stichting
mathematisch
centrum

MC

AFDELING MATHEMATISCHE BESLISKUNDE BW 37/74 SEPTEMBER

P.J. WEEDA

SOME COMPUTATIONAL EXPERIMENTS WITH A SPECIAL GENERALIZED MARKOV PROGRAMMING MODEL

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.

Some computational experiments with a special generalized Markov programming model

by

P.J. Weeda

ABSTRACT

The principles of generalized Markov programming were developed by DE LEVE [3] to solve continuous time Markov decision problems under the long run average return criterion. In this report the special generalized Markov decision model is investigated that arises if the natural process is given by a finite state semi Markov process and interventions are restricted to the points in time just after a state transition in the natural process. The iteration method for this model induced by the general iteration scheme of DE LEVE is given. Four variants on the iteration method are developed which all have the pleasant property in this special model of convergence within a finite number of steps to an optimal strategy. The results of computational experience with these variants are presented. The problems solved include randomly generated problems as well as three numerical versions of a preformulated problem from the field of production control. The numerical results are compared with those obtained by applying existing policy iteration methods to these problems.

KEYWORDS: *Generalized Markov programming; Computational; Finite state semi Markov decision problems.*

1. INTRODUCTION

The principles of generalized Markov programming were developed by DE LEVE [3] to solve continuous time Markov decision problems under the long run average return criterion. In this report the special generalized Markov decision model is investigated, that arises if the natural process is given by a finite state semi Markov process and interventions are restricted to the points in time just after a state transition in the natural process.

The general iteration scheme of generalized Markov programming DE LEVE, [3] induces an iteration method for this special model which consists of three operations to be executed at each iteration step and which converges within a finite number of steps to an optimal strategy. Two of these three operations are related to the value determination and policy improvement operation in the methods of HOWARD [1] and JEWELL [2]. If only these two operations are applied at each iteration step the iteration method still converges, but not necessarily, to an optimal strategy unless the third operation (cutting operation) is applied. This cutting operation in its original form is not directly amenable for computation. In a previous report WEEDA [6], the relation between the cutting operation and optimal stopping in a Markov chain has been stated and proved for this special model. This result yields a useful algorithm (to be called optimal cutting) for this cutting operation.

Besides optimal cutting (variant 1) three other variants are tried out in this report. The computational results show that the number of iteration steps required for convergence is reduced by weakening the importance of the cutting operation. This may be done either by weakening the cutting operation itself or by reducing its frequency of use or both.

Weakening of the cutting operation itself can be attained by aiming at a suboptimal cutting set (variant 2) rather than an optimal cutting set or by performing the cutting operation on the values of the current strategy rather than on the improved values obtained by the policy improvement operation (variant 3). Variant 4 reduces the frequency of use of the cutting operation.

The motivation to use this relatively simple model is primarily the fact that the same ALGOL 60 procedures can be used for any problem satisfying this special model. The numerical solution of the more general type of problems, covered by generalized Markov programming, requires problem-dependent numerical techniques. Yet the author has reasons to expect that most of the conclusions will be of value for the more general case.

Because each problem satisfying this special generalized Markov programming model can be solved by the policy iteration method of JEWELL (or by HOWARD's in case the natural process is described by a Markov chain), these methods are also concerned in the computational comparison.

2. 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. In a finite state semi Markov process the system makes random transitions among a finite number of states. Let J denote the set of states. The states are numbered by $i=1,\dots,|J|$. 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 $\underline{j} \in J$ *) 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 $|J| \times |J|$ - matrix of transition probabilities q_{ij}, $i,j \in J$, satisfying $0 \le q_{ij} \le 1$ and $\sum_{j \in J} q_{ij} = 1$; $u > 0$ denotes the $|J|$- dimensional vector of expected sojourn times and h denotes the $|J|$- dimensional vector with elements $-\infty < h_i < \infty$ representing the expected return of the process during the sojourn time in state i including the transition to the next state.

*) Random variables are underlined.

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, which is 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 the present 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 and the matrix Q have to satisfy the requirement that the inverse exists of the matrix $(I-Q)_{\overline{A_0}}$ with entries $\delta_{ij} - q_{ij}$ for $i,j \in \overline{A_0}$, with δ_{ij} satisfying $\delta_{ii} = 1$ and $\delta_{ij} = 0$ for $j \neq i$. 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 $g_i(x)$. If the system assumes state $\underline{m} = m$ after an intervention then it remains in state m until the next transition in the natural process has occurred. The sojourn time in state m has expectation $u_m = E\underline{\tau}_m$. By the foregoing the nulldecision can be viewed as an intervention satisfying

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

and

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

Strategies

A stationary deterministic strategic 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. The definitions of A_0 and A_z imply

(2) $$A_z \supseteq A_0 \; .$$

In the next section the iteration method induced by generalized Markov programming on this special model will be presented. It computes an optimal strategy, i.e. a strategy which maximizes the expected average return per unit of time. Because the existence of an optimal stationary deterministic strategy is guaranteed by the finiteness of the model, the computation is restricted to strategies of this class, denoted by Z.

3. 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

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

$$(t_0)_{A_0} := 0.$$

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 numbers $t(i,x)$ 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 as follows: Each element $k_0(i)$ $(t_o(i))$ 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 $k_0(i)$ $(t_o(i))$ 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 the set 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 the following three operations are executed.

Value determination operation

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 $(Q)_{\bar{A}_z A_z}$ denotes 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 the matrix $(I-Q)^{-1}_{\bar{A}_z}$ is implied by the existence of $(I-Q)^{-1}_{\bar{A}_0}$ and relation (2).

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 $(v(z))_{A_z}$ by solving 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 $K(\ell)$, $\ell=1,\dots,L(z)$ of the imbedded process an arbitrary state $i(\ell) \in K(\ell)$ for which we put $v_{i(\ell)}(z) = 0$, $\ell=1,\dots,L(z)$.

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

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

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

Policy improvement operation

Compute:

a. The $|J|$- dimensional vector y' with elements $y'_i, i \in J$ defined by

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

b. The subset $X_1(i)$ of $X(i)$ defined for each $i \in J$ by

[*)] It is assumed in generalized Markov programming that $p_{im}(z(i)) = 0$ for $i,m \in A_z$ for each stationary deterministic strategy z.

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

c. The $|J|$- dimensional vector v' with elements $v'_i, i \in J$ defined by

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

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

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

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 nulldecision 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 at the computation 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 $i \in A_z$ or equivalently

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

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) respectively by

$$\begin{cases} (y''(A))_{\bar{A}} := S(A)(y')_A \\ (y''(A))_A := (y')_A \end{cases}$$

and

$$\begin{cases} (v''(A))_{\bar{A}} := S(A)(v')_A \\ (v''(A))_A := (v')_A \end{cases}$$

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 is terminated. Otherwise the value determination operation is reentered with z := z".

4. FOUR VARIANTS ON THE ITERATION METHOD

1. *Usual policy improvement and optimal cutting.*

This variant uses the policy improvement operation of the preceeding paragraph followed by the optimal cutting algorithm stated and proved in [6]. This optimal cutting algorithm computes the set A^* by solving two optimal stopping problems in the Markov chain defined by the matrix Q of the natural process with y' and v' as return vectors respectively, see [5] or [6].

2. *Usual policy improvement and suboptimal cutting.*

This variant replaces the cutting operation by a suboptimal cutting algorithm. By this algorithm a suboptimal cutting set C is computed in the following way (see also [5]):

Compute

a. The set B(y') defined by

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

b. If $B(y') \neq A_{z'}$ and/or $y' > y(z)$ then take $C := B(y')$, otherwise continue with step c.

c. The set C defined by

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

This algorithm simply computes a member of the class M. If the class M consists only of sets A, $A_0 \subseteq A \subseteq A_{z'}$ satisfying

$$y''(A) = y''(A^*)$$

$$v''(A) = v''(A^*)$$

then C is defined to be identical to $A_{z'}$.

3. *Compound policy improvement*

This variant can be derived from the usual policy improvement operation by replacing for each $i \in \bar{A}_0$ the definition of $p_{ij}(x_0(i))$ (1) by

$$p_{ij}(x_0(i)) := q_{ij} \qquad j \in J.$$

Note that in the single chain case the compound policy improvement operation is identical to the usual policy improvement operation followed by suboptimal cutting on v(z) instead of v'.

4. *Reduction of the frequency of use of the cutting operation.*

A reduction of the frequency of use of the cutting operation can be done in many ways. By example we may use the policy improvement operation at each iteration step and the cutting operation only periodically with a period of 1,2,3 or more iteration steps. The period may also be changed during the iteration. Better is to let the choice of the period depend on the course of the iteration. The most natural way of doing this is to omit the cutting operation until the iteration has converged. Then use the cutting operation once. If this does not alter the lastly obtained strategy then this strategy is optimal. Otherwise, the iteration is resumed until again convergence is obtained by the policy improvement operation only. Then again the cutting operation is used once and so on. This procedure also converges within a finite number of steps to an optimal strategy. The numerical results by variant 4 are obtained with the suboptimal cutting algorithm of variant 2 substituting the cutting operation.

5. NUMERICAL RESULTS FROM RANDOMLY GENERATED PROBLEMS

The experiments carried out in this section are restricted to randomly generated problems satisfying the model considered in this report. In the problems generated all interventions imply a deterministic change of the state of the system. Consequently we shall denote each intervention $x \in X(i)$ by the state m it leads into.

To generate a problem, primarily the natural process given by the triple (Q,u,h) is generated. Each row of the matrix Q is obtained by generating $|J|$ random numbers and dividing them by their sum. The vectors u and h consist of random numbers multiplied by a suitable factor (here 1000 in both cases). The set of actions in each state $i \in J\backslash A_0$ is given by

$$X(i) := \{m=1,\dots,J\}$$

where $m = i$ corresponds to the nulldecision. The set A_0 is given by

$$A_0 := \{|J|\}$$

and because the nulldecision is infeasible in A_0 we have

$$X(|J|) := \{m=1,\dots,|J| - 1\}.$$

Note that each problem generated in this way will have exactly one ergodic set for each strategy of the class Z. In general the policy improvement operation may generate strategies with a sequence of interventions in zero time. To avoid violation of the basic notions of the model we will rule out strategies of this kind by imposing a condition on the numbers g_{im}. Suppose strategy z applies intervention $z(\ell) = m$ in state ℓ, then $v_\ell(z)$ satisfies

$$(3) \qquad v_\ell(z) = - g_{\ell m} + k_0(m) - k_0(\ell) - y(z)(t_0(m) - t_0(\ell)) + v_m(z).$$

If in state i the intervention $\ell \in X(i)$ is compared with intervention $m \in X(i)$ then intervention i → m is preferred over the sequence of interventions i → ℓ → m if

$$- g_{im} - k_0(m) - k_0(i) - y(z)(t_0(m) - t_0(i)) + v_m(z)$$

$$> - g_{i\ell} + k_0(\ell) - k_0(i) - y(z)(t_0(\ell) - t_0(i)) + v_\ell(z)$$

which yields using (3) and multiplying by - 1

$$(4) \qquad g_{im} < g_{i\ell} + g_{\ell m}.$$

Hence the numbers g_{im} should satisfy the strict triangular inequality. We note that condition (4) looks more stringent than is really necessary.

If the numbers g_{im} satisfy only

(5) $g_{im} \geq 0$ for $i,m \in J$,

then we determine for each pair of states the sequence of interventions which minimizes the total intervention costs. This problem is in fact the shortest route problem considered in graph theory. The resulting "shortest route" matrix then satisfies the triangular inequality. Hence what we suggest to do in problems satisfying (5) is to replace the matrix of intervention costs by its "shortest route" matrix and replace each sequence of interventions which yields this "shortest route" by one intervention. Then we solve the new problem. The iteration method then yields an optimal strategy which specifies an optimal strategy to the original problem by replacing each new intervention by the corresponding sequence of original interventions.

The number g_{im}, $i,m \in J$ were 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 scalar.

Two series of problems were generated.

Series a: 10 problems with 10 states and 10 actions per state
Series b: 5 problems with 50 states and 50 actions per state.

The following iteration methods were used:

1. Usual policy improvement and optimal cutting
2. Usual policy improvement and suboptimal cutting
3. Compound policy improvement
4. Using a cutting operation (here suboptimal cutting) only each time convergence with the policy improvement operation is obtained.
5. The policy iteration method of JEWELL [2].

Each problem of series a has been solved by each of the five methods for two initial strategies being:

(1) $z_1(i) = x_0(i)$ for $i = 1$ and $z_1(i) = 1$ for $i=2,\dots,|J|$.

and

$$(2) \quad \begin{cases} z_1(i) = x_0(i) & \text{for } i=1,\dots,|J|-1 \\ z_1(|J|) = 1 & \end{cases}$$

Results of series a:

Presented are the number of iteration steps for each problem, the total number of iteration steps as well as the total CPU-time minus time for compilation required to solve all 10 problems and the average time per step.

Initial strategy (1):

Method	Number of iteration steps per problem											Time	Time
	1	2	3	4	5	6	7	8	9	10	total		(per step)
1	5	5	6	4	5	4	4	5	6	1	45	36 sec.	.80 sec.
2	5	4	5	4	4	4	4	4	5	1	40	29 "	.71 "
3	4	3	5	4	3	3	3	3	3	1	32	24 "	.75 "
4	4	3	5	4	3	3	3	3	3	1	32	25 "	.77 "
5	3	2	3	3	3	2	2	2	2	1	23	20 "	.85 "

Initial strategy (2):

Method	Number of iteration steps per problem											Time	Time
	1	2	3	4	5	6	7	8	9	10	total		(per step)
1	4	4	5	5	4	4	3	4	5	4	42	33 sec.	.80 sec.
2	4	3	4	5	4	3	3	3	4	3	36	27 "	.75 "
3	3	2	4	5	3	2	2	2	2	2	27	22 "	.83 "
4	3	2	4	5	3	2	2	2	2	2	27	23 "	.88 "
5	3	2	3	4	3	2	2	2	2	2	25	25 "	.98 "

Each problem of series b has only been solved with initial strategy, because series a indicates that the results do not depend very much on the initial strategy.

Results of series b

Initial strategy (2)

Method	Number of iteration per problem						Time	Time
	1	2	3	4	5	total		(per step)
1	6	6	5	6	5	28	480 sec.	17.1 sec.
2	5	5	4	5	4	23	315 "	13.5 "
3	4	4	4	5	3	20	295 "	14.8 "
4	4	5	4	5	3	21	301 "	14.3 "
5	3	3	3	3	3	15	610 "	41.2 "

The results were obtained on the CDC Cyber 73 while previous results in WEEDA [5] were obtained on the Electrologica X8.

6. NUMERICAL RESULTS FROM A PRODUCTION PROBLEM

The following problem has also been presented in WEEDA [5] with numerical results only on variants 1 and 2. A continuous version of this problem has been solved in DE LEVE, TIJMS & WEEDA [4].

Problem formulation

A product can be produced at m production rates, $r=0,\dots,m$. $r = 0$ corresponds to the situation that the production is switched off and $r > 0$ to a production rate of r units of product per unit of time. The demand is supplied immediately from the available stock s. If the demand exceeds the available stock the shortage is replenished by an emergency purchase. The production is controlled by changing the production rate. As soon as the maximum stocklevel M is reached the production is switched off until the applied strategy prescribes a restart of the production. Stockholding cost amount c_1 per unit of time and per unit of product in stock at the end of the unit time period. An emergency purchase costs c_2 per unit product. Production cost is given by c_3r per unit of time for production rate r. Changing the production rate from r' to r" costs an amount of $b(r',r")$. Find a strategy that minimizes the average cost per unit of time.

State space

$$J := \{i = (r,s)\colon r=0,1,\dots,m,\ s=0,1,\dots,M\}.$$

It will be convenient to distinguish the subsets of states with fixed $r \in \{0,1,\dots,m\}$

$$J^{(r)} := \{i = (r,s) \in J\colon r \text{ fixed},\ s=0,1,\dots,M\}.$$

Natural process

The natural process is defined for each initial state $(r,s) \in J$.
The natural process with initial state (r,s) visites only the states of subset $J^{(r)}$. Let i=)r,s) be a given state and let $\underline{j} = (r,\underline{s}')$ be the state

assumed after the next transition. Then $\underline{s}'$ is given by

$$\underline{s}' := \begin{cases} s + r - \underline{k} & \text{if } 0 < s + r - \underline{k} < M \\ M & \text{if } s + r - \underline{k} \geq M \\ 0 & \text{if } s + r - \underline{k} \leq 0 \end{cases}$$

where $\underline{k}$ denotes the size of the demand with probability

$$a_k := P\{\underline{k} = k\} = \frac{\lambda^k}{k!} e^{-\lambda}.$$

Then we have

$$P\{\underline{s}' = M|(r,s)\} = P\{\underline{k} \leq s + r - M\} = \sum_{k=0}^{s+r-M} a_k$$

$$P\{\underline{s}' = 0|(r,s)\} = P\{s + r - \underline{k} \leq 0\} = P\{\underline{k} \geq r + s\} = \sum_{k=s+r}^{\infty} a_k.$$

$$P\{\underline{s}' = s'|(r,s)\} = P\{\underline{k} = r + s - s'\} = a_{r+s-s'}, \quad \text{for } 0 < s' < M.$$

These relations define the probabilities of the natural process q_{ij} for $i = (r,s)$ and $j = (r,s')$. For $i = (r,s)$ and $j = (r',s')$ with $r' \neq r$ we have $q_{ij} = 0$. By the formulation of the problem we have

$$u_i := 1 \qquad \text{for } i \in J.$$

The expected return h_i in state $i = (r,s)$ consists of stockholding cost, emergency purchase expenses and production cost and is given by

$$h_i := -c_1 E\{\underline{s}'|(r,s)\} + c_2 E\{\min[s + r - \underline{k}, 0]\} - c_3 r =$$

$$= -c_1 M \sum_{k=0}^{s+r-M} a_k - c_1 \sum_{s'=1}^{M-1} a_{r+s-s'} \cdot s' +$$

$$- c_2 \sum_{k=s+r+1}^{\infty} a_k (s+r-k) - c_3 r.$$

Interventions

In each state $i = (r,s) \in J$ each feasible intervention implies a deterministic change of the state of the of the system to state (r',s') satisfying $r' \neq r$ and $s' = s$. The intervention cost $g_i(x)$ is given by

$$g_i(x) := b(r,r') \qquad \text{for } i = (r,s) \text{ and } x = (r',s).$$

The numbers $b(r,r')$ are assumed to satisfy

$$b(r,r') \leq b(r,r'') + b(r'',r')$$

for $r,r',r'' \in \{0,1,\dots,m\}$.

The set A_0

$$A_0 := \{i = (r,M)\colon r=1,\dots,m\} \cup \{(0,0)\} \cup \{(1,0)\}.$$

Three different numerical versions of the problem were solved by the 4 variants of section 4 and the method of HOWARD (method 5).

Numerical problem 1

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

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

Initial strategy:

$$\begin{aligned} z_1(r,0) &= (3,0) && \text{for } r = 0,1 \\ z_1(r,20) &= (0,20) && \text{for } r = 1,2,3 \\ z_1(r,s) &= x_0(r,s) && \text{otherwise.} \end{aligned}$$

Optimal strategy:

s	r'	s	r'	s	r'	s	r'
0,1	3	0,1	3	0,...,3	2	0,1,2	3
2	2	2,...,7	1	4,5,6	1	3,...,7	1
3,...,20	0	8,...,20	0	7,...,20	0	8,...,20	0

Computational performance:

Method	Number of iterations	CPU-time iteration
1	6	105 (+4) sec. *)
2	6	98 (+4) sec.
3	5	98 (+4) sec.
4	9	130 (+4) sec.
5	6	170 sec.

Convergence of y(z):

Strategy	Method 1	Method 2	Method 3	Method 4	Method 5
1	-3.674	-3.674	-3.674	-3.674	-3.674
2	-2.836	-2.836	-2.710	-2.710	-2.710
3	-2.484	-2.470	-2.489	-2.593	-2.438
4	-2.346	-2.340	-2.351	-2.459	-2.360
5	-2.339	-2.339	-2.339	-2.409	-2.341
6	-2.339	-2.339		-2.396	-2.339
7				-2.339	
8				-2.339	
9				-2.339	

*) The number between parenthesis is the computation time for the vectors k_0 and t_0 together.

Numerical problem 2

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

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

Initial strategy:

$$\begin{array}{ll} z_1(r,0) = (3,0) & \text{for } r = 0,1 \\ z_1(r,20) = (0,20) & \text{for } r = 1,2,3 \\ z_1(r,s) = x_0(r,s) & \text{otherwise} \end{array}$$

Optimal strategy:

r = 0		r = 1		r = 2		r = 3	
s	r'	s	r'	s	r'	s	r'
0,1	3	0,1	3	0	3	0,...,6	3
2,3	2	2	2	1,...,7	2	7,...,10	1
4,...20	0	3,...,13	1	8,...,10	1	11,...,20	0
		14,...,20	0	11,...,20	0		

Computational performance:

Method	Number of iterations	CPU-time iteration
1	6	119 (+4) sec.
2	4	73 (+4) sec.
3	6	119 (+4) sec.
4	9	136 (+4) sec.
5	6	171 sec.

Convergence of y(z):

Strategy	Method 1	Method 2	Method 3	Method 4	Method 5
1	-4.453	-4.453	-4.453	-4.453	-4.453
2	-3.653	-3.560	-3.600	-3.600	-3.600
3	-3.392	-3.267	-3.400	-3.600	-3.380
4	-3.293	-3.249	-3.249	-3.400	-3.291
5	-3.260		-3.249	-3.312	-3.249
6	-3.249		-3.249	-3.310	-3.249
7				-3.253	
8				-3.253	
9				-3.249	

Numerical problem 3

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

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

Initial strategy:

$z_1(r,0) = (3,0)$ for $r = 0,1$

$z_1(r,25) = (0,25)$ for $r = 1,2,3$

$z_1(r,s) = x_0(r,s)$ otherwise.

Optimal strategy:

r = 0		r = 1		r = 2		r = 3	
s	r'	s	r'	s	r'	s	r'
0,1	3	0,1	3	0	3	0,...,4	3
2,3,4	2	2,3,4	2	1,...,10	2	5,6	2
5,...,25	0	5,...,18	1	11,12	1	7	3
		19,...,25	0	13,...,25	0	8,...,12	1
						13,...,25	0

Computational performance:

Method	Number of iterations	CPU-time iteration
1	7	220 (+6.4) sec.
2	4	117 (+6.4) sec.
3	6	189 (+6.4) sec.
4	9	212 (+6.4) sec.
5	8	349 sec.

Convergence of y(z):

Strategy	Method 1	Method 2	Method 3	Method 4	Method 5
1	-5.147	-5.147	-5.147	-5.147	-5.147
2	-4.313	-4.196	-4.550	-4.550	-4.550
3	-4.007	-3.742	-4.099	-4.550	-4.094
4	-3.850	-3.733	-3.797	-4.099	-3.770
5	-3.741		-3.733	-3.801	-3.744
6	-3.733		-3.733	-3.801	-3.733
7	-3.733			-3.733	-3.733
8				-3.733	-3.733
9				-3.733	

The numerical results presented in this report are obtained on the CDC Cyber 73 while previous results in WEEDA [5] were obtained on the former Electrologica X8 of the Mathematical Centre.

7. CONCLUSIONS

From all examples solved, we observe that the computation time per iteration step for each of the four variants of generalized Markov programming is smaller than required for the conventional policy iteration methods of HOWARD [1] and JEWELL [2]. The explanation is evidently the fact that the value determination operation of the GMP iteration method in the single chain case requires the inversion of two square matrices of size $|\bar{A}_z|$ and $|A_z|$ respectively while the conventional methods require the inversion of one square matrix of size $|J|$. Because the computation time required to invert a NxN - matrix is proportional to N^3, the GMP variants will usually be faster per step. In the sequel we will denote these variants by GMP1... GMP4. The second important contributor in the overall computation time is the number of iteration steps required. For the whole sample of randomly generated problems we have *uniformly* in the number of iteration steps

$$\text{JEWELL} \le \text{GMP3} \le \text{GMP4} \le \text{GMP2} \le \text{GMP1}$$

This uniformicity is not merely a consequence of a too small sample size, because previously in WEEDA [5] an experiment, restricted to GMP1 and GMP2, has been performed for which the sample size of series a has been extended to 65 problems. The result has been that uniformly in the number of iteration steps for the whole sample

$$\text{GMP2} \le \text{GMP1}$$

For the three numerical versions of the production problem there is less uniformicity. Here we have uniformly

$$\text{GMP2 and 3} \le \text{GMP1} \le \text{HOWARD} < \text{GMP4}$$

Here also GMP2 and GMP3 have to be preferred over GMP1 while conventional policy iteration even looses its advantage of requiring a smaller number of iteration steps. Note that the position of GMP2 is strengthened and the position of GMP4 is worsened compared with the random sample. We note further that small deviations in the number of iteration steps between this and the previous study are caused by the fact that a newly developed numerical procedure to solve a linear system has been used here.
From the experiments as a whole we may conclude that in view of the overall computation time generalized Markov programming has to be preferred over conventional policy iteration in this model. Further if one agrees to use GMP then preferably use variants 2 or 3 rather than conventional generalized Markov programming (variant 1).

8. REFERENCES

[1] HOWARD, R.A., *Dynamic Programming and Markov Processes*, M.I.T. Press, Cambridge, Massachusetts, 1960.

[2] JEWELL, W.S., *"Markov-Renewal Programming, I: Formulation, Finite Return Models, II: Infinite Return Models"*, Operations Research 10 (1963), Vol. 6, pp. 938-972.

[3] LEVE, G. DE, *Generalized Markovian Decision Processes, Part I: Model and Method, Part II: Probabilistic Background*, Mathematical Centre Tracts 3 and 4, Amsterdam, 1964.

[4] LEVE, G. DE, TIJMS, H.C. & WEEDA, P.J., *Generalized Markovian Decision Processes, Applications*, Mathematical Centre Tracts 5, Amsterdam, 1970.

[5] WEEDA, P.J., *Generalized Markov Programming with a finite state semi Markov process as natural process (Extended abstract)*, Rapport BW 32/74, Prepublication, Mathematisch Centrum, Amsterdam, 1974.

[6] WEEDA, P.J., *On the relationship between the cutting operation of generalized Markov Programming and optimal stopping.* Rapport BW 36/74, Mathematisch Centrum, Amsterdam, 1974.