

A Computational Note on  
Markov Decision Processes  
without Discounting

by

Paul E. Pfeiffer

and

J.E. Dennis, Jr.

Technical Report 87-19, July 1987



# A Computational Note on Markov Decision Processes without Discounting

Paul E. Pfeiffer and J. E. Dennis, Jr.

## Abstract

The Markov decision process is treated in a variety of forms or cases: finite or infinite horizon, with or without discounting. The finite horizon cases and the case of infinite horizon with discounting have received considerable attention. In the infinite horizon case, with discounting, the problem either receives a linear programming treatment or is treated by the elegant and effective *policy-iteration procedure* introduced by Ronald Howard. In the undiscounted case, however, a special form of this procedure is required, which detracts from the directness and elegance of the method. The difficulty comes in the step generally called the *value-determination procedure*. The equations used in this step are linearly dependent, so that the solution of the system of linear equations requires some adjustment. We propose a new computational procedure which avoids this difficulty and works directly with the average next-period gains and powers of the transition probability matrix. The fundamental computational tools are matrix multiplication and addition.

## Evolution of ergodic Markov chains

Consider an ergodic, homogeneous Markov chain  $X_N = \{X_n: 0 \leq n\}$  with finite state space  $E = \{0, 1, \dots, M\}$  and transition matrix  $P = [p(i, j)]$ , where  $p(i, j) = P(X_{n+1} = j | X_n = i)$ . Let  $p_j(n) = P(X_n = j)$  for all  $j \in E$  and all  $n \in N = \{0, 1, 2, \dots\}$ . Put

$$\pi(n) = [p_0(n), p_1(n), \dots, p_M(n)]$$

Then  $\pi(0)$  is the initial probability distribution,  $\pi(1) = \pi(0)P$  is the first period distribution, and  $\pi(n) = \pi(n-1)P = \pi(0)P^n$  is the  $n$ th-stage distribution.

A generating function analysis of the evolution of  $\pi(n)$  shows that these distributions may be expressed in terms of the eigenvalues  $\lambda_i$  of the transition matrix  $P$ . If the chain is ergodic, then  $\lambda_0 = 1$  is an eigenvalue, and all other eigenvalues have  $|\lambda_i| < 1$ . Usually, the eigenvalues are distinct, in which case

$$P^n = P_0 + \sum_{k \neq 0} P_k \lambda_k^n \quad \text{where} \quad P_0 = \lim_{n \rightarrow \infty} P^n = P^\infty$$

The convergence of the distributions to a stationary distribution is determined by the convergence of  $P^n$  to  $P_0$ . If  $|\lambda|$  is the largest of the  $|\lambda_k|$ ,  $k \neq 0$ , then on an element-by-element basis,

$$|P^n - P_0| \leq |\lambda|^n \sum_{k \neq 0} P_k |\lambda_k / \lambda|^n < a |\lambda|^n$$

so that the rate of convergence is determined by  $|\lambda|$ . Each row of  $P_0$  is the long run distribution  $\pi = [\pi_0, \pi_1, \dots, \pi_M]$ . The rows of  $P^n$  must become more nearly equal as  $n$  increases. This fact provides a ready practical test for a satisfactory level of convergence.

## Markov decision processes.

In a Markov decision process, we assume there is a cost or reward pattern of one of the following four types.

$$G_{n+1} = g(X_n, X_{n+1}) \quad G_{n+1} = g(X_n) \quad G_{n+1} = g(X_{n+1}) \quad \text{or}$$

$$G_{n+1} = g(X_n, D_{n+1}) \quad \text{where} \quad \{D_n: 1 \leq n\} \text{ is independent, and each } D_{n+1} \text{ is independent of the past}$$

The second and third cases may be viewed as special cases of the first. We let

$$q_i = E[G_{n+1} | X_n = i] = v_i^{(1)} \quad \text{and} \quad v_i^{(m)} = E[G_{n+1} + \dots + G_{n+m} | X_n = i]$$

That is,  $q_i$  is the expected next-period gain, given that the present state is  $i$ , and  $v_i^{(m)}$  is the expected gain in the next  $m$  periods, given that the present state is  $i$ . The following recursion relation is established for the expected future gains  $v_i^{(m)}$ .

$$v_i^{(n)} = q_i + \sum_{j \in E} v_j^{(n-1)} p(i, j)$$

The only differences in the four cases is the manner in which the  $q_i$  are obtained. In terms of the matrices

$$\mathbf{v}(n) = [v_0^{(n)} \ v_1^{(n)} \ \cdots \ v_M^{(n)}]^T \quad \text{and} \quad \mathbf{q} = [q_0 \ q_1 \ \cdots \ q_M]^T$$

these recursive equations may be written

$$\mathbf{v}(n) = \mathbf{q} + \mathbf{P}\mathbf{v}(n-1) \quad \text{for all } n \geq 1$$

A generating function analysis similar to that for the  $\mathbf{P}^n$  shows that

$$\mathbf{v}(n) = n\mathbf{P}_0\mathbf{q} + \mathbf{v} + \text{transients}$$

The transients die out as  $|\lambda|^n$ , so that for sufficiently large  $n$  they may be neglected. The matrix product  $\mathbf{P}_0\mathbf{q}$  is a column matrix, each of whose elements is the long run average gain per period  $g = \sum_{i \in \mathbf{E}} \pi_i q_i$ . The members  $v_i$  of  $\mathbf{v}$  are constants whose values are determined by  $\mathbf{q}$  and  $\mathbf{P}$ , in a manner described below. The results of the generating function analysis may, therefore, be written

$$\mathbf{v}(n) = ng\mathbf{1} + \mathbf{v} + \text{transients}$$

### Stationary decision policies.

In a decision process, for each state  $i \in \mathbf{E}$ , there is a class  $A_i$  of possible **actions** which may be taken when the process is in state  $i$ . In general, the choice of action in a state affects both the gain and the transition probabilities. A **decision policy** is a sequence of decision functions  $d_0, d_1, \dots$  such that

$$\text{The action at stage } n \text{ is } d_n(X_0, X_1, \dots, X_n).$$

The action selected is in  $A_i$  whenever  $X_n = i$ . We limit consideration to the special class of policies known as **stationary decision policies**. These are described by

$$d_n(X_0, X_1, \dots, X_n) = d(X_n) \quad \text{with } d \text{ invariant with } n$$

The possible action depends only on the current state, regardless of the period, with  $d(X_n) \in A_i$  whenever  $X_n = i$ . We seek a stationary decision policy which maximizes the long run average gain per period  $g$ . An optimal policy may not be unique, although the associated gain  $g$  is unique. We suppose that each possible policy yields an ergodic chain.

### Policy iteration method.

In his pioneering work, *Dynamic Programming and Markov Processes*, published in 1960, Ronald Howard developed a simple two-stage procedure for determining an optimum stationary policy. To set up this procedure, we begin by combining the recursive equations and the asymptotic results of the generating function analysis for  $\mathbf{v}(n)$  to show that  $g$  and  $\mathbf{v}$  satisfy

$$g\mathbf{1} + \mathbf{v} = \mathbf{q} + \mathbf{P}\mathbf{v} \quad \text{or} \quad g + v_i = q_i + \sum_{j \in \mathbf{E}} p(i, j)v_j \quad 0 \leq j \leq M$$

Suppose a stationary policy  $d$  is selected. That is, action  $d(i)$  is taken whenever the process is in state  $i$ . To simplify writing, we drop the indication of the action and simply write  $p(i, j)$  for  $p_{ij}(d(i))$ , etc. Associated with this policy, there is a gain  $g$ . We should like to determine whether or not this is the maximum possible gain, and if it is not, to find a policy which does give the maximum gain. Howard showed the following two-phase procedure to be effective.

1. **Value-determination procedure.** The policy  $d$  with  $d(i) = a_i$  will determine a choice of  $q_i(a_i)$  and  $p_{ij}(a_i)$  for each state  $i$ . Use these to determine the long run average gain  $g$  and the  $v_i$ .

2. **Policy-improvement procedure.** Suppose policy  $d$  has been used through period  $n$ . We seek to improve policy  $d$  by selecting a new policy  $d^*$ , with  $d^*(i) = a_i^*$ , to satisfy

$$q_i^* + \sum_j p_{ij}^* v_j = \max\{q_i(a_i) + \sum_j p_{ij}(a_i)v_j; a_i \in A_i\}$$

In the procedure for selecting  $d^*$ , we use the "old"  $v_j$ . Howard has shown that if  $g^*$  is the long run average gain per period for the new policy, then  $g^* \geq g$ , with equality iff  $g$  is optimal. Since there is a finite number of policies, the procedure must converge in a finite number of steps.

**A new method for determining the value matrix.**

In his original treatment, Howard dealt with the equations in  $g$  and the  $v_i$  as a system of  $M + 1$  linear algebraic equations in  $M + 2$  unknowns. He and most subsequent expositors handle this by considering the values of the  $v_i$  relative to one of the fixed values. That is he considered the variable  $g$  and the  $M$  variables  $v_i - v_M$ ,  $0 \leq i \leq M - 1$ . It may appear that the problem could be avoided by direct computation of  $g = \sum_j \pi_j q_j$ , so that there are only as many unknowns as variables. However, closer examination shows the resulting equations are not linearly independent.

The procedure we establish for determining the  $v_i$  requires the values of the next-period expected gains  $q_i$  and powers of the transition probability matrix  $\mathbf{P}$  for the decision rule under consideration. The principal computational operations are addition and multiplication of matrices, which modern computers perform quite rapidly and efficiently.

The generating function analysis shows that  $\mathbf{v} = \mathbf{B}_0 \mathbf{q}$ , where

$$\mathbf{B}_0 = \lim_{s \rightarrow 1} \mathbf{B}(s) \quad \text{with} \quad \mathbf{B}(s) = [\mathbf{I} - s\mathbf{P}]^{-1} - \frac{\mathbf{P}_0}{1 - s}$$

Use of the geometric series for  $\frac{1}{1 - s}$  and the well known expansion

$$[\mathbf{I} - s\mathbf{P}]^{-1} = \mathbf{I} + s\mathbf{P} + (s\mathbf{P})^2 + \dots = \sum_{k=0}^{\infty} (s\mathbf{P})^k$$

yields

$$\mathbf{B}(s) = \sum_{k=0}^{\infty} (s\mathbf{P})^k - \sum_{k=0}^{\infty} s^k \mathbf{P}_0 = \sum_{k=0}^{\infty} (\mathbf{P}^k - \mathbf{P}_0) s^k \quad \text{for } |s| < 1$$

Since  $|\mathbf{P}^n - \mathbf{P}_0| \rightarrow 0$  like  $a |\lambda|^n$ , the latter series converges for  $s = 1$ , so that

$$\mathbf{B}_0 = \mathbf{B}(1) = \sum_{k=0}^{\infty} (\mathbf{P}^k - \mathbf{P}_0)$$

We may approximate the infinite series by the partial sum

$$\mathbf{B}_0 \approx \sum_{k=0}^m (\mathbf{P}^k - \mathbf{P}_0) = \sum_{k=0}^m \mathbf{P}^k - (m + 1)\mathbf{P}_0 \approx \sum_{k=0}^m \mathbf{P}^k - (m + 1)\mathbf{P}^m$$

The error of approximation by the partial sum satisfies

$$\left| \sum_{k=m+1}^{\infty} (\mathbf{P}^k - \mathbf{P}_0) \right| \leq a |\lambda|^{m+1} \sum_{k=0}^{\infty} |\lambda|^k = \frac{a |\lambda|^{m+1}}{1 - |\lambda|}$$

In addition, if we approximate  $\mathbf{P}_0$  by  $\mathbf{P}^n$ ,  $n > m$ , we have an additional error less than  $a(m + 1)|\lambda|^n$ . It is clear that if  $|\lambda|$  is substantially less than one, these errors become negligible for reasonable values of  $m, n$ .

**Example**

A Markov decision process has three states: State space  $\mathbf{E} = \{0, 1, 2\}$ .

Actions: State 0:  $A_0 = \{0, 1, 2\}$  State 1:  $A_1 = \{0, 1\}$  State 2:  $A_2 = \{0, 1\}$

Transition probabilities and rewards are:

$$\begin{array}{ll} p_{0j}(0): [1/3 & 1/3 & 1/3] & g_{0j}(0): [1 & 3 & 4] \\ p_{0j}(1): [1/4 & 3/8 & 3/8] & g_{0j}(1): [2 & 2 & 3] \\ p_{0j}(2): [1/3 & 1/3 & 1/3] & g_{0j}(2): [2 & 2 & 3] \\ \\ p_{1j}(0): [1/8 & 3/8 & 1/2] & g_{1j}(0): [2 & 1 & 2] \\ p_{1j}(1): [1/2 & 1/4 & 1/4] & g_{1j}(1): [1 & 4 & 4] \\ \\ p_{2j}(0): [3/8 & 1/4 & 3/8] & g_{2j}(0): [2 & 3 & 3] \\ p_{2j}(1): [1/8 & 1/4 & 5/8] & g_{2j}(1): [3 & 2 & 2] \end{array}$$

Use the policy iteration method to determine the policy which gives the maximum gain  $g$ .

**SOLUTION**

*Classical procedure*

We set  $v_M = v_2 = 0$ . First, we must determine the expected next-period expected gains  $q_i$ . These depend upon the actions taken in each state. If action  $a$  is taken when the process is in state  $i$ ,

$$q_i(a) = \sum_{j=0}^M p_{ij}(a)g_{ij}(a)$$

For  $i = 0, j = 0$ , we have  $q_0(0) = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 4 = 8/3$ . The other values are determined in a similar manner. We tabulate the results below

$$\begin{array}{lll} q_0(0) = 8/3 & q_0(1) = 19/8 & q_0(2) = 7/3 \\ q_1(0) = 13/8 & q_1(1) = 5/2 & \\ q_2(0) = 21/8 & q_2(1) = 17/8 & \end{array}$$

We start with the policy  $d$  which gives the greatest immediate expected gains. Policy  $d = [0 \ 1 \ 0]$ .

1. *Value determination.* Set  $v_2 = 0$ . Solve  $g + v_i = q_i + p_{i0}(a)v_0 + p_{i1}(a)v_1$ . The equations are

$$\begin{aligned} g + v_0 &= \frac{8}{3} + \frac{1}{3}v_0 + \frac{1}{3}v_1 \\ g + v_1 &= \frac{5}{2} + \frac{1}{2}v_0 + \frac{1}{4}v_1 \\ g &= \frac{21}{8} + \frac{3}{8}v_0 + \frac{1}{4}v_1 \end{aligned}$$

These are solved algebraically to give  $v_0 = \frac{1}{33}$ ,  $v_1 = -\frac{4}{33}$ , and  $g = \frac{86}{33}$ .

2. *Policy improvement.* Use  $v_0 = \frac{1}{33}$   $v_1 = -\frac{4}{33}$ . For each state  $i$  and each action  $a$ , determine

$$c_i(a) = q_i(a) + p_{i0}(a)\frac{1}{33} - p_{i1}(a)\frac{4}{33}$$

The results are tabulated as follows

State	Action	$c_i(a)$
$i = 0$	$a = 0$	2.64*
	1	2.34
	2	2.30
$i = 1$	$a = 0$	1.58
	1	2.48*
$i = 2$	$a = 0$	2.61*
	1	2.10

The starred values represent the greatest  $c_i(a)$  for each state. Thus,  $d^* \sim [0 \ 1 \ 0]$ . Since this is the same as the original policy  $d$ , it must be optimal. For this system, under an optimal policy, the long-run gain per stage is  $g = 86/33 \approx 2.61$ .

#### New procedure

We let  $\mathbf{P}_a$  be the entire set of transition probabilities and  $\mathbf{g}_a$  be the entire set of gain values. We proceed in the following steps.

#### 1. Data input

$$\mathbf{P}_a = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/4 & 3/8 & 3/8 \\ 1/3 & 1/3 & 1/3 \\ 1/8 & 3/8 & 1/2 \\ 1/2 & 1/4 & 1/4 \\ 3/8 & 1/4 & 3/8 \\ 1/8 & 1/4 & 5/8 \end{bmatrix} \quad \text{and} \quad \mathbf{g}_a = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 2 & 3 \\ 2 & 2 & 3 \\ 2 & 1 & 2 \\ 1 & 4 & 4 \\ 2 & 3 & 3 \\ 3 & 2 & 2 \end{bmatrix}$$

The first three rows correspond to the three possible actions in state 0, the next two rows correspond to the two possible actions in state 1, and the final two rows correspond to the two possible actions in state 2. We can calculate the corresponding expected next-period gains for each state and action in one operation

$$\mathbf{q}_a = \text{diagonal of } \mathbf{P}_a \mathbf{g}_a^T = [2.6667 \ 2.3750 \ 2.333 \ 1.6250 \ 2.5000 \ 2.6250 \ 2.1250]^T.$$

2. **Policy input.** We make a preliminary choice of policy. This is usually done by taking that action in each state which yields the greatest expected next-period gain  $q_i$ . In this example, this is the policy  $[0 \ 1 \ 0]$ , which corresponds to the choice of rows 1, 5, and 6 in the matrices  $\mathbf{P}_a$ ,  $\mathbf{g}_a$ ,  $\mathbf{q}_a$ . If we pick out the rows corresponding to this policy, we have

$$\mathbf{P} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/4 & 1/4 \\ 3/8 & 1/4 & 3/8 \end{bmatrix} \quad \text{and} \quad \mathbf{q} = \begin{bmatrix} 2.6667 \\ 2.5000 \\ 2.6250 \end{bmatrix}$$

3. **Policy improvement.** We calculate the following:

- $\mathbf{P}_0 = \lim_k \mathbf{P}^k \approx \mathbf{P}^n$  (we use  $n = 16$ )
- $\boldsymbol{\pi} = \text{first row of } \mathbf{P}_0 \quad g = \boldsymbol{\pi} \mathbf{q}$
- $\mathbf{v} = \mathbf{B}_0 \mathbf{q} \approx [\mathbf{I} + \mathbf{P} + \mathbf{P}^2 + \cdots + \mathbf{P}^s - (s+1)\mathbf{P}^s] \mathbf{q}$  (we use  $s = 4$ )
- $\mathbf{T} = \mathbf{q}_a + \mathbf{P}_a \mathbf{v}$
- Pick out the best row in  $\mathbf{T}$  for each state, thus determining a new policy.

4. **Continuation or termination.**

- If the "new policy" is the same as the previous policy, record this policy and the gain  $g$ , and stop. These are an optimum policy and maximum gain.
- If the "new policy" is different, repeat steps 2, 3, 4, until an optimum policy is found.

For the selected policy, and the corresponding  $\mathbf{P}$ ,  $\mathbf{q}$ , above, use of a computer matrix program yields

$$\mathbf{T} = \begin{bmatrix} 2.6587 \\ 2.3595 \\ 2.3254 \\ 1.6057 \\ 2.5072 \\ 2.6284 \\ 2.1208 \end{bmatrix}$$

The first of the first three rows is maximum; the second of the next two rows is maximum; and the first of the final two rows is maximum. Thus, we select rows 1, 5, 6, corresponding to policy  $[0 \ 1 \ 0]$ . This "new policy" is the same as the original; hence, it is optimal. The results may be summarized:

- The policy  $[0 \ 1 \ 0]$  is optimal
- The average long run gain per period  $g \approx 2.61$ .
- The value matrix is

$$\mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0.0527 \\ -0.0989 \\ 0.0223 \end{bmatrix}$$

The gain value  $g = 2.6061 \approx 86.0013/33$  compares quite closely with the value  $86/33$  obtained by the classical method. To make a comparison of the  $v_i$  with the previous results, we must subtract  $v_2$  from each of the  $v_i$  to get

$$v_0 - v_2 = 0.0304 \approx \frac{1.0032}{33} \quad \text{and} \quad v_1 - v_2 = -0.1212 \approx \frac{-3.9996}{33}$$

These compare quite closely with the corresponding values  $1/33$  and  $-4/33$  obtained in the classical calculation above.

— □  
*Remarks*

1. We made a somewhat arbitrary choice of the powers of  $\mathbf{P}$  used in the approximation of  $\mathbf{P}_0$  and  $\mathbf{B}_0$ . The convergence of  $\mathbf{P}^n$  is governed by the magnitude of the largest eigenvalue other than one. We can always check on the reliability of the calculations by checking the eigenvalues for  $\mathbf{P}$  corresponding to the presumed optimal policy. For the choice above, we find the eigenvalues to be 1, -0.1250, 0.0833. Since  $0.125^4 \approx 0.0002$ , the choices of exponents should be quite satisfactory. In fact, we could probably use  $\mathbf{P}^8$  as a satisfactory approximation to  $\mathbf{P}_0$ . The margin allows for the possibility that for some policies the eigenvalues may not be so small. We need sufficient accuracy in the determination of the  $v_i$  to be able to distinguish between various possibilities in the test matrix  $\mathbf{T}$ .
2. The classical procedure could, of course, be given matrix formulation for direct machine computation. To obtain the  $\mathbf{q}_a$  matrix, we use  $\mathbf{P}_a$  and  $\mathbf{g}_a$  as in our formulation. However, we need to express the reduced equations (with  $v_M$  set to zero as a reference) in the form  $\mathbf{A}\mathbf{w} = \mathbf{q}$ , where  $\mathbf{w}$  is the  $\mathbf{v}$  matrix with the reference value  $v_M = 0$  replaced by  $g$  and  $\mathbf{A}$  is the matrix  $\mathbf{I} - \mathbf{P}$  with column  $M$ , replaced by a column of 1's. The computation requires inversion of  $\mathbf{A}$ . The test matrix  $\mathbf{C}$  for determining a new policy is obtained by  $\mathbf{C} = \mathbf{q}_a + \mathbf{P}_a\mathbf{v}'$ , where  $\mathbf{v}'$  is  $\mathbf{w}$  with  $g$  replaced by 0.

APPENDIX. Matlab implementation of the calculations.

The actual calculations make use of the versatile matrix program MATLAB developed by Cleve Moler of the University of New Mexico. The Matlab implementation we use is the following.

**Data** (in file: data)

PA = <1/3 1/3 1/3; 1/4 3/8 3/8; 1/3 1/3 1/3;  
1/8 3/8 1/2; 1/2 1/4 1/4;  
3/8 1/4 3/8; 1/8 1/4 5/8> (transition probabilities for all states and actions)

GA = <1 3 4; 2 2 3; 2 2 3; 2 1 2; 1 4 4; 2 3 3; 3 2 2> (gains for all states and actions)

QA = diag(PA\*GA') (average next-period gain for all states and actions)

**Policy improvement** (in file: newpol)

for i=1:m, P(i,:) = PA(D(i,:),:);

for i=1:m, Q(i,:) = QA(D(i,:),:);

P0 = P\*\*n (approximation for P0)

PI = P0(1,:) (long run distribution)

G = PI\*Q (long run expected gain per period)

C = eye + P

for j=2:s, C = C + P\*\*j

V = (C - (s+1)\*P0)\*Q

T = QA + PA\*V (test values for determining new policy)

**Procedure with matlab**

matlab

exec('data')

enter m (number of states)

enter n (power of P to approximate P0)

enter s (highest power of P in approximation of V)

enter D (column matrix with choice of actions for policy under consideration)

exec('newpol')

select new policy

---

enter new D

exec('newpol')

select new policy

---

repeat last three steps until the latest "new policy" matches the previous policy