Journal of the Operations Research
Society of Japan
Vol. 27, No. 2, June 1984

ON THE CLASS OF CLOSED DYNAMIC PROGRAMS

Katsushige Sawaki
Nanzan University

(Received February 24, 1982: Final March 12, 1984)

Abstract This paper considers a class of general dynamic programs which satisfies the monotonicity and contraction assumption, and in which the sets of cost functions and policies are closed under the monotone contraction operators. This class of dynamic programs includes, piecewise linear, affine dynamic programs, partially observable Markov decision processes, and many sequential decision processes under uncertainty such as machine maintenance control models and search problems with incomplete information.

An algorithm based on generalized policy improvement has the property that it only generates cost functions and policies belonging to distinguished subsets of cost functions and policies, respectively.

1. Introduction

Special classes of dynamic programs were proposed by [5], [6], [12] and [16]. On the other hand, an algorithm for dynamic programs was developed in [16] (also, see [13].) This algorithm, called generalized policy improvement, includes policy improvement [2], [10] and successive approximation [1], [4] as special cases. This paper considers a class of dynamic programs, called closed, with the property that the generalized policy improvement algorithm stays within a certain "small" subset of cost functions and policies. In other words, the sets of cost functions and policies generated by the algorithm are closed under the monotone contraction operators. Furthermore, it is possible to keep such sets within "distinguished small" subsets of the sets of all bounded cost functions and all stationary policies, respectively. This property is very important to dynamic programming from a computational aspect. The class of closed dynamic programs includes piecewise linear dynamic programming [16], affine dynamic programming [5], [6], partially observable Markov decision processes [7], [15], [17] and many sequential decision processes with imperfect information such as machine maintenance models [17] and search models. The approximation of dynamic programs is

analyzed by [19], [20], [11], which are different from our approach. This paper concerns with the method of constructing of ε-optimal policy and cost.

Closed dynamic programs are defined in Section 2. We also discuss conditions that ensure the existence of an ε-optimal policy within the distinguished subset of policies and of an ε-optimal cost within the distinguished subset of cost functions. An algorithm for finding an ε-optimal policy and the proof of the convergence are given in Section 3. Examples, special cases of closed dynamic programs, are given in Section 4. In the closed dynamic program the distinguished subsets of cost functions and policies generated by the algorithm are easily stored in a computer even for uncountable state space dynamic programs.

2. Closed Dynamic Programs

First of all, we define the general class of dynamic programs which satisfy the monotone and contraction assumption of Denardo [4]. Secondly the class of closed dynamic programs is defined. The state space Ω is an arbitrary non-empty set. Let V be the set of all bounded real valued functions on Ω. An element v of V is a cost function. The norm defined by $\|v\| = \sup\{|v(x)|:x\varepsilon\Omega\}$ makes V a Banach space. For u and v in V we write $u\leq v$ if $u(x)\leq v(x)$ for each $x\varepsilon\Omega$. The norm of V is monotone in the sense that $0\leq u\leq v$ implies $\|u\|\leq\|v\|$. For each $x\varepsilon\Omega$ there is a set A_x of actions. Let Δ be the Cartesian product $\underset{x\varepsilon\Omega}{\times} A_x$. An element $\delta\varepsilon\Delta$ is called a policy. The loss function h is defined to be a mapping from $\underset{x\varepsilon\Omega}{\cup}\{x\} \times A_x \times V$ into a real number. In Markov decision processes the loss function h can be written as $h(x,a,v) = c(x,a) + \beta\int_\Omega v(y)q(dy|x,a)$ where $c(x,a)$ is the one period cost, β the discount factor, v a terminal cost and $q(\cdot|x,a)$ the transition probability measure on Ω when the pair (x,a) causes a transition to new state y.

The loss function h is assumed to satisfy the contraction and monotonicity assumptions as follows:

The contraction assumption: For some $\beta \varepsilon [0,1)$ $|h(x,a,u)-h(x,a,v)| \leq \beta\|u-v\|$ for each u, $v\varepsilon V$, $x\varepsilon\Omega$, $a\varepsilon A_x$.

The monotonicity assumption: For each $x\varepsilon\Omega$ and $a\varepsilon A_x$, $h(x,a,u) \leqq h(x,a,v)$ whenever $u\leq v$ in V.

We define two operators H_δ for $\delta\varepsilon\Delta$ and H_* as follows:

$$(H_\delta v)(x) = h(x,\delta(x),v),\ x\varepsilon\Omega,\ v\varepsilon V,$$

and

$$(H_* v)(x) = \inf_{\delta \varepsilon \Delta} H_\delta v(x), \quad x \varepsilon \Omega, \quad v \varepsilon V .$$

Assume that for each $v \varepsilon V$ there is some $\bar{\delta} \varepsilon \Delta$ such that $H_{\bar{\delta}} v = \inf_{\delta \varepsilon \Delta} H_\delta v$. Denardo [4] gives a useful sufficient condition for this to hold. An operator $H : V \to V$ is monotone if $u \leq v$ implies $Hu \leq Hv$, and is a contraction if for some $\beta \varepsilon [0,1)$, $\|Hu - Hv\| \leq \beta \|u - v\|$ for each u, v in V. Denardo [4] verifies under the monotone contraction assumption that H_* and H_δ are monotone contraction operators.

By Banach's fixed point theorem for contraction operators, for each $\delta \varepsilon \Delta$ there is a unique $v_\delta \varepsilon V$ such that $H_\delta v_\delta = v_\delta$, which is called the cost of the policy δ. Similarly, v^* is called the optimal cost if $H_* v^* = v^*$. If $v_\delta = v^*$, then the policy δ is called optimal. If $\|v - v^*\| \leq \varepsilon$, then v is an ε-optimal cost function. The dynamic programs defined so far, together with the assumption imposed are called the general class of dynamic programs (abbreviated by GDP). The class of closed dynamic programs (abbreviated by CDP) is a subclass of GDP which has a subset of cost functions $V' \subset V$ and subset of policies $\Delta' \subset \Delta$ which satisfy the following two conditions:

(i) $H_\delta v \varepsilon V'$ whenever $\delta \varepsilon \Delta'$ and $v \varepsilon V'$,

(ii) if $v \varepsilon V'$, then there exists some $\delta \varepsilon \Delta'$ such that $H_\delta v = H_* v$.

The reason why the dynamic program is called closed is because it is closed under operators H_δ, H_*. Elements of V' and Δ' are called closed cost functions and closed policies, respectively. In this paper we explore how GDP can be approximated by CDP, that is, CDP always possesses an ε-optimal cost and ε-optimal policies to the optimal cost and optimal policy of GDP, respectively. In the following section an algorithm is developed to generate a sequence of ε-optimal costs staying within CDP.

3. The Generalized Policy Improvement Algorithm for CDP

GDP may be defined by the triple $\{\Omega, \Delta, V\}$. Similarly, CDP is defined by the triple $\{\Omega, \Delta', V'\}$, where Δ' and V' may be distinguished subsets of Δ and V, respectively. In this section we develop an algorithm which enables us to preserve cost functions and policies within V' and Δ', respectively.

Lemma 1. For a given GDP $\{\Omega, \Delta, V\}$ and CDP $\{\Omega, \Delta', V'\}$, define $\delta_n \varepsilon \Delta$ such that $H_{\delta_n} v^n = H_* v^n$, and define $v^{n+1} = H^k_{\delta_n} v^n, v^n \varepsilon V$, $n=0,1,2,\ldots$, where k, depending on n, is a number of iterations of H_{δ_n} to be applied.

(i) If $v^0 \varepsilon V'$, then $\delta_n \varepsilon \Delta'$ and $v^{n+1} \varepsilon V'$ for $n=0,1,2,\ldots$.

(ii) If there exists $v^0 \in V'$ satisfying $v^0 \geq H_* v^0$, then $\{v^n\}$ is a decreasing sequence in V' and $v^n \geq v^* \in V$.

Proof: (i) The proof is by induction on n. From the condition (ii) of CDP, if $v^0 \in V'$, then there exists a policy $\delta_0 \in \Delta'$ such that $H_{\delta_0} v^0 = H_* v^0$. For such $v^0 \in V'$ and $\delta_0 \in \Delta'$, we have $v^1 \equiv H^k_{\delta_0} v^0 = H^{k-1}_{\delta_0}(H_{\delta_0} v^0) \in V'$ by using the condition (i) of CDP in a sequential fashion on k. Suppose that $v^n \in V'$. Then, by the same argument as in the step $n = 0$, there exists a policy $\delta_n \in \Delta'$ such that $H_{\delta_n} v^n = H_* v^n$. For such $\delta_n \in \Delta'$ and $v^n \in V'$, we have $v^{n+1} = H^k_{\delta_n} v^n \in V'$ by using the condition (i) of CDP.

(ii) First, we shall show by induction on n that $v^n \geq H_{\delta_n} v^n$ for $n=0,1,2,\ldots$. For $n = 0$ we have $v^0 \geq H_* v^0 = H_{\delta_0} v^0 \in V'$. Assume for $n > 0$ that $v^n \geq H_{\delta_n} v^n$. For $(n+1)$ we obtain $v^{n+1} \equiv H^k_{\delta_n} v^n \geq H^k_{\delta_n}(H_{\delta_n} v^n) = H_{\delta_n}(H^k_{\delta_n} v^n) = H_{\delta_n} v^{n+1} \geq H_* v^{n+1} = H_{\delta_{n+1}} v^{n+1}$. So, we have $v^n \geq H_{\delta_n} v^n$ for all n. Therefore, $v^{n+1} = H^k_{\delta_n} v^n \leq H_{\delta_n} v^n \leq v^n$, which implies that $\{v^n\}$ is a decreasing sequence. Hence, $v^n \geq v^{n+1} = H^k_{\delta_n} v^n \geq H^k_* v^n$. For a fixed n $H^k_* v^n$ converges pointwise to v^* as $k \to \infty$. Consequently, $v^n \geq v^* \in V$. The way of constructing of v^n guarantees that $v^n \in V'$, but the limit v^* may not belong to V'. (Q.E.D.)

An algorithm for approximating the fixed point v^* of H in finite steps is presented. A terminal criterion is given by

(1) $\quad \|v - Hv\| \leq (1-\beta)\varepsilon$ implies $\|v - v^*\| \leq \varepsilon$

where β is a contraction coefficient of operator H. An upper bound on the number of iterations starting from v required to obtain an ε-approximation to v^* can be derived from (1), namely,

(2) $\quad \|v - Hv\| \leq (1-\beta)/\beta^n$ implies $\|H^n v - v\| \leq \varepsilon$.

Lemma 2. $\|v^* - v^n\| \leq (1-\beta)\varepsilon/2\beta$ implies $\|v^* - v_{\delta_n}\| \leq \varepsilon$.

Proof:
$$\begin{aligned}\|v^* - v_{\delta_n}\| &= \|H_* v^* - H_{\delta_n} v_{\delta_n}\| \\ &\leq \|H_* v^* - H_{\delta_n} v^n\| + \|H_{\delta_n} v^n - H_{\delta_n} v^*\| + \|H_{\delta_n} v^* - H_{\delta_n} v_{\delta_n}\| \\ &\leq \beta\|v^* - v^n\| + \beta\|v^n - v^*\| + \beta\|v^* - v_{\delta_n}\|\end{aligned}$$

where we use the equality $H_{\delta_n} v^n = H_* v^n$. Arranging the above inequality,

we have

$$\|v^* - v_{\delta_n}\| \le \frac{2\beta}{1-\beta}\|v^* - v^n\| \le \varepsilon. \qquad \text{(Q.E.D.)}$$

Putting equations (1), (2) and lemma 2 together in restating the stopping rule in terms of n, we have

$$\|H^n v - v\| \le \varepsilon \text{ for } n > \log \frac{(1-\beta)^2\varepsilon}{2\beta\|v - Hv\|} \Big/ \log \beta.$$

If H_δ is applied for $\delta \in \Delta'$, then the method of generating v^n is a policy improvement part. If H_* is applied, then the method is a successive approximation part. In CDP either H_* or H_δ is applied. Thus the following algorithm is called generalized policy improvement.

Algorithm: Start with $v_0 \in V'$ satisfying $v^0 \ge H_* v^0$. Set $n = 0$.

Step 1 Find $\delta_n \in \Delta'$ such that $H_{\delta_n} v^n = H_* v^n$.

Step 2 If $\|v^n - H_{\delta_n} v^n\| \le (1-\beta)\,\varepsilon$, then go to Step 4.

Step 3 Otherwise, choose some positive integer k_n and evaluate $v^{n+1} \equiv H_{\delta_n}^{k_n} v^n$. Increment n by 1 and go to Step 1.

Step 4 δ_n is an ε-optimal closed policy, and $H_* v^n$ and v^n are ε-optimal closed cost functions. Furthermore, $v^n \ge H_* v^n \ge v^*$.

In CDP the algorithm provides a procedure of approximating either v_δ or v^* by iterating H_δ or H_*, respectively, until (1) is satisfied. If $v \in V'$, then $H_*^n v \in V'$ for each n. If $\delta \in \Delta'$, then $H_\delta^n v \in V'$ for each n. Such algorithm involves only functions in V' and policies in Δ'.

Lemma 3. Let $\{v^n\}$ be a sequence of closed cost functions generated by the algorithm and define δ_n by $H_{\delta_n} v^n = H_* v^n$. Then, $v^n \ge v_{\delta_n} \ge v^*$. Furthermore, if $\|v^n - v^{n+1}\| \le (1-\beta)\varepsilon$, δ_n is an ε-optimal closed policy.

Proof: Let n be arbitrary but fixed. From the proof for lemma 1 (ii) and the monotonicity of H, we have

$$v^n \ge H_{\delta_n}^{k_n} v^n \ge H_*^{k_n} v^n \quad \text{for each } k_n.$$

Since $H_{\delta_n}^{k_n} v^n \to v_{\delta_n}$ and $H_*^{k_n} v^n \to v^*$ as $k_n \to \infty$ and n is arbitrary, we obtain

$$v^n \ge v_{\delta_n} \ge v^* \text{ for all } n.$$

On the other hand, we have

$$\|v^n - v^*\| \le \|v^n - H_* v^n\| + \|H_* v^n - H_* v^*\|$$
$$\le \|v^n - H_{\delta_n} v^n\| + \beta\|v^n - v^*\|$$
$$\le \|v^n - H_{\delta_n}^{k_n} v^n\| + \beta\|v^n - v^*\| \quad \text{for each } k_n$$

because $v^n \ge H_{\delta_n} v^n \ge H_{\delta_n}^{k_n} v^n$ for $k_n = 1,2,\ldots$. Thus, $(1-\beta)\|v^n - v^*\| \le \|v^n - H_{\delta_n}^{k_n} v^n\| \equiv \|v^n - v^{n+1}\| \le (1-\beta)\varepsilon$, which implies that v^n is ε-optimal. So is v_{δ_n} because v_{δ_n} is smaller than v_n. The way of constructing v_n and δ_n preserves them to be closed for finite number of n. (Q.E.D.)

From lemmas 1, 3 we may conclude that the algorithm converges. This argument verifies the following theorem.

Theorem 1. CDP has ε-optimal closed cost functions and ε-optimal closed policies, provided that there exists some $v_0 \in V'$ such that $v_0 \ge H_* v_0$.

Remarks: (i) If $k_n = 1$ for each n, then the algorithm reduces to successive approximation and Step 1 is to evaluate $H_* v^n$. If in Step 3 $\lim_{k\to\infty} H_{\delta_n}^k v^n = v^{n+1}$ can be evaluated, then $v^{n+1} = v_{\delta_n}$ and the algorithm is policy improvement. In that case, however, v^{n+1} may not belong to V' because V' is not necessarily closed in V, and so v_δ is not necessarily in V' even if $\delta \in \Delta$. Therefore, in the algorithm we must choose a finite number of k_n for each step n in order to keep v^n staying only in V'.

(ii) To start the algorithm we must find $v^0 \in V'$ satisfying $v^0 \ge H_* v^0$. In the appendix we discuss how to find such function v^0 for the three examples of CDP.

4. Special Classes of Closed Dynamic Programs

For a given GDP we formulate CDP and then discuss the relationship between them. The algorithm shows how to nicely generate closed cost functions for CDP and how GDP can be approximated by CDP in finite iterations. In this context, someone may raise a question how large the class of CDP is. This section will give you an answer by showing several examples of CDP.

4.1. Piecewise dynamic programs

A finite collection $B = \{B_1, B_2, \ldots, B_m\}$ of subsets of Ω is a partition of Ω if $B_i \cap B_j = \phi$ for $i \ne j$ and $\bigcup_{i=1}^m B_i = \Omega$. The product of two partitions

P_1 and P_2 is $P_1 \times P_2 = \{B \cap D : B \varepsilon P_1,\ D \varepsilon P_2\}$. The product of $P_1, P_2, \ldots, P_n$ is defined inductively by $\mathop{\times}_{i=1}^{n} P_i = P_n \times \mathop{\times}_{i=1}^{n-1} P_i$. A function $v \varepsilon V$ is called piecewise if there exists a partition $\{B_1, B_2, \ldots, B_m\}$ of Ω, a subset V' of V and a vector of functions $\{v_1, v_2, \ldots, v_m\}$ such that $v(x) = v_i(x)$ on each B_i and each $v_i \varepsilon V'$. A policy $\delta \varepsilon \Delta$ is called piecewise if there exists a partition $\{B_1, B_2, \ldots, B_m\}$ of Ω and a set of actions $\{a_1, a_2, \ldots, a_m\}$ such that $\delta(x) = a_i$ on each B_i and $a_i \varepsilon \bigcap_{x \varepsilon B_i} A_x$, $i=1,2,\ldots,m$. A piecewise dynamic program is a closed dynamic program with V' as the set of piecewise functions in V and Δ' as the set of piecewise policies in Δ. A piecewise linear dynamic program [16] is an example of the piecewise dynamic program and hence of CDP if we take V' as the set of piecewise linear functions, Δ' as the set of piecewise constant policies and each cell of a partition B_i as a convex polyhedron. The paper by Denardo and Rothblum [5] discuss piecewise dynamic programs with V' as the set of affine functions in V and Δ' as the set of constant policies in Δ. Finite states Markov decision processes are piecewise dynamic programs with V' as the set of finite numbers, Δ' as the set of piecewise constant policies and each cell B_i of a partition as a singleton subset. Another example, which has theoretical interest, arises when $A_x = A$ for all $x \varepsilon \Omega$ where A is a measurable space, V' is the set of Borel measurable functions in V, and Δ' is the set of measurable policies in Δ.

The next theorem provides a sufficient condition for a general dynamic program to be a piecewise (closed) dynamic program.

Theorem 2. Suppose that GDP has the property that for each $x \varepsilon \Omega$, A_x is the same finite set $A = (a_1, a_2, \ldots, a_p)$. Let V' be the set of piecewise cost functions and let Δ' be the set of piecewise policies. If $h(\cdot, a, v) \varepsilon V'$ for each $a \varepsilon A$ and $v \varepsilon V'$, then the GDP is piecewise.

Proof: Choose $v \varepsilon V'$ and $\delta' \varepsilon \Delta'$. Suppose $\delta(x) = a_i$ on each B_i where $\{B_1, B_2, \ldots, B_m\}$ is a partition of Ω. Since $h(\cdot, a_i, v)$ is piecewise for each i, there exists a partition $\{C_{i1}, C_{i2}, \ldots, C_{in}\}$ of Ω, a subset V_i' of V' and a vector of functions $\{w_{i1}, w_{i2}, \ldots, w_{in}\}$ such that $h(x, a_i, v) = w_{ij}(x)$ on each C_{ij} and each $w_{ij} \varepsilon V_i'$. Let $P_i = \{C_{ij} \cap B_i : j=1,2,\ldots,n\}$ and $P = \bigcup_{i=1}^{m} P_i$ is also a partition of Ω. In addition, since $\delta(x) = a_i$ on each B_i,

$$(H_{\delta(x)} v)(x) = h(x, a_i, v) \quad \text{on } B_i$$
$$= w_{ij}(x) \quad \text{on } B_i \cap C_{ij}$$

, which is again piecewise. Thus $H_\delta v \varepsilon V'$, which satisfies (i) of the definition of CDP.

Let $v \varepsilon V'$. We next show how to find $\delta \varepsilon \Delta'$ such that $H_\delta v = H_* v$. For

$a \in A$, $h(\cdot,a,v)$ is piecewise, say $h(x,a,v) = d_{ja}(x)$ on the j-th cell of a partition P_a. Form the product partition $\overset{p}{\underset{i=1}{\times}} P_{a_i} = P$ and put $P = \{B_1, B_2, \ldots, B_m\}$, after reordering the cells of the partition. If B_i is a subset of the j-th cell of P_a, let $d_{ja}(\cdot) = w_{ia}(\cdot)$. For each $a \in A$, P is plainly finer than P_a, so that $h(x,a,v) = w_{ia}(x)$ on B_i, $i=1,2,\ldots,m$. For each $i=1,2,\ldots,m$ and $j=1,2,\ldots,p$ define the set

$$G_{ij} = \{x \in B_i : w_{ia_j}(x) \leq w_{ia_k}(x) \quad \text{for} \quad k=1,2,\ldots,p, k \neq j\}.$$

Then, put $Q_i = \{G_{ij} : j=1,2,\ldots,p\}$ is a partition of B_i and $Q \equiv \bigcup_{i=1}^{m} Q_i$ is a partition of Ω with the property that

$$(H_* v)(x) = \inf_{\delta} H_\delta(x) = \min_k h(x,a_k,v)$$

$$= w_{ij}(x) \quad \text{if} \quad x \in G_{ij} \quad \text{which is a cell of } Q.$$

Thus, the policy $\delta \in \Delta'$ defined by $\delta(x) = a_j$ on G_{ij} satisfies $H_\delta v = H_* v$, which completes the proof. (Q.E.D.)

4.2. Partially observable Markov decision processes

To introduce partially observable Markov decision processes, we first discuss a machine maintenance and repair model similar to that in Smallwood and Sondik [17]. A machine consists of two internal components. The state of the machine is the number of working components. The machine produces four finished items at each period and the machine cannot be inspected. However, a random sample from the four items can be selected and the number of defectives determined. The number of defective items out of the four has a binomial distribution with mean $4\pi_i$ if the state of the machine is i. At the beginning of a period, a prior probability distribution as to the state of the machine is known. Based on this distribution, an action is taken whether or not to overhaul the machine and the number of items to be sampled from the next lot. An action $a = (1,k)$ represents the action to overhaul and sample k from the next lot and $a = (0,k)$ represents the action of not overhauling and sampling k from the next lot. A cost $c(i,a)$ is incurred if action a is made when the machine is in state i. The dynamics of the process $\{Z_n : n = 0,1,2,\ldots\}$ giving the state of the machine in period n are governed by two probability transition matrices P_1 if the machine is overhauled and P_0 if the machine is not overhauled. The loss function is $h(i,(\delta,k),v) = c(i,(\delta,k)) + \beta \sum_{j=0}^{2} P_\delta(i,j)v(j)$ where $\beta \in (0,1)$ is the discount factor and δ is zero or one.

In general consider a dynamic program with state space $S = \{1,2,\ldots,\}$ (called the *core process*), with a finite action set A such that each action is admissible in each state, and a loss function

$$\hat{h}(i,a,v) = c(i,a) + \beta(P_a v)(i), \quad (i,a,v) \in S \times A \times R^N$$

where $\beta \in (0,1)$ is the discount factor, each P_a is a probability transition matrix, and $(P_a v)(i)$ is the i-th component of the vector $P_a v$. Thus the objective is to maximize $E \sum_{n=0}^{\infty} \beta^n c(Z_n,a_n)$ where Z_n is the state of the core process and a_n is the action at period n. However, Z_n cannot be observed. Instead a signal Y_n which takes values in the finite set Θ is generated by the conditional probability distribution function

$$\Gamma(\theta|Z_n,a_n) = Pr[Y_n = \theta|Z_n,a_n] = Pr[Y_n = \theta|Z_n,a_n,Z_k,a_k,Y_k,k \leqq n-1].$$

Assume that the probability distribution of Z_0 is known, say

$$X_0(i) = Pr[Z_0 = i] \qquad i = 1,2,\ldots,N.$$

The n-th action, a_n, is based on the history of the process $H_n = (X_0; Y_1,Y_2,\ldots,Y_n; a_0,a_1,\ldots,a_{n-1})$. Let X_n be the probability vector defined by $X_n(i) = Pr[Z_n = i|H_n]$ for $i \in S$. It can be shown (cf. Dynkin [7]) that $X_{n+1}(i) = Pr[Z_{n+1} = i|H_{n+1}] = Pr[Z_{n+1} = i|Z_n,Y_{n+1},a_n]$. Thus X_n is a sufficient statistic for the history H_n. It follows that $\{X_n : n = 0,1,2,\ldots\}$ is a Markov process and is called the *observed process*. Its space is $\Omega = \{x \in R^N : x_i \geqq 0, \sum_1^N x_i = 1\}$. Its loss function is $h(x,a,v) = r_a \cdot x + \beta\int_\Omega v(y)q(dy|x,a)$ where $q(B|x,a) = Pr[X_{n+1} \in B|X_n = x, a_n = a]$, r_a is the vector $(c(i,a) : i \in S)$, and v is a bounded real-valued function defined on Ω.

A formula expressing the probability transition function $q(B|x,a)$ in terms of Γ and P_a is derived in [14]. The vector X_{n+1} is a deterministic function of X_n,a_n, and Y_{n+1}. Let $P_{\theta,a}$ be the matrix with components $P_{\theta,a}(i,j) = \Gamma(\theta|j,a)P_a(i,j)$. Let $g(x,a,\theta) = \frac{P_{\theta,a}x}{P_{\theta,a}x}$, so that $X_{n+1} = g(X_n,a_n, Y_{n+1})$. For $B \subset R^N$ and $x \in \Omega$, define the set valued function $\Phi_a(B,x) = \{\theta: g(x,a,\theta) \in B\}$ and for $\psi \subset \Theta$ define $\Phi_a^{-1}(\psi,B) = \{x \in \Omega: \Phi_a(B,x) = \psi\}$. Then $\{\Phi_a^{-1}(\psi,B): \psi \subset \Theta\}$ is a finite partition of Ω for each $a \in A$ and $B \subset R^N$. Using $P[Y_{n+1}=\theta|X_n=x,a_n=a] = P_{\theta,a}x$, it follows from [15] that for each $\psi \subset \Theta$,

$$(1) \qquad q(B|x,a) = \sum_{\theta\in\psi} P_{\theta,a}x \quad \text{for} \quad x \in \Phi_a^{-1}(\psi,B).$$

The next theorem provides a formula for the loss function which is convenient for machine implementation. Since the theorem demonstrates that the loss function is piecewise linear, it follows from Theorem 2 that the observed process is a piecewise linear dynamic program.

Theorem 3. Suppose $v(x) = v_i \cdot x$ for $x \in B_i$ with $P_v = \{B_1, B_2, \ldots, B_m\}$ a partition of Ω. Then

$$(2) \qquad h(x,a,v) = [r_a + \beta \sum_i v_i \sum_{\theta \in \psi_i} P_{\theta,a}] \cdot x \text{ for } x \in \bigcap_{i=1}^{m} \Phi_a^{-1}(\psi_i, B_i)$$

where $\bigcap_{i=1}^{m} \Phi_a^{-1}(\psi_i, B_i)$ is a cell in the partition P of Ω defined by

$$P = \bigtimes_{i=1}^{m} \{\Phi_a^{-1}(\psi, B_i) : \psi \subset \Theta\}.$$

In other words, a partially observable Markov decision process is a piecewise linear (closed) dynamic program.

Proof: First observe that from (1),

$$\int_B yq(dy|x,a) = \sum_{\theta \in \psi} g(x,a,\theta) P_{\theta,a} x = \sum_{\theta \in \psi} P_{\theta,a} x \quad \text{for} \quad x \in \Phi_a^{-1}(\psi, B).$$

Consequently, (2) follows by substituting the above into

$$\begin{aligned} h(x,a,v) &= r_a \cdot x + \beta \int_\Omega v(y) q(dy|x,a) \\ &= r_a \cdot x + \beta \sum_{i=1}^{m} v_i \int_{B_i} yq(dy|x,a). \end{aligned}$$

That $\{\Phi_a^{-1}(\psi, B_i) : \psi \subset \Theta\}$ is a partition was noted in the discussion preceding (1). This completes the proof. (Q.E.D.)

4.3. A stochastic inventory model

Let x be the inventory level at the beginning of a period, a be the inventory level immediate after producing the goods, that is, $a - x$ is the amount produced and s the amount of demand with the distribution function $F(\cdot)$. A function v on Ω is called K-convex if there exists a constant value K such that $K + v(\alpha x_1 + (1-\alpha)x_2) \leq \alpha v(x_1) + (1-\alpha) v(x_2)$ for any $x_1, x_2 \in \Omega$, $\alpha \in (0,1)$. Note that a 0-convex function is convex in the ordinary definition. As a matter of fact, K can be interpreted as a set up cost, which will be seen in the following paragraph. It is easy to show that if v is K-convex,

so is $\int_0^\infty v(a-s)dF(s)$ and that if v_1 and v_2 are K-convex, so is $v_1 + \beta v_1$ for $\beta > 0$. Define the loss function

$$h(x,a,v) = \begin{cases} K + c(x,a) + \beta\int_0^\infty v(a-s)dF(s) & \text{if } a > x \\ c(x,0) + \beta\int_0^\infty v(x-s)dF(s) & \text{if } a = x \end{cases}$$

where $c(x,a)$ is the sum of the expected inventory cost and the expected shortage cost. Assume that $c(x,a)$ is convex in x for each a. Take V' as the set of K-convex functions. From the properties of a K-convex function mentioned above $h(x,a,v)$ is K-convex in x for each a whenever v is K-convex. Take Δ' as the set of piecewise constant policies. It is well known (see Hadley and Whiten [9]) that in such an inventory model with a set up cost the $(S-s)$ policy is optimal, provided $c(x,a)$ is convex in x for each a. Such $(S-s)$ policy is certainly piecewise constant because $\delta(x) = S$ for $x < s$, and $\delta(x) = 0$ for $x \geq s$. This implies that if $v \in V'$ there exists a policy $\delta \in \Delta'$ such that $H_* v = H_\delta v$. Furthermore, if $\delta \in \Delta'$, then

$$H_\delta v = h(x,\delta(x),v) = \begin{cases} h(x,S,v) & \text{for } x \leq s, \\ h(x,0,v) & \text{for } x > s. \end{cases}$$

Therefore, $H_\delta v \in V'$ for $v \in V'$ and $\delta \in \Delta'$. This concludes that the stochastic inventory model is piecewise (closed) with V' as the set of K-convex functions and Δ' as the set of piecewise constant policies.

Acknowledgement

The auther would like to express his gratitude to an anonymous referee for his helpful comments and suggestions. This research was partially supported by the Nanzan University Research Grant (1983). Also, special thanks go to Professor Y. Iihara, Nanzan University, and Professor S. Brumelle, University of British Columbia, for their helpful comments and encouragements.

References

[1] Bellman, R.: *Dynamic Programming*, Princeton University Press, Princeton, N.J., 1957.

[2] Blackwell, D.: Discounted Dynamic Programming, *Annals of Mathematical Statistics*, Vol.36 No.1 (1965), 226-235.

[3] Brumelle, S. L. and Sawaki, K.: Generalized Policy Improvement for Simple Dynamic Programs, Working Paper 546, Faculty of Commerce, University of British Columbia, Vancouver (1978).

[4] Denardo, E. V.: Contraction Mapping in the Theory Underlying Dynamic Programming, *SIAM Review* Vol.9 No.2 (1967), 165-177.

[5] Denardo, E. V. and Rothblum, U. G.: Affine Dynamic Programming, *Dynamic Programming and Its Applications*, (ed. M. L. Puterman), Academic Press (1978), 255-267.

[6] Denardo, E. V. and Rothblum, U. G., Affine Structure and Invariant Policies for Dynamic Programs, *Mathematics of Operations Research*, Vol.8 No.3 (1983), 342-365.

[7] Dynkin, E. B: Controlled Random Sequences, *Theory of Probability and Its Applications*, Vol.X (1965), 1-14.

[8] Fox, B. L.: Finite-State Approximations to Denumerable-State Dynamic Programs, *Journal of Mathematical Analysis and Applications*, Vol.34 No.3 (1971), 665-670.

[9] Hadley, G. and Whitin, T. M.: *Analysis of Inventory Systems*, Prentice Hall, Englewood Cliffs, N.J., 1963.

[10] Howard, R. A.: *Dynamic Programming and Markov Processes*, Wiley, New York (1960).

[11] Langen, H. J.: Convergence of Dynamic Programming Models, *Mathematics of Operations Research*, Vol.6 No.4 (1981), 493-512.

[12] Larson, R. E.: *State Increment Dynamic Programming*, Elsevier, New York (1968).

[13] Ohno, Katsuhisa, A Unified Approach to Algorithms with Suboptimality Test in Discounted Semi-Markov Decision Processes, *Journal of Operations Research Society of Japan*, Vol.24 No.4 (1981), 296-324.

[14] Puterman, M. L. and Brumelle, S. L.: On the Convergence of Policy Iteration in Stationary Dynamic Programming, *Mathematics of Operations Research*, Vol.4 No.1 (1979), 60-69.

[15] Sawaki, K. and Ichikawa, A.: Optimal Control for Partially Observable Markov Decision Processes Over an Infinite Horizon, *Journal of Operations Research Society of Japan*, Vol.21 No.1 (1978), 1-16.

[16] Sawaki, K.: Piecewise Linear Dynamic Programs with Applications, *Journal of Operations Research Society of Japan*, Vol.23 No.2 (1980), 91-110.

[17] Smallwood, R. D. and Sondik, E. J.: The Optimal Control of Partially Observable Markov Processes over a Finite Horizon, *Operations Research*, Vol.21 No.5 (1973), 1071-1088.

[18] Strauch, R. E.: Negative Dynamic Programming, *Annals of Mathematical Statistics*, Vol.37 No.4 (1966), 871-890.

[19] Whitt, W.: Approximations of Dynamic Programs I, *Mathematics of Operations Research*, Vol.3 No.3 (1978), 231-256.

[20] Whitt, W.: Approximations of Dynamic Programs II, *Mathematics of Operations Research*, Vol.4 No.2 (1979), 176-185.

Katsushige SAWAKI: Faculty of Business Administration, Nanzan University, 18 Yamazato-Cho, Showa-Ku, Nagoya 466 Japan.

Appendix

Initialization of the algorithm for the three special classes

To start the algorithm we must choose some $v_0 \in V'$ such that $v_0 \geq H_* v_0$. In this appendix we show how to find such an initial cost function v_0 for certain classes of CDP. To this aim we need a specification of loss function h. Suppose that $h(x,a,v) = c(x,a) + \beta\int_\Omega v(y)q(dy|x,a)$ as in Markov decision processes. Assume that V' is a subset of V which contains constant functions.

4.1. Piecewise dynamic programs

Define a constant function v_0 by

$$v_0 = \sup_x \min_a \frac{c(x,a)}{1-\beta}$$

whcih reduces to the following inequalities,

$$(1-\beta)v_0 \geqq \min_a c(x,a) \quad \text{for all} \quad x.$$

Hence, $(H_* v_0)(x) = \min_a\{c(x,a) + \beta\int_\Omega v_0\, q(dy|x,a)\} \leq (1-\beta)v_0 + \beta v_0 = v_0$ for all x. Therefore, such a constant function $v_0 \in V'$ satisfies $v_0 \leq H_* v_0$.

4.2. Partially observable Markov decision processes

Since in this class of CDP we have $h(x,a,v) = r_a \cdot x + \beta\int v(y)q(dy|x,a)$, we put

$$v_0 = \min_a \max_{x\in\Omega} \frac{r_a \cdot x}{1-\beta}$$

where $\Omega = \{x=(x_1,\ldots,x_N): x_i \geq 0, \sum_i x_i = 1\}$. $\max_{x\in\Omega} r_a \cdot x \equiv r_a \cdot x^*$ is simply the optimal value of a linear programming problem for each a, $(1-\beta)v_0 = \min_a r_a x^* \equiv \min_a r_a \cdot x$ for all x. Hence $H_* v_0 = \min_a\{r_a \cdot x + \beta v_0\} \leqq (1-\beta)v_0 + \beta v_0 = v_0$.

4.3. A stochastic inventory model

Define

$$v_0 = \frac{1}{1-\beta} \max\{\sup_x \min_a [K + c(x,a)],\ \sup_x c(x, 0)\}.$$

By the same arguments as the above, such v_0 is satisfactory to start the algorithm.