)) + \omega_k \sum q_{ij}(u^n) \max |V^n - V|.
\]

Thus, there is convergence for any \(\omega_k\) for which

\[
(13) \quad 1 \leq \omega_k < \frac{2}{1 - \omega_k(1-\omega_k) + \sum q_{ij}(u^n)}
\]

for all \(u^n\). But, by \((C1), (C3)\) (see \((7)\)), the supremum of the denominator of \((13)\) is less than 2. Thus there is an interval of convergence which contains more than the point \(\omega_k = 1\). Numerical experience suggests that this interval is very conservative. Thus, trial and error is required to determine the useful \(\omega_k^*\).

Similarly, for the \(\text{AJ} \) procedure, if

\[
(14) \quad \sup |1 - p_{ij}(v_k) + \sum q_{ij}(u_k)| < 2.
\]

The interval of convergence contains more than \(\omega_k = 1\). We have not succeeded in treating AGS (except under the restrictive condition \((13)\)), which, in
fact, computational experience indicates the best of all the accelerated procedures. Theorem 1 suggests that the accelerated procedures may well converge for all $\alpha$ for which the spectral radii of the matrices in (11) are less than unity. Numerical experience also suggests this, since, for such $\alpha$, no subsequence of the $\sqrt{\alpha}$ can tend to $\infty$. We have not been successful in exploiting this fact.


A Continuous Time Problem. Consider the following continuous time problem. Write

$$y = u \psi t$$

(15)

where $u$ is a control, $|u| \leq 1$ and $\psi$ is "white Gaussian noise". A state variationalization of (15), in Itô stochastic differential equation form, is

$$dx_1 = dx_2 dt$$
$$dx_2 = u dt + \sigma dz_2,$$

where $z_2$ is a Wiener process. The object of the control is to drive the state $x_2$ to the internal line in the box in Figure 1, in minimum average time. The process is confined to the box, and, upon reaching the boundary, it is either reflected in, or moves along the boundary, each with a probability which is consistent with the dynamics internal to the box; there is also a cost (1.5 - time) associated with movement on the boundary. The boundary properties are not too important here, since we are interested here only in a comparison between different computational techniques.


of Figure 1. (See [1] for details). The target nodes correspond to state 0. The general form for the cost equations (for the approximating chain are, of course,

$$V_1 = \min \left\{ \sum_{j=1}^{N} p_{ij}(c) v_j + h(i) \right\},$$

for a suitable $p_{ij}(\cdot)$ and $h(\cdot)$, where $i$ ranges over the nodes (other than the target) in Figure 1.

More particularly, let $x = (x_1, x_2)$ be a nodal point, and let $e_i$ denote the unit vector in the $i$th coordinate direction. Thus $x + e_i h$ is also a nodal point, for $x$ internal in the box. For $x$ internal in the box, and $\sigma^2 > h$ (write $v_k$ for $v_x$, where $x$ is the cartesian coordinate of node $i$ in Fig. 1

$$v_x = \min \left\{ \sum_{i=1}^{N} p_{ij}(c) v_j + h(i) \right\},$$

(17)

where $Q = \sigma^2 + 1 h(\cdot)$. The upper entries in the brackets pertain when $x_2 > 0$, the lower otherwise. A similar equation can be written for $x$ on the outer boundary. The chain satisfies (11), (12) and, for a suitable ordering (13).

In particular the ordering in Figure 2 was used. Also, we set $\alpha = u\lambda$.

The numerical results are described by Figures 3 and 4, where the error

$$E = \sum_{i=0}^{N} |v_0^0 - v_0^k|$$

is plotted vs. the iteration number $n$ - on a semi-log scale. Figure 3
plots the results for AGS, GS, and J. All diverged for all \( \omega \) which we tried. In other, simpler problems, Al did not diverge for some values of \( \omega \), but was not as good as the forms of the GS. For the AGS, the performance deteriorated rapidly as \( \omega \) increased beyond 1.4, this being really the best value. The I/B notations give the number of iterations required per significant digit improvement in precision. Figure 4 plots the SAGS which, while better than J and G.S., is not as good as the best AGS. Note, however, that the asymptotic behavior is better for the AGS.

The AGS procedure is clearly superior. This was somewhat expected at the terminal stages, when assuming that the control converged, the linear system behavior prevailed. Even then it is not really obvious, since the \( \omega \) for which the control converged, could be poor \( \omega \) for the asymptotic (or linear) part of the iteration. Note, however, that the runs on Fig. 5 which are better asymptotically are also better initially - right from the beginning.

The best AGS gave a 10-fold reduction in computation over the J

(which is the usual backward iteration procedure of dynamic programming) and a four-fold improvement over the ordinary GS, the procedure of [1].

It is also apparent that the range of \( \omega \) for which the result of Section (4) guarantees convergence of SAGS is grossly underestimated.

Let \( \mathbf{u} \) denote the optimal control. Some interesting properties of the accelerated procedures can be seen by comparing their errors to the errors for the linear procedures (18) (AGS) and (19) (SAGS) for computing the least cost when \( \mathbf{u} \) is given

(18) \[
V_{n+1} = \omega \sum_{j=1}^{N-1} \sum_{i=1}^n p_{ij}(\mathbf{u}_{i,j})V_{n+1}^j + \sum_{j=1}^{N-1} p_{ij}V_{n+1}^j + h(i)u_j \]

\[
V_{n+1} = \omega V_n + (1-\omega)\mathbf{v}_n.
\]

Theorem 1 proved that for all \( \omega \) for which (10) (or (19)) converges, the iterates (20) (or (5)) are bounded from above (for \( \omega = \omega \)). A comparison of Figures 3 and 5 suggests that (10) converges for all \( \omega \) for which (18) converges, and that the \( \omega \) which are preferable for (18) are also preferable for (10) (for \( \omega = \omega \)). The asymptotic rates (slopes) in Figures 3 and 5 are the same, an expected result given that convergence occurs, for ultimately \( V_{n+1} = \mathbf{v}_n \). However, the asymptotically preferable \( \omega \) are also preferable for all iterations - even before the sequence \( V_n \) has converged.

Next, compare the SAGS iteration (19) in Figure 6, with (9) (Figure 4, \( \omega = \omega \)). Up to \( \omega = 1.6 \), Figures 4 and 6 are similar. For \( \omega = 1.8 \), 1.9, the linear procedure (19) oscillates substantially before settling down. Observe, in Figure 7, the oscillation in the spectral radius in this region.

This is typical for linear problems. See [2] for more information.

This is also true for the non-linear problem (9). It can still be said that, for all runs taken, (9) converges for all \( \omega \) (\( \omega = \omega \)) for which (18) converged. For \( \omega = 1.5 \), 1.9, the iteration (9) took much longer for the control to converge, than for smaller values of \( \omega \). But the initial behavior of (19) for these \( \omega \) is not as smooth as for smaller \( \omega \) and it is still possible that the \( \omega \) which are best for (19) for moderate values of \( n \), are best for (9) for moderate values of \( n \), but a precise analysis has eluded us.

A graphic illustration of the advantage of acceleration is given in Figure 7, where the spectral radii of the linear operators in (18), (19) are plotted.
Conclusions

Accelerated procedures for computing optimal controls for a Markov chain model are discussed, and numerical results are presented. For example, one of the methods described gave a 10 fold decrease in computation time over a more usual procedure of dynamic programming.

References

FIG. 1

FIG. 2. THE ORDERING OF THE STATES
FIG. 3. COMPUTATION FOR THE OPTIMAL CONTROL PROBLEM.
AGS and J

FIG. 4. COMPUTATION FOR THE CONTROL PROBLEM. SAGS
FIG. 5. COMPUTATION FOR FIXED $R(\bar{u})$, $\bar{u} =$ OPTIMAL CONTROL, AGS

FIG. 6. COMPUTATION FOR FIXED $R(\bar{u})$, $\bar{u} =$ OPTIMAL CONTROL, SAGS
Accelerated procedures for computing optimal controls for a Markov chain model are discussed, and numerical results are presented. For example, one of the methods described gave a 10-fold decrease in computation time over a more usual procedure of dynamic programming.