This article was downloaded by: [216.73.216.182] On: 04 June 2026, At: 04:36
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Technical Note—Undiscounted Markov Renewal Programming Via Modified Successive Approximations

Thomas E. Morton,

To cite this article:
Thomas E. Morton, (1971) Technical Note—Undiscounted Markov Renewal Programming Via Modified Successive Approximations. Operations Research 19(4):1081-1089. https://doi.org/10.1287/opre.19.4.1081

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

© 1971 INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

UNDISCOUNTED MARKOV RENEWAL PROGRAMMING VIA MODIFIED SUCCESSIVE APPROXIMATIONS

Thomas E. Morton

Carnegie-Mellon University, Pittsburgh, Pennsylvania

(Received June 26, 1970)

This note describes an efficient class of procedures for finding a solution to the functional equations

$$v_i^* = \max_k [q_i^k - g^* T_i^k + \sum_{j=1}^{j=N} P_{ij}^k v_j^*], \quad 1 \leqq i \leqq N,$$

of undiscounted Markov renewal programming. First, for the special case of a single possible policy, the problem is proved equivalent to solving two related ordinary Markov chain problems, which leads to an algorithm for the general problem whose exact form depends on the specification of a decision rule for alternation of two types of iterations. At one extreme, the technique is exactly 'policy iteration,' with iterative techniques replacing solution of N equations for each improved policy; at the other extreme, the algorithm becomes essentially 'value iteration,' generalizing the method of successive approximations proposed by D. J. WHITE for Markovian decision processes. The latter version of the technique is related to another generalization being currently proposed by PAUL J. SCHWEITZER; the methods being proposed here, however, do not deteriorate when the minimum transition time between states becomes very small.

THIS NOTE provides an efficient class of iterative solution procedures for the functional equations of undiscounted Markov renewal programming:

$$v_i^* = \max_{k \in \alpha_i} [q_i^k - g^* T_i^k + \sum_{j=1}^{j=N} P_{ij}^k v_j^*], \; v_N^* = 0, \; 1 \leqq i \leqq N, \tag{1}$$

where $T_i^k > 0$, $P_{ij}^k \geqq 0$, and $\sum_{j=1}^{j=N} P_{ij}^k = 1$. They represent a generalization of ordinary Markovian decision processes to situations where the time between state changes is itself a random variable. Under rather general assumptions, it has been shown that the expected time spent in state i when decision k is made T_i^k is sufficient to specify the problem (essentially because of the central limit theorem). The variable q_i^k is the expected reward (cost) accruing in a single period if, in state i, decision k is made. Similarly, P_{ij}^k are the corresponding transition probabilities to new states j. The variables to be solved for are g^* (the gain of the optimal policy), v_i^* (the relative total reward of being in state i relative to some base state N), and, implicitly, the (possibly not unique) optimal policy k_i^*. The problem has been described in detail elsewhere (*see* references 4, 5, and 11). (Because of the close connection between methods used here and those currently being proposed by PAUL J. SCHWEITZER,[11] identical notations will be employed.) Here α_i denotes the given finite number of policy choices available when in state i; the case where each set consists of exactly one element is of special interest, and is discussed in Section 1.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

We will assume here that, for each possible policy A, the associated stochastic matrix P^A has a subdominant eigenvalue (which may be complex) λ^A satisfying $|\lambda^A| \leqq \lambda < 1$ (*see* Note 1). As long as there is only one communicating class, (1) is known always to possess a solution with g^* unique and v_i^* unique to an additive constant.[11]

Conventional policy-iteration and linear-programming techniques for these problems require the solution of N equations and N unknowns in each iteration, and hence are impractical both in terms of computer storage and computational efficiency for large N, since a core storage of N^2 and a number of calculations of order N^3 would be required for each iteration (*see* Note 2).

By contrast, Schweitzer's proposed method and the methods proposed here require only $2N$ to $4N$ in storage (*see* Note 3). Total numbers of calculations required per iteration are about linear in N, if the number of decisions possible in each state and the number of possible transitions from each state are held constant. Furthermore, the number of iterations required does not depend directly on N, but on the underlying problem structure, the subdominant eigenvalue in particular (*see* Note 4). In contrast to Schweitzer's method, the currently proposed scheme requires about double the computer storage, and slightly more calculations per iteration. However, since the time slice per iteration is, in effect, the mean transition time between states rather than the minimum of all possible transition times, the number of iterations required should usually be less by a large factor.

Section 1 proves that the Markov renewal programming problem for the special case of a single fixed policy equivalent to solving two related ordinary Markov-chain problems. To solve this restricted problem, we then suggest a formal iterative scheme whose convergence rate is known to be geometric and equal to the absolute value of the subdominant eigenvalue of the transition matrix. Section 2 generalizes the scheme to include an occasional policy-'improvement' step. The less often this step is performed, the more it can be guaranteed that each new policy produced indeed possesses a higher gain. In particular, if the policy improvement step is only performed each time after complete convergence of the relative rewards and the gain, then an improved policy is guaranteed each time; one is really performing conventional policy iteration with an iterative procedure substituted for solution of the N equations. At the other extreme, if the policy 'improvement' step is done after *each* minor iteration, then the technique becomes a generalization of WHITE's modified successive-approximation approach (value iteration) for ordinary Markov decision processes.[15]

Intermediate techniques, which can be useful utilizing the same machinery, are mentioned. The usefulness of the technique relative to Schweitzer's method is also discussed.

1. THE CASE OF A UNIQUE POLICY

WHEN EACH OF the α_i contains only one choice, so that there is only one policy, equation (1) may be written more simply in vector notation as:

$$gT + v = q + Pv, \qquad v_N = 0. \tag{2}$$

It turns out that this equation may be solved rather easily for g and v by solving

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

two related ordinary Markov chain problems, as indicated by Theorem 1, where e is identically equal to a column of ones.

THEOREM 1. *Let $\bar{q}$ and w satisfy*

$$\bar{q}e+w=q+Pw, \qquad w_N=0, \tag{3}$$

while $\bar{T}$ and t satisfy

$$\bar{T}e+t=T+Pt, \qquad t_N=0. \tag{4}$$

and define

$$g=\bar{q}/\bar{T}, \qquad v=w-gt. \tag{5}$$

Then g, v satisfy equation (2).

Proof. We see that

$$\begin{aligned} q+Pv&=(q+Pw)-(gPt)=(\bar{q}e+w)-g(\bar{T}e+t-T)\\ &=\bar{q}e+w-(\bar{q}/\bar{T})\bar{T}e-gt+gT=gT+v. \end{aligned}$$

Also, $v_N=w_N-gt_N=0$.

The intuitive explanation of Theorem 1 is as follows. The desired relative-cost vector v gives relative costs of being in each state over a given long period of *time*, while w gives relative costs of being in each state over a given long number of *transitions*. If every transition took the same length of time, or if the system were in stationary equilibrium, there would be no distinction. However the vector t in equation (5) is designed precisely to relate the *relative* differences in time required for a given long number of transitions. Thus, if t is weighted by g, the average cost per unit time of the system far in the future in equilibrium, one gets precisely the correction needed to transform w to the desired v. If $\bar{q}$ and $\bar{T}$ satisfy (3) and (4), then they clearly represent the average cost per transition and the average time per transition, respectively, and it is a well known result that g is given by their ratio.

Now it is known[6] that (3) and (4) may both be solved very efficiently, both in terms of computer storage and in computational effort, by iterative means.

COROLLARY 1. *Suppose the subdominant eigenvalue of P is given by β, $|\beta|<1.0$, that P, q, and T are represented implicity in function form, and that J is the maximum number of nonzero elements of a row of P. Then the computer storage required to solve* (2) *is of order $4N$, while the number of calculations required is of order $2JN/(1-\beta)$.*

Proof. Reference 6 shows that the number of iterations required to solve each of (3) and (4) by modified successive approximations is of order $1/(1-\beta)$. In each iteration, $Pw(n)$ and $Pt(n)$ must be calculated, requiring J calculations for each row. The only principal storage required is for the vectors $w(n)$, $w(n+1)$, $t(n)$, and $t(n+1)$.

The iterative scheme corresponding to Theorem 1 is shown in Fig. 1. The subscripts on q, T, and P are redundant (since there is a unique policy); however, they are shown, since this block will later be incorporated in the full routine discussed in the next section. Step (i) represents an iteration in the process of successive approximation to (3); similarly, for step (ii) and equation (4). Step (iii)

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

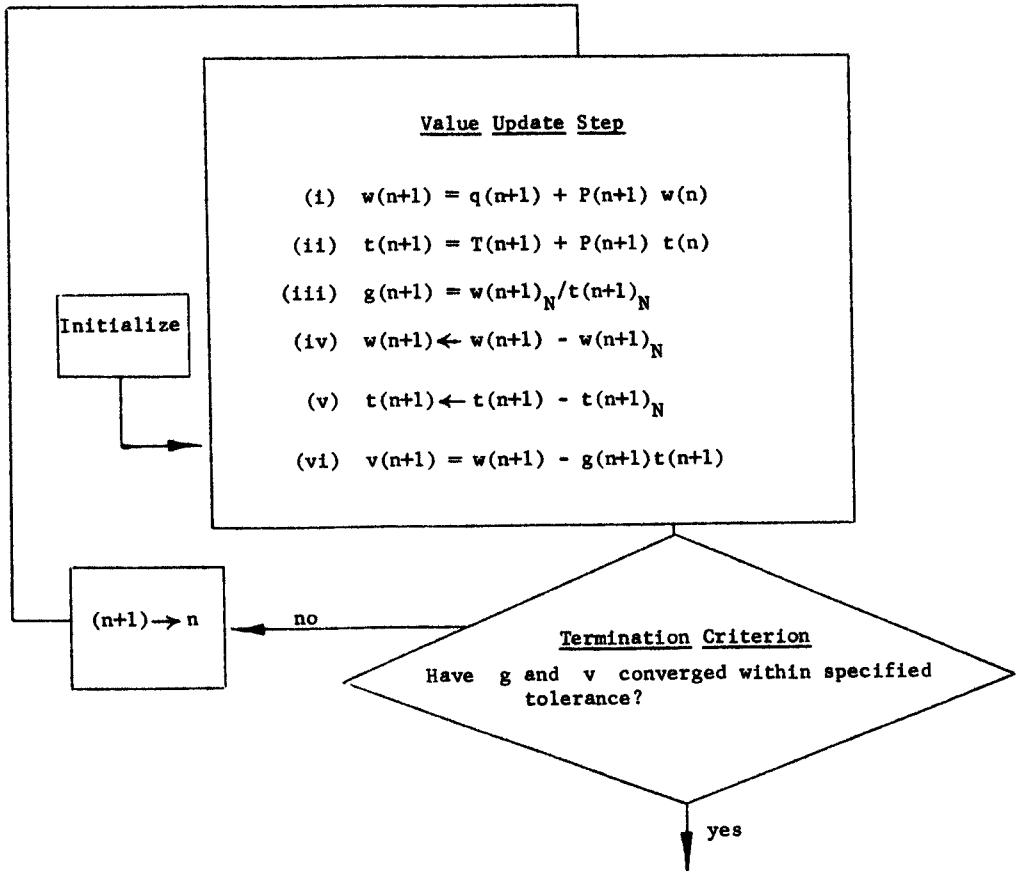

Fig. 1. The unique-policy scheme.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

gives the current approximation to the gain as seen from (5). Steps (iv) and (v) are not essential at each step, but make the comparison of $v(n)$ and $v(n+1)$ somewhat easier for the termination check. Step (vi) calculates the current approximation to the relative value function as seen from (5). Since g and the v_i are known to converge asymptotically geometrically, a safe termination rule is something like 'stop if the maximum absolute percentage change among g and the v_i is less than 0.00001.' If there is any worry that the dominant eigenvalue is very close to 1.0, then, to be perfectly safe, the program could estimate the apparent geometric factor, extrapolate the problem geometrically to the limit, and stop only if current values are very close to this limit. Note that steps (i) and (ii) are the bulk of computation, requiring $2JN$ multiplications for each iteration.

2. THE GENERAL CASE

IT SHOULD BE clear that the procedure suggested by Theorem 1 and presented schematically in Fig. 1 can be implemented directly as the 'value-determination' step of Howard's policy-iteration scheme.[4] Furthermore, it is probably not typically necessary to iterate g and v 'perfectly' for a given policy in order that the policy-'improvement' step yield a superior policy. This suggests some flexibility in our thinking as to the relative frequency of the value-update step and the policy-'update' step. (From here on we shall use 'update' rather than 'improvement' to allow for some ambiguity on this point.)

Figure 2 presents the general scheme being proposed. It actually represents a 'family' of such schemes, since the decision-rule circle has not been specified. If the decision between policy update and value update as the next step is 'policy update only when g and v have currently converged to a fine tolerance,' then indeed *Fig. 2 represents policy iteration with iterative solution of the value-determination step.* It is a major improvement over conventional policy iteration as indicated by Corollary 2:

COROLLARY 2. *If the policy-iteration scheme given requires M policy-improvement steps, and there are an average of K choices in each state, then this modified scheme requires about $4N$ in storage requirements, and the computational effort is on the order of $M[JKN+c(2JN/(1-\beta)]$, where c is a constant.*

Note that, for modified policy iteration, the policy-improvement step will typically be the dominant computational effort involved for very large problems, while, for conventional policy improvement, it is the value-determination step. Thus a rough ratio of the relative computations required will be on the order of $MJKN/MN^3$ or JK/N^2. In a large-scale problem with perhaps $J=100$, $K=10$, and $N=10{,}000$ (which become feasible), the saving would be something like a factor of 100,000.

At the other extreme in Fig. 2, suppose that the decision rule were to be given by 'always go to the policy-update routine.' The relative rewards produced by the *value-update* routine then actually correspond to rewards obtainable from an $(n+1)$-period problem by following the sequence of policies produced by the policy-update routine. Thus, Fig. 2 with this scheme is executing a form of modified successive approximations, or value iteration. If, indeed, we consider the special

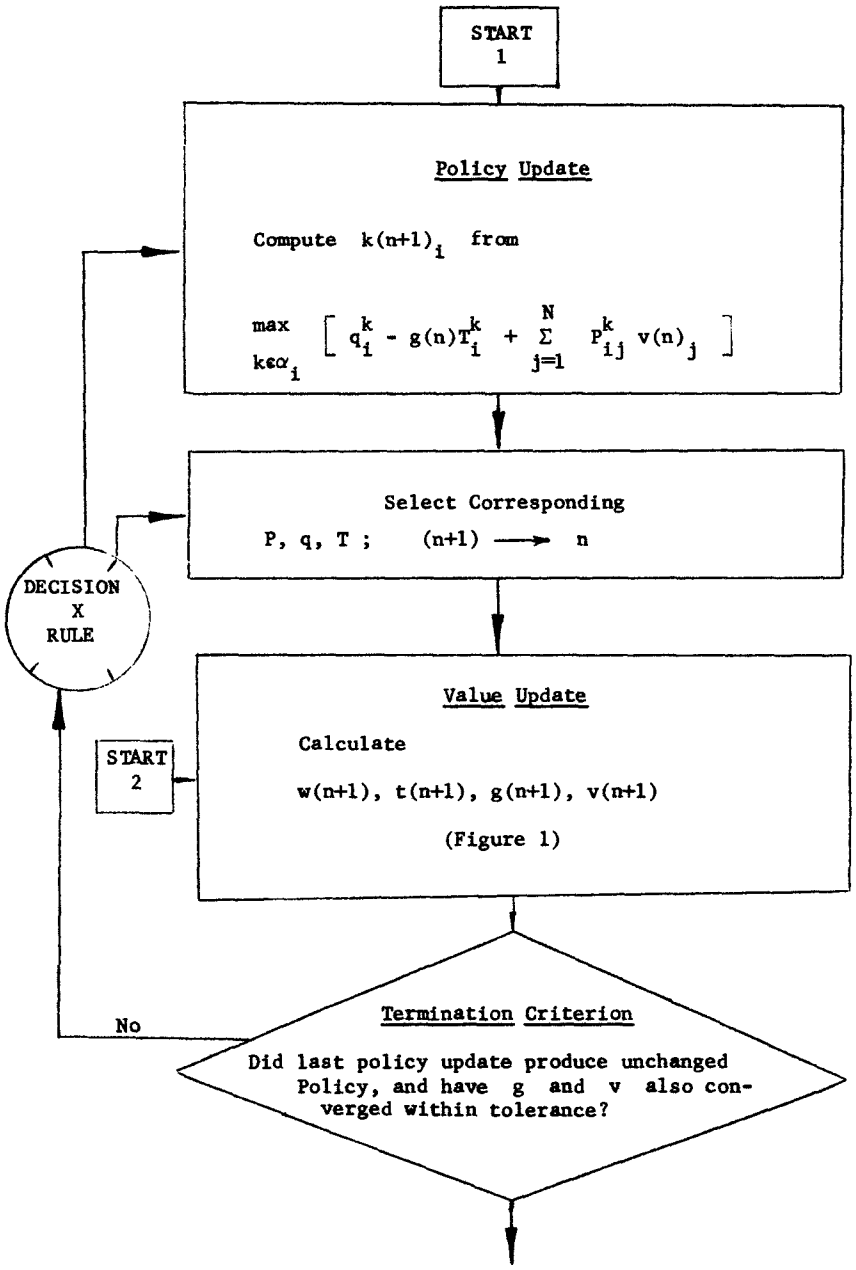

Fig. 2. The general scheme.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

case where $T_i^k \equiv 1$, then the Markov-renewal-programming problem reduces to an ordinary Markov decision problem and it may easily be checked that the scheme reduces to a (slightly clumsy) version of White's modified successive-approximation method.[15] (It is redundant to calculate the vector $t(n+1)$, which is always zero.)

I have not been able to prove that the pure value-iteration generalization of White's scheme converges under very general conditions. However, the following facts powerfully support its further investigation as a 'good' scheme:

(a) For the special case of a fixed policy, Corollary 1 proves the method converges very rapidly. By contrast, Schweitzer's 'time-slice' method was developed only after *a naive extension of White's method proved to be widely divergent for a fixed policy*. Thus, the current scheme at least corrects for the difficulty presented by Schweitzer.

(b) Schweitzer's time-slice method, unfortunately, requires a time slice per iteration less than the minimum mean transition time for any state and any policy, compared essentially with the 'average' transition time under the method proposed here. This is a very serious limitation for problems with highly variable transition times.

Finally, of course, it is *not* necessary to use the decision rule 'use policy update every iteration.' At the other extreme, the technique becomes *policy iteration* with guaranteed monotone convergence to an optimal policy within a finite number of policy updates. The estimates of computational effort required for this version represent a dramatic improvement in themselves over Schweitzer's time-slice method, with the additional comfort of monotone policy convergence.

Figure 2 also suggests a number of possible intermediate schemes. It is clear that each repetition of the value-update routine has a 'stabilizing' effect in terms of the policy-update routine. Because of the geometric convergence of g and v with repetition of the value-update routine, it is also clear that each successive repetition of the value-update routine contributes less and less. This might suggest, for example, a coarse criterion for convergence in the decision rule, and a fine convergence criterion for termination, as a slightly modified version of policy iteration that might cut computation considerably.

As a single example of a more complicated mixed rule that could be constructed with *guaranteed convergence*, consider the following:

(a) Start with any sloppy tolerance, such as 10 per cent maximum change in the relative cost function as the criterion for moving into the 'improvement' routine. Of course, at this point a truly improved policy is therefore not guaranteed; however, computation per iteration is relatively cheap.

(b) If either (i) successive update policies ever repeat, or (ii) 10 policy updates elapse, then constrict the required tolerance by a factor of 100.

(c) Repeat (a) and (b) until the tolerance level drops to, say 10^{-9}.

(d) By this point, one probably has the optimal policy. If not, one is at least performing pure policy iteration, which is guaranteed to converge in a finite number of update steps; thus:

(e) Drop the 10-policy-update requirement and terminate on obtaining successively identical policies.

An experimental program is currently being undertaken to investigate the

relative computational efficiency of a variety of such decision rules for a variety of large-scale problems considered by other authors (*see* Note 5).

NOTES

1. This formulation may be extended easily to the case where the absolute value of the eigenvalue is equal to one by the standard transformation,[12] which transforms every transition probability P_{ij}^k to $(1-s)P_{ij}^k+s\delta_{ij}$ (the Kronecker delta), where $0<s<1$. This transforms all eigenvalues to $\lambda_i^A(1-s)+s$. This change slows convergence in all but 'pathological' cases, and hence should be used only as a last resort.

2. This estimate assumes one is using something like the revised simplex method, so that only the current inverse need be stored in fast access. (Column vectors might be stored on medium access or generated as needed.) For some problems with special structures, the linear programming version can be made much more efficient, however (*see* DE GHELLINCK AND EPPEN[1]). Decomposition could be used to reduce storage requirements, but experience seems to be mixed in terms of its computational efficiency for very large problems.

3. This estimate assumes that q_i^k, T_i^k, and P_{ij}^k are easily generated as needed, as is often the case. Given a current state i, for example, the new state j is often given

TABLE I

THE AVERAGE NUMBER OF TOTAL ITERATIONS TO OPTIMAL POLICY AND THE RELATIVE COST FUNCTION FROM AN IDENTICALLY ZERO INITIAL-COST FUNCTION

	$N=70$	$N>2000$
Normal $(\mu/\sigma=0.5)$	17	24
Exponential $(\mu/\sigma=1.0)$	29	29
Long tail $(\mu/\sigma=1.7)$	51	50

NOTE: The tolerance is 10^{-4}.

as $J(i, k, d)$, where d is a random variable independent of i, j, or k, and J is a simple function requiring no multiplications.

4. There seems to be a prevalent misconception that the number of iterations required must go up dramatically with the state-space size. This is not true, in the author's experience, for problems with the same underlying structure. In the computational study of reference 7, I used the straight White modified-value-iteration procedure (a special case of the model here with all transition times identically equal to 1.0) for a one-period-lag no-backlogging inventory problem with about 70 states, and a similar two-period-lag problem with over 2000 states (both on-hand inventory and the outstanding order are then needed to specify the state). The cases utilized four cost sets and three distribution sets, and obtained the results shown in Table I. In addition, the one-period normal problem was re-run after subdividing the states, the demand distribution, and possible orders each into about three times as many values. The number of required iterations was not affected. The running time went up by about a factor of 10 as expected. (Problem structure allows searching over only two or three policy choices per state on policy update.) Note that the LP formulation for the 2000-state problem would have been about 2,000 by 80,000, and thus quite impractical. The author's version fit easily in core and used roughly 7 minutes of 7094 time per case.

5. A mixed backlogging problem with setup cost and zero or one period lag has been coded and is currently being tested. A generalized machine-replacement

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

problem will be tested similar to that developed by Eppen.[3] A cash-balance problem of the type studied by Eppen and Fama[3] will be included to check out the relative effectiveness of the method for separable problems.

REFERENCES

1. G. T. DE GHELLINCK AND G. D. EPPEN, "Linear Programming Solutions for Separable Markovian Decision Problems," *Management Sci.* **13**, 371–394 (1967).
2. G. D. EPPEN, "A Dynamic Analysis of a Class of Deteriorating Systems," *Management Sci.* **12**, 223–240 (1965).
3. ——— AND E. F. FAMA, "Solutions for Cash Balance and Simple Portfolio Problems," *J. of Business* **41**, 94–112 (1968).
4. R. A. HOWARD, *Dynamic Programming and Markov Processes*, The M.I.T. Press, Cambridge, 1960.
5. W. S. JEWELL, "Markov-Renewal Programming. I and II." *Opns. Res.* **11**, 938–948, 949–971 (1963).
6. T. E. MORTON, "On the Asymptotic Convergence Rate of Cost Differences for Markovian Decision Processes," *Opns. Res.* **19**, 244–249 (1971).
7. ———, "The Near Myopic Nature of the Lagged Proportional Cost Inventory Problem with Lost Sales," Management Sciences Research Report No. 182, Carnegie-Mellon University, September, 1969.
8. A. ODONI, "On Finding the Maximal Gain for Markov Decision Processes," *Opns. Res.* **17**, 857–860 (1969).
9. P. J. SCHWEITZER, "Perturbation Theory and Markovian Decision Processes," M.I.T. Operations Research Center Technical Report No. 15, 1965.
10. ———, "Perturbation Theory and Undiscounted Markov Renewal Programming," *Opns. Res.* **17**, 716–727 (1969).
11. ———, "Iterative Solution of the Functional Equations of Undiscounted Markov Renewal Programming," Institute for Defense Analysis, Arlington, Virginia 22202, March 31, 1970.
12. J. L. SMITH, "Markov Decisions on a Partitioned State Space," (to appear in *IEEE G-SSC Transactions*), 1970.
13. D. J. WHITE, "Optimal Revision Periods," *J. Math. Anal. and Appl.* **4**, 353–365 (1962).
14. ———, *Studies in Dynamic Programming*, Ph.D. Thesis, Birmingham University (1962).
15. ———, "Dynamic Programming, Markov Chains, and the Method of Successive Approximations," *J. Math Anal. and Appl.* **6**, 373–376 (1963).
16. ———, *Dynamic Programming*, Oliver and Boyd, 1969.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.

Copyright 1971, by INFORMS, all rights reserved. Copyright of Operations Research is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.

Downloaded from informs.org by [216.73.216.182] on 04 June 2026, at 04:36 . For personal use only, all rights reserved.