Journal of the Operations Research Society of Japan
Vol. 24, No. 1, March 1981

AN INVERSE CONTROL PROCESS AND AN INVERSE ALLOCATION PROCESS

Seiichi Iwamoto
Hiroshima University

(Received February 5, 1979; Final July 18, 1980)

Abstract An inverse theory of sequential decision processes, including the standard control process and allocation process, is developed. A finite-stage deterministic invertible (main) dynamic program (DP) whose reward functions depend not only on action but also on state is formulated as a sequential decision process. The main DP is transformed into an equivalent inverse DP by an algebraic inversion. The main DP maximizes a generalized total reward, while the inverse DP minimizes a generalized total state. An inverse theorem is established. It characterizes optimal solutions (optimal reward functions and optimal policy) of inverse DP by those of main DP through inverse and composition. The main DP includes a linear equation and quadratic criterion (main) control process on the half-line and a typical multi-stage (main) allocation process. Therefore, the inverse DP generates an inverse control process and an inverse allocation process, respectively. Not solving the recursive equation directly but applying the inverse theorem, optimal solutions of both inverse processes are easily calculated by use of those of the corresponding main processes.

1. Introduction

Recently Iwamoto [5], [6], [7], [8], [9] has developed an inverse theory of dynamic programs which is applicable to a number of mathematical programming problems with a *single* constraint-function. As will be shown at the concluding remarks, the well-known linear equation and quadratic criterion control problem and the multi-stage allocation problem are transformed into equivalent mathematical programming problems with *no* constraint-function and *multiple* constraint-functions, respectively. Therefore, the inverse theory is not applicable to control and allocation problems. Furthermore, it makes no doubt that both are most interesting and typical problems which are formulated as sequential decision processes (see Bellman [1, Chap 1], [2, p.116], [3, p.329] and [4, p.10]). These two facts motivate the continuing interests in obtaining a "further" inverse theory for a class of sequential decision processes including control and allocation processes.

This paper deals with an inversion of finite-stage deterministic dynamic programs (DP's) on one-dimensional state space whose n-th reward function depends on state and action. These DP's are not included in those of [5], [6], [7], [8], [9]. The inverse theory is applied to a linear equation, quadratic criterion and finite-horizon control process on the state space $[0, \infty)$ and to the well-known multi-stage allocation process. This generates two new processes, i.e., an inverse control process and an inverse allocation process. As far as the author knows, both processes have never been discussed elsewhere.

Thus we have established a kind of duality theory for DP's, which is different from Bellman's duality (upper and lower bounds), quasilinearization and inverse problem [2], [4].

In §2, specifying seven components, we formulate a (main) DP as a sequential decision process. The recursive formula for the main DP is obtained. In §3, for the invertible DP, we specify an inverse DP by the components of the main DP. The recursive formula for the inverse DP is also obtained. This equation is not so well-known as the usual equations in [1]. In §4, we establish an inverse theorem between main and inverse DP's. The optimal reward functions of the inverse DP are inverse functions to those of the main DP. The main result is to apply the inverse theorem to a linear equation and quadratic criterion deterministic control process on state space $[0, \infty)$ (in §5) and to a multi-stage allocation process (in §6). Each process together with its inverse process is solved analytically.

2. Main dynamic program

Let R and S be two intervals of one-dimensional Euclidean space R^1. Then note that if $f : S \to R$ is onto strictly increasing function, then it is continuous and there exists the inverse function $f^{-1} : R \to S$ which is onto strictly increasing. Therefore such an f yields a homeomorphism from S onto R.

A *dynamic program (DP)* $\mathcal{D}$ is specified by an ordered seven-tuple (Opt, $\{S_n\}_1^{N+1}$, $\{R_n\}_1^{N+1}$, $\{A_n\}_1^N$, $\{f_n\}_1^N$, k, $\{T_n\}_1^N$), where

(i) N is a positive integer, the *number of stages*.

(ii) S_n is an interval of R^1, the *n-th state space*. This element s_n is called the *n-th state*.

(iii) R_n is an interval of R^1, the *n-th reward space*. This element r_n is called the *n-th reward*.

(iv) A_n is a non-empty subset of the p_n-dimensional Euclidean space R^{p_n}, the *n-th action space*. Further there corresponds for each n-th state $s_n \in S_n$ a nonempty subset $A_n(s_n)$ of A_n, the *n-th action space at state* s_n. This element a_n is called the *n-th action* available at state s_n. We usually write $A_n(.) : S_n \to 2^{A_n}$, where 2^A denotes the set of all nonempty subsets of the set A. It will be clear from the context whether A_n is considered the set or the point-to-set valued mapping. We define the graph of the mapping $A_n(.) : S_n \to 2^{A_n}$ by

$$\text{graph}(A_n) = \{(s_n, a_n) \mid a_n \in A_n(s_n),\ s_n \in S_n\} \subset S_n \times A_n.$$

(v) f_n : $\text{graph}(A_n) \times R_{n+1} \to R_n$ is an onto continuous function such that each $f_n(s_n,a_n;.)$ $((s_n,a_n) \in \text{graph}(A_n))$ is strictly increasing, the *n-th reward function*.

(vi) $k : S_{N+1} \to R_{N+1}$ is an onto strictly increasing function, the *terminal reward function*.

(vii) $T_n : S_n \times A_n \to R^1$ is a continuous function such that its restriction $T_n|\text{graph}(A_n)$ is a function from $\text{graph}(A_n)$ to S_{n+1} and that each $T_n(.;a_n)$ $(a_n \in A_n)$ is strictly increasing, the *n-th state transformation*.

(viii) Opt is either Max or Min, the *optimizer*. According as Opt = Max or Min, it *represents* the optimization (maximization or minimization) problem :

(2.1) $$\text{Optimize}\quad f_1(s_1,a_1;f_2(s_2,a_2;\ \ldots\ ;f_N(s_N,a_N;k(s_{N+1}))\ \ldots))$$

$$\text{subject to}\quad \text{(i)}\ T_n(s_n;a_n) = s_{n+1} \qquad 1 \leq n \leq N$$

(2.2) $$\text{(ii)}\ a_n \in A_n(s_n) \qquad 1 \leq n \leq N.$$

We call the DP $\mathcal{D}$ the *main DP*. Let us now define the (N-n+1)-*subproblem* of (2.1), (2.2) by the problem :

(2.3) $$\text{Optimize}\quad f_n(s_n,a_n;\ \ldots\ ;f_N(s_N,a_N;k(s_{N+1}))\ \ldots)$$

$$\text{subject to}\quad \text{(i)}\ T_m(s_m;a_m) = s_{m+1} \qquad n \leq m \leq N$$

(2.4) $$\text{(ii)}\ a_m \in A_m(s_m) \qquad n \leq m \leq N,$$

where $s_n \in S_n$, $1 \leq n \leq N$. Let $u^{N-n+1}(s_n)$ be the optimum value of (2.3), (2.4). Further we define $u^0(s_{N+1})$ by

$$u^0(s_{N+1}) = k(s_{N+1}) \qquad s_{N+1} \in S_{N+1}.$$

The function $u^{N-n+1} : S_n \to R_n$ is called the (N-n+1)-*st optimal reward function* of $\mathcal{D}$. Thus the functions $\{u^0, u^1, \ldots, u^N\}$ are called the *optimal reward functions* of $\mathcal{D}$. We have the recursive equation between two adjacent optimal reward functions.

Theorem 1. (RECURSIVE FORMULA FOR $\mathcal{D}$)

$$u^{N-n+1}(s_n) = \underset{a_n \varepsilon A_n(s_n)}{\mathrm{Opt}} \; f_n(s_n,a_n;u^{N-n}(T_n(s_n;a_n))) \quad s_n \varepsilon S_n, \; 1 \leqq n \leqq N \tag{2.5}$$

$$u^0(s_{N+1}) = k(s_{N+1}) \qquad s_{N+1} \varepsilon S_{N+1}$$

Proof: Easy.

3. Inverse dynamic program

For a two-variable function $h : A \times B \longrightarrow C$ we define two one-variable functions $h^a : B \longrightarrow C$ and $h_b : A \longrightarrow C$ by

$$h^a(b) = h(a;b), \qquad h_b(a) = h(a;b),$$

respectively. The main DP $\mathcal{D} = (\mathrm{Opt}, \{S_n\}_1^{N+1}, \{R_n\}_1^{N+1}, \{A_n\}_1^N, \{f_n\}_1^N, k, \{T_n\}_1^N)$ is called *invertible* if it has onto strictly increasing optimal reward functions $\{u^0, u^1, \ldots, u^N\}$. An *inverse* $\mathcal{D}^{-1}$ to the invertible main DP $\mathcal{D}$ is specified by the following ordered seven-tuple :

$$\mathcal{D}^{-1} = (\overline{\mathrm{Opt}}, \{R_n\}_1^{N+1}, \{S_n\}_1^{N+1}, \{B_n\}_1^N, \{g_n\}_.^N, \ell, \{U_n\}_1^N)$$

where

(i) $\overline{\mathrm{Opt}} = \mathrm{Min}$ if $\mathrm{Opt} = \mathrm{Max}$
$\quad = \mathrm{Max}$ if $\mathrm{Opt} = \mathrm{Min}$

(ii) $B_n = S_n \times A_n$
$B_n(r_n) = \{(s_n, a_n) \mid s_n = (u^{N-n+1})^{-1}(r_n),\ a_n \varepsilon A_n(s_n),\ (f_n^{(s_n,a_n)})^{-1}(r_n) \varepsilon R_{n+1}\}$

(iii) $g_n(a_n;s_{n+1}) = (T_{na_n})^{-1}(s_{n+1})$

(iv) $\ell(r_{N+1}) = k^{-1}(s_{N+1})$

(v) $U_n(r_n;s_n,a_n) = (f_n^{(s_n,a_n)})^{-1}(r_n)$.

We call $\mathcal{D}^{-1}$ the *inverse DP*. It represents the problem :

$$\overline{\mathrm{Optimize}} \quad g_1(a_1;g_2(a_2; \ldots ;g_N(a_N;\ell(r_{N+1})) \ldots)) \tag{3.1}$$

(3.2) subject to (i) $U_n(r_n;s_n;a_n) = r_{n+1}$ $1 \leqq n \leqq N$

(ii) $(s_n,a_n) \in B_n(r_n)$ $1 \leqq n \leqq N.$

Note that the objective function (3.1) does not depend on the sequence of states $\{r_n\}_1^N$. On the other hand, the n-th action at the n-th state r_n for the inverse DP $\mathcal{D}^{-1}$ is formally considered as a direct product (s_n,a_n). However, the first action s_n has no freedom to be selected. That is, from the definition of $B_n(r_n)$, it is uniquely determined by the relation $s_n = (u^{N-n+1})^{-1}(r_n)$. This notion is not applied to the previous "inverse DP" in [5], [6], [8] and [9]. Only the second action a_n is to be controlled so as to $\overline{\text{opt}}$imize (3.1).

We have the following economic interpretations. The main DP $\mathcal{D}$ is, given an initial state s_1, to choose the sequence of actions $\{a_n\}_1^N$ so as to maximize a generalized total reward r_1, while the inverse DP $\mathcal{D}^{-1}$ is, given an initial reward r_1, to choose the sequence of actions $\{s_n,a_n\}_1^N$ so as to minimize a generalized total state s_1. Here state corresponds to cost, manpower, energy, position (in a negative sense), post (in a negative sense), and others. These are compatible with money in a sense. Both the interpretations above for $\mathcal{D}$ and $\mathcal{D}^{-1}$ follow directly from forward and backward recursive relations

$$\mathcal{D} : \begin{cases} T_n(s_n;a_n) = s_{n+1}, \ a_n \in A_n(s_n) & 1 \leqq n \leqq N \\ k(s_{N+1}) = r_{N+1} & \\ f_n(s_n,a_n;r_{n+1}) = r_n & N \geqq n \geqq 1 \end{cases}$$

and

$$\mathcal{D}^{-1} : \begin{cases} U_n(r_n;s_n,a_n) = r_{n+1}, \ (s_n,a_n) \in B_n(r_n) & 1 \leqq n \leqq N \\ \ell(r_{N+1}) = s_{N+1} & \\ g_n(a_n;s_{n+1}) = s_n & N \geqq n \geqq 1 \end{cases}$$

respectively, where $N \geqq n \geqq 1$ means that the time n runs backwards N+1, N, ..., 2, 1.

The problem (3.1), (3.2) may also be expressed in terms of the components of $\mathcal{D}$ as follows :

(3.3) $\overline{\text{Opt}}$imize $(T_{1a_1})^{-1} \circ (T_{2a_2})^{-1} \circ \cdots \circ (T_{Na_N})^{-1} \circ k^{-1}(r_{N+1})$

subject to (i) $(f_n^{(s_n,a_n)})^{-1}(r_n) = r_{n+1}$ $1 \leqq n \leqq N$

(ii) $a_n \in A_n(s_n), \ s_n = (u^{N-n+1})^{-1}(r_n)$ $1 \leqq n \leqq N$

(iii) $(f_n^{(s_n,a_n)})^{-1}(r_n) \in R_{n+1}$ $1 \leqq n \leqq N.$

Similarly, the (N-n+1)-*subproblem* of (3.1), (3.2) is defined by the problem :

$$(3.5)\qquad \overline{\text{Optimize}}\ \ g_n(a_n; \ldots ; g_N(a_N;\ell(r_{N+1}))\ldots)$$

$$\text{subject to}\quad \text{(i)}\ \ U_m(r_m;s_m,a_m) = r_{m+1} \qquad n \leq m \leq N$$

$$\text{(ii)}\ \ (s_m,a_m) \in B_m(r_m) \qquad n \leq m \leq N,$$

where $r_n \in R_n$, $1 \leq n \leq N+1$. Let $v^{N-n+1}(r_n)$ be the $\overline{\text{optimum}}$ value of (3.5), (3.6). Further, we define $v^0(r_{N+1})$ by

$$v^0(r_{N+1}) = \ell(r_{N+1}) \qquad r_{N+1} \in R_{N+1}.$$

The function $v^{N-n+1} : R_n \longrightarrow S_n$ is called the (N-n+1)-*st optimal reward function* of $\mathcal{D}^{-1}$. Thus the functions $\{v^0, v^1, \ldots, v^N\}$ are called the *optimal reward functions* of $\mathcal{D}^{-1}$. The recursive equation becomes as follows :

Theorem 2. (RECURSIVE FORMULA FOR $\mathcal{D}^{-1}$)

$$(3.7)\qquad v^{N-n+1}(r_n) = \overline{\underset{\substack{a_n \in A_n((u^{N-n+1})^{-1}(r_n)) \\ U_n(r_n;(u^{N-n+1})^{-1}(r_n),a_n) \in R_{n+1}}}{\text{Opt}}}\ g_n(a_n; v^{N-n}(U_n(r_n;(u^{N-n+1})^{-1}(r_n),a_n))) \qquad r_n \in R_n,\ 1 \leq n \leq N$$

$$v^0(r_{N+1}) = \ell(r_{N+1}) \qquad r_{N+1} \in R_{N+1}$$

Proof: Easy.

4. Inverse theorem

In order to state an inverse theorem describing the relationship between the main DP $\mathcal{D}$ and the inverse DP $\mathcal{D}^{-1}$, let us now define an optimal policy for each DP. A *policy* of $\mathcal{D}$ is a sequence $\{\pi_1, \pi_2, \ldots , \pi_N\}$ such that the mapping $\pi_n : S_n \longrightarrow A_n$ has the property $\pi_n(s_n) \in A_n(s_n)$ for $s_n \in S_n$, $1 \leq n \leq N$. A policy $\{\pi_1^*, \pi_2^*, \ldots , \pi_N^*\}$ is *optimal* for $\mathcal{D}$ if for each $s_n \in S_n$, $1 \leq n \leq N$ $\pi_n^*(s_n)$ attains the optimum value of (2.5).

On the other hand, a *policy* of $\mathcal{D}^{-1}$ is a sequence $\{\sigma_1, \sigma_2, \ldots , \sigma_N\}$ such that the mapping $\sigma_n : R_n \longrightarrow A_n$ has the property $\sigma_n(r_n) \in A_n((u^{N-n+1})^{-1}(r_n))$ and $U_n(r_n;(u^{N-n+1})^{-1}(r_n),\sigma_n(r_n)) \in R_{n+1}$, $1 \leq n \leq N$. A policy $\{\hat{\sigma}_1, \hat{\sigma}_2, \ldots , \hat{\sigma}_N\}$ is *optimal* for $\mathcal{D}^{-1}$ if for each $r_n \in R_n$, $1 \leq n \leq N$ $\hat{\sigma}_n(r_n)$ attains the

Optimum value of (3.7).

Our fundamental result is an inverse theorem in dynamic programming. The differences between the following INVERSE THEOREM and inverse theorems in [5], [6], [8], [9] and [10] are as follows. First, this paper, [5], [9] and [10] discuss sequential decision processes, while [6] and [8] do mathematical programming problems. Second, our theorem treats the case where the objective function is dependent on state sequence, while the others do the case where it is not. Finally, our theorem, as will be shown, is only applicable to control and allocation processes. The others are not. Furthermore both processes have been considered as typical sequential decision processes ([1], [2], [3] and [4]). These are main reasons why we are willing to establish an inverse theory of sequential decision process and apply it to both processes.

Theorem 3. (INVERSE THEOREM) (i) If the main DP $\mathcal{D}$ has onto strictly increasing optimal reward functions $\{u^0, u^1, \ldots, u^N\}$ and an optimal policy $\{\pi_1^*, \pi_2^*, \ldots, \pi_N^*\}$, then the inverse DP $\mathcal{D}^{-1}$ has onto strictly increasing optimal reward functions $\{(u^0)^{-1}, (u^1)^{-1}, \ldots, (u^N)^{-1}\}$ and an optimal policy $\{\pi_1^*\circ(u^N)^{-1}, \pi_2^*\circ(u^{N-1})^{-1}, \ldots, \pi_N^*\circ(u^1)^{-1}\}$.

(ii) Let $\{u^0, u^1, \ldots, u^N\}$ be onto strictly increasing optimal reward functions of the main DP $\mathcal{D}$. If the inverse DP $\mathcal{D}^{-1}$ has onto strictly increasing optimal reward functions $\{v^0, v^1, \ldots, v^N\}$ and an optimal policy $\{\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_N\}$, then it holds that

$$(v^{N-n+1})^{-1} = u^{N-n+1} \qquad 1 \leqq n \leqq N+1.$$

Furthermore, the main DP $\mathcal{D}$ has an optimal policy $\{\hat{\sigma}_1\circ(v^N)^{-1}, \hat{\sigma}_2\circ(v^{N-1})^{-1}, \ldots, \hat{\sigma}_N\circ(v^1)^{-1}\}$.

Proof: The proof is by induction on n. It suffices to prove the theorem only for the case Opt = Max. (i) Let the main DP have onto strictly increasing optimal reward functions $\{u^n\}_0^N$ and an optimal policy $\{\pi_n^*\}_1^N$. Then we have

$$(4.1) \qquad u^{N-n+1}(s_n) = \mathop{\mathrm{Max}}_{a_n \in A_n(s_n)} f_n(s_n,a_n;u^{N-n}(T_n(s_n;a_n)))$$
$$= f_n(s_n,\pi_n^*(s_n);u^{N-n}(T_n(s_n:\pi_n^*(s_n)))) \qquad s_n \in S_n, \quad 1 \leqq n \leqq N$$
$$u^0(s_{N+1}) = k(s_{N+1}) \qquad s_{N+1} \in S_{N+1}.$$

First, from the definition, we get

$$v^0(r_{N+1}) = (u^0)^{-1}(r_{N+1}) \qquad r_{N+1} \in R_{N+1}.$$

Second, let us consider the case n = N of (2.5). Fix $s_N \in S_N$. Let $u^1(s_N) = r_N$, $T_N(s_N;\pi_N^*(s_N)) = s_{N+1}$ and $k(s_{N+1}) = r_{N+1}$. Then $r_N \in R_N$ and $r_N = f_N(s_N, \pi_N^*(s_N))\ (r_{N+1})$. Therefore it holds that

$$\begin{aligned} s_N &= (u^1)^{-1}(r_N) \\ r_{N+1} &= (f_N^{(s_N,\pi_N^*(s_N))})^{-1}(r_N) \\ &= U_N(r_N;s_N,\pi_N^*(s_N)) \\ s_N &= (T_{N\pi_N^*(s_N)})^{-1}(s_{N+1}) \\ &= g_N(\pi_N^*(s_N);s_{N+1}) \\ &= g_N(\pi_N^*(s_N);\ell(r_{N+1})). \end{aligned} \tag{4.2}$$

Let us define $w^1(r_N)$ by

$$w^1(r_N) = \operatorname*{Inf}_{\substack{a_N \in A_N((u^1)^{-1}(r_N)) \\ U_N(r_N;s_N,a_N) = r_{N+1} \\ s_N = (u^1)^{-1}(r_N)}} g_N(a_N;\ell(r_{N+1})).$$

Hence we get $w^1(r_N) \leqq s_N$. If $w^1(r_N) < s_N$, then there exists an $\hat{a}_N \in A_N((u^1)^{-1}(r_N))$ such that

$$g_N(\hat{a}_N;\ell(\hat{r}_{N+1})) < s_N$$

where $\hat{r}_{N+1} = U_N(r_N;s_N,\hat{a}_N)$, $s_N = (u^1)^{-1}(r_N)$. Letting $\ell(\hat{r}_{N+1}) = \hat{s}_{N+1}$, we in turn obtain

$$\hat{s}_{N+1} < T_N(s_N;\hat{a}_N) \tag{4.3}$$

$$f_N(s_N,\hat{a}_N;\hat{r}_{N+1}) = r_N \tag{4.4}$$

and

$$u^1(s_N) = r_N. \tag{4.5}$$

Therefore the strict increasingness of $f_N(s_N,\hat{a}_N:.)$ and k, $\hat{a}_N \in A_N(s_N)$, and (4.3), (4.4) imply that

$$f_N(s_N,\hat{a}_N;k(T_N(s_N;\hat{a}_N))) > r_N.$$

This contradicts (4.5). Hence we have $w^1(r_N) = s_N$. Since $s_N \in S_N$ is arbitrary, we get $w^1 = (u^1)^{-1}$. This equality together with (4.2) also implies that $(u^1)^{-1}(\pi_N^*(s_N))$ attains the minimum of (3.7) for n = N. Finally we get $v^1 = (u^1)^{-1}$.

In general, it is inductively shown that

$$v^{N-n+1} = (u^{N-n+1})^{-1} \qquad n = N-1,\ N-2,\ \dots,\ 1.$$

and that $(u^{N-n+1})^{-1}(\pi_n^*(s_n))$ attains the minimum of (3.7) for n = N-1, N-2, ... , 1. This completes the proof of (i).

(ii) Let the main DP $\mathcal{D}$ and the inverse DP $\mathcal{D}^{-1}$ have onto strictly increasing optimal reward functions $\{u^n\}_0^N$ and $\{v^n\}_0^N$, respectively. Then they satisfy the recursive formulas (2.5), (3.7), respectively. From the analysis in (i), it turns out that the functions $\{(u^n)^{-1}\}_0^N$ also satisfy (3.7) and $(u^0)^{-1} = v^0$. This implies that $(u^{N-n+1})^{-1} = v^{N-n+1}$ namely $(v^{N-n+1})^{-1} = u^{N-n+1}$ for $1 \leqq n \leqq N$. The similar argument as in (i) with the roles of $\{u^n\}_0^N$ and $\{v^n\}_0^N$ exchanged leads the equality

$$v^{N-n+1}(r_n) = \underset{a_n \in A_n((u^{N-n+1})^{-1}(r_n))}{\text{Min}} g_n(a_n;v^{N-n}(U_n(r_n;(u^{N-n+1})^{-1}(r_n),a_n)))$$
$$= g_n(\hat{\sigma}_n(r_n);v^{N-n}(U_n(r_n:(u^{N-n+1})^{-1}(r_n),\hat{\sigma}_n(r_n))))$$

to the equality

$$u^{N-n+1}(s_n) = \underset{a_n \in A_n(s_n)}{\text{Max}} f_n(s_n,a_n;u^{N-n}(T_n(s_n;a_n)))$$
$$= f_n(s_n,\hat{\sigma}_n\circ(v^{N-n+1})^{-1}(s_n);u^{N-n}(T_n(s_n;\hat{\sigma}_n\circ(v^{N-n+1})^{-1}(s_n)))).$$

Therefore the main DP $\mathcal{D}$ has an optimal policy $\{\hat{\sigma}_n\circ(v^{N-n+1})^{-1}\}_1^N$. This completes the proof of (ii).

The INVERSE THEOREM gives us useful informations on one DP, provided that the optimal behavior of the other DP is known. Let $\mathcal{D}$ have the desired optimal solutions $\{u^n\}$, $\{\pi_n^*\}$. Then the total reward from state r_1 for $\mathcal{D}^{-1}$ is $(u^N)^{-1}(r_1)$. Further, an optimal action at state r_n for $\mathcal{D}^{-1}$ is $\hat{a}_n = \pi_n^*(s_n)$, where $s_n = (u^{N-n+1})^{-1}(r_n)$. Conversely, let $\mathcal{D}$ and $\mathcal{D}^{-1}$ have the desired optimal solutions $\{u^n\}$, $\{\pi_n^*\}$ and $\{v^n\}$, $\{\hat{\sigma}_n\}$, respectively. Then u^{N-n+1} is the inverse function of

v^{N-n+1} and vice versa. Further, an optimal action at state s_n for $\mathcal{D}$ is $a_n^* = \hat{\sigma}_n(r_n)$, where $r_n = (v^{N-n+1})^{-1}(s_n)$, as well as $\hat{a}_n = \pi_n^*(s_n)$.

5. Inverse control process

Throughout this section let $b > 0$ and N be a positive integer. First we consider the following linear equation and quadratic criterion, finite-stage and deterministic control process (see [2, p.116])

$$\text{Minimize} \quad \sum_{n=1}^{N} (x_n^2 + y_n^2) + x_{N+1}^2$$

$$\text{subject to} \quad \text{(i)} \quad bx_n + y_n = x_{n+1} \qquad 1 \leqq n \leqq N$$

$$\text{(ii)} \quad -\infty < y_n < -\infty \qquad 1 \leqq n \leqq N, \quad x_1 = c.$$

It is well-known that this problem has a quadratic minimum value $u^N(c) = p_N c^2$, where p_N is determined by (5.2) which will be shown later. Note that the function $u^N : (-\infty,\infty) \rightarrow [0,\infty)$ is not strictly increasing on $(-\infty,0)$.

Therefore, we further *assume* the condition $0 \leqq x_n < \infty$ for $1 \leqq n \leqq N$. This restricted problem is written in terms of state s_n and action a_n as follows :

$$\text{Minimize} \quad \sum_{n=1}^{N} (s_n^2 + a_n^2) + s_{N+1}^2$$

$$\text{subject to} \quad \text{(i)} \quad bs_n + a_n = s_{n+1}, \quad s_n \geqq 0 \qquad 1 \leqq n \leqq N$$

$$s_{N+1} \geqq 0$$

$$\text{(ii)} \quad -\infty < a_n < \infty \quad 1 \leqq n \leqq N, \quad s_1 = c.$$

Consider a simple inventry model with the following meanings :

$1 - b$ = the deterioration rate of the goods, $0 \leqq b < 1$

s_n = the stock level at the n-th period subtracted by constant demand

a_n = the production quantity at the n-th period.

Then the interpretations for system dynamics and objective function are straightforward.

This problem is represented by an N-stage main DP $\mathcal{D}$ = (Min, $\{S_n\}_1^{N+1}$, $\{R_n\}_1^{N+1}$, $\{A_n\}_1^N$, $\{f_n\}_1^N$, k, $\{T_n\}_1^N$), where

$$S_n = R_n = [0,\infty), \qquad A_n = (-\infty,\infty)$$

$$A_n(s_n) = [-bs_n,\infty), \qquad f_n(s_n,a_n;r_{n+1}) = s_n^2 + a_n^2 + r_{n+1}$$

$$k(s_{N+1}) = s_{N+1}^2, \qquad T_n(s_n;a_n) = bs_n + a_n.$$

The main DP $\mathcal{D}$ is called the *main control process*. The corresponding recursive formula

$$
(5.1)\quad \begin{aligned} u^{N-n+1}(s_n) &= \underset{a_n \geq -bs_n}{\text{Min}} \; [s_n^2 + a_n^2 + u^{N-n}(bs_n + a_N)] \quad s_n \geq 0, \quad 1 \leq n \leq N \\ u^0(s_{N+1}) &= s_{N+1}^2 \qquad s_{N+1} \geq 0 \end{aligned}
$$

has quadratic optimal reward functions $\{u^0, u^1, \ldots, u^N\}$ and a linear optimal policy $\{\pi_1^*, \pi_2^*, \ldots, \pi_N^*\}$:

$$
u^{N-n+1}(s_n) = p_{N-n+1}s_n^2, \qquad \pi_n^*(s_n) = \alpha_n s_n
$$

where

$$
(5.2)\qquad p_0 = 1, \qquad p_b = 1 + b^2 - \frac{b^2}{1 + p_{n-1}} \qquad 1 \leq n \leq N
$$

$$
\alpha_n = -\frac{p_{N-n}}{1 + p_{N-n}}\, b \qquad 1 \leq n \leq N.
$$

Since each $u^{N-n+1} : [0,\infty) \rightarrow [0,\infty)$ is onto strictly increasing, the inverse DP $\mathcal{D}^{-1}$ is specified by the following components :

$$
\begin{aligned}
&\overline{\text{Opt}} = \text{Max}, \quad R_n = S_n = [0,\infty), \quad B_n = [0,\infty)\times(-\infty,\infty) \\
&B_n(r_n) = \{(s_n,a_n) \mid -bs_n \leq a_n, \quad s_n = \sqrt{r_n/p_{N-n+1}}, \quad s_n^2 + a_n^2 \leq r_n\} \\
&g_n(a_n;s_{n+1}) = (-a_n + s_{n+1})/b, \qquad \ell(r_{N+1}) = \sqrt{r_{N+1}} \\
&U_n(r_n;s_n,a_n) = r_n - (s_n^2 + a_n^2).
\end{aligned}
$$

The inverse DP $\mathcal{D}^{-1}$ is called the *inverse control process*. It represents the problem :

$$
\text{Maximize} \quad -\frac{a_1}{b} - \frac{a_2}{b^2} - \cdots - \frac{a_N}{b^N} + \frac{\sqrt{r_{N+1}}}{b^N}
$$

$$
\begin{aligned}
\text{subject to} \quad &\text{(i)} \quad r_n - (s_n^2 + a_n^2) = r_{n+1} \qquad 1 \leq n \leq N \\
&\text{(ii)} \quad s_n = \sqrt{r_n/p_{N-n+1}} \qquad 1 \leq n \leq N \\
&\text{(iii)} \quad s_n^2 + a_n^2 \leq r_n, \quad -bs_n \leq a_n \qquad 1 \leq n \leq N.
\end{aligned}
$$

Then the recursive formula becomes as follows :

$$
(5.3)\qquad v^{N-n+1}(r_n) = \underset{\substack{s_n=\sqrt{r_n/p_{N-n+1}} \\ s_n^2+a_n^2\leq r_n \\ -bs_n\leq a_n}}{\text{Max}} \left[\frac{1}{b}(-a_n + v^{N-n}(r_n - s_n^2 - a_n^2))\right] \qquad r_n \geq 0, \quad 1 \leq n \leq N
$$

$$v^0(r_{N+1}) = \sqrt{r_{N+1}} \qquad r_{N+1} \geqq 0.$$

However, not solving this equation backwards, but applying the INVERSE THEOREM, we have onto strictly increasing optimal reward functions $\{v^0, v^1, \dots, v^N\}$ and an optimal policy $\{\hat{\sigma}_1, \hat{\sigma}_2, \dots, \hat{\sigma}_N\}$:

$$v^{N-n+1}(r_n) = (u^{N-n+1})^{-1}(r_n) = \sqrt{r_n/p_{N-n+1}}$$

$$\hat{\sigma}_n(r_n) = \pi_n^* \circ (u^{N-n+1})^{-1}(r_n) = (\alpha_n/\sqrt{p_{N-n+1}})\sqrt{r_n}.$$

Of course these optimal solutions are obtained by solving directly the recursive equation (5.3). The reader will find that solving (5.3) is more difficult than (5.1). Therefore the application of the INVERSE THEOREM is more effective than solving (5.3).

In particular, two-stage main control process $\mathcal{D}$ and its inverse control process $\mathcal{D}^{-1}$ have

$$u^0(s_3) = s_3^{\ 2}$$

$$u^1(s_2) = (s + \tfrac{1}{2}b^2)s_2^{\ 2} \qquad \pi_2^*(s_2) = -\tfrac{1}{2}bs_2$$

$$u^2(s_1) = \frac{2 + \frac{3}{2}b^2 + \frac{1}{2}b^4}{2 + \frac{1}{2}b^2} s_1^{\ 2} \qquad \pi_1^*(s_1) = -\frac{b(1 + \frac{1}{2}b^2)}{2 + \frac{1}{2}b^2} s_1$$

and

$$v^0(r_3) = \sqrt{r_3}$$

$$v^1(r_2) = \frac{1}{\sqrt{1 + \frac{1}{2}b^2}}\sqrt{r_2} \qquad \hat{\sigma}_2(r_2) = \frac{-\frac{1}{2}b}{\sqrt{1 + \frac{1}{2}b^2}}\sqrt{r_2}$$

$$v^2(r_1) = \frac{\sqrt{2 + \frac{1}{2}b^2}}{\sqrt{2 + \frac{3}{2}b^2 + \frac{1}{2}b^4}}\sqrt{r_1} \qquad \hat{\sigma}_1(r_1) = \frac{-b(1 + \frac{1}{2}b^2)}{\sqrt{(2+\frac{1}{2}b^2)(2+\frac{3}{2}b^2+\frac{1}{2}b^4)}}\sqrt{r_1},$$

respectively.

6. Inverse allocation process

Throughout this section let $0 \leqq a < 1$, $0 < b < 1$, $c_1, c_2, c_3, d > 0$ and N be a positive integer. Consider the following typical N-stage allocation problem (see [1, p.44]) :

$$\text{Maximize} \quad \sum_{n=1}^{N} [c_1 a_n{}^d + c_2(s_n - a_n)^d] + c_3 s_{N+1}{}^d$$

$$\text{subject to} \quad \text{(i)} \quad aa_n + b(s_n - a_n) = s_{n+1} \qquad 1 \leqq n \leqq N$$

$$\text{(ii)} \quad 0 \leqq a_n \leqq s_n \qquad 1 \leqq n \leqq N.$$

The economic interpretation is stated in [1, p.4]. This problem is represented by an N-stage main DP $\mathcal{D}$ whose components are specified as follows :

$$\overline{\text{Opt}} = \text{Max}, \quad S_n = R_n = A_n = [0,\infty) \quad A_n(s_n) = [0,\ s_n]$$

$$f_n(s_n, a_n; r_{n+1}) = c_1 a_n{}^d + c_2(s_n - a_n)^d + r_{n+1}$$

$$k(s_{N+1}) = c_3 s_{N+1}{}^d, \quad T_n(s_n; a_n) = aa_n + b(s_n - a_n).$$

We call $\mathcal{D}$ the *main allocation process*. It is easily shown that the main allocation process $\mathcal{D}$ has onto strictly increasing optimal reward functions $\{u^0, u^1, \ldots, u^N\}$ and an optimal policy $\{\pi_1^*, \pi_2^*, \ldots, \pi_N^*\}$:

$$u^{N-n+1}(s_n) = p_{N-n+1} s_n{}^d, \qquad \pi_n^*(s_n) = \alpha_n s_n$$

where

$$p_0 = c_3$$

$$(6.1) \quad p_{N-n+1} = \underset{0 \leqq x \leqq 1}{\text{Max}} \quad [c_1 x^d + c_2(1-x)^d + p_{N-n}(ax + b(1-x))^d] \qquad 1 \leqq n \leqq N$$

and α_n is the value of x which attains the maximum of (6.1).

On the other hand, the components of the inverse DP $\mathcal{D}^{-1}$, called the *inverse allocation process*, become as follows :

$$\overline{\text{Opt}} = \text{Min}, \quad S_n = R_n = [0,\infty), \quad B_n = [0,\infty)\times[0,\infty)$$

$$B_n(r_n) = \{(s_n, a_n) \mid 0 \leqq a_n \leqq s_n = (r_n/p_{N-n+1})^{1/d},\ c_1 a_n{}^d + c_2(s_n - a_n)^d \leqq r_n\}$$

$$g_n(a_n; s_{n+1}) = (-(a-b)a_n + s_{n+1})/b, \quad \ell(r_{N+1}) = (r_{N+1}/c_3)^{1/d}$$

$$U_n(r_n; s_n, a_n) = r_n - c_1 a_n{}^d - c_2(s_n - a_n)^d.$$

The inverse allocation process $\mathcal{D}^{-1}$ represents the problem :

$$\text{Minimize} \quad -\frac{a-b}{b} a_1 - \frac{a-b}{b^2} a_2 - \cdots - \frac{a-b}{b^N} a_N + \frac{1}{b^N}\left(\frac{r_{N+1}}{c_3}\right)^{1/d}$$

$$\text{subject to} \quad \text{(i)} \quad r_n - c_1 a_n{}^d - c_2(s_n - a_n)^d = r_{n+1} \qquad 1 \leqq n \leqq N$$

(ii) $s_n = (\frac{r_n}{p_{N-n+1}})^{1/d} \qquad 1 \leq n \leq N$

(iii) $0 \leq a_n \leq s_n, \quad c_1 a_n{}^d + c_2(s_n - a_n)^d \leq r_n \qquad 1 \leq n \leq N.$

The corresponding recursive formula becomes

$$(6.2) \quad v^{N-n+1}(r_n) = \operatorname*{Min}_{\#} \left[\frac{1}{b}(-(a-b)a_n + v^{N-n}(r_n - c_1 a_n{}^d - c_2(s_n - a_n)^d))\right] \qquad r_n \geq 0, \quad 1 \leq n \leq N$$

$$\# := \begin{cases} s_n = (r_n/p_{N-n+1})^{1/d} \\ c_1 a_n{}^d + c_2(s_n - a_n)^d \leq r_n \\ 0 \leq a_n \leq s_n \end{cases}$$

$$v^0(r_{N+1}) = (r_{N+1}/c_3)^{1/d} \qquad r_{N+1} \geq 0.$$

The INVERSE THEOREM gives the inverse allocation process $\mathcal{D}^{-1}$ and the following optimal solutions :

$$v^{N-n+1}(r_n) = (u^{N-n+1})^{-1}(r_n) = (r_n/p_{N-n+1})^{1/d}$$

$$\hat{\sigma}_n(r_n) = \pi_n^* \circ (u^{N-n+1})^{-1}(r_n) = (\alpha_n/((p_{N-n+1})^{1/d}) r_n{}^{1/d}.$$

Note that the recursive equation (6.2) has the solution

$$v^{N-n+1}(r_n) = q_{N-n+1} r_n{}^{1/d}, \qquad \hat{\sigma}_n(r_n) = \beta_n r_n{}^{1/d}$$

where

$$q_0 = 1/(c_3{}^{1/d})$$

$$(6.3) \quad q_{N-n+1} = \operatorname*{Min}_{\substack{0 \leq y \leq (\frac{1}{p_{N-n+1}})^{1/d} \\ c_1 y + c_2((\frac{1}{p_{N-n+1}})^{1/d} - y) \leq 1}} \left[-\frac{(a-b)}{b} y + \frac{1}{b} q_{N-n}(1 - c_1 y^d - c_2((\frac{1}{p_{N-n+1}})^{1/d} - y)^d)^{1/d}\right] \qquad 1 \leq n \leq N$$

and β_n is the value of y which attains the minimum of (6.3). Thus we obtain

$$q_{N-n+1} = 1/((p_{N-n+1})^{1/d}), \qquad \beta_n = \alpha_n/((p_{N-n+1})^{1/d}).$$

In particular, for the case a = 0 and d = 1/2, we have

$$p_0 = c_3$$

$$p_{N-n+1} = c_1^2 + (c_2 + \sqrt{b}\,p_{N-n})^2 \qquad 1 \leq n \leq N$$

$$\alpha_n = \frac{c_1^2}{c_1^2 + (c_2 + \sqrt{b}\,p_{N-n})^2} \qquad 1 \leq n \leq N$$

and

$$q_0 = \frac{1}{c_3^2}$$

$$q_{N-n+1} = \frac{1}{c_1^2 + (c_2 + \sqrt{b}/\sqrt{q_{N-n}})^2} \qquad 1 \leq n \leq N$$

$$\beta_n = \frac{c_1^2}{c_1^2 + (c_2 + \sqrt{b}/\sqrt{q_{N-n}})^2}\left(\frac{1}{p_{N-n+1}}\right)^2 \qquad 1 \leq n \leq N,$$

respectively.

Concluding remarks

Eliminating the state variables $s_2, s_3, \ldots, s_{N+1}$, and identifying $s_1, a_1, a_2, \ldots, a_N$ with $c, x_1, x_2, \ldots, x_N$ we transform the problem represented by the main control process in §5 into an equivalent constrained mathematical programming problem :

$$\begin{aligned}
\text{Minimize} \quad & (b^N c + b^{N-1}x_1 + \ldots + bx_{N-1} + x_N)^2 \\
& + (b^{N-1}c + b^{N-2}x_1 + \ldots + bx_{N-2} + x_{N-1})^2 + x_{N-1}^2 \\
& + (b^{N-2}c + b^{N-3}x_1 + \ldots + bx_{N-3} + x_{N-2})^2 + x_{N-2}^2 \\
& \vdots \\
& + (bc + x_1)^2 + x_1^2
\end{aligned}$$

$$\begin{aligned}
\text{subject to} \quad & (1) \quad b^N c + b^{N-1}x_1 + \ldots + bx_{N-1} + x_N \geq 0 \\
& (2) \quad b^{N-1}c + b^{N-2}x_1 + \ldots + bx_{N-2} + x_{N-1} \geq 0 \\
& \vdots \\
& (N) \quad bc + x_1 \geq 0 \\
& (N+1) \quad -\infty < x_i < \infty \qquad 1 \leq i \leq N
\end{aligned}$$

where positive constant b is given and parameter c ranges on half line $[0, \infty)$. Thus we have the following one-parametric, quadratic and multi-constrained problem :

$$\begin{aligned} &\text{Minimize} \quad (x, A(c)x) + 2(b(c), x) + d(c) \\ &\text{subject to} \quad \text{(i)} \quad B(c)x \geqq e(c) \\ &\qquad\qquad\quad \text{(ii)} \quad x \in R^N, \end{aligned}$$

where A(c) is positive definite and B(c) is upper-triangular and nonsingular. Note that the original control process without nonnegativity of state variables represents the equivalent unconditional problem without constraint (i). Similarly, the main allocation process in §6 represents an equivalent, one-parametric, and multi-constrained problem. These problems leave us an open problem of developing a general inverse theory for parametric multi-constrained mathematical programming problems.

The inverse theory has generated a counterpart in dynamic programming problem whose solution is obtained through inverse and composition from the solution of the original dynamic programming problem. Furthermore, the theory generates a new class of dynamic programming problems whose solution is not charactized by the solution of the original porblem. These problems are obtained from the inverse problem by exchanging the constraints $s_n = (u^{N-n+1})^{-1}(r_n)$ $1 \leqq n \leqq N$ for another constraints.

Finally we remark that with appropriate modifications the precèding argument will remain valid for a number of sequential decision processes on the one-dimensional state space.

Acknowledgement

The author would like to express his hearty thanks to anonymous referees whose helpful suggestions and valuable comments have improved the manuscript.

Reference

[1] Bellman, R.: *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

[2] Bellman, R.: *Introduction to the Mathematical Theory of Control Processes. Vol. I, Linear Equations and Quadratic Criteria.* Academic Press, New York, 1967.

[3] Bellman, R.: *Introduction to Matrix Analysis. 2nd Ed.* McGraw-Hill, New York, 1970.

[4] Bellman, R.: *Introduction to the Mathematical Theory of Control Processes. Vol. II, Nonlinear Processes.* Academic Press, New York, 1971.

[5] Iwamoto, S.: Inverse Dynamic Programming. *Memoirs of the Faculty of Science. Kyushu University. Series A. Mathematics.* Vol. 30, No. 1 (1976), 24-42.

[6] Iwamoto, S.: Inverse Theorem in Dynamic Programming I. *Journal of Mathematical Analysis and Applications.* Vol. 58, No. 1(1977), 113-134.

[7] Iwamoto, S.: Inverse Theorem in Dynamic Programming II. *Journal of Mathematical Analysis and Applications.* Vol. 58, No. 2(1977), 247-279.

[8] Iwamoto, S.: Inverse Theorem in Dynamic Programming III. *Journal of Mathematical Analysis and Applications.* Vol. 58, No. 3(1977), 439-448.

[9] Iwamoto, S.: Inverse Dynamic Programming II. *Memoirs of the Faculty of Science. Kyushu University. Series A. Mathematics.* Vol. 31, No. 1 (1977), 24-44.

[10] Iwamoto, S.: An Inverse Theorem Between Main and Inverse Dynamic Programming : Infinite-Stage Case. *Dynamic Programming and Its Applications; Proceedings of the International Conference on Dynamic Programming (ed. M. L. Puterman).* Academic Press, New York, 1978, 319-334.

Seiichi IWAMOTO: Study of Information and Behavioral Sciences, Faculty of Integrated Arts and Sciences, Hiroshima University, Hiroshima, 730, Japan.