# Linear and Dynamic Programming in Markov Chains*

YOAV KISLEV AND AMOTZ AMIAD

**Some essential elements of the Markov chain theory are reviewed, along with programming of economic models which incorporate Markovian matrices and whose objective function is the maximization of the present value of an infinite stream of income. The linear programming solution to these models is presented and compared to the dynamic programming solution. Several properties of the solution are analyzed and it is shown that the elements of the simplex tableau contain information relevant to the understanding of the programmed system. It is also shown that the model can be extended to cover, among other elements, multiprocess enterprises and the realistic cases of programming in the face of probable deterioration of the productive capacity of the system or its total destruction.**

RECENTLY there has been growing interest in programming of economic processes which can be formulated as Markov chain models. One of the pioneering works in this field is Howard's *Dynamic Programming and Markov Processes* [6], which paved the way for a series of interesting applications. Programming techniques applied to these problems had originally been the dynamic, and more recently, the linear programming approach. Practically, a computer program to execute the dynamic programming calculation is simpler to prepare than one for the linear programming procedure. On the other hand, linear programming routines are readily available and allow great flexibility, as in parametric programming and sensitivity analysis. These features can be introduced into dynamic programming routines, although at an increasing cost. In this article we will show the lines of similarity between the two techniques and investigate some possible extensions and applications.

A finite Markov chain is a statistical model useful in describing various economic phenomena.[1] In this model, we envisage a process which is in a certain *state* $i$, where $i=1, 2, \cdots, n$ ($n$ finite), in a particular *period* or *stage*, and is transformed in the next period to a state $j$ ($j=i$ is permissible). The chain is described by an $n$-order *transition*, or *Markov* matrix, whose elements $p_{ij}$ are the probabilities that the process will go from state $i$ to state $j$. These probabilities are independent of the past history of the process.

* We are indebted to Hanna Lifson, who read previous versions of this article and helped to clarify many of its points. The article also benefited from discussions with Amnon Amir, Yakir Plessner, and the late Yoseph Levi, and from comments by the *Journal*'s readers. The work for the article was financed, in part, by a grant from the United States Department of Agriculture under P. L. 480.

[1] For a rigorous and complete treatment of Markov chains see Kemeny and Snell [8].

YOAV KISLEV *is a lecturer in the Department of Agricultural Economics, The Hebrew University, Rehovot, Israel.* AMOTZ AMIAD *is a graduate student in the School of Business Administration, The University of California, Berkeley.*

For example, let us consider a field whose state is defined by the level of humidity of the soil (measured in discrete units). The field may be transformed from one state to another with certain probabilities, depending on crop and weather conditions [1]. Additional illustrations might be a system of pieces of equipment whose failures are a stochastic process [2], or a warehouse where the state is given by the level of inventory [3, 4, 7].

In economic processes, with every state is associated a reward—or cost—for example, yield of the field, repair of machine, profits from sales of items out of inventory. The interesting cases are those in which the transition probabilities can be affected by *action*. A *policy* will then be the rule which dictates an action to be taken in every state. An *optimal* policy will be the policy under which total expected income from the process is maximized. In this framework, programming is the choice of an optimal policy from a given set of alternatives. The choice can be made efficiently by either dynamic or linear programming methods. We will investigate the relations between the two methods and interpret the results of the linear programming calculations. We hope to show not only that linear programming is applicable in this context, as has already been shown [3, 4, 7, 9], but also that its interpretation throws light on the "anatomy" of the system and clarifies understanding of its properties.

In order to simplify the discussion, we will make several assumptions to be relaxed later in the article. First, we assume regular Markov chains, that is, any state is probable far enough in the future. Also we explicitly assume that the transition matrices are not decomposable, that is, that the process cannot be split into two or more isolated chains. We further assume that a series of processes has a unique maximum present value. The discussion is limited to processes of indefinite duration—that is, an infinite economic horizon is assumed.

## Income Streams

We start the discussion by noting the mathematical equivalence of three analogous income streams[2] and naming these parallel cases for future reference. As usual, an income stream is defined by its annuity—$a_t$ in period $t$.

### The discounting case

Assume that a process yielding income lasts forever and that $a_t = a_0$ for all $t$. Let $r$ be the appropriate rate of interest and $\alpha = 1/(1+r)$ be the discounting factor. Then the worth of the source of income—its present value—is

$$z_\alpha = \sum_{t=0}^{\infty} \alpha^t a_0 = a_0/(1-\alpha), \tag{1}$$

since $0 < \alpha < 1$.

[2] In the present context, the analogy was first introduced by D'Epenoux [4].

### The deterioration case

Assume now that income from the source is not constant, but deteriorates at the rate $\beta$, where $\beta = a_{t+1}/a_t$ and $0<\beta<1$ (radioactive decay). Then the present *not discounted* value of the source of income is

$$z_\beta = \sum_{t=0}^{\infty} a_t = \sum_{t=0}^{\infty} \beta^t a_0 = a_0/(1-\beta). \tag{2}$$

### The breakdown case

In this third case, consider a constant annuity, $a_0$, as long as the source of income exists. There is, however, a constant probability $1-\gamma$, at every period $t$, that the source will be destroyed before the coming of the next period. Hence, $\gamma$ is the probability of survival. Here expected worth of the income stream (not discounted) is

$$z_\gamma = \sum_{t=0}^{\infty} \gamma^t a_0 = a_0/(1-\gamma). \tag{3}$$

These three cases are mathematically equivalent. Of course, they could be consolidated into one general case which would constitute a mixture of the three. In the course of our discussion, we shall make use of the analogy of the separate cases, as well as of the mixed case.

It will also be useful if we note that the previous equations can be rewritten in a slightly different form. Instead of (1), for example, write the recurrence relation

$$\begin{aligned} z_\alpha &= a_0 + \alpha \sum_{t=0}^{\infty} \alpha^t a_0 \\ &= a_0 + \alpha z_\alpha. \end{aligned} \tag{1'}$$

We have named this new form the *two-steps form* of (1). It emphasizes that the present value of the infinite income stream is composed of an immediate annuity, plus the present value of the same income stream started one period later. Similar forms and interpretations can be given to (2) and (3).

## Markov Chains in Economic Systems

Consider a Markov chain with an $n$-order transition matrix $P(n\times n) = [p_{ij}]$. Since the $p_{ij}$ elements are probabilities,

$$\sum_j p_{ij} = 1 \qquad (i = 1, 2, \cdots n). \tag{4}$$

Let the current state of the process—state $i$—be denoted by a *state vec-*

*tor*[3] $E_i$ $(1 \times n)$. $E_i$ is the unit row vector with the unity in position $i$. Given a state vector $E_i$, the vector $E_iP$ is the probability vector for the states of the process in the succeeding stage. In the stage after that, the probabilities will be $(E_iP)P = E_iP^2$. In general, the probabilities for the $t$th period constitute the vector $E_iP^t$. Also, let a rewards row vector $C(1 \times n) = [c_i]$ associate an immediate reward[4] with every state $i$. The present value of the next period's reward is, therefore, $\alpha E_iPC'$. Thus, if the process continues indefinitely, the expected present value of all future incomes—the worth of the process currently in state $i$—is

$$z_i = \sum_{t=0}^{\infty} E_i(\alpha P)^t C' = E_i(I - \alpha P)^{-1}C' \tag{5}$$

where $\alpha$ is, as previously, the discounting factor.

Utilizing scalar notation, we may introduce the two-steps form of (5):

$$z_i = c_i + \alpha \sum_{j=1}^{n} p_{ij}z_j. \tag{5'}$$

Starting from a state $i$, the worth of the process is the immediate reward $c_i$, plus the expected worths of the states of the next stage, discounted one period.

To consider all starting states, we replace $E_i$ by the unit matrix $I$ and write

$$Z' = I(I - \alpha P)^{-1}C' = (I - \alpha P)^{-1}C', \tag{6}$$

where $Z$ is the $(1 \times n)$ vector whose elements are the $z_i$ values of (5).

In terms of the previous section, the case presented here is the discounting case, within the framework of the Markovian model. We shall now make use of the analogy to the breakdown case; this will link us directly to the general theory of Markov chains and provide us with convenient terminology and greater insight. Toward this end, consider a process with a transition matrix $T$, of the order $n+1$, which can be partitioned:

$$T = \begin{bmatrix} Q & H' \\ 0 & 1 \end{bmatrix}.$$

In $T$, $H(1 \times n)$ is a probability vector, 0 a zero vector, and 1 a scalar. The Markov chain defined by $T$ consists of two sets of states: one, *transient*, with the $n$ states in $Q$, and one, the $(n+1)$ state—*ergodic*. Once the process

[3] $E_i$ can be regarded as a particular case of a *state probability vector*.

[4] The assumption in the text is that the reward is associated with the occupation of the state. It is not difficult to incorporate the alternative assumption that the reward is due to a particular transition from state $i$ to state $j$ [7, p. 460].

reaches the ergodic state, it will be *absorbed* there and will never re-enter any of the transient states. The elements of $H$ are, therefore, the probabilities that the process would be transformed from each of the transient states into the ergodic state. $Q$ is the transition matrix of the transient states.

Associated with every transient set—with every matrix $Q$—is a *fundamental* square matrix, $V=[v_{ij}]$.

$$V = (I - Q)^{-1} = \sum_{t=0}^{\infty} Q^t. \tag{7}$$

The elements $v_{ij}$ indicate the expected number of times that a process, currently in state $i$, will be in state $j$ before being absorbed in the ergodic state (including the current stage in the count of $v_{ii}$). To complete the analogy, let every transient state $i$ carry a reward $c_i$, and the ergodic state represent total breakdown of the system—zero income. Total expected income (*not discounted*) for a process starting in state $i$, is

$$\begin{aligned} z_i &= \sum_{t=0}^{\infty} E_i Q^t C' \\ &= E_i (I - Q)^{-1} C'. \end{aligned} \tag{8}$$

By defining $Q$ of (7) and (8) as $Q=\alpha P$, we return to the discounting case and may treat the matrix $\alpha P$ as if it were the transient part of a Markov process. Here we shall name the $v_{ij}$ elements of $V=(I-\alpha P)^{-1}$, the *expected discounted* number of times that a process, currently in state $i$, will be in state $j$. These numbers are finite, while physically the process will continue for an infinite duration.

Since $P$ is a transition matrix, the sum of every row of $\alpha P$ is $\alpha$ (see equation 4), and therefore all elements of the corresponding $H$ vector are $1-\alpha$, which is also the sum of all rows in the matrix $I-\alpha P$. Hence, total discounted number of stages in any state, starting from state $i$, is by (A.2) in the Appendix

$$\sum_{j=1}^{n} v_{ij} = 1/(1 - \alpha) \qquad (i = 1, 2, \cdots, n). \tag{9}$$

We can interpret this result, again utilizing the analogy to the breakdown case, as follows: $1-\alpha$ is the probability of breakdown of the system in any stage; therefore, $\alpha$ is the probability of survival. Hence, the total expected number of stages before breakdown will be

$$\sum_{t=0}^{\infty} \alpha^t = 1/(1 - \alpha). \tag{10}$$

The interpretation we gave to the elements of the fundamental matrix $V$ permits the rewriting of (8) as

(8′)
$$z_i = \sum_{j=1}^{n} v_{ij}c_j \qquad (i = 1, 2, \cdots, n),$$

which can easily be verified algebraically and interpreted economically.

Programming will be meaningful in those cases in which a certain process can be chosen from several alternatives. Instead of enumerating all possible transition matrices, we consider an *expanded* matrix $R\ (m\times n) = [p_{ij}^{d(i)}]$, which consists of $k_i$ different probability rows for every state $i$, $m = \sum_{i=1}^{n} k_i$. The superscript $d(i)$ indicates an *action* to take in state $i$ where $d(i) = 1, 2, \cdots, k_i$. Generally, we shall eliminate, for brevity, the index $i$ of $d(i)$ and write $p_{ij}^d$. The action indicated by the superscript will affect the transition probabilities (probabilities of failure of equipment, for example, can be affected by actions of maintenance). The immediate reward of the state $i$ is also affected by the action; for example, cost of action is deducted from the gross value of the reward. Thus, the vector $C$ is also expanded and its elements are now $c_i^d$. An expanded probability matrix $R$ of the dimension $6\times 2$, with the corresponding immediate rewards vector $C$, is given in Table 1. Thus, in the table, if in state 1 action $a_1$ is taken, $d(1) = 1$, the transition probabilities are $p_{11}^1 = 0.20$, $p_{12}^1 = 0.80$ and the expected immediate reward is $c_1^1 = \$5.00$.

**Table 1. An expanded transition matrix with rewards**

| State | Actions[a] | Probabilities of transition (Matrix $R$) | | Immediate rewards (Vector $C'$) |
|---|---|---|---|---|
| | | to state 1 | to state 2 | |
| State 1 | $a_1$ | 0.20 | 0.80 | \$5.00 |
| | $a_2$ | 0.00 | 1.00 | 4.50 |
| | $a_3$ | 1.00 | 0.00 | 0.00 |
| State 2 | $b_1$ | 0.60 | 0.40 | \$2.00 |
| | $b_2$ | 0.40 | 0.60 | 2.30 |
| | $b_3$ | 0.00 | 1.00 | 0.00 |

[a] Actions are listed by names. For example, $a_1$ is the name of the action in state 1 for which $d(1) = 1$.

The Markov process will be determined when a decision vector $D(1\times n)$ is chosen, designating a $d(i)$ value for every $i$, that is, specifying a policy—an action to take in every possible state.[5] By deciding on a $D$, one chooses a particular transition matrix $P$, out of $R$, for the process at hand and a corresponding vector $C$ of immediate rewards.

Programming for maximal expected income can be performed by the budgeting method—by listing all possible $P$ square matrices out of $R$, calculating, by (5), expected worth of each, and selecting the one with the

[5] We shall regard the vector $D$, interchangeably, as either the vector consisting of the indices $d(i)$ or of the names of the actions $a_1$, $b_2$, etc.

highest $z_i$. This might be extremely laborious. Instead, dynamic or linear programming methods may be applied.

## Dynamic Programming

In this section we will follow Hadley [7, pp. 454–460], who also provides the proofs for the procedure described here.

To select an optimal decision vector $D$ by the dynamic programming method, start from an arbitrary $D$, call it $D(1)$, thus selecting a corresponding matrix $P(1)$ and a vector $C(1)$. Now calculate a vector $Z(1)$ of expected present values for all starting states.

$$\begin{aligned} Z(1)' &= [I - \alpha P(1)]^{-1} C(1)' \\ &= C(1)' + \alpha P(1) Z(1)'. \end{aligned} \tag{11}$$

The last line—the two-steps form of (11)—is the matrix form of (5′).

Next, check whether $D(1)$ is optimal. This is done by the following recurrence procedure: define a *test policy* to be the policy $D(1)$ for all future stages but not necessarily for the current one. For the current stage, the test policy associates an alternative action $d(i)$—not necessarily in $D(1)$—with state $i$. Now evaluate

$$z_i = \max_d \left[ c_i^d + \alpha \sum_{j=1}^{n} p_{ij}^d z_j(1) \right] \qquad (i = 1, 2, \cdots, n). \tag{12}$$

A new decision vector $D(2)$ emerges, consisting, for every $i$, of the $d(i)^*$ element that maximizes the expression in (12). If $D(1)$ is an optimal policy, then $D(2) = D(1)$. If not, calculate

$$Z(2)' = [I - \alpha P(2)]^{-1} C(2)', \tag{13}$$

and repeat (12) and (13) until $D(k) = D(k-1) = D^*$.[6] $D^*$ is the optimal policy which maximizes present value of expected income from the process.

In this procedure, all possible starting states are considered. Thus, $D^*$ is invariant under different starting states—the set of optimal actions to take in every possible state is independent of the current state of the process.

## Linear Programming

Our linear programming problem [5] will be

$$\begin{cases} a.\ \max C\Pi' \\ \quad \text{subject to} \\ b.\ M\Pi' = E_i' \\ c.\ \Pi \geq 0. \end{cases} \tag{14}$$

[6] The optimal policy need not be unique; several $D$ vectors might lead to the same maximal present value. It is, however, not difficult to protect the computer program against cycling.

In (14), $C$ is the expanded immediate rewards vector; $\Pi$ is the solution vector to the linear programming problem; $E_i$ is, as previously, the unit state vector with unity in position $i$. The matrix $M(n\times m)$ is constructed of the expanded transition matrix $R$ by first expanding a unit matrix to a matrix $J(m\times n)$, which consists of $k_i$ identical $E_i$ unit row vectors for every $i$, and then

$$M = (J - \alpha R)'. \tag{15}$$

The matrices $J$, $R$, and $M$, for a problem with two states and two actions in each state, are illustrated below.

$$J = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \qquad \alpha R = \begin{bmatrix} \alpha p_{11}{}^1 & \alpha p_{12}{}^1 \\ \alpha p_{11}{}^2 & \alpha p_{12}{}^2 \\ \alpha p_{21}{}^1 & \alpha p_{22}{}^1 \\ \alpha p_{21}{}^2 & \alpha p_{22}{}^2 \end{bmatrix}$$

$$M = \begin{bmatrix} 1-\alpha p_{11}{}^1 & 1-\alpha p_{11}{}^2 & -\alpha p_{21}{}^1 & -\alpha p_{21}{}^2 \\ -\alpha p_{12}{}^1 & -\alpha p_{12}{}^2 & 1-\alpha p_{22}{}^1 & 1-\alpha p_{22}{}^2 \end{bmatrix}$$

Table 2 is the simplex table for the example of Table 1. The matrix $M$ constitutes the bulk of the first section—the input–output coefficients—to which a unit matrix of slack variables (artificial activities) was added. The assumption in the table is that the process is started in state 1.

We shall now show that the solution to the linear programming problem (14), like the dynamic programming solution, will select a policy that will maximize expected present value of income from the process at hand.

Following the usual linear programming convention, we add slack variables and partition the vectors $\Pi$ and $C$ and the matrix $M$:

$$\Pi = [\Pi_s \quad \Pi_o \quad \Pi_1], \qquad C = [C_s \quad C_o \quad 0], \qquad M = [M_s \quad M_o \quad I], \tag{16}$$

where $s$ is the index of the part in the basis, and $o$ is the index of the part not in the basis. By (14) and (16),

$$M_s\Pi_s' + M_o\Pi_o' = E_i' \tag{17}$$

and

$$\begin{aligned} \Pi_s' &= M_s^{-1}E_i' - M_s^{-1}M_o\Pi_o' \\ &= M_s^{-1}E_i' \end{aligned} \tag{18}$$

since $\Pi_o = 0$.

It was shown by Wolfe and Danzig [9] that the linear programming procedure assures that, in (18), $M_s^{-1} = [(I-\alpha P_s)^{-1}]'$, where $P_s$ is a transition matrix selected from $R$. This means that there will be exactly one column in $M_s$ for every possible starting state. We repeat, for completeness, the es-

**Table 2. First and last simplex sections**[a]

| | $C_s$ | $C\rightarrow$ | | State 1 | | | State 2 | | | "Slacks" | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 5.0 | 4.5 | 0.0 | 2.0 | 2.3 | 0.0 | 0 | 0 |
| | | Basis | $\Pi_s$ | $a_1$ | $a_2$ | $a_3$ | $b_1$ | $b_2$ | $b_3$ | $d_1$ | $d_2$ |
| First section | 0 | $d_1$ | 1 | 0.82 | 1.00 | 0.10 | −0.54 | −0.36 | 0.00 | 1 | 0 |
| | 0 | $d_2$ | 0 | −0.72 | −0.90 | 0 | 0.64 | 0.46 | 0.10 | 0 | 1 |
| Last section | 5.0 | $a_1$ | 4.706 | 1.00 | 1.133 | 0.471 | 0.00 | 0.132 | 0.397 | 4.706 | 3.971 |
| | 2.0 | $b_1$ | 5.294 | 0.00 | −0.133 | 0.529 | 1.00 | 0.868 | 0.603 | 5.294 | 6.029 |
| | | $z_j$ | 34.118 | 5.0 | 5.397 | 3.412 | 2.0 | 2.397 | 3.191 | 34.118 | 31.912 |
| | | $z_j$-$c_j$ | | 0.0 | 0.897 | 3.412 | 0.0 | 0.097 | 3.191 | 34.118 | 31.912 |

[a] Based on Table 1, with $\alpha=0.9$. For additional explanations see text.

sence of the proof: since $E_i \geq 0$ and $\Pi_o = 0$, then (14.c) and (17) can be simultaneously maintained only if every row of $M_s$ contains at least one nonnegative element. The only positive elements in $M$ are of the form $1-\alpha p_{ii}{}^d$, of which there is one in every column. The matrix $M_s$ is of the order $n$; it has $n$ columns, each with exactly one element of the form $1-\alpha p_{ii}{}^d$. It also has $n$ rows, and must, as stated, have at least one nonnegative element in every row. Hence, it will have exactly one element of the form $1-\alpha p_{ii}{}^d$ in every row. Therefore, there will be exactly one element $1-\alpha p_{ii}{}^d$ in every row and column of $M_s$, which completes the proof.

Equation (18) can now be written as

$$\Pi_s' = [(I - \alpha P_s)^{-1}]' E_i', \tag{19}$$

and, therefore,

$$C\Pi' = C[(I - \alpha P_s)^{-1}]' E_i'. \tag{20}$$

Comparing (20) to (5), we see that $C\Pi'$ is the worth of a Markov process currently in state $i$. The maximal value of $C\Pi'$—the value of the objective function in the solution to (14)—is the maximal worth of a system of Markov processes.

The solution to (14) determines a policy vector, $D_s$, which can be constructed by observing the vectors in the basis. It stems from Property 7 of the next section that $D_s$ is not affected by the starting state of the process. Thus, $D_s$ of linear programming, like $D^*$ of the dynamic programming solution, is an optimal policy vector. The same expected maximal present value is reached by the linear and the dynamic programming methods and, if there is only one unique optimal policy vector, then $D_s = D^*$.

In the next section we shall investigate some of the properties and possible interpretations of the simplex routine and elaborate further on the lines of similarity between the dynamic and the linear programming methods.

## Properties of the Simplex Solution

It will be convenient if we state here the criterion function of the simplex routine—the $Z-C$ row vector—

$$\begin{aligned} Z - C &= C_s M_s^{-1} [M_s \ M_o \ I] - [C_s \ C_o \ 0] \\ &= C_s [I \ M_s^{-1} M_o \ M_s^{-1}] - [C_s \ C_o \ 0] \\ &= [0 \ C_s M_s^{-1} M_o - C_o \ C_s M_s^{-1}]. \end{aligned} \tag{21}$$

Reference to the element of (21) is made in the discussion that follows.

### Property 1

As was previously explained, by programming for a $D_s$ we select a transition matrix $P_s$ and $M_s = (I - \alpha P_s)'$. Therefore, by (7),

(22)
$$\begin{aligned} M_s^{-1} &= [(I - \alpha P_s)^{-1}]' \\ &= [(I - Q)^{-1}]' \\ &= V', \end{aligned}$$

where $V$ is the fundamental matrix associated with the "transient" matrix $\alpha P_s$. Thus, in Table 2, consistent with the terminology introduced in the section "Markov Chains in Economic Systems," the expected discounted number of times that a process, currently in state 2, will be in state 1 is 3.971, and in state 2 is 6.029.

### Property 2

By equations (22) and (9), the sums of the columns of $M_s^{-1}$ are $1/(1-\alpha)$. In Table 2, $\alpha = 0.9$, $1/(1-\alpha) = 10$, and the sums are

$$\text{column } d_1\text{:}\quad 4.706 + 5.294 = 10$$

$$\text{column } d_2\text{:}\quad 3.971 + 6.029 = 10.$$

### Property 3

Let $u_{ik}^o$ be the simplex table element for row $i$, state $k$, and $o$ a value for $d(k)$ outside the basis. Thus $u_{ik}^o$ is defined by $M_s^{-1}M_o = [u_{ik}^o]$. For example, in Table 2, column $b_2$, last section, $u_{12}^2 = 0.132$.

By Property 1, $M_s^{-1}M_o = V'M_o$. Therefore, in scalar notations and denoting by $p_{ij}^o$ the transition probabilities in $M_o$ (thus $p_{ij}^o$ is the probability of transition from $i$ to $j$ with action $o$),

(23)
$$\begin{aligned} u_{ik}^o &= -\sum_{j \neq k} v_{ji}\alpha p_{kj}^o + v_{ki}(1 - \alpha p_{kk}^o) \\ &= v_{ki} - \alpha \sum_j p_{kj}^o v_{ji} \qquad (k = 1, 2, \cdots, n). \end{aligned}$$

Examining the last line—the two-steps form of (23)—one recognizes that $u_{ik}^o$ is the difference between ($a$) the expected discounted number of times that a process, currently in state $k$, will be in state $i$—if the present policy is adopted ($v_{ki}$), and ($b$) the expected discounted number of times that a process starting in state $k$ will be in state $i$ if the test policy, with action $d(k) = o$ for the current stage and the basic policy for all future stages, is adopted. Action $o$ is taken *once* and the basic policy $D_s$ is followed for all other stages. Hence, $u_{ik}^o$ is the marginal rate of substitution of the present (basic) policy to the *alternative policy* with action $o$ for state $k$ in all stages. The substitution is in the decision vector $D$, and it is "marginal" in that the alternative policy is adopted for only one stage—the current stage.

### Property 4

The sum of the elements in every column of the simplex table is unity.

For actions in the basis this is obvious—these columns are unit columns. For actions not in the basis, the sums of the elements of the matrix $M_s^{-1}M_o$ are also unity. Since the sum of every column of the matrix $M$ is $1-\alpha$, therefore, by *A.2* in the Appendix, the sums of the columns of $M_s^{-1}$ are all $1/(1-\alpha)$. Hence, by *A.1* of the Appendix, the column sums in $M_s^{-1}M_o$ are

$$(1-\alpha)/(1-\alpha) = 1.$$

For example, in Table 2, column $a_1$, the sum is

$$1.133 - 0.133 = 1.0.$$

Making use of (23), we write the column sum as

$$\sum_i u_{ik}{}^o = \sum_i \left(v_{ki} - \alpha \sum_i p_{kj}{}^o v_{ji}\right) \tag{24}$$
$$= 1 \qquad (k = 1, 2, \cdots, n).$$

The sum in the right-hand side of the first line of (24) is the difference in the total discounted number of stages under the two policies—the basic policy and the test policy. In general, the total discounted number of stages is the same under any policy (Property 2). The difference in (24), which is unity, stems from the fact that the count of stages for the basic policy includes the current stage (the sum in equation 7, for example, goes from *zero* to infinity), whereas for the test policy the count starts from the next stage and omits the current one.

**Property 5**

The dual values, the elements of the row vector $C_sM_s^{-1}$, are the values of the alternative objective function, under the basic policy, for all possible starting states. If we write the element $k$ of this vector as $z_k{}^s$ and denote by $c_i{}^s$ the element of $C_s$, the dual values are

$$z_k{}^s = \sum_i c_i{}^s v_{ki} \qquad (k = 1, 2, \cdots, n), \tag{25}$$

which is exactly (8′). In the table, $z_1{}^s = \$34.118$—the value of the objective function for a process starting in state 1; $z_2{}^s = \$31.912$—the objective function for a process starting in state 2.

**Property 6**

The elements in the $Z-C$ row for actions not in the basis (21) are $C_sM_s^{-1}M_o - C_o$.

For a state $k$ and action $o$, we shall denote these elements in the criterion function as $z_k{}^o - c_k{}^o$ and write in scalar notation

$$(26) \quad z_k^o - c_k^o = -\sum_{j\neq k} \alpha p_{kj}^o z_j^s + z_k^s(1 - \alpha p_{kk}^o) - c_k^o$$
$$= z_k^s - \left(c_k^o + \alpha \sum_j p_{kj}^o z_j^s\right) \qquad (k = 1, 2, \cdots, n).$$

The term in the parenthesis in the second version of (26) is the two-steps form of the objective function, for a process in state $k$, under the test policy. The alternative policy—with action $o$ for state $k$—will be adopted throughout all future periods if the value of (26) is negative, that is, if the test policy is superior to the basic policy. Since the process lasts forever, if action $o$ for state $k$ is superior for the current state it will also be superior in any future state. This principle is, of course, the rationale behind the dynamic programming procedure, outlined in the section, "Dynamic Programming." It is evident now that the criteria for changing a policy, from iteration to iteration, are the same in the linear and in the dynamic programming techniques. The one difference, however, is that in the simplex method of linear programming one element of $D$ is replaced at a time, whereas in dynamic programming a new vector $D$ is constructed at every iteration, which can differ from the previous policy by several elements.

**Property 7**

The optimal policy is not affected by the starting state of the process. To see this, one must show that a change of $E_i$ to $E_j$ will not alter the basis of the linear programming solution. Denote a solution vector associated with the starting state $i$ by $\Pi_s(i)$. We know [5, p. 133] that

$$(27) \qquad \Pi_s(i) = M_s^{-1}E_i' \geq 0$$

is a feasible solution for a starting state $i$, and that a change of $E_i$ to $E_j$ will not alter the optimal basis, $M_s$, if, in addition to (27),

$$(28) \qquad \Pi_s(j) = M_s^{-1}E_j' \geq 0.$$

The condition in (28) is maintained, since all elements in $M_s^{-1}$—the $v_{ij}$ elements—are nonnegative.

## Extensions and Applications

### A multiprocess system

Generally an enterprise will not be a single process but will constitute a system of many processes—fields in a farm, for example, or machines in a factory, or units of an operating army. If we assume that these processes are independent and let $e_i$ be the number of processes, at present in state $i$, in an enterprise, then the total worth of the enterprise is

(29) $$W = \sum_{i=1}^{n} e_i z_i^s,$$

where $z_i^s$ is defined as in (25). $W$ can be easily calculated from the dual values of the linear programming solution.

Alternatively, a direct approach can be implemented: define a state vector $E(1 \times m)$ whose elements are the $e_i$ values (the vector $E_i$ is now a particular value of $E$), and instead of (14) solve as follows:

(14′) $$\begin{cases} a.\ a \max C\Pi' \\ \quad \text{subject to} \\ b.\ M\Pi' = E' \\ c.\ \Pi \geq 0. \end{cases}$$

The maximal value of the objective function in (14′) will be the $W$ of (29).

## A decomposable system

Up to now, we have assumed a system that is not decomposable. This need not be the only case. If the matrix $M$ is decomposable, and if, say, $E_i = E_1$, then (14.$b$) will be

(30) $$\begin{bmatrix} M_1 & 0 \\ & \\ 0 & M_2 \end{bmatrix} \begin{bmatrix} \Pi_1' \\ \\ \Pi_2' \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}.$$

The elements of $\Pi_2$ in (30) must be zeros by the formulation of the problem. The second chain will not be programmed at all.

To avoid this difficulty, it has been suggested [4, 7] that, even in cases of single-process systems, (14′) be solved with an arbitrary nonzero $E$—the vector on the right-hand side. The optimal policy is not affected by this device. The calculated value of the objective function depends, of course, on the selected values for $E$.

## An inferior state

Another assumption was that chains were regular, that their fundamental matrices had no zero entries, that all states were probable far enough in the future. In practice, one might encounter states which are economically inferior and can be avoided—small inventories, for example, or old machinery. If it is possible, and the appropriate actions are specified, a policy will be selected that will avoid the inferior states. If the process is started in such a state, it will leave that state in one or a few periods. As an example, consider, in Table 3, a new expanded matrix constructed from Table 1 by eliminating, for simplicity, $a_3$ and $b_3$ and adding a third state.

**Table 3. An expanded $R$ matrix with an inferior state**

| States | Actions | Probabilities of transition | | | Immediate rewards |
|---|---|---|---|---|---|
| | | to state 1 | to state 2 | to state 3 | |
| State 1 | $a_1$ | 0.20 | 0.80 | 0.00 | \$5.00 |
| | $a^2$ | 0.00 | 1.00 | 0.00 | 4.50 |
| | $a^4$ | 0.00 | 0.00 | 1.00 | 0.00 |
| State 2 | $b_1$ | 0.60 | 0.40 | 0.00 | \$2.00 |
| | $b_2$ | 0.40 | 0.60 | 0.00 | 2.30 |
| | $b^4$ | 0.80 | 0.00 | 0.20 | 1.00 |
| State 3 | $c_1$ | 1.00 | 0.00 | 0.00 | \$4.00 |
| | $c_2$ | 0.10 | 0.70 | 0.20 | 4.50 |

Programming,[7] one finds that the optimal policy vector, $D_s$, of this process consists of $a_1$, $b_1$, and $c_1$ and the corresponding transition matrix is, therefore,

$$P_s = \begin{bmatrix} 0.20 & 0.80 & 0 \\ 0.60 & 0.40 & 0 \\ 1.00 & 0.00 & 0 \end{bmatrix}.$$

## An absorbing state

As experience teaches, some policies may lead to irreversible, and sometimes destructive, results. A particular crop rotation will not protect the soil and a heavy rain may cause erosion and destroy all future possibility of cultivating the field. A monopolist may charge high prices that will breed rival firms. These are breakdown cases whose Markov matrices are like $T$ of the section "Markov Chains in Economic Systems." Some reflection, and the example below, will show that "destructive" policies may sometimes be optimal. In fact, whether they will be chosen or rejected depends, all other things being the same, on the discounting rate—the higher the rate of interest, the more probable it is that a "suicidal" policy, which yields high income until destruction, will be adopted.

As an example, consider the expanded $R$ matrix given in Table 4. Note that the reward for the third, absorbing state is zero and that no possible action is attached to this state, which stands for the collapse of the economic system. The optimal policies for this system are listed in Table 5. Also given in Table 5 are the probabilities that a process starting in state 1 will be in any of the states at some specified $t$ period. Once action $a_1$ is introduced, the process must end in state 3.

[7] We took $\alpha=0.9$ in this case too.

**Table 4. Expanded matrix with rewards, possible breakdown case**

| States | Actions | Probabilities of transition | | | Immediate rewards |
|---|---|---|---|---|---|
| | | to state 1 | to state 2 | to state 3 | |
| State 1 | $a_1$ | 0.40 | 0.55 | 0.05 | $6.00 |
| | $a_2$ | 0.70 | 0.30 | 0.00 | 4.00 |
| State 2 | $b_1$ | 0.30 | 0.50 | 0.10 | $5.00 |
| | $b_2$ | 0.40 | 0.60 | 0.00 | 3.00 |
| State 3 | | 0.00 | 0.00 | 1.00 | $0.00 |

The right-hand section of the table lists the expected number of times (not discounted) that the process will be in any of the states, under the optimal policies. The numbers in the parentheses are the standard deviations of these numbers [8, Chap. 3]. Thus, in Table 5, in the lower section, under policy $a_1b_1$, the number of times that a process starting in state 1 will be in state 2 is $7.35 \pm 7.49$: the standard deviations are quite high in relation to the expected values. Under policy $a_2b_2$, the process will never reach the absorbing state and will be an infinite number of times in both states 1 and 2.

## Depletion and deterioration

The last section dealt with a system with a possible breakdown case. More probable than the sudden "death" or collapse of the economic process is the possibility of depletion or decay of productivity—the deterioration case. A particular crop rotation will gradually impoverish the field; pumping of coastal groundwater damages the quality of that source; a certain maintenance routine results in a gradual reduction of income from an asset. In some respects depletion and deterioration are "historical" phenomena, alien to the Markovian assumption of independence. However, by utilizing the analogy of the deterioration case to the other two cases (in the section "Income Streams"), one may incorporate realistic types of these phenomena into our model.

Assume, for simplicity, a zero rate of interest, namely $\alpha = 1$, and let income, productivity, service, etc. from the economic process deteriorate at a rate $1-\beta (0<\beta<1)$ per period. Expected, not discounted, worth of the income stream is

(31)
$$\begin{aligned} z_i &= \sum_{t=0}^{\infty} E_i(\beta P)^t C' \\ &= E_i(I - \beta P)^t C'. \end{aligned}$$

More interesting will be the case in which the rate of deterioration is not

**Table 5. Characteristics of optimal policies for data in Table 4**

| Range of interest rate | State | Optimal policy | Transition matrix | | | Probability of state ($E_l p^t$) | | | | | | | Expected number of transitions in state (standard deviation) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | $t=0$ | 1 | 2 | 3 | 5 | 10 | ∞ | State 1 | State 2 |
| *percent* | | | | | | | | | | | | | | |
| 0–10 | 1 | $a_2$ | 0.70 | 0.30 | 0.00 | 1.00 | 0.70 | 0.610 | 0.583 | 0.572 | 0.571 | 0.571 | ∞ | ∞ |
| | 2 | $b_2$ | 0.40 | 0.60 | 0.00 | 0.00 | 0.30 | 0.390 | 0.417 | 0.428 | 0.429 | 0.429 | ∞ | ∞ |
| | 3 | | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0 | 0 |
| 11–20 | 1 | $a_1$ | 0.40 | 0.55 | 0.05 | 1.00 | 0.40 | 0.380 | 0.372 | 0.357 | 0.321 | 0.000 | 20 (19.5) | 27.5 (25.8) |
| | 2 | $b_2$ | 0.40 | 0.60 | 0.00 | 0.00 | 0.55 | 0.550 | 0.539 | 0.517 | 0.465 | 0.000 | 20 (19.5) | 30.0 (29.5) |
| | 3 | | 0.00 | 0.00 | 1.00 | 0.00 | 0.05 | 0.070 | 0.089 | 0.126 | 0.214 | 1.000 | ∞ | ∞ |
| 21 and up | 1 | $a_1$ | 0.40 | 0.55 | 0.05 | 1.00 | 0.40 | 0.325 | 0.295 | 0.248 | 0.162 | 0.000 | 5.35 (3.64) | 7.35 (7.49) |
| | 2 | $b_1$ | 0.30 | 0.60 | 0.10 | 0.00 | 0.55 | 0.550 | 0.509 | 0.429 | 0.280 | 0.000 | 4.00 (4.77) | 8.00 (7.48) |
| | 3 | | 0.00 | 0.00 | 1.00 | 0.00 | 0.05 | 0.125 | 0.196 | 0.323 | 0.558 | 1.000 | ∞ | ∞ |

just one rate for the process but differs from state to state. Now, at the period in which the process occupies state $i$, its productivity deteriorates at the rate $\beta_i$. For example, expected income from the next stage of a process, currently in state $i$, is $\beta_i \sum_j p_{ij}c_j = \sum_j \beta_i p_{ij} c_j$, or, in matrix notation, $E_iBPC'$, where $B$ is a diagonal matrix with $\beta_i$ on the diagonal and zeros elsewhere.

Expected value of an everlasting process is, therefore,

$$(32) \qquad z_i = \sum_{t=0}^{\infty} E_i(BP)^t C' = E_i(I - BP)^{-1}C'.$$

It is easily seen now that to allow nonzero rates of interest, one simply incorporates $\alpha$ in (32) to form

$$(33) \qquad z_i = E_i(I - \alpha BP)^{-1}C'.$$

For alternative policies and programming, $B$ is expanded to allow $\beta_i^{(di)}$—deterioration is a function of state and action.

### Growth and appreciation

If deterioration is represented by $\beta_i < 1$, growing productivity or appreciation can be represented by $\beta_i > 1$. In fact, (33) applies to cases of appreciation so long as $\alpha\beta_i^d < 1$ for all $i$ and $d$. If $\alpha\beta_i^d \geq 1$ for some $i$ and $d$, the existence of the inverse matrix of (33) is not assured; that is, $z_i$ in (33) need not be finite. Programming is, however, still possible by, for example, considering a finite horizon. We shall not pursue this subject further here.

## Concluding Remarks

We have tried to show that the Markov chain model may be used in a variety of economic applications. The discussion of the linear programming solution has facilitated, we trust, better understanding of the Markov process and of the rival dynamic programming method. An unsolved problem is that of the incorporation of the regular linear programming limitations and requirements into the present model. The difficulty lies in the fact that the solutions to the Markovian systems are in terms of expected numbers, while the actual magnitudes will change from period to period and may under- or overshoot limitations and requirements, if such exist. We hope to return to this question in the future.

## Appendix

### A. 1

Let $B = [b_{ij}]$ and $F = [f_{ij}]$ be $n$-order square matrices with constant column sums: $\sum_i b_{ij} = s$ $(j = 1, 2, \cdots, n)$ and $\sum_i f_{ij} = t$ $(j = 1, 2, \cdots, n)$. If

we let the matrix $G=[g_{ij}]$ be the product matrix of $B$ and $F(G=BF)$, then the column sums of $G$ are all $st$.
Proof:

$$\begin{aligned}\sum_i g_{ij} &= \sum_i \sum_k b_{ik} f_{kj} \\ &= \sum_k f_{kj} \sum_i b_{ik} \\ &= s \sum_k f_k \\ &= st.\end{aligned}$$

**A. 2**

If we let $H=[h_{ij}]$ be the inverse matrix of $B(H=B^{-1})$, then $\sum_i h_{ij}=1/s$ $(j=1, 2, \cdots, n)$.
Proof:

$$\begin{aligned}BH &= I \\ \sum_i \sum_j b_{ij} h_{jk} &= 1 \\ \sum_j h_{jk} \sum_i b_{ij} &= 1 \\ s \sum_j h_{jk} &= 1 \\ \sum_j h_{jk} &= 1/s.\end{aligned}$$

## References


[1] BURT, OSCAR R., AND JOHN R. ALLISON, "Farm Management Decisions with Dynamic Programming," *J. Farm Econ.* 45:121–136, Feb. 1963.
[2] ———, "Optimal Replacement Under Risk," *J. Farm Econ.* 47:324–346, May 1965.
[3] DE GHELLINCK, GUY T., AND GARY D. EPPEN, "Linear Programming Solutions for Separable Markovian Decision Problems," *Mgt. Sci.* 13:371–394, Jan. 1967.
[4] D'EPENOUX, F., "A Probabilistic Production and Inventory Problem," *Mgt. Sci.* 10:98–108, 1963.
[5] GASS, SAUL I., *Linear Programming: Methods and Applications*, 2d ed., New York, McGraw-Hill Book Co., 1964.
[6] HOWARD, R., *Dynamic Programming and Markov Processes*, Cambridge, Mass., Technology Press, 1960.
[7] HADLEY, G., *Non-Linear and Dynamic Programming*, Reading, Mass., Addison-Wesley Publishing Co., Inc., 1964.
[8] KEMENY, JOHN G., AND J. LAURIE SNELL, *Finite Markov Chains*, Princeton, N. J., D. Van Nostrand Co., Inc., 1960.
[9] WOLFE, PHILIP, AND G. B. DANZIG, "Linear Programming in a Markov Chain," *Operations Res.* 10:702–710, 1962.



Reprinted from
AMERICAN JOURNAL OF AGRICULTURAL ECONOMICS
Vol. 50, No. 1, February, 1968