stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 32/74 MARCH

P.J. WEEDA

GENERALIZED MARKOV PROGRAMMING WITH A FINITE STATE SEMI MARKOV PROCESS AS NATURAL PROCESS (Extended abstract)

Prepublication

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS (MOS) subject classification scheme (1970): 90C45

GENERALIZED MARKOV PROGRAMMING WITH A FINITE STATE SEMI MARKOV PROCESS AS NATURAL PROCESS *)

P.J. Weeda

ABSTRACT

The principles of generalized Markov programming were developed by DE LEVE(4) to solve continuous time Markov decision problems under the long run average return criterion. Here we investigate the generalized Markov decision model that arises if the natural proces is given by a finite state semi Markov process and actions are restricted to the points in time just after a state transition.
The iteration method induced by the general iteration scheme of DE LEVE(4) for this specialization distinguishes three operations at each iteration step which are called respectively: the value determination -, the policy improvement - and the cutting operation. The first two are related to similar operations in the iteration methods of HOWARD(2) and JEWELL(3) and are directly amenable for computation. This, however, is not true for the third one. For this cutting operation new algorithms are developed which are based upon the relationship between the cutting operation and optimal stopping. Some computational results with computer implementations of these algorithms are presented.

INTRODUCTION

In generalized Markov programming (4) the state of the system is described by a point in a finite dimensional Cartesian space at each point of time. For each initial state the evolution of the system is described by a stochastic process which is called the *natural process*. This natural process is assumed to be a strong Markov process. The decisionmaker may interrupt the natural process in each state by an *intervention* which implies an instantaneous (possibly random) change of the state of the system. In each state the decisionmaker has a set of feasible interventions at his disposal. The only alternative to interventions is to leave the natural process untouched. This alternative is called the *nulldecision* in that state. Except for a (nonempty) subset of states the nulldecision is feasible in each state. After an intervention the evolution of the system is again described by the natural process until the next intervention is effectuated. It is assumed that a finite number of interventions is taken in each finite period of time. A general iteration scheme is presented in (4) which approaches a strategy, which is optimal within the class of stationary deterministic strategies, arbitrarily close. The optimality criterion is to maximize the expected average return per time unit in the long run. Some applications of the method are presented in (5). In this paper we consider the special model that arises if the natural process is given by a *finite state semi Markov process*. The decisionmaker is only allowed to intervene at the points of time a state transition in the natural process has just occurred. The iteration method induced by the general iteration scheme for this special model is formulated. Like the general scheme this iteration method distinguishes three operations per iteration step: the value determination -, the policy improvement - and the cutting operation. The main interest in this paper is focussed on the cutting operation of generalized Markov programming. New is the relation between the cutting operation and *optimal stopping* which is stated here and proved in (6) for this special model. A second new idea is to replace the original cutting operation by a more simple operation which is called *suboptimal cutting*. A proof that the iteration method for this special model with suboptimal

*) This paper is not for review; it is meant for publication in the Proceedings of the IFAC symposium on "Stochastic Control", Budapest, September 25-27, 1974.

cutting converges within a finite number of steps to an optimal strategy will be given in a coming report. In this paper the suboptimal cutting algorithm, which is the most simple from a computational point of view, will be presented. Finally the computational performance of the algorithms will be compared.

THE MODEL

Natural process

The natural process of this generalized Markov decision model is supposed to be given by a finite state semi Markov process. Such a stochastic process makes random transitions among a finite number of states J. Let J denote also the set of states. If a transition to some state $i \in J$ has just occurred at time t the system remains in state i until the next transition to a random state $\underline{j}$ [*)] occurs at a random time $t+\underline{\tau}_i$ where $\underline{\tau}_i$ is the sojourn time in state i. Sufficient information for our purposes about the behaviour of the process is provided by the triple (Q,u,h) where Q denotes the J×J-matrix of transition probabilities q_{ij}, i and $j \in J$ satisfying $0 \le q_{ij} \le 1$ and $\sum_{j\in J} q_{ij} = 1$, u denotes the J-dimensional vector of the expected sojourn times and h denotes the J-dimensional vector of the expected returns. Each element h_i represents the expected return of the process during the sojourn time in state i including the transition to the next state.

Interventions and nulldecisions

In each state $i \in J$ the decisionmaker has a finite set of actions X(i) at his disposal consisting of interventions and at most one nulldecision, denoted by $x_0(i)$. The nulldecision leaves the state of the system unchanged, which implies here that the natural process remains untouched during the sojourn time in that state, including the next state transition. The nulldecision satisfies

$$x_0(i) \notin X(i) \qquad\qquad \text{for } i \in A_0$$

where A_0 is a nonempty subset of states. Further A_0 has to satisfy the requirement that the inverse exists of the matrix $(I-Q)_{\bar{A}_0}$ with entries $\delta_{ij}-q_{ij}$ for $i,j \in \bar{A}_0$. To each intervention $x \in X(i)$ is associated a probability distribution $p_{im}(x)$ of the state $\underline{m}$ into which the intervention leads and an expected cost $g_i(x)$, a finite real number. After the intervention the evolution of the system is determined by the natural process at least until just after the first future state transition.
The nulldecision $x_0(i)$ can be viewed as an intervention satisfying

$$p_{im}(x_0(i)) = \begin{cases} 1 & \text{if } i=m \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_i(x_0(i)) = 0.$$

Strategies

A stationary deterministic strategy z applies the same action $z(i) \in X(i)$ each time a transition to state i has just occurred. By a strategy of this type the state space is dichotomized into a set A_z defined by

$$A_z := \{i \in J : z(i) \neq x_0(i)\}$$

and its complement $\bar{A}_z$. The definitions of A_0 and A_z imply

$$A_z \supseteq A_0.$$

THE ITERATION METHOD

Preliminary computations

Compute:

a. The J-dimensional vector k_0 defined by

$$(k_0)_{\bar{A}_0} := (I-Q)^{-1}_{\bar{A}_0}\,(h)_{\bar{A}_0}$$
$$(k_0)_{A_0} := 0.$$

b. The J-dimensional vector t_0 defined by

$$(t_0)_{\bar{A}_0} := (I-Q)^{-1}_{\bar{A}_0}\,(u)_{\bar{A}_0}$$
$$(t_0)_{A_0} := 0.$$

[*)] Random variables are underlined in this paper.

c. The numbers $k(i,x)$ defined for each $x \in X(i)$ and $i \in J$ by

$$k(i,x) := -g_i(x) + \sum_{m \in J} p_{im}(x)\; k_0(m) - k_0(i).$$

d. The number $t(i,x)$ defined for each $x \in X(i)$ and $i \in J$ by

$$t(i,x) := \sum_{m \in J} p_{im}(x)\; t_0(m) - t_0(i).$$

The interpretation of the vectors k_0 and t_0 is that each element $k_0(i)$ $(t_0(i))$ represents the expected return (expected time elapsed) in the natural process with initial state $i \in \bar{A}_0$ until the first state in A_0 is assumed. The elements $k_0(i)$ and $t_0(i)$ for $i \in A_0$ vanish. The numbers $k(i,x)$ $(t(i,x))$ represent the difference in expected return (expected duration) between two stochastic walks. The first walk applies action $x \in X(i)$ in initial state i and is subsequently described by the natural process until the first state in A_0 is taken on. The second walk is completely described by the natural process from initial state i until the first state in A_0 is taken on. The definitions of $k(i,x)$ and $t(i,x)$ imply $k(i,x_0(i)) = t(i,x_0(i)) = 0$.

After these preliminary computations the iteration cycle is entered with an arbitrarily chosen initial strategy. During each iteration step three operations are performed.

<u>Value determination operation</u>

Compute:

a. The $|A_z|$-dimensional vector $k(z)$ with elements $k(i,z(i))$, $i \in A_z$.

b. The $|A_z|$-dimensional vector $t(z)$ with elements $t(i,z(i))$, $i \in A_z$.

c. The $|\bar{A}_z| \times |A_z|$-matrix $S(A_z)$ defined by

$$S(A_z) := (I-Q)^{-1}_{\bar{A}_z}\;(Q)_{\bar{A}_z A_z}$$

where $(Q)_{\bar{A}_z A_z}$ is the $|\bar{A}_z| \times |A_z|$-matrix with entries q_{ij}, $i \in \bar{A}_z$, $j \in A_z$. The existence of the matrix $(I-Q)^{-1}_{\bar{A}_z}$ is implied by the existence of $(I-Q)^{-1}_{\bar{A}_0}$ and $A_z \supseteq A_0$.

d. The $|A_z| \times |A_z|$-matrix $R(z)$ defined by

$$R(z) := P(z)\; S(A_z)$$

where $P(z)$ denotes the $|A_z| \times |\bar{A}_z|$-matrix with entries $p_{im}(z(i))$, $i \in A_z$, $m \in \bar{A}_z$. $R(z)$ is the matrix of transition probabilities of the imbedded process defined by the states $i \in A_z$.

e. The subvectors $(y(z))_{A_z}$ and $(v(z))_{A_z}$ from the following set of equations

$$(y(z))_{A_z} = R(z)\;(y(z))_{A_z}$$

$$(v(z))_{A_z} = k(z) - (y(z))_{A_z} \square\, t(z) + R(z)(v(z))_{A_z}$$

where the notation $a \square b$ stands for the vector with elements $a_i b_i$. A unique solution to this set is obtained by choosing in each ergodic set $K(l)$, $l=1,\dots,L(z)$ of the imbedded process an arbitrary state $i(l) \in K(l)$ for which we put $v_{i(l)}(z) = 0$, $l=1,\dots,L(z)$.

f. The subvectors $(y(z))_{\bar{A}_z}$ and $(v(z))_{\bar{A}_z}$ from

$$(y(z))_{\bar{A}_z} = S(A_z)\;(y(z))_{A_z}$$

$$(v(z))_{\bar{A}_z} = S(A_z)\;(v(z))_{A_z}$$

<u>The policy improvement operation</u>

Compute:

a. The J-dimensional vector y' with elements y'_i defined by

$$y'_i := \max_{x \in X(i)} \Big[\sum_{j \in J} p_{ij}(x)\; y_j(z)\Big].$$

b. The subset $X_1(i)$ of $X(i)$ defined by

$$X_1(i) = \{x \in X(i) : \sum_{j \in J} p_{ij}(x)\; y_j(z) = y'_i\}.$$

c. The J-dimensional vector v' whose elements v'_i are defined by

$$v'_i := \max_{x \in X_1(i)} \Big[k(i,x) - y'_i t(i,x) + \sum_{j \in J} p_{ij}(x)\, v_j(z)\Big].$$

d. The subset $X_2(i)$ of $X_1(i)$ defined by

$$X_2(i) := \{x \in X_1(i) : k(i,x) - y'_i t(i,x) + \sum_{j \in J} p_{ij}(x)\; v_j(z) = v'_i\}.$$

e. Strategy z' defined by the following rule: Take $z'(i) = z(i)$ if $z(i) \in X_2(i)$; otherwise take $z'(i)$ equal to an arbitrary action from $X_2(i)$.

We note that at the computation of y' the null-decision for a state $i \in A_z \cap \bar{A}_0$ yields

$$\sum_{j\in J} p_{ij}(x_0(i))\ y_j(z) = y_i(z)$$

while the intervention z(i) yields

$$\sum_{j\in J} p_{ij}(z(i))\ y_j(z) = y_i(z).$$

The same holds for the determination of v'. Because the policy improvement operation implies $z'(i) =$ $= z(i)$ if $y'_i = y_i(z)$ and $v'_i = v_i(z)$ we conclude that in any case $z'(i) \neq x_0(i)$ for $i \in A_z$ implying

$$A_{z'} \supseteq A_z.$$

Cutting operation

Let A be an arbitrary set of states satisfying $A_0 \subseteq A \subseteq A_{z'}$. Define the J-dimensional vectors $y''(A)$ and $v''(A)$ by

$$\left\{\begin{array}{l} (y''(A))_{\bar{A}} := S(A)\ (y')_A \\ (y''(A))_A := (y')_A \end{array}\right\} \quad \text{resp.}$$

$$\left\{\begin{array}{l} (v''(A))_{\bar{A}} := S(A)\ (v')_A \\ (v''(A))_A := (v')_A \end{array}\right\} .$$

Let M be the collection of sets A satisfying either $y''_i(A) > y'_i$ or $y''_i(A) = y'_i$ and $v''_i(A) \geq v'_i$ for each $i \in A_{z'}$.

Compute:

a. The set A^* defined by

$$A^* := \bigcap_{A\in M} A.$$

b. The strategy z'' defined by

$$z''(i) := \begin{cases} z'(i) & \text{for } i \in A^* \\ x_0(i) & \text{for } i \in \overline{A^*}. \end{cases}$$

If z'' = z then the iteration cycle has terminated. Otherwise the value determination operation is re-entered with z := z''.

While the value determination and policy improvement operation are directly amenable for computation, this is not true for the cutting operation. This gap is removed in the next section.

THE CUTTING OPERATION AND OPTIMAL STOPPING

In this section we state the relationship between the cutting operation of the preceding section and optimal stopping of a Markov chain. Primarily optimal stopping of a Markov chain is briefly reviewed. For a more extensive treatment see [1].

An optimal stopping problem in a finite Markov chain is a Markov decision problem with at most two feasible actions x_0 and x_1 in each state $i \in J$, where J denotes the set of states of the chain. If action x_0 is applied in state i then the original chain is continued at least until the next transition occurs. If action x_1 is applied in state i then the chain is stopped and a return w_i is obtained. An optimal stopping problem in a finite Markov chain is completely defined by the quadruple (A_s, A_c, Q, w) where A_s is the nonempty subset of states in which only action x_1 is feasible; A_c is the (possibly empty) set of states, $\bar{A}_c \supseteq A_s$, containing all the states in which only action x_0 is feasible; Q is the matrix of transition probabilities of the original chain and w is the $|\bar{A}_c|$-dimensional vector with elements $-\infty < w_i < \infty$. We require the existence of the matrix $(I-Q)^{-1}_{\bar{A}_s}$.

A strategy which maximizes the expected return for each initial state is called optimal. It is well-known [1] that there exists an optimal strategy within the class Z of stationary deterministic strategies. Each strategy dichotomizes the set of states J into a stopping set B defined by

$$B := \{i \in J : z(i) = x_1\}$$

and its complement $\bar{B}$. To each stopping set B is associated an expected return vector f(B). Obviously there is a 1-1 correspondence between strategies of the class Z and the class of feasible stopping sets B satisfying $A_s \subseteq B \subseteq \bar{A}_c$. An optimal stopping set (notation B_m) can be computed by a simplified form of the policy iteration method of HOWARD [2].

Next we state the cutting operation in terms of optimal stopping.

Compute:

a. An optimal stopping set $B_m(y')$ to stopping problem $(A_0,\bar{A}_{z'},Q,y')$ by the method of HOWARD.

b. The smallest and the largest optimal stopping set (notation respectively $B_m^s(y')$ and $B_m^l(y')$) defined by

$$B_m^s(y') :=$$

$$B_m(y') \setminus \{i \in B_m(y') \cap \bar{A}_0 : \sum_{j\in J} q_{ij}y''_j(B_m(y')) = y'_i\}.$$

$$B_m^l(y') :=$$

$$B_m(y') \cup \{i \in \overline{B_m(y')} \cap A_{z'} : \sum_{j\in J} q_{ij}y''_j(B_m(y')) = y'_i\}.$$

c. An optimal stopping set $B_m(v')$ to stopping problem $(B_m^s(y'),\overline{B_m^l(y')},Q,v')$ by the method of HOWARD.

d. The smallest optimal stopping set (notation $B_m^s(v')$) defined by

$$B_m^s(v') :=$$

$$B_m(v') \setminus \{i \in B_m(v') \cap \overline{B_m^s(y')} : \sum_{j\in J} q_{ij}v''_j(B_m(v'))=v'_i\}.$$

e. The strategy z" defined by

$$z''(i) := \begin{cases} z'(i) & \text{for } i \in B_m^s(v') \\ x_0(i) & \text{for } i \in \overline{B_m^s(v')}. \end{cases}$$

This cutting algorithm is based upon the following theorem.

Theorem 1: $B_m^s(v') \equiv A^*$.

A proof is given in [6]. We note that the iteration method also converges if $B_m^s(v')$ is replaced by $B_m(v')$. A stopping set which is optimal to stopping problem $(B_m^s(y'),\overline{B_m^l(y')},Q,y')$ will be called an *optimal cutting set.*

SUBOPTIMAL CUTTING

In this section we introduce more simple algorithms for the cutting operation which are based upon the computation of a suboptimal cutting set rather than an optimal cutting set.

Definition: A *preferable stopping set* B to stopping problem (A_s,A_c,Q,w) is a feasible stopping set satisfying either $(f(B))_{\bar{A}_c} > (w)_{\bar{A}_c}$ or $B \equiv \bar{A}_c$ iff $\bar{A}_c$ is optimal to (A_s,A_c,Q,w).

Let $B(y')$ be a preferable stopping set to $(A_0,\bar{A}_{z'},Q,y')$ and let $B^s(y')$ and $B^l(y')$ be respectively the smallest and the largest set satisfying $f(B^l(y')) = f(B^s(y')) = f(B(y'))$. Let $B(v')$ be a preferable set to stopping problem $(B^s(y'),\overline{B^l(y')},Q,v')$.

Definition: A *suboptimal cutting set* C is defined by
1) if $B(y') \not\equiv A_{z'}$, then $C := B(y')$ or $C := B(v')$
2) if $B(y') \equiv A_{z'}$, then $C := B(v')$.

Note that the definition of suboptimal cutting set includes optimal cutting sets. The following theorem justifies the use of suboptimal cutting sets in computing optimal strategies.

Theorem 2: The iteration model induced by the general scheme for this special model with the cutting operation replaced by suboptimal cutting converges to an optimal strategy within a finite number of steps.

A proof of this theorem can be given and will be presented in a coming report.
There are several ways to compute suboptimal cutting sets. One way is to stop the computation of $B_m^s(y')$ and/or $B_m^s(v')$ by the policy iteration method of HOWARD after the n^{th} step, n=1,2,... . Here we present the algorithm which requires the least computational effort to obtain a suboptimal cutting set C.

Compute:

a. The set $B(y')$ defined by

$$B(y') := A_{z'} \setminus \{i \in A_{z'} \cap \bar{A}_0 : \sum_{j\in J} q_{ij}y'_j > y'_i\}.$$

b. If $B(y') \not\equiv A_{z'}$ and/or $y' > y(z)$ then take $C := B(y')$; otherwise continue with computation c.

c. The set C defined by

$$C := A_{z'} \setminus \{i \in A_{z'} \cap \bar{A}_0 : \sum_{j\in J} q_{ij}v'_j > v'_i\}.$$

COMPUTATIONAL PERFORMANCE

In this section two versions of the iteration method are compared. In the first version the set A^* and in the second version the set C is computed at each iteration step. The computational performance is tested on randomly generated problems as well as three numerical versions of a production problem. The randomly generated problems arise by generating the matrix Q and the vectors u and h of the natural process, defining the set of actions X(i) for each $i \in J$ and the set A_0. We restrict ourselves here to problems satisfying: 1) each intervention $x \in X(i)$, $i \in J$ implies a deterministic change of the state of the system, 2) the state space is an ergodic set for each strategy and 3) the set A_0 consists only of the state J.
Each row of the matrix Q is generated by taking J random numbers and dividing them by their sum. The vectors u and h consists of random numbers multiplied by a suitable factor (here 1000 in both cases). Because interventions are deterministic we may denote each intervention x by the state m it leads into. For X(i) we take

$$X(i) := \begin{cases} \{m = 1,\dots,J\} & i \neq J \\ \{m = 1,\dots,J-1\} & i = J. \end{cases}$$

The J×J-matrix with entries g_{im} $i \in J$, $m \in X(i)$ is generated by taking J random points in the unit square and taking g_{im} equal to the distance between point i and point m. After that the matrix may be multiplied by a suitable positive factor. The numbers g_{im} are generated in this way because they have to satisfy the triangular inequality. This condition should be satisfied to prevent the iteration method of generating strategies with sequences of interventions in zero time, see [6]. Two series of problems were generated:

1. 65 problems with 10 states and 10 actions per state.
2. 5 problems with 50 states and 50 actions per state.

The results were:

Series 1:

Method	Average comp. time	Average number of steps
with A^*	5.87 (+.75) sec.	4.64
with C	3.46 (+.75) sec.	3.77

Series 2:

Method	Average comp. time	Average number of steps
with A^*	161.49 (+20.31) sec.	5.4
with C	106.07 (+20.31) sec.	5.0

The numbers between parenthesis are the average computation time for the vectors k_0 and t_0.

A production problem

A product can be produced at m+1 production rates, r=0,...,m, r=0 corresponds with the situation that the production is switched off and r > 0 with a production rate of r units of product per unit of time. The demand is Poisson distributed with a mean of λ units of product per unit of time. The demand is supplied immediately from the available stock s. If the demand exceeds the available stock then the shortage is replenished by an emergency purchase. As soon as the maximum stock level is reached the production is switched off. The production is controlled by changing the production rate.
Stockholding costs are c_1 per unit of time and per unit of product in stock at the end of the unit time period. The emergency purchase expenses are c_2 per unit of product. Production costs are proportional to the production rate r and given by $c_3 r$. Changing the production rate from r_1 to r_2 costs an amount $a(r_1,r_2)$.
Find a strategy that minimizes the average cost per unit time in the long run.
The continuous version of this problem is solved [5]. The problem stated above can also be solved by the method of HOWARD [2]. Three numerical versions were solved by generalized Markov programming. Again the computational implications of computing the set A^* at each step have been compared with computing the set C at each step.

Numerical version 1

$M = 20$, $m = 3$, $c_1 = .2$, $c_2 = 15$, $c_3 = 1$, $\lambda = 1.2$ and

$$a = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 1 & 0 & 2 & 2 \\ 1 & 1 & 0 & 2 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Method	Computation time	Number of steps
with A^*	391 (+17) sec.	5
with C	364 (+17) sec.	5

Numerical version 2

$M = 20$, $m = 3$, $c_1 = .2$, $c_2 = 15$, $\lambda = 1.7$, $c_3 = 1$ and

$$a = \begin{bmatrix} 0 & 3 & 3 & 3 \\ 3 & 0 & 3 & 3 \\ 3 & 3 & 0 & 3 \\ 3 & 3 & 3 & 0 \end{bmatrix}$$

Method	Computation time	Number of steps
with A^*	441 (+17) sec.	6
with C	268 (+17) sec.	4

Numerical version 3

$M = 25$, $m = 3$, $c_1 = .2$, $c_2 = 15$, $\lambda = 1.9$, $c_3 = 1$ and

$$a = \begin{bmatrix} 0 & 5 & 5 & 5 \\ 5 & 0 & 5 & 5 \\ 5 & 5 & 0 & 5 \\ 5 & 5 & 5 & 0 \end{bmatrix}$$

Method	Computation time	Number of steps
with A^*	822 (+22) sec.	7
with C	453 (+22) sec.	4

CONCLUSIONS

The relationship between the cutting operation and optimal stopping is interesting because it relates a fundamental aspect of generalized Markov programming to a wellknown problem in probability theory. Moreover, known algorithms for solving optimal stopping problems can be used. In this way the cutting operation becomes an efficient standard procedure which can be applied to any problem satisfying the special model in this paper.

The concept of suboptimal cutting is computationally more simple than the computation of the set A^* and does not disturb the pleasant property in this model of convergence within a finite number of steps to an optimal strategy. Moreover the experiments show that suboptimal cutting even reduces the number of iteration steps. In all problems solved the number of iterations using suboptimal cutting has been less than or equal to the number of iterations required if the set A^* is computed at each step. Further the results of this paper may be useful in obtaining computational solutions to the more general type of problems covered by generalized Markov programming. Because any problem satisfying the model considered in this paper can also be solved by the iteration method of JEWELL (3), a comparison between the two techniques is interesting and will be carried out in the near future.

NOMENCLATURE

	page
A	4
A^*	4
A_0	2
A_c	4
A_s	4
A_z	2
B	4
B_m	4
$B_m(v')$	5
$B_m(y')$	5
$B_m^s(v')$	5
$B_m^s(y')$	5
$B_m^1(y')$	5
cutting operation	4
C	5
δ_{ij} (Kronecker delta)	2
$f(B)$	4
h	2
$g_i(x)$	2
g_{im}	6

	page
I (unit matrix)	2
intervention	2
J	2
k_0	2
$k(i,x)$	3
$K(l)$	3
m	2
M	4
natural process	2
null decision	2
optimal stopping	4
policy improvement operation	3
preferable stopping set	5
$P(z)$	3
q_{ij}	2
Q	2
$R(z)$	3
$S(A_z)$, $S(A)$	3
stationary deterministic strategy	2
suboptimal cutting	5
t_0	2
$t(i,x)$	3
$t(z)$	3
u	2
value determination operation	3
v'	3
$v''(A)$	4
$v(z)$	3
w	4
x	2
x_0	4
x_1	4
$x_0(i)$	2
$X(i)$	2
y'	3
$y''(A)$	4
$y(z)$	3
z	2
z'	4
z''	4
Z	4

REFERENCES

1 Derman, C., Finite State Markovian Decision Processes, Mathematics in Science and Engineering, Volume 67, Academic Press, New York and London, 1970.

2 Howard, R.A., Dynamic Programming and Markov Processes, M.I.T. Press, Cambridge, Massachusetts, 1960.

3 Jewell, W.S., "Markov-Renewal Programming, I: Formulation, Finite Return Models, II: Infinite Return Models", Operations Research 10 (1963), 6, pp. 938-972.

4 Leve, G. de, Generalized Markovian Decision Processes, Part I: Model and Method, Part II: Probabilistic Background, Mathematical Centre Tracts 3 and 4, Amsterdam, 1964.

5 Leve, G. de, Tijms, H.C. and Weeda, P.J., Generalized Markovian Decision Processes, Applications, Mathematical Centre Tracts 5, Amsterdam, 1970.

6 Weeda, P.J., "Generalized Markov Programming with a finite state semi Markov process as natural process", Report (in preparation), Mathematisch Centrum, Amsterdam, 1974.