stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 24/73 JUNE

P.J. WEEDA

GENERALIZED MARKOV-PROGRAMMING APPLIED TO SEMI-MARKOV DECISION PROBLEMS AND TWO ALGORITHMS FOR ITS CUTTING OPERATION

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

The application of generalized Markov-programming to semi-Markov decision problems and two algorithms for its cutting operation

by

P.J. Weeda

SUMMARY. This paper deals with the application of generalized Markov-programming, developed by *de Leve* for continuous time Markov decision problems, to semi-Markov decision problems. The criterium is the long run average cost. It is shown that any semi-Markov decision problem can be converted to a problem satisfying the generalized Markov-programming model by an extension of the state space. Applied on problems requiring a complete extension of the state space the iteration method of generalized Markov-programming yields exactly Jewell's method. However in many problems this extension of the state space is either not at all or only partly required. Especially this class of problems is intersting because then the two methods are different. In this and a coming report it is investigated whether generalized Markov-programming is a useful alternative to Jewell's method. An important difference between the two methods is that generalized Markov-programming requires a second policy improvement operation, called the cutting operation. Two new algorithms for this cutting operation are developed. They are based on the relation between the original cutting operation of *de Leve* and a special type of optimal stopping problem. A proof of this relation for the considered model is given. Computational results will be given in a coming report.

1. Introduction

This report deals with the application of generalized Markov-programming developed by de Leve [6] to semi-Markov decision problems. The semi-Markov (or Markov-renewal) decision model was introduced by Jewell [5], who also presented an algorithm to compute an optimal strategy. We restrict ourselves here to undiscounted models. After the introduction of some notation and a basic lemma stating the properties of the unique fixed point of a type of operator frequently used in Markov-programming, the semi-Markov decision model (SMD) is introduced in section 3.1, while in section 3.2 SMD iteration method developed by Jewell is presented. Section 4.1 presents the relevant generalized Markov-programming model (GMP) and section 4.2 the corresponding iteration method. Although generalized Markov-programming was originally developed to solve continuous time decision problems, it is demonstrated in section 5 that it can be applied to any semi-Markov decision problem.

It is shown in section 5.1 that any SMD problem can be converted into an equivalent GMP problem by an extension of the state space. The iteration method of GMP applied to the GMP version of the problem yields then exactly Jewell's method applied to the original problem. In section 5.2 it is shown that any GMP problem (satisfying the model of section 4.1) can be converted into an equivalent SMD problem. However the SMD iteration method of Jewell applied to the converted problem differs from the GMP iteration method applied to the original GMP problem.

The question now arises whether the GMP method is a useful alternative to the SMD iteration method. From section 5 it can be deduced that if the complete extension of the state space, described in section 5.1, is not required to convert a SMD problem (which then should satisfy the GMP model partly or completely) to a GMP problem, then the GMP iteration method differs from the SMD iteration method. Hence the question of usefulness is most relevant in this case, because when a complete extension of the state space is required both methods are identical.

A first requirement to be useful will be that the method yields an algorithm which is general enough to be applied within the class of problems just defined. Until now this has not been the case. The application of GMP in some examples (see [7], [9] or [3]) show that for its second policy improvement operation (or better: cutting operation) no general but only special algorithms exist, which exploit the special properties of the specific problem to be solved. In section 7 two general algorithms for the cutting operation are presented. One is based upon the relation between the original cutting operation of de Leve and a special type of optimal stopping problem considered in section 6. In section 7 it is proved that the original cutting operation is equivalent to two optimal stopping problems to be solved successively. The second cutting method differs from the original cutting operation in that it computes no smallest optimal set but a suboptimal set. The suboptimal cutting method reduces the required computations to such an extent that it is about as efficient as the special devices developed in [3] and [9] which are not applicable to other problems. In a coming publication it will be proved that both GMP algorithms converge within a finite number of steps to an optimal strategy.

A second requirement is that the GMP algorithm is at least as fast as Jewell's. Here again the attention is focussed on problems satisfying the GMP model without requiring the complete extension of the state space. Comparing the structure of both algorithms the GMP method is probably faster in these problems. This conjecture is supported by comparing the required computations per step as well as by computational experience. Some numerical examples will be presented in a separate report.

2. Some notations and preliminaries

Primarily some notational principles will be introduced concerning subvectors and submatrices. Let a and b be two N-vectors with elements a_i, b_i, $i = 1,\dots,N$ and let P be a N×N matrix with entries p_{ij}, $i,j = 1,\dots,N$. Let J denote the set of (N) states and let A and B be two arbitrary non-empty subsets of J. The following notation will be used.

$(a)_A$: subvector of a with elements a_i, $i \in A$.

$(P)_A$: square submatrix of P with entries p_{ij}, $i \in A$, $j \in A$.

$(P)_{AB}$: submatrix of P with entries p_{ij}, $i \in A$, $j \in B$ and $A \nparallel B$.

The vector with elements $a_i \cdot b_i$, $i = 1,\dots,N$ will be denoted by $a \,\square\, b$.

Further $a > b$ means $a \geq b$ and $a \neq b$.

Secondly some useful properties of finite Markov-chains will be stated. The N×N matrix of transition probabilities P will be called a probability matrix. Its entries satisfy

$$(2.1) \qquad p_{ij} \geq 0$$

$$(2.2) \qquad \sum_{j=1}^{N} p_{ij} \leq 1.$$

The n-th power of P is P^n. Its entries will be denoted by $p_{ij}^{(n)}$. The numbers $p_{ij}^{(n)}$ represent the n-step transition probabilities, i.e. the probability that the state is j after n transitions if i is the initial state.

A probability matrix P is transient if its entries satisfy (1) and (2) and moreover.

$$(2.3) \qquad \max_i \sum_{j=1}^{N} p_{ij}^{(N)} < 1.$$

The n-th power of P, P^n goes to zero for $n \to \infty$. Then the matrix $(I-P)^{-1}$ exists and the following identity holds

$$(2.4) \qquad \sum_{n=0}^{\infty} P^n = (I-P)^{-1}$$

with $P^0 = I$, the identity matrix.

Let H be the operator in N-dimensional Eucledian space on an arbitrary N-dimensional vector r defined by

$$(2.5) \qquad Hr := s + Pr$$

with s a given N-dimensional vector and P a given N×N transient matrix. The operator H has a unique fixed point u^* given by

$$(2.6) \qquad u^* := \lim_{n\to\infty} H^n r = (I-P)^{-1} s.$$

Frequent use will be made of the following lemma.

Lemma 2.1

Let H be defined by (2.5) and its unique fixed point u^* be given by (2.6). Then

a. $Hr = r \Longleftrightarrow u^* = r$

b. $Hr \geq r \Longrightarrow u^* \geq r$ and $Hr \leq r \Longrightarrow u^* \leq r$

c. $Hr > r \Longrightarrow u^* \geq Hr > r$ and $Hr < r \Longrightarrow u^* \leq Hr < r$.

Proof

a.
Suppose $Hr = r$ then by induction $H^n r = r$ and $\lim_{n\to\infty} H^n r = r$ or $u^* = r$. Reversely suppose $r = u^*$ then $Hu^* = s + Pu^* = s + P(I-P)^{-1}s = (I-P+P)(I-P)^{-1}s = u^*$.

b.
$q \geq t$ implies $Hq \geq Ht$ hence $Hr \geq r$ implies that $\{H^n r\}$ is a non-decreasing sequence. Since $H^n r \to u^*$ we have $u^* \geq r$. By the same argument: $u^* \leq r$ if $r \leq Hr$.

c.
$Hr > r$ implies $H^n r \geq Hr$ for $n = 2,3,\ldots$ and consequently $u^* \geq Hr > r$.

3. The semi-Markov decision model

3.1. The model

This model was introduced by Jewell [5], who also developed an iterative algorithm to compute an optimal strategy for this model. The model is described as follows. A system makes state transitions only at discrete points in time among a finite member of states $\tilde{N}$. The time intervals between these transitions are stochastic. The set of states is denoted by $\tilde{J}$. Just after a transition to state i, say, the decisionmaker has to choose a decision from a finite set of feasible decisions $\tilde{X}(i)$. A decision $x \in \tilde{X}(i)$ in state $i \in \tilde{J}$ specifies

(1) A probability distribution $\tilde{p}_{ij}(x)$ of the state j to which the next transition leads.

(2) The expected length of the stochastic time interval until the next transition: $\tilde{u}_i(x)$.

(3) The expected return $\tilde{h}_i(x)$ during this stochastic time interval.

We restrict ourselves to the undiscounted model. The purpose is to find a strategy which maximizes the expected average return of this process in the long run. Such a strategy is called optimal.

This model includes the previously developed models of Howard [4]. In the first model of Howard the time intervals are deterministic with equal length for all feasible decisions. In the second model of Howard the time intervals are stochastic and have exponential distributions depending on the decisions.

The computation of an optimal strategy can be restricted to the class Z of stationary deterministic strategies because there exists an optimal strategy in this class. A stationary deterministic strategy z applies the same decision $z(i) \in X(i)$ each time the system is in state i. The policy iteration method of Jewell which computes an optimal strategy in Z is summarized in the next section.

3.2. Jewell's policy iteration algorithm

Let z_n be the strategy obtained after the (n-1)-th step. Let $\tilde{h}(z_n)$ and $\tilde{u}(z_n)$ be the vectors with elements $\tilde{h}_i(z_n(i))$ and $\tilde{u}_i(z_n(i))$, $i \in J$, respectively and let $\tilde{P}(z_n)$ be the stochastic matrix with the i-th row having the elements $\tilde{p}_{ij}(z(i))$, $j \in J$. Then two operations are executed.

1. Value determination operation.

Solve the vectors $y(z_n)$ and $v(z_n)$ from the set of equations

$$(3.1) \qquad \begin{cases} y(z_n) = \tilde{P}(z_n)\; y(z_n) \\ \\ v(z_n) = \tilde{h}(z_n) - y(z_n) \square \tilde{u}(z_n) + \tilde{P}(z_n)\; v(z_n). \end{cases}$$

This set has a unique solution if in each ergodic set $K(l)$, $l = 1,\dots,L(z_n)$ of strategy z_n an arbitrary state $i(l)$ is chosen unambiguously. For each $i(l)$, $l = 1,\dots,L(z_n)$ we put $v_{i(l)}(z_n) = 0$.

2. Policy improvement operation

Compute the vectors y' and v' with elements y_i' and v_i' defined by

$$y_i' = \max_{x \in \tilde{X}(i)} \Big[\sum_{j \in J} \tilde{p}_{ij}(x)\, y_j(z_n) \Big]$$

and compute the set $X_1(i)$ defined by

$$X_1(i) := \{x \in X(i) : \sum_{j \in J} \tilde{p}_{ij}(x)\, y_j(z_n) = y_i'\}.$$

Define the vector v' with elements v_i', $i = 1,\dots,N$ by

$$v_i' = \max_{x \in \tilde{X}_1(i)} \Big[\tilde{h}_i(x) - y_i'\, \tilde{u}_i(x) + \sum_{j \in J} \tilde{p}_{ij}(x)\, v_j(z_n) \Big].$$

Let the set $X_2(i)$ contain the decisions $x \in X_1(i)$ which yield v_i' for each $i \in J$, then strategy z_{n+1} is defined as follows. If $z_n(i) \in X_2(i)$ then take $z_{n+1}(i) = z_n(i)$, otherwise take $z_{n+1}(i)$ equal to an arbitrary $x \in X_2(i)$.

If $z_{n+1} = z_n$ then z_n is optimal, otherwise re-enter the value determination operation with strategy z_{n+1}. Within a finite number of steps, the algorithm converges to an optimal strategy, as was proved by Denardo [2].

4. Generalized Markov-programming applied to semi-Markov decision problems

4.1. The model and basic definitions

Generalized Markov-programming was developed by de Leve [6] for continuous time decision problems. The special generalized Markov-programming model, which can be applied to any semi-Markov decision problem, (as will be shown in section 5.1), is constructed as follows. Also in this model the system makes state transitions among a finite number of states only at discrete points in time. Let J denote the set of these states. A semi-Markov process is defined which is called the natural process. This natural process specifies

(1) A probability distribution q_{ij} of the state $\underline{j}$ to which the next transition in the natural process leads.

(2) The expected length of the stochastic time interval between successive transitions u_i.

(3) The expected return h_i during this time interval.

Just after a transition in the natural process to state i, the decision-maker has to choose x from a finite set of feasible decisions X(i). We distinguish two types of decisions: nulldecisions and interventions. Each set X(i), $i \in J$, contains at most one nulldecision which is denoted by x_0. All other decisions $x \in X(i)$, $x \neq x_0$ are called interventions. The nulldecision $x_0 \in X(i)$ in state i implies that the natural process is followed at least until the next transition has taken place. An intervention $x \in X(i)$, $x \neq x_0$ in state $i \in J$ implies an instantaneous transformation to a stochastic state $\underline{k}$ with probability distribution $p_{ik}(x)$. To this transformation itself a return $g_i(x)$ is associated. After this transformation the system is subject to the natural process at least until the next transition has taken place.

A further requirement is the existence of a non-empty set A_0 defined by

$$A_0 := \{i \in J : x_0 \notin X(i)\}.$$

Another requirement to be fulfilled by the definition of A_0 is that $\bar{A}_0$ contains only transient states in the natural process.

A stationary deterministic strategy $z \in Z$ dichotomizes the set of states J into a set of states A_z in which interventions are applied by strategy z and its complement $\bar{A}_z$ in which the nulldecision is applied in each state. A_z is defined by

$$A_z := \{i \in J : z(i) \neq x_0, z(i) \in X(i)\} \qquad \text{for } z \in Z.$$

The definitions of A_0 and A_z imply

$$A_z \supseteq A_0 \qquad \text{for } z \in Z.$$

Furthermore one requirement has still to be fulfilled. The state $\underline{k}$ resulting from the intervention $z(i) \in X(i)$ for $i \in A_z$ should satisfy

$$P\{\underline{k} \in \bar{A}_z\} = 1.$$

The process resulting from the interventions of a strategy and the natural process will be called the <u>decision process</u>

Let A_{01} and A_{02} be two subsets of A_0 with the property that their complements $\bar{A}_{01}$ and $\bar{A}_{02}$ contain only transient states in the natural process. Then the matrices $(Q)_{\bar{A}_{01}}$ and $(Q)_{\bar{A}_{02}}$ are transient matrices. Define $(k_0)_{\bar{A}_{01}}$ as being the unique fixed point of the operator H_1 on a vector r in $|\bar{A}_{01}|$-dimensional Euclidian space.

(4.1) $$H_1 r := (h)_{\bar{A}_{01}} + (Q)_{\bar{A}_{01}} r$$

and define $(t_0)_{\bar{A}_{02}}$ analogously in $|\bar{A}_{02}|$-dimensional space by the unique fixed point of the operator

(4.2) $$H_2 r := (u)_{\bar{A}_{02}} + (Q)_{\bar{A}_{02}} r.$$

The subvectors $(k_0)_{A_{01}}$ and $(t_0)_{A_{02}}$ are defined to be $|A_{01}|$- and $|A_{02}|$-dimensional nullvectors.

For each $x \in X(i)$, $i \in J$ the functions $k(i,x)$ and $t(i,x)$ are defined by

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

and

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

Let $k(z)$ and $t(z)$ denote the N-dimensional vectors with elements $k(i,z(i))$ and $t(i,z(i))$, $i \in J$ respectively. For non-intervention states $i \in \bar{A}_z$ it can be shown that $k(i,z(i)) = t(i,z(i)) = 0$.

Let $S(A)$ denote the non-square matrix with entries $s_{ij}(A)$, $i \in \bar{A}$, $j \in A$ for an arbitrary non-empty set of states $A_0 \subseteq A \subset J$. The $s_{ij}(A)$ represent the probability that $j \in A$ is the first state assumed in the set A if the natural process is followed from initial state $i \in \bar{A}$ on. The matrix $S(A)$ is obtained from the equation

$$S(A) = (Q)_{\bar{A}A} + (Q)_{\bar{A}}\, S(A). \tag{4.5}$$

Because the matrix $(I-Q)^{-1}_{\bar{A}}$ exists, $S(A)$ can be uniquely solved from (4.5) by

$$S(A) = (I-Q)^{-1}_{\bar{A}}\, (Q)_{\bar{A}A}. \tag{4.6}$$

Let $P(z)$ denote the non-square matrix with entries $p_{ik}(z(i))$ for $i \in A_z$ and $k \in \bar{A}_z$. Let $R(z)$ denote the square matrix of size $|A_z| \times |A_z|$ defined by

$$R(z) := P(z)\, S(A_z). \tag{4.7}$$

Its entries $r_{ij}(z)$, $i,j \in A_z$ represent the probability that j is the first future state assumed in A_z if the decisionprocess of strategy z is followed from initial state $i \in A_z$ on.

A strategy z which maximizes the expected average return per time unit in the long run is called optimal. Again there exists a strategy in the class Z of stationary deterministic strategies, which is optimal. *) A policy iteration algorithm based on generalized Markov-programming to perform this computation is presented in the next section.

*) This follows from section 5.2 which shows that any problem satisfying this model can be converted into an equivalent semi-Markov decision problem.

4.2. A policy iteration algorithm based on generalized Markov-programming

Preperatory part

a. Compute the vectors k_0 and t_0 from (4.1) and (4.2)

b. Compute $k(i;x)$ and $t(i;x)$ from (4.3) and (4.4) each $x \in X(i)$, $x \neq x_0$, $i \in J$.

After the preperatory part the iteration cycle is entered. Let z_n, $n = 1,2,\ldots$ be the strategies successively obtained, with initial strategy z_1. Then the n-th iteration step consists of three operations executed with the strategy z_n obtained by the (n-1)-th step.

1. Value determination operation

a. Compute the matrix $S(A_{z_n})$ from (4.6).

b. Compute the matrix $R(z_n)$ from (4.7).

c. Solve unknown subvectors $(y(z_n))_{A_{z_n}}$ and $v(z_n))_{A_{z_n}}$ from the set of equations

$$(4.8)\quad \begin{cases} (y(z_n))_{A_{z_n}} = R(z_n)\,(y(z_n))_{A_{z_n}} \\ (v(z_n))_{A_{z_n}} = (k(z_n) - y(z_n)\,t(z_n))_{A_{z_n}} + R(z_n)\,(v(z))_{A_{z_n}}\ . \end{cases}$$

d. Compute

$$(y(z_n))_{\bar{A}_{z_n}} = S(A_{z_n})\,(y(z_n))_{A_{z_n}}$$

$$(v(z_n))_{\bar{A}_{z_n}} = S(A_{z_n})\,(v(z_n))_{A_{z_n}}\ .$$

A unique solution to (4.8) is obtained by choosing in each ergodic set

$K(l)$, $l = 1,\dots,L(z_n)$ of strategy z_n an arbitrary intervention state $i(l)$ for which we put $v_{i(l)}(z_n) = 0$, $l = 1,\dots,L(z_n)$.

2. First policy improvement operation

Compute the vector $y(z)$ with elements $y_i(z)$ defined by

$$y'_i := \begin{cases} \max\limits_{x \in X(i)\setminus\{x_0\}} \left[\sum\limits_{j \in J} p_{ij}(x)\, y_j(z_n) \right] & \text{for } i \in A_z \\ \max\limits_{x \in X(i)} \left[\sum\limits_{j \in J} p_{ij}(x)\, y_j(z_n) \right] & \text{for } i \in \bar{A}_z. \end{cases}$$

Compute for each $i \in J$ the set $X_1(i)$ defined by

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

Compute the vector v' with elements v'_i defined by

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

Let $X_2(i)$ be the set of decisions $x \in X_1(i)$ which yield v'_i. The strategy z'_n is defined as follows: If $z_n(i) \in X_2(i)$ then take $z'_n(i) = z_n(i)$, otherwise take $z'_n(i)$ equal to an arbitrary decision from $X_2(i)$.

3. Second policy improvement operation (cutting operation)

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

$$y''_i(A) = \begin{cases} \sum\limits_{j \in A} s_{ij}(A)\, y'_j & \text{for } i \in \bar{A} \\ y'_i & \text{for } i \in A \end{cases}$$

and

$$v_i''(A) := \begin{cases} \sum\limits_{j \in A} s_{ij}(A)\, v_j' & \text{for } i \in \bar{A} \\ v_i' & \text{for } i \in A. \end{cases}$$

Let M be the collection of sets A satisfying

$$A_0 \subseteq A \subseteq A_z.$$

and either

$$y_i''(A) > y_i'$$

or

$$\begin{cases} y_i''(A) = y_i' \\ v_i''(A) \geq v_i' \end{cases}$$

for each $j \in J$.

Compute the set A^* defined by

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

Then strategy z_n'' is defined by

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

If $z_n'' = z_n$ then strategy z_n is optimal. If $z_n'' \neq z_n$ then take $z_{n+1} = z_n''$ and re-enter the value determination operation.

5. Conversion of a semi-Markov decision problem (SMD) into a generalized Markov-programming problem (GMD) and reversely

5.1. The conversion SMD → GMP

Each SMD problem can be converted into a GMP problem by an extension of the set of states $\tilde{J}$. However this extension is not required in many problems. The set of states $\tilde{J}$ is extended with the new states (i,x) for each $x \in \tilde{X}(i)$ and $i \in \tilde{J}$. The set of states of the GMP problem J then bec becomes

$$J := \tilde{J} \cup \{(i,x) : x \in X(i),\ j \in J\}.$$

The natural process needs to be defined only in the states $(i,X) \in J \setminus \tilde{J}$. We define

$$q_{(i,x),j} := \tilde{p}_{ij}(x) \qquad \text{for } (i,x) \in J \setminus \tilde{J},\ j \in \tilde{J},$$

$$q_{(i,x),(k,w)} := 0 \qquad \text{for } (i,x),(k,w) \in J \setminus \tilde{J},$$

$$u_{(i,x)} := \tilde{u}_i(x) \qquad \text{for } (i,x) \in J \setminus \tilde{J}$$

and

$$h_{(i,x)} := \tilde{h}_i(x) \qquad \text{for } (i,x) \in J \setminus \tilde{J}.$$

In the states $i \in \tilde{J}$ an original decision $x \in \tilde{X}(i)$ becomes an intervention which implies a deterministic transformation to the state $(i,x) \in J \setminus \tilde{J}$. We define for $i \in J$

$$X(i) := \tilde{X}(i)$$

with

$$p_{i,m}(x) := \begin{cases} 1 & \text{for } m = (i,x) \\ 0 & \text{otherwise} \end{cases}$$

and

$$g_i(x) := 0.$$

In each state $i \in \tilde{J}$ the nulldecision and in each state $i \in J \setminus \tilde{J}$ interventions are impossible. Hence

(5.1) $$A_z \equiv A_0 \equiv \tilde{J} \qquad \text{for } z \in Z.$$

Because $q_{(i,x),(k,w)} = 0$ for (i,x) and $(k,w) \in \bar{A}_0$ we have

(5.2) $$(Q)_{\bar{A}_0} = 0.$$

We take $A_{01} \equiv A_{02} \equiv A_0$. Then by (4.1), (4.2) and (5.2) it is easily established that

(5.3) $$(k_0)_{\bar{A}_{01}} = (k_0)_{\bar{A}_0} = (h)_{\bar{A}_0}$$

and

(5.4) $$(t_0)_{\bar{A}_{02}} = (t_0)_{\bar{A}_0} = (u)_{\bar{A}_0}.$$

From (4.3), (5.3) and the definition of $h_{(i,x)}$

$$k(i;x) := h_{(i,x)} = \tilde{h}_i(x)$$

and from (4.4), (5.4) and the definition of $u_{(i,x)}$

$$t(i;x) := u_{(i,x)} = \tilde{u}_i(x)$$

for $x \in X(i)$, $i \in A_0$. So we have for each $z \in Z$

(5.5) $$(k(z))_{A_0} = \tilde{h}(z); \quad (k(z))_{\bar{A}_0} = 0$$

and

$$(5.6) \qquad (t(z))_{A_0} = \tilde{u}(z); \quad (t(z))_{\bar{A}_0} = 0.$$

Because of (4.6) and (5.2) we have

$$(5.7) \qquad S(A_0) = (Q)_{\bar{A}_0 A_0}$$

and because of (4.7) and (5.7) we have

$$(5.8) \qquad R(z) = \tilde{P}(z).$$

Then (5.5), (5.6), (5.7) and (5.8) imply that the set of equations

$$(y(z))_{A_0} = R(z)\ (y(z))_{A_0}$$

$$(v(z))_{A_0} = (k(z) - y(z) \square t(z))_{A_0} + R(z)\ (v(z))_{A_0}$$

is equivalent to the set of equations of the SMB iteration method given in section 3.2. The $y_{(i,x)}(z)$ and $v_{(i,x)}(z)$, $(i,x) \in \bar{A}_0$ are also computed in the GMP value determination operation by means of

$$y_{(i,x)}(z) = \sum_{j \in A_0} q_{(i,x),j}\ y_j(z) = \sum_{j \in A_0} \tilde{p}_{ij}(x)\ y_j(z)$$

and

$$v_{(i,x)}(z) = \sum_{j \in A_0} q_{(i,x),j}\ v_j(z) = \sum_{j \in A_0} \tilde{p}_{ij}(x)\ v_j(z).$$

This computation is implicitly executed in the SMB policy improvement operation. The second policy improvement operation of the GMP method is superfluous because each strategy has the same intervention set A_0 according to (5.1). Hence we may conclude two things

1. The conversion of a GMP problem to a SMD problem can always be done.
2. The GMP iteration method applied to the converted problem and the SMD iteration method applied to the original problem yield equivalent algorithms.

5.2. The conversion GMP → SMD

In the conversion SMD → GMP the extension of the state space was needed to dichotomize an SMD decision into an intervention immediately followed by a nulldecision. Here we have to unite an intervention and the nulldecision in the stochastic state k just after the intervention to a SMB decision. No extension of the set of states is needed so we define

$$\tilde{J} := J.$$

Also the sets of feasible decisions remain unchanged

$$\tilde{X}(i) := X(i) \qquad\qquad \text{for } i \in J.$$

For each $x \in \tilde{X}(i)$ we define for $i,j \in \tilde{J}$ and $x \in \tilde{X}(i)$

$$(5.9) \qquad \tilde{p}_{ij}(x) := \begin{cases} \sum\limits_{k} p_{ik}(x)\; q_{kj} & x \neq x_0 \\ \\ q_{ij} & x = x_0 \end{cases}$$

$$(5.10) \qquad \tilde{u}_i(x) := \begin{cases} \sum\limits_{k} p_{ik}(x)\; u_k & x \neq x_0 \\ \\ u_i & x = x_0 \end{cases}$$

$$(5.11) \qquad \tilde{h}_i(x) := \begin{cases} \sum\limits_{k} p_{ik}(x)\; h_k + g_i(x) & x \neq x_0 \\ \\ h_i & x = x_0 . \end{cases}$$

The two operations of the SMD iteration method expressed in the original notation of the GMP model then become

1. Value determination operation

Let z be the current strategy. Solve the set of equations in y(z) and v(z) given by

$$
\begin{aligned}
(y(z))_{\bar{A}_z} &= (Q)_{\bar{A}_z} \ (y(z))_{\bar{A}_z} + (Q)_{\bar{A}_z A_z} \ (y(z))_{A_z} , \\
(y(z))_{A_z} &= P(z) \ (Q)_{\bar{A}_z} \ (y(z))_{\bar{A}_z} + P(z) \ (Q)_{\bar{A}_z A_z} \ (y(z))_{A_z} , \\
(v(z))_{\bar{A}_z} &= (h-y(z)\square u)_{\bar{A}_z} + (Q)_{\bar{A}_z} \ (v(z))_{\bar{A}_z} + (Q)_{\bar{A}_z A_z} \ (v(z))_{A_z} , \\
(v(z))_{A_z} &= g(z) + P(z) \ (h-y(z)\square u)_{\bar{A}_z} + \\
&\qquad + P(z) \ (Q)_{\bar{A}_z} \ (v(z))_{\bar{A}_z} + P(z) \ (Q)_{\bar{A}_z A_z} \ (v(z))_{A_z} .
\end{aligned}
$$

2. Policy improvement operation

Compute the vector y' with elements y_i', defined by

$$
y_i' := \max[\max_{\substack{x \in X(i) \\ x \neq x_0}} [\sum_{k \in J} p_{ik}(x) \sum_{j \in J} q_{kj} \ y_j(z)], \sum_{j \in J} q_{ij} \ y_j(z)] \qquad \text{for } i \in \bar{A}_0 ,
$$

$$
y_i' = \max_{x \in X(i)} [\sum_{k \in J} p_{ik}(x) \sum_{j \in J} q_{kj} \ y_j(z)] \qquad \text{for } i \in A_0 .
$$

Compute the set $X_1(i)$ defined by

$$
X_1(i) := \{x \in X(i) : \max[\max_{\substack{x \in X(i) \\ x \neq x_0}} [\sum_{k \in J} p_{ik}(x) \sum_{j \in J} q_{kj} \ y_j(z)], \sum_{j \in J} q_{ij} \ y_j(z)] = y_i' \} \qquad \text{for } i \in \bar{A}_0
$$

and

$$X_1(i) := \{x \in X(i) : \max_{x \in X(i)} [\sum_{k \in J} p_{ik}(x) \sum_{j \in J} q_{kj}\ y_j(z)] = y_i'\}$$

for $i \in A_0$.

Compute the vector v' with elements v_i' defined by

$$(5.12)\quad v_i' := \max[\max_{\substack{x \in X_1(i) \\ x \neq x_0}} [g_i(x) + \sum_{k \in J} p_{ik}(x) \{h_k - y_i'\ u_k + \sum_{j \in J} q_{kj}\ v_j(z)]\},$$

$$, h_i - y_i'\ u_i + \sum_{j \in J} q_{ij}\ v_j(z)] \quad \text{if } x_0 \in X(i)$$

and

$$(5.13)\quad v_i' := \max_{x \in X_1(i)} [g_i(x) + \sum_{k \in J} p_{ik}(x) \{h_k - y_k'\ u_k + \sum_{j\ J} q_{kj}\ v_j(z)\}]$$

otherwise. Let $X_2(i)$ be the set of decisions satisfying (5.12) or (5.13) for state i. Then the next strategy is defined in the same way as in section 3.2.

By these results two conclusions can be drawn.

1. The conversion of a GMP problem to a SMD problem can always be done.
2. The SMD iteration method applied to the converted problem and the GMP iteration method applied to the original GMP problem yield <u>different</u> algorithms.

Among the most remarkable differences are the device of the functions k(i,x) and t(i,x), the value determination operation and the absence of the second policy improvement operation (cutting operation) of GMP in the resulting SMD iteration method.

6. A special type of optimal stopping problem

6.1. The model

The special type of optimal stopping problem to be considered in this context is defined as follows. Let Q be the matrix of transition probabilities of a finite Markov-chain. Each state of the chain $i \in J$ is either transient or absorbing. At least one state is absorbing. In each state i a decision has to be chosen from a set $X(i)$ of feasible decisions. At most two decisions x_0 and x_1 are feasible. If the decision $x_0 \in X(i)$ is applied then the probability distribution of the state assumed after the next transition is given by the i-th row of the matrix Q with entries q_{ij} satisfying $\sum_{j \in J} q_{ij} = 1$ and $0 \leq q_{ij} \leq 1$ for $j \in J$. In this case no return is obtained. If the decision $x_1 \in X(i)$ is applied then the system remains in state i and an average return w_i per time unit is obtained. The following sets of states are defined.

(6.1) $$A_d := \{i \in J : X(i) \equiv \{x_0, x_1\}\},$$

(6.2) $$A_c := \{i \in J : X(i) \equiv \{x_0\}\},$$

(6.3) $$A_s := \{i \in J : X(i) \equiv \{x_1\}\}.$$

The set A_s contains the absorbing states of the original chain and is assumed to be non-empty. The goal is to find a strategy which maximizes the expected average return for each state $i \in J$. Such a strategy is called optimal.

This type of optimal stopping problem is a special type of a Markov decision problem with a finite state space and finite decision sets. Because a stationary deterministic strategy is optimal over the whole class of randomized history remembering strategies the computation of an optimal strategy can be restricted to the class Z of stationary deterministic strategies. Any strategies $z \in Z$ dichotomizes the state space J into two complementary sets: a stopping set to be denoted by $B(z)$ and its complement the continuation set $\overline{B(z)}$. The definitions (6.1) ... (6.3) imply that $B(z) \supseteq A_s$ and $\overline{B(z)} \supseteq A_c$ for any $z \in Z$.

An optimal strategy z_0 can be computed by the policy iteration method of Howard. Because for each strategy $z \in Z$ the states $i \in B(z)$ are absorbing and the states $i \in \overline{B(z)}$ are transient only the $y_i(z)$, $i \in J$, have to be used. For notational reasons we shall write $f_i(z)$ instead of $y_i(z)$. The unique maximum expected average return vector will be denoted by f^* and satisfies the following functional equation

$$(6.4)\qquad f_i^* = \begin{cases} \sum_{j\in J} q_{ij}\, f_j^* & i \in A_c \\ \max[w_i, \sum_{j\in J} q_{ij}\, f_j^*] & i \in A_d \\ w_i & i \in A_s. \end{cases}$$

Let the sets B_l^* and B_s^* be defined by

$$(6.5)\qquad B_l^* := \{i \in A_d : \sum_{j\in J} q_{ij}\, f_j^* \le w_i\} \cup A_s$$

and

$$(6.6)\qquad B_s^* := \{i \in A_d : \sum_{j\in J} q_{ij}\, f_j^* < w_i\} \cup A_s.$$

B_l^* and B_s^* are easily identified as the largest and the smallest optimal stopping set respectively. Note that for $i \in \overline{B_l^*} \cap \bar{A}_c$ we have

$$\sum_{j\in J} q_{ij}\, f_j^* > w_i.$$

6.2. A policy iteration algorithm

Suppose at the n-th iteration the strategy z_n with stopping set B_n is obtained. Perform the following two operations:

1. Value determination operation

Solve the vector $f(z_n)$ from the set of linear equations

$$f_i(z_n) = w_i \qquad \text{for } i \in B_n$$

$$f_i(z_n) = \sum_{j\in J} q_{ij}\, f_j(z_n) \qquad \text{for } i \in \bar{B}_n.$$

2. Policy improvement operation

Compute f_i' for $i \in J$ defined by

$$f_i' := \begin{cases} \max[w_i, \sum_{j\in J} q_{ij} f_j(z_n)] & \text{for } i \in A_d \\ \\ f_i(z_n) & \text{for } i \in \bar{A}_d. \end{cases}$$

If $f_i' = f_i(z_n)$ then take $z_{n+1}(i) = z_n(i)$ otherwise take $z_{n+1}(i)$ equal to the decision which yields f_i'.

If $z_{n+1}(i) = z_n(i)$ for $i \in J$ then z_n is optimal, otherwise re-enter the value determination operation.

6.3. A proof that the policy iteration algorithm yields an optimal stopping set within a finite number of steps

In this section let z and z' be two successive strategies at any step of the iteration with stopping sets B and B' respectively.

LEMMA 6.1.

At each step of the iteration we have either $f_i' > f_i(z)$ *or* $z'(i) = z(i)$ *for each* $i \in J$.

PROOF

For $i \in A_d$ with $z'(i) \neq z(i)$ we either have

$$f_i' = w_i > \sum_{j\in J} q_{ij} f_j(z) = f_i(z)$$

or

$$f_i' = \sum_{j\in J} q_{ij} f_j(z) > f_i(z) = w_i.$$

In all other cases we have $z'(i) = z(i)$. *

LEMMA 6.2.

Two successive strategies z *and* z' *satisfy either* f(z') > f(z) *or* z' = z.

PROOF

Suppose the set $\bar{B} \cap B'$ is nonempty, then for $i \in \bar{B} \cap B'$

$$(6.6) \qquad f_i(z') = w_i > f_i(z) = \sum_{j\in J} q_{ij}\ f\ (z).$$

Also we have for $i \in B \cap B'$

$$(6.7) \qquad f_i(z') = w_i = f_i(z).$$

From (6.6) and (6.7) follows $(f(z'))_{B'} > (f(z))_{B'}$. Suppose the set $B \cap \overline{B'}$ is nonempty then for $i \in B \cap \overline{B'}$

$$(6.8) \qquad f_i' = \sum_{j\in J} q_{ij}\ f_j(z) > w_i = f_i(z)$$

Also we have

$$(6.9) \qquad f_i' = f_i(z) = \sum_{j\in J} q_{ij}\ f_j(z) \qquad\qquad \text{for } i \in \bar{B} \cap \overline{B'}.$$

Hence from (6.8) and (6.9) and because in any case $f(z'))_{B'} \geq (f(z))_{B'}$

$$(6.10) \qquad (f')_{\overline{B'}} = (Q)_{\overline{B'}}\ (f(z))_{\overline{B'}} + (Q)_{\overline{B'}B'}\ (f(z'))_{B'} > (f(z))_{\overline{B'}}\ .$$

By the value determination operation applied on f(z') we have

$$(6.11) \qquad (f(z'))_{\overline{B'}} = (Q)_{\overline{B'}}\ (f(z'))_{\overline{B'}} + (Q)_{\overline{B'}B'}\ (f(z'))_{B'}.$$

Lemma 2.1 applied on (6.10) and (6.11) with $u^* = (f(z'))_{\overline{B'}}$, $r = (f(z))_{\overline{B'}}$, $s = (Q)_{\overline{B'}B'}\ (f(z'))_{B'}$ and $P = (Q)_{\overline{B'}}$ implies

$$(f(z'))_{\overline{B'}} > (f(z))_{\overline{B'}}\ .$$

If $z' \neq z$ then either $\bar{B} \cap B'$ or $B \cap \overline{B'}$ or both are nonempty, implying $f(z') > f(z)$. *

THEOREM 6.1.

The policy iteration algorithm of section 3.1 converges within a finite number of steps to a strategy z_o *satisfying*

$$z_o' = z_o$$

which is optimal.

PROOF

At each step the expected average return $f(z)$ of the current strategy z with stopping set B satisfies

$$(6.12) \qquad \begin{cases} (f(z))_{\bar{B}} = (Q)_{\bar{B}}\,(f(z))_{\bar{B}} + (Q)_{\bar{B}B}\,(f(z))_B \\ \\ (f(z))_B = (w)_B. \end{cases}$$

The solution to (6.12) is unique because $(Q)_{\bar{B}}$ is a transient matrix. At each step we have $f(z') > f(z)$ for strategy z and its successor z', except at the terminal step. These two facts imply that no previously obtained strategy may turn up again before a strategy z_o satisfying $z_o' = z_o$ is obtained. Because Z contains a finite number $(= 2^{|A_d|})$ of strategies the strategy z_o is obtained within a finite number of steps. Because $z_o' = z_o$ strategy z_o satisfies

$$(6.13) \qquad (Q)_{\bar{B}}\,(f(z_o))_{\bar{B}} + (Q)_{\bar{B}B}\,(f(z_o))_B \leq (f(z_o))_{\bar{B}}$$

$$(6.14) \qquad (w)_B \leq (f(z_o))_B$$

for any stopping set B satisfying $A_s \subseteq B \subseteq \bar{A}_c$. Let z be the strategy with stopping set B then $(f(z))_{\bar{B}}$ satisfies (6.12). Lemma 2.1 applied on (6.12) and (6.13) yields

$$(f(z))_{\bar{B}} \leq (f(z_o))_{\bar{B}}.$$

This together with $(f(z))_B = (w)_B \leq (f(z_o))_B$ implies

$$f(z) \leq f(z_o) \qquad \text{for } z \in Z$$

or equivalently z_o is optimal. *

It is clear from the definitions (6.5) and (6.6) of B_1^* and B_s^* that any optimal strategy z^* with stopping set B^* satisfies

$$(6.15) \qquad B_s^* \subseteq B^* \subseteq B_1^*.$$

The policy improvement operation of section 6.2 can be simplified by the following lemma.

LEMMA 6.3.

Let B_n *denote the stopping sets of the successive strategies* z_n, n=1,2,... *obtained at the successive steps of the policy iteration algorithm of section 6.2. Let* $B_1 \equiv \bar{A}_c$ *be the initial stopping set and let* $m \geq 1$ *be the integer for which* $B_m \equiv B_{m+1}$. *Then*

(a) $B_{n+1} \subset B_n$ *for* $1 \leq n < m$,

(b) $B_m \equiv B_1^*$, *the largest optimal stopping set.*

PROOF

(a)

Suppose that at the n-th step a state $i \in A_d$ satisfies $i \in B_n \cap \bar{B}_{n+1}$. Then

$$w_i = f_i(z_n) < \sum_{j \in J} q_{ij}\, f_j(z_n) \qquad \text{for } i \in B_n \cap \bar{B}_{n+1}.$$

For the successors of z_n the strategies z_k, $k = n+1, n+2, \ldots$ we have for $i \in B_n \cap \bar{B}_{n+1}$

$$(6.16) \qquad \sum_{j \in J} q_{ij}\, f_j(z_k) \geq \sum_{j \in J} q_{ij}\, f_j(z_n) > w_i$$

or equivalently $i \in \bar{B}_k$ for $k = n+1, n+2, \ldots$. Hence a state which is thrown out of the stopping set at any step does not return to the stopping set anymore. Hence if $m > 1$ then

$$B_n \supset B_{n+1} \qquad \text{for } 1 \leq n < m$$

which states the nesting of successive stopping sets.

(b)

According to (a) there exists for each $i \in \bar{B}_m \cap A_c$ an integer n, $1 \leq n < m$ satisfying

$$(6.17) \qquad f_i(z_m) = \sum_{j \in J} q_{ij}\, f_j(z_m) \geq \sum_{j \in J} q_{ij}\, f_j(z_n) > w_i .$$

The optimality of B_m and the definition of B_1^* imply $B_m \subseteq B_1^*$. On the other hand the strict inequality in (6.17) and the definition of B_1^* imply that $\bar{B}_m \cap B_1^*$ is empty or equivalently $B_m \supseteq B_1^*$. Hence $B_m \equiv B_1^*$. *

By lemma 6.3 the policy improvement operation of section 6.2 can be simplified by computing f_i' only for $i \in B_n \cap A_d$. f_i' is then defined by

$$f_i' := \begin{cases} \max[w_i, \sum_{j \in J} q_{ij}\, f_j(z_n)] & \text{for } i \in B_n \cap A_d \\ \\ f_i(z_n) & \text{otherwise,} \end{cases}$$

if one starts the iteration with $\bar{A}_c$ as initial stopping set.

7. Two algorithms for the cutting operation of generalized Markov-programming.

7.1. An algorithm for the original cutting operation

In this section we present an algorithm for the cutting operation of generalized Markov-programming which consists of solving two optimal stopping problems successively. Also in this section it will be proved that this algorithm yields the set A^* of section 4.2 which is the goal of the original cutting operation defined by de Leve [6]. The algorithm consists of the following two optimal stopping problems to be solved by the policy iteration algorithm presented in section 6.2:

1. Optimal stopping problem I (OSP I)

Solve the optimal stopping problem applied to the imbedded Markov chain of the natural process with

a. $A_s := A_0$

b. $A_c := \bar{A}_{z'}$

c. $w_i := y_i'$ for $i \in A_{z'}$.

This yields an optimal stopping set $B^*(y')$. Determine the largest and the smallest optimal stopping set, to be denoted by $B_l^*(y')$ and $B_s^*(y')$ respectively.

2. Optimal stopping problem II (OSP II)

Solve the optimal stopping problem applied to the imbedded Markov chain of the natural process with

a. $A_s := B_s^*(y')$

b. $A_c := \overline{B_l^*(y')}$

c. $w_i := v_i'$ for $i \in A_{z'}$.

This yields an optimal stopping set $B^*(v')$. Determine $B_s^*(v')$ and $B_l^*(v')$ which are respectively the smallest and the largest optimal stopping set.

The following theorem states the equivalence of the original cutting operation with these two optimal stopping problems.

THEOREM 7.1

$$B_s^*(v') \equiv \bigcap_{A \in M} A.$$

To prove this theorem we prove first three lemma's.

LEMMA 7.1

Let A *be a set of states satisfying* $A_0 \subseteq A \subseteq J$. *Let the vector* w *be a vector in* $|A|$*-dimensional space and let* f *be a vector in* $|J|$*-dimensional space defined by the unique solution of the set*

$$f_i = \sum_{j \in J} q_{ij} f_j \qquad i \in \bar{A}$$

$$f_i = w_i \qquad i \in A$$

then f *is also the unique solution of*

$$f_i = \sum_{j \in J} s_{ij}(A)\, w_j \qquad i \in \bar{A}$$

$$f_i = w_i \qquad i \in A$$

and reversely. (See section 4.1 for the definition of the probabilities $s_{ij}(A)$*).*

PROOF

We only need to show that $(f)_{\bar{A}}$ satisfying

$$(7.1) \qquad (f)_{\bar{A}} = (Q)_{\bar{A}}\, (f)_{\bar{A}} + (Q)_{\bar{A}A}\, (w)_A$$

is the unique solution of

$$(7.2) \qquad (f)_{\bar{A}} = S(A)\ (w)_A$$

and reversely. Because $(I-Q)_{\bar{A}}^{-1}$ exists we can solve $(f)_{\bar{A}}$ uniquely from (7.1) and obtain

$$(7.3) \qquad (f)_{\bar{A}} = (I-Q)_{\bar{A}}^{-1}\ (Q)_{\bar{A}A}\ (w)_A.$$

Relation (4.6) together with (7.3) yield (7.2) and (7.2) with (4.6) yield (7.3) implying (7.1). *

LEMMA 7.2

Let M *be the collection of sets* A *defined in the original cutting operation of section 4.2 and let* $A_{z'}$ *be the intervention set of strategy* z' *obtained by the first policy improvement operation then*

$$A_{z'} \in M.$$

PROOF

We have (referring to section 4.2)

$$(7.4) \qquad \left\{ \begin{aligned} y_i''(A_{z'}) &= y_i' \\ v_i''(A_{z'}) &= v_i' \end{aligned} \right\} \qquad \text{for } i \in A_{z'}.$$

Because $y_i' \geq y_i$ and $v_i' \geq v_i$ for $i \in A_{z'}$ and $y_i' = y_i$ and $v_i' = v_i$ for $i \in \overline{A_{z'}}$ we have

$$(7.5) \qquad \left\{ \begin{aligned} y_i' = y_i &= \sum_{j \in J} q_{ij}\ y_j \leq \sum_{j \in J} q_{ij}\ y_j' \\ v_i' = v_i &= \sum_{j \in J} q_{ij}\ v_j \leq \sum_{j \in J} q_{ij}\ v_j' \end{aligned} \right\} \qquad \text{for } i \in \bar{A}_{z'}.$$

On the other hand by the definitions of y"(A) and v"(A) and lemma 7.1 we have

$$(7.6)\qquad \left\{\begin{array}{l} y''_i(A_{z'}) = \sum\limits_{j\in J} q_{ij}\; y''_j(A_{z'}) \\ \\ v''_i(A_{z'}) = \sum\limits_{j\in J} q_{ij}\; v''_j(A_{z'}) \end{array}\right\} \qquad \text{for } i \in \bar{A}_{z'}.$$

Lemma 2.1 applied on (7.4) and (7.6) implies

$$(7.7)\qquad \left\{\begin{array}{l} y''_i(A_{z'}) \geq y'_i \\ \\ v''_i(A_{z'}) \geq v'_i \end{array}\right\} \qquad \text{for } i \in \bar{A}_{z'}$$

(7.4) and (7.7) imply that all conditions for $A_{z'}$ to be a member of the collection M are satisfied.

LEMMA 7.3

$$B^*_s(v') \in M.$$

PROOF

Because $B^*_s(v')$ is optimal for OSP II and OSP I and because of lemma 7.2 we have

$$(7.8)\qquad \left\{\begin{array}{l} y''_i(B^*_s(v')) \geq y''_i(A_{z'}) \geq y'_i \\ \\ v''_i(B^*_s(v')) \geq v''_i(A_{z'}) \geq v'_i \end{array}\right\} \qquad \text{for } i \in \bar{A}_{z'}.$$

The optimality of $B^*_s(v')$ for OSP I and OSP II implies

$$(7.9)\qquad y''_i(B^*_s(v')) > y'_i \qquad\qquad \text{for } i \in \overline{B^*_1(y')} \cap A_{z'}$$

and

$$(7.10)\qquad \left\{\begin{array}{l} y''_i(B^*_s(v')) = y'_i \\ \\ v''_i(B^*_s(v')) \geq v'_i \end{array}\right\} \qquad \text{for } i \in \overline{B^*_s(v')} \cap B^*_1(y').$$

The definitions of $y''_i(B^*_s(v'))$ and $v_i(B^*_s(v'))$ imply

$$(7.11)\qquad \left\{\begin{array}{l} y''_i(B^*_s(v')) = y'_i \\ \\ v''_i(B^*_s(v')) = v'_i \end{array}\right\} \qquad \text{for } i \in B^*_s(v')$$

(7.8) ... (7.11) with $A_0 \subseteq B^*_s(v') \subseteq A_z$, imply $B^*_s(v') \in M$.

THEOREM 7.1.

$$B^*_s(v') \equiv \bigcap_{A \in M} A.$$

PROOF

By lemma 7.3 $B^*_s(v') \in M$. Suppose $\bar{A} \cap B^*_s(v')$ nonempty for some $A \in M$. Then by the optimality of $B^*_s(v')$ for OSP I and OSP II we have

$$(7.12)\qquad y''_i(A) = \sum_{j \in J} q_{ij}\, y''_j(A) \leq \sum_{j \in J} q_{ij}\, y''_j(B^*_s(v')) \leq y'_i$$

and

$$(7.13)\qquad v''_i(A) = \sum_{j \in J} q_{ij}\, v''_j(A) \leq \sum_{j \in J} q_{ij}\, v''_j(B^*_s(v')) < v'_i$$

for $i \in \bar{A} \cap B^*_s(v')$. (7.12) and (7.13) imply $A \notin M$, a contradiction. Hence $\bar{A} \cap B^*_s(v')$ is empty for $A \in M$ and consequently $A \supseteq B^*_s(v')$ for $A \in M$.

7.2. Suboptimal cutting

In the preceding section an algorithm has been developed which yields the smallest stopping set which is optimal for OSP I and OSP II. By theorem 7.1 this set is equivalent to the set A^* which is the goal of the original cutting operation of generalized Markov programming. In this section we propose a procedure called suboptimal cutting which differs from the original cutting operation but is more attractive from a computational point of view. The goal is then to compute a stopping set which is a member of the collection M but not necessarily optimal for OSP I and OSP II.

Definition 7.2.1.

Let B and C be two stopping sets satisfying $A_s \subseteq B, C \subseteq \bar{A}_c$ and let w be the return vector with elements w_i defined for $i \in \bar{A}_c$. Let f(B) and f(C) denote the expected average return vectors for B and C respectively. If $f(B) > f(C)$ with $f_i(B) > f_i(C)$ for at least one $i \in \bar{A}_c$ then we call B better than C with respect to w. If $f(B) = f(C)$ then B is called equivalent to C with respect to w.

The algorithm of section 6.2 yields a stopping set at each non-terminal step which is better than its predecessors by lemma 6.2. If we start the iteration with $B_1 \equiv \bar{A}_c$ and apply one policy improvement step then the resulting set B_2 is already a better stopping set than $B_1 \equiv \bar{A}_c$ with respect to w if $\bar{A}_c$ itself is not optimal. If $\bar{A}_c$ is optimal then $B_1^* \equiv \bar{A}_c$ according to the definition of B_1^* and no better stopping sets can be obtained. This procedure to compute a better (or if no one exists an equivalent) stopping set with respect to w is used in the following

Suboptimal cutting algorithm

1. Suboptimal stopping problem I

   Apply one policy improvement operation to the imbedded Markov-chain of the natural process with initial stopping set $\bar{A}_c$ and

   a. $A_s := A_0$

   b. $A_c := \bar{A}_{z'}$

   c. $w_i := y'_i$ for $i \in A_{z'}$.

   This yields a stopping set B(y') which is better than or equivalent to

$A_{z'}$ with respect to y'. Determine the largest and the smallest stopping set denoted by $B_l(y')$ and $B_s(y')$ respectively which are equivalent to $B(y')$ with respect to y'. Note that the computation of $y''(B(y'))$ is required to obtain $B_l(y')$ and $B_s(v')$.

2. Suboptimal stopping problem II

Apply one policy improvement operation to the imbedded Markov-chain of the natural process with initial stopping set $B_l(y')$ and

a. $A_s := B_s(y')$

b. $A_c := \overline{B_l(y')}$

c. $w_i := v_i$ for $i \in B_l(y')$.

This yields a stopping set $B(v')$ which is better than or equivalent to $B_l(y')$ with respect to v'.

A proof that the GMP iteration method with either cutting algorithm converges to an optimal strategy within a finite number of steps will be presented in a coming publication.

8. References

[1] Derman, C., Finite state Markovian decision processes, Mathematics in science and engineering, volume 67, 1970, Academic Press, New York and London.

[2] Denardo, E.V. and Fox, B.L., Multichain Markov renewal programs, SIAM Journal of Applied Mathematics, Vol. 16, No. 3, May 1968.

[3] Hollenberg, J.P., Een twee-dimensionaal voorraadprobleem, Rapport BN 2/71, Mathematisch Centrum, Amsterdam, 1971.

[4] Howard, R.A., Dynamic programming and Markov processes, Technology Press M.I.T. and Wiley and Sons, New York - London, 1960.

[5] Jewell, W.S., Markov-renewal programming, Opns. Res. 10 (1963), 938-972.

[6] Leve, G. de, Generalized Markovian decision processes, Part I and Part II, Mathematical Centre Tracts, No. 3 and 4, Amsterdam, 1964.

[7] Leve, G. de, Tijms, H.C. and Weeda, P.J., Generalized Markovian decision processes, Applications, Mathematical Centre Tracts No. 5, Amsterdam, 1970.

[8] Leve, G. de en Tijms, H.C., Leergang Besliskunde, deel 7c, Mathematisch Centrum, Amsterdam, 1971.

[9] Tijms, H.C., Een kwaliteitscontrole probleem, Rapport BN 5/71, Mathematisch Centrum, Amsterdam, 1971.