九州大学学術情報リポジトリ
Kyushu University Institutional Repository

THE SECOND PRINCIPLE OF OPTIMALITY

Iwamoto, Seiichi
Department of Mathematics, Faculty of Science, Kyushu University

https://doi.org/10.5109/13118

出版情報：統計数理研究. 17 (3/4), pp.101-114, 1977-03. Research Association of Statistical Sciences
バージョン：
権利関係：

KYUSHU UNIVERSITY

THE SECOND PRINCIPLE OF OPTIMALITY

By

Seiichi IWAMOTO*

(Received October 22, 1976)

1. Introduction.

Dynamic programming is closely related to the sequential decision process. They and the relations between them have been extensively investigated by many research workers [1,2,3,4,7,9,10]. It was explicitly or sometimes implicitly assumed that the reward function (or return, objective function) possesses both recurisveness (or separability) and monotonicity. The main result is Bellman's Principle of Optimality or its version in a sense [1, 2, 3, 4, 7, 9, 10]. Here we remark that the monotonicity is nothing but the "nondecreasingness".

In this paper we study N-stage sequential decision processes whose reward functions satisfy both recurisveness and "nonincreasingness". G. L. Nemhauser [10; p.44] gave a few interesting examples satisfying these two properties. Defining $(N-n+1)$-subprocesses $(1\leqq n\leqq N)$ by taking account of the optimizer (maximizing process or minimizing one) and monotonicity (nondecreasingness or nonincreasingness) we can combine $(N-n+1)$-subprocess and $(N-n)$-subprocess by the formula containing either maximum operator or minimum one. It is noted that the operator coincides with the optimizer of the $(N-n+1)$-subprocess. Further it is shown that, under strict monotonicity (strict increasingness or strict decreasingness) condition, Second Principle of Optimality holds true—the subpolicies of optimal policy are also optimal.

Calculating the optimal value functions and optimal policy of the reduced process we can find the solution of mathematical programming problem whose objective or constraint function is a finite continued fraction. Section 3 gives five illustrative examples. The last section comments on two meanings of Second Principle of Optimality.

2. Subprocesses and recursive formula.

An *N-stage sequential decision process* consists of a sixtuple (Opt, $\{S_n\}_{1\leqq n\leqq N+1}$, $\{A_n\}_{1\leqq n\leqq N}, \{g_n\}_{1\leqq n\leqq N}, k, \{T_n\}_{1\leqq n\leqq N}$) with the following specifications:

(i) N is a natural number, the *number of stage.*

(ii) S_n is a non-empty subset of the p_n-dimensional Euclidean space $R^{p_n}(p_n\geqq 1,$

* Department of Mathematics, Faculty of Science, Kyushu University, Fukuoka 812, Japan.

integer), the *n-th state space.*

(iii) A_n is a point-to-nonempty set valued mapping from S_n to a non-empty subset A_n of $R^{q_n}(q_n \geqq 1)$. $A_n(s_n)$ is the *n-th feasible action space at state* $s_n \in S_n$. It will be clear from the context whether A_n is considered mapping or set.

(iv) g_n: graph $(A_n) \times S_{n+1} \times$ range $(g_{n+1}) \to R^1$ $(1 \leqq n \leqq N-1)$ and g_N: graph $(A_N) \times S_{N+1} \times$ range $(k) \to R^1$ are the *n-th* and *N-th reward functions*, where graph $(A_n) = \{(s_n, a_n) \in S_n \times A_n \mid a_n = A_n(s_n)\}$ and range $(g_n) = \{g_n(s_n, a_n, s_{n+1}; g_{n+1}) \mid (s_n, a_n, s_{n+1}, g_{n+1}) \in$ graph $(A_n) \times S_{n+1} \times$ range $(g_{n+1})\}$ $1 \leqq n \leqq N-1$.

(v) k: $S_{N+1} \to R^1$ is the *terminal reward function,* where range $(k) = \{k(s_{N+1}) \mid s_{N+1} \in S_{N+1}\}$.

(vi) T_n: graph $(A_n) \to S_{n+1}$ is a mapping, the *n-th state transformation.*

(vii) Opt is either "Max" or "Min", the *optimizer.* It means to optimize, that is, either to maximize or to minimize the following problem:

$$\begin{aligned} &\text{Optimize} \quad g_1(s_1, a_1, s_2; g_2(s_2, a_2, s_3; \cdots; g_N(s_N, a_N, s_{N+1}; k(s_{N+1})) \cdots)) \\ &\text{subject to} \quad \text{(i)} \quad T_n(s_n, a_n) = s_{n+1} \quad 1 \leqq n \leqq N \\ &\qquad\qquad \text{(ii)} \quad a_n \in A_n(s_n) \quad 1 \leqq n \leqq N. \end{aligned}$$

$\pi = \{\pi_1, \pi_2, \cdots, \pi_N\}$ is called a *policy* if each π_n: $S_n \to A_n$ is a mapping such that $\pi_n(s_n) \in A_n(s_n)$ for $s_n \in S_n$. Π denotes the set of all policies. A policy $\{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\} \in \Pi$ is called *optimal* if for each initial state s_1 the sequence $\{\pi_1^*(s_1), \pi_2^*(\hat{s}_2), \pi_3^*(\hat{s}_3), \cdots, \pi_N^*(\hat{s}_N)\}$ attains the optimum value of the problem above, where $\hat{s}_2 = T_1(s_1, \pi_1^*(s_1))$, $\hat{s}_3 = T_2(\hat{s}_2, \pi_2^*(\hat{s}_2)), \cdots, \hat{s}_N = T_{N-1}(\hat{s}_{N-1}, \pi_{N-1}^*(\hat{s}_{N-1}))$.

Throughout the paper we assume that all the optimum values of the real-valued function over the regions considered always exist and hencefore that the point which attains optimum value exists. Hence we may assume that the optimal policy always exists.

First we consider an N-stage process (Max, $\{S_n\}_{1 \leqq n \leqq N+1}, \{A_n\}_{1 \leqq n \leqq N}, \{g_n\}_{1 \leqq n \leqq N}, k,$ $\{T_n\}_{1 \leqq n \leqq N}$) satisfying that each $g_k(s_k, a_k, s_{k+1}; \cdot)$ $((s_k, a_k, s_{k+1}) \in$ graph $(A_k) \times S_{k+1})$ is nonincreasing for $k = 1, 2, \cdots, N$, that is, for any $(s_k, a_k, s_{k+1}) \in$ graph $(A_k) \times S_{k+1}$, $c > c'$ with $c, c' \in$ range (g_{k+1}) implies $g_k(s_k, a_k, s_{k+1}; c) \leqq g_k(s_k, a_k, s_{k+1}; c')$.

For $n = 1, 3, 5, \cdots (\leqq N+1)$, we define $(N-n+1)$-subprocess by

$$(\text{Max}, \{S_m\}_{n \leqq m \leqq N+1}, \{A_m\}_{n \leqq m \leqq N}, \{g_m\}_{n \leqq m \leqq N}, k, \{T_n\}_{n \leqq m \leqq N}).$$

This subprocess starting in state s_n means to maximize the following problem:

$$\begin{aligned} &\text{Maximize} \quad g_n(s_n, a_n, s_{n+1}; g_{n+1}(s_{n+1}, a_{n+1}, s_{n+2}; \cdots; g_N(s_N, a_N, s_{N+1}; k(s_{N+1})) \cdots)) \\ &\text{subject to} \quad \text{(i)} \quad T_m(s_m, a_m) = s_{m+1} \quad n \leqq m \leqq N \\ &\qquad\qquad \text{(ii)} \quad a_m \in A_m(s_m) \quad n \leqq m \leqq N. \end{aligned}$$

This maximum value is denoted by $u^{N-n+1}(s_n)$. The function u^{N-n+1} is called the *$(N-n+1)$-th optimal value function.*

On the other hand, for $n = 2, 4, 6, \cdots, (\leqq N+1)$, $(N-n+1)$-subprocess is defined by

$$(\text{Min},\ \{S_m\}_{n\leqq m\leqq N+1}, \{A_m\}_{n\leqq m\leqq N}, \{g_m\}_{n\leqq m\leqq N}, k, \{T_m\}_{n\leqq m\leqq N}),$$

which in turn means to minimize the following problem:

$$\text{Minimize } g_n(s_n, a_n, s_{n+1}; g_{n+1}(s_{n+1}, a_{n+1}, s_{n+2}; \cdots; g_N(s_N, a_N, s_{N+1}; k(s_{N+1}))\cdots))$$

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

$$\text{(ii)} \quad a_m\in A_m(s_m) \quad n\leqq m\leqq N.$$

This minimum value is also denoted by $u^{N-n+1}(s_n)$. The function u^{N-n+1} is called the *$(N-n+1)$-th optimal value function.* Since 0-subprocess (the case $n=N+1$) includes no optimization factor, the definition yields the first (optimal) value function

$$u^0(s_{N+1})=k(s_{N+1}).$$

Note that the last optimal value $u^N(s_1)$ is the optimal value of the original process starting in (initial) state s_1.

Then according as n is odd or even we have the alternately recursive formula:

THEOREM 1.

$$u^{N-n+1}(s_n)=\max_{a_n\in A_n(s_n)} g_n(s_n, a_n, T_n(s_n, a_n); u^{N-n}(T_n(s_n, a_n))).$$

$$s_n\in S_n,\ n=1,3,5,\cdots,(\leqq N)$$

$$u^{N-n+1}(s_n)=\min_{a_n\in A_n(s_n)} g_n(s_n, a_n, T_n(s_n, a_n); u^{N-n}(T_n(s_n, a_n))).$$

$$s_n\in S_n,\ n=2,4,6,\cdots,(\leqq N)$$

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

PROOF. It suffices to prove only the case $n=1$, i.e.,

$$u^N(s_1)=\max_{a_1\in A_1(s_1)} g_1(s_1, a_1, T_1(s_1, a_1); u^{N-1}(T_1(s_1, a_1))) \quad s_1\in S_1, \tag{1}$$

since the other cases can be proved by similar method. Let $\{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\}$ be the optimal policy for the original process. Fix an arbitrary $s_1\in S_1$. Then we have

$$u^N(s_1)=g_1(s_1, a_1^*, s_2^*; g_2(s_2^*, a_2^*, s_3^*; \cdots; g_N(s_N^*, a_N^*, s_{N+1}^*; k(s_{N+1}^*))\cdots)), \tag{2}$$

where $T_n(s_n^*, a_n^*)=s_{n+1}^*$, $a_n^*=\pi_n^*(s_n^*)\in A_n(s_n^*)$ $1\leqq n\leqq N$, $s_1^*=s_1$. Since $\{a_n^*\}_{2\leqq n\leqq N}$ satisfies the conditions (i) and (ii) of the $(N-1)$-subprocess starting from state s_2^* and $u^{N-1}(s_2^*)$ is the minimum value of this process, we have

$$u^{N-1}(s_2^*)\leqq g_2(s_2^*, a_2^*, s_3^*; g_3(s_3^*, a_3^*, s_4^*; \cdots; g_N(s_N^*, a_N^*, s_{N+1}^*; k(s_{N+1}^*))\cdots)). \tag{3}$$

Hence the nonincreasingness of $g_1(s_1, a_1^*, s_2^*; \cdot)$ together with (2) and (3) yields

$$g_1(s_1, a_1^*, s_2^*; u^{N+1}(s_2^*))$$

$$\geqq g_1(s_1, a_1^*, s_2^*; g_2(s_2^*, a_2^*, s_3^*; \cdots; g_N(s_N^*, a_N^*, s_{N+1}^*; k(s_{N+1}^*))\cdots))=u^N(s_1).$$

Consequently

$$\underset{a_1\in A_1(s_1)}{\text{Max}}\, g_1(s_1,a_1,T_1(s_1,a_1);u^{N-1}(T_1(s_1,a_1)))\geqq u^N(s_1)\,. \tag{4}$$

Now let us establish the converse inequality. Fix an arbitrary $s_1\in S_1$ and an arbitrary $a_1\in A_1(s_1)$. Let $\{\hat{\pi}_2,\hat{\pi}_3,\cdots,\hat{\pi}_N\}$ be the optimal policy for the $(N-1)$-subprocess. Define $s_2=T_1(s_1,a_1)$. Then $s_2\in S_2$. By considering the $(N-1)$-subprocess starting in state s_2, we have

$$u^{N-1}(s_2)=g_2(\hat{s}_2,\hat{a}_2,\hat{s}_3;g_3(\hat{s}_3,\hat{a}_3,\hat{s}_4;\cdots;g_N(\hat{s}_N,\hat{a}_N,\hat{s}_{N+1};k(\hat{s}_{N+1}))\cdots)) \tag{5}$$

where $T_n(\hat{s}_n,\hat{a}_n)=\hat{s}_{n+1}$, $\hat{a}_n=\hat{\pi}_n(\hat{s}_n)$ $2\leqq n\leqq N$, $\hat{s}_2=s_2$. Since $\{a_1,\hat{a}_2,\hat{a}_3,\cdots,\hat{a}_N\}$ satisfies the conditions (i) and (ii) of the original process starting in s_1 and $u^N(s_1)$ is the maximum value of this process, we have

$$u^N(s_1)\geqq g_1(s_1,a_1,s_2;g_2(s_2,\hat{a}_2,\hat{s}_3;g_3(\hat{s}_3,\hat{a}_3,\hat{s}_4;\cdots;g_N(\hat{s}_N,\hat{a}_N,\hat{s}_{N+1};k(\hat{s}_{N+1}))\cdots))\,.$$

Then substituting (5) it holds that

$$u^N(s_1)\geqq g_1(s_1,a_1,s_2;u^{N-1}(s_2))\quad s_2=T_1(s_1,a_1)\,.$$

Since $a_1\in A_1(s_1)$ is arbitrary, we have

$$u^N(s_1)\geqq \underset{a_1\in A_1(s_1)}{\text{Max}}\, g_1(s_1,a_1,T_1(s_1,a_1);u^{N-1}(T_1(s_1,a_1)))\,. \tag{6}$$

Combining (4) and (6) we obtain the desired result (1). This completes the proof.

Second we consider a minimizing process (Min, $\{S_n\}_{1\leqq n\leqq N+1},\{A_n\}_{1\leqq n\leqq N},\{g_n\}_{1\leqq n\leqq N}$, $k,\{T_n\}_{1\leqq n\leqq N}$) satisfying that each $g_k(s_k,a_k,s_{k+1};\cdot)((s_k,a_k,s_{k+1})\in\text{graph}\,(A_k)\times S_{k+1})$ is nonincreasing for $k=1,2,\cdots,N$. Then for $n=1,3,\cdots,(\leqq N+1)$ we define $(N-n+1)$-subprocess by (Min, $\{S_m\}_{n\leqq m\leqq N+1},\{A_m\}_{n\leqq m\leqq N},\{g_m\}_{n\leqq m\leqq N},k,\{T_m\}_{n\leqq m\leqq N}$), which means to minimize the resulting objective function

$$g_n(s_n,a_n,s_{n+1};g_{n+1}(s_{n+1},a_{n+1},s_{n+2};\cdots;g_N(s_N,a_N,s_N;k(s_{N+1}))\cdots))$$

subject to the same conditions as the $(N-n+1)$-subprocess in the case Opt=Max. On the other hand for $n=2,4,\cdots,(\leqq N+1)$ we define $(N-n+1)$-subprocess by (Max, $\{S_m\}_{n\leqq m\leqq N+1},\{A_m\}_{n\leqq m\leqq N},\{g_m\}_{n\leqq m\leqq N},k,\{T_m\}_{n\leqq m\leqq N}$) which is a maximization process. Let $u^{N-n+1}(s_n)$ denote the *minimum* (resp. *maximum*) *value of the* $(N-n+1)$*-subprocess starting from* s_n for $n=1,3,\cdots$ (resp. $2,4,\cdots$). Then we obtain the following relation between $(N-n+1)$ and $(N-n)$-subprocesses.

THEOREM 2.

$$u^{N-n+1}(s_n)=\underset{a_n\in A_n(s_n)}{\text{Min}}\, g_n(s_n,a_n,T_n(s_n,a_n);u^{N-n}(T_n(s_n,a_n)))$$
$$s_n\in S_n,\quad n=1,3,5,\cdots(\leqq N)$$
$$u^{N-n+1}(s_n)=\underset{a_n\in A_n(s_n)}{\text{Max}}\, g_n(s_n,a_n,T_n(s_n,a_n);u^{N-n}(T_n(s_n,a_n)))$$
$$s_n\in S_n,\quad n=2,4,6,\cdots(\leqq N)$$
$$u^0(s_{N+1})=k(s_{N+1})\,.$$

PROOF. It sufficies to exchange Max for Min and $\leqq$ for $\geqq$ in the proof of Theorem 1.

Third we consider a $2N$-stage sequential decision process (Max, $\{S_n\}_{1\leqq n\leqq 2N+1}$, $\{A_n\}_{1\leqq n\leqq 2N}, \{g_n\}_{1\leqq n\leqq 2N}, k, \{T_n\}_{1\leqq n\leqq 2N}$) satisfying that each $g_k(s_k, a_k, s_{k+1}; \cdot)((s_k, a_k, s_{k+1}) \in \text{graph}\,(A_k)\times S_{k+1})$ is nonincreasing for $k=2n$ $1\leqq n\leqq N$ and nondecreasing for $k=2n+1$ $0\leqq n\leqq N-1$, where N is a natural number. We define $(2N-n+1)$-subprocess by $(P_n, \{S_m\}_{n\leqq m\leqq 2N+1}, \{A_m\}_{n\leqq m\leqq 2N}, \{g_m\}_{n\leqq m\leqq 2N}, k, \{T_m\}_{n\leqq m\leqq 2N})$ where

$$P_n=\begin{cases}\text{Max} & \text{if } n=4l+1,\ 4l+2\\ \text{Min} & \text{if } n=4l+3,\ 4l+4.\end{cases} \qquad l=0,1,2,\cdots$$

Let $u^{2N-n+1}(s_n)$ denote the *optimum (either maximum or minimum) value of the* $(2N-n+1)$*-subprocess starting from* s_n. Then corresponding to n we have the following formula:

THEOREM 3.

$$u^{2N-n+1}(s_n)=\operatorname*{Max}_{a_n\in A_n(s_n)} g_n(s_n, a_n, T_n(s_n, a_n); u^{2N-n}(T_n(s_n, a_n)))$$
$$s_n\in S_n,\ n=1,2,5,6,9,10,\cdots (\leqq 2N),$$
$$u^{2N-n+1}(s_n)=\operatorname*{Min}_{a_n\in A_n(s_n)} g_n(s_n, a_n, T_n(s_n, a_n); u^{2N-n}(T_n(s_n, a_n)))$$
$$s_n\in S_n,\ n=3,4,7,8,11,12,\cdots (\leqq 2N)$$
$$u^0(s_{2N+1})=k(s_{2N+1}).$$

Finally we consider an N-stage sequential decision process (Opt, $\{S_n\}_{1\leqq n\leqq N+1}$, $\{A_n\}_{1\leqq n\leqq N}, \{g_n\}_{1\leqq n\leqq N}, k, \{T_n\}_{1\leqq n\leqq N}$) satisfying that each $g_k(s_k, a_k, s_{k+1}; \cdot)((s_k. a_k, s_{k+1})\in \text{graph}\,(A_k)\times S_{k+1})$ is either nondecreasing or nonincreasing for $1\leqq k\leqq N$. This case includes not only the preceding three cases but also the dynamic programming one. Here the latter case means that each $g_k(s_k, a_k, s_{k+1}; \cdot)$ is nondecreasing for $1\leqq k\leqq N$, which was analyzed by N. Furukawa and S. Iwamoto [2].

The key point of this case is to determine $(N-n+1)$-subprocesses for $0\leqq n\leqq N+1$. Taking account of the optimizer (either Max or Min) and monotonicity (either nondecreasingness or nonincreasingness) of $g_k(s_k, a_k, s_{k+1}; \cdot)$ $1\leqq k\leqq N$, these subprocesses can be defined in the same spirit as the three cases stated above. Then depending on n and Opt we have between $(N-n+1)$- and $(N-n)$-subprocesses either

$$u^{N-n+1}(s_n)=\operatorname*{Max}_{a_n\in A_n(s_n)} g_n(s_n, a_n, T_n(s_n, a_n); u^{N-n}(T_n(s_n, a_n))) \quad s_n\in S_n$$

or

$$u^{N-n+1}(s_n)=\operatorname*{Min}_{a_n\in A_n(s_n)} g_n(s_n, a_n, T_n(s_n, a_n); u^{N-n}(T_n(s_n, a_n))) \quad s_n\in S_n,$$

where $u^{N+n+1}(s_n)$ is the optimum (either maximum or minimum) value of the resulting $(N-n+1)$-subprocess starting in s_n.

As was shown by Teorem 2.4 in [2], we obtain for the final case

THEOREM 4 (*Second Principle of Optimality*). *If, in addition, each* $g_k(s_k, a_k, s_{k+1}; \cdot)$ $((s_k, a_k, s_{k+1})\in \text{graph}\,(A_k)\times S_{k+1})$ *is either strictly increasing or strictly decreasing for* $k=1,2,\cdots,N$, *then the subpolicy* $\{\pi_n^*, \pi_{n+1}^*, \cdots, \overset{*}{N}\}$ *of the optimal policy* $\{\pi_1^*, \pi_2^*, \cdots, \pi_N^*\}$ *for the original process is optimal for the* $(N-n+1)$*-subprocess, where* $n=1,2,\cdots,N$.

PROOF. Taking account of the strict monotonicity, the method of proofs of Theorems 2.3 and 2.4 in [2] is extended to our case.

3. Applications.

It should be noted that the class of mathematical programming problems whose objective and constraint functions are expressed in the form

$$h(x_1, x_2, \cdots, x_N) = h_1(x_1; h_2(x_2; \cdots; h_{N-1}(x_{N-1}; h_N(x_N))\cdots))$$

where each $h_n(x_n; \cdot)$ $1 \leqq n \leqq N-1$ and h_N are continuous and either strictly increasing or strictly decreasing can be formulated as an $(N-1)$-stage sequential decision process. This result is a generalization of Propositions 3.1 and 3.2 in [7]. Further the optimum solution of the problem can be found through the last optimal value function, optimal policy and state transformations of the reduced process. This algorithm is the same as dynamic programming one [Section 5 of 2, Section 6 of 7]. We gave several examples for which Theorem 1 or 2 holds true [6, 8]. Here we illustrate another mathematical programming problems whose objective or constraint function is a finite continued fraction. The technique of reduction into a finite-stage sequential decision process has been already shown by the author [Section 3 of 3, Section 5 of 4]. The following examples are analyzed in detail in [5], which collects many other examples.

EXAMPLE 1. Maximize $\dfrac{x}{1+x+2\dfrac{y}{2+y+2(3+z)}}$

subject to (i) $x+y+z \leqq c(\geqq 0)$

(ii) $x \geqq 0,\ y \geqq 0,\ z \geqq 0.$

This problem reduces to a 2-stage sequential decision process (Max, $\{S_1, S_2, S_3\}, \{A_1, A_2\}, \{g_1, g_2\}, k, \{T_1, T_2\}$) where $S_n=[0,\infty]$ $1 \leqq n \leqq 3$, $A_n=[0,\infty]$, $A_n(s_n)=[0,s_n]$ $n=1,2$, $g_n(s_n, a_n, s_{n+1}; g)=\dfrac{a_n}{n+a_n+2g}$ $n=1,2$, $k(s_3)=3+s_3$, $T_n(s_n, a_n)=s_n-a_n$ $n=1,2$, $a_1=x$, $a_2=y$.

Applying Theorem 1 to the process, we find

$$u^0(s_3)=3+s_3$$

$$u^1(s_2)=\operatorname*{Min}_{a_2 \in A_2(s_2)} g_1(s_2, a_2, T_2(s_2, a_2); u^0(T_2(s_2, a_2)))$$

$$=\operatorname*{Min}_{0 \leqq a_2 \leqq s_2}\left[\frac{a_2}{2+a_2+2(3+s_2-a_2)}\right]=0 \quad \pi_2^*(s_2)=0$$

$$u^2(s_1)=\operatorname*{Max}_{a_1 \in A_1(s_1)} g_1(s_1, a_1, T_1(s_1, a_1); u^1(T_1(s_1, a_1)))$$

$$=\operatorname*{Max}_{0 \leqq a_1 \leqq s_1}\left[\frac{a_1}{1+a_1+2\times 0}\right]=\frac{s_1}{1+s_1}, \quad \pi_1^*(s_1)=s_1.$$

Hence the problem attains the maximum $\dfrac{c}{1+c}$ at $(x^*(c), y^*(c), z^*(c))=(c,0,0)$.

EXAMPLE 2. Maximize $\dfrac{1+x}{x+\dfrac{1+y}{y+1+z}}$

subject to (i) $x+y+z\leqq c(\geqq 0)$

(ii) $x\geqq 0,\ y\geqq 0,\ z\geqq 0.$

This also reduces to the previous process except for $g_n(s_n,a_n,s_{n+1};g)=\dfrac{1+a_n}{a_n+g}$ $n=1,2$, $k(s_3)=1+s_3$. The optimum value functions $\{u^0,u^1,u^2\}$ and (stationary) optimal policy $\{\pi_1^*,\pi_2^*\}$ are given as follows:

$$u^0(s_3)=1+s_3$$

$$u^1(s_2)=\frac{1}{1+s_2}\qquad \pi_2^*(s_2)=0$$

$$u^2(s_1)=1+s_1\qquad \pi_1^*(s_1)=0.$$

Then the problem attains the maximum value $1+c$ at $(x^*(c),y^*(c),z^*(c))=(0,0,c)$.

EXAMPLE 3. Minimize $\dfrac{x^3}{x+2\dfrac{y^2}{y+2\dfrac{z}{z+2w}}}$

subject to (i) $x+y+z+w\leqq c(\geqq 4)$

(ii) x,y,z,w; natural numbers,

where c ranges on $\{4,5,6,\cdots\}$. This problem reduces to a 3-stage process whose components are specified as follows:

$$\text{Opt}=\text{Min},\ S_n=\{5-n,6-n,7-n,\cdots\}\quad n=1,2,3,4,$$

$$A_n=\{1,2,3,\cdots\},\ A_n(s_n)=\{1,2,\cdots,s_n-(4-n)\},$$

$$g_n(s_n,a_n,s_{n+1};g_{n+1})=\frac{a_n^{4-n}}{a_n+2g_{n+1}}\quad n=1,2,3,\ k(s_4)=s_4,$$

$$T_n(s_n,a_n)=s_n-a_n\quad n=1,2,3.$$

By application of Theorem 2 the process yields

$$u^0(s_4)=s_4$$

$$u^1(s_3)=1/(2s_3-1)\quad \pi_3^*(s_3)=1$$

$$u^2(s_2)=(s_2-2)^2/\left(s_2-\frac{4}{3}\right)\quad \pi_2^*(s_2)=s_2-2$$

$$u^3(s_1)=1/\left(1+2\frac{(s_1-3)^2}{s_1-\frac{7}{3}}\right)\quad \pi_1^*(s_1)=1.$$

Hence when $(x,y,z,w)=(1,c-3,1,1)$ the problem has the minimum value $\dfrac{1}{1+2\dfrac{(c-3)^2}{c-\frac{7}{3}}}$.

EXAMPLE 4. Minimize $\dfrac{y_1y_3}{y_2y_4y_5}$

$$\text{subject to (i)}\quad \frac{1}{y_1}+\cfrac{2}{\frac{1}{y_2}+\cfrac{2}{\frac{1}{y_3}+\cfrac{2}{\frac{1}{y_4}+\frac{1}{y_5}}}}\leqq d$$

$$\text{(ii)}\quad 1\leqq y_j\leqq 2\quad 1\leqq i\leqq 5,$$

where d ranges on $\left[\frac{47}{38}, \frac{47}{19}\right]$ and the objective function $\frac{y_1y_3}{y_2y_4y_5}$ on $\left[\frac{1}{8}, 4\right]$. Rewriting

$$\frac{y_1y_3}{y_2y_4y_5}=\cfrac{y_1}{\cfrac{y_2}{\cfrac{y_3}{\cfrac{y_4}{\cfrac{1}{y_5}}}}},$$

the problem is formulated as the following 4-stage process:

$$\text{Opt}=\text{Min},\quad S_1=\left[\frac{47}{38}, \frac{47}{19}\right],\quad S_2=\left[\frac{19}{14}, \frac{19}{7}\right],\quad S_3=\left[\frac{7}{6}, \frac{7}{3}\right],$$

$$S_4=\left[\frac{3}{2}, 3\right],\quad S_5=[1,2],\quad A_1(s_1)=\left[\frac{1}{1\wedge\left(s_1-\frac{14}{19}\right)}, \frac{1}{\frac{1}{2}\vee\left(s_1-\frac{28}{19}\right)}\right],$$

$$A_2(s_2)=\left[\frac{1}{1\wedge\left(s_2-\frac{6}{7}\right)}, \frac{1}{\frac{1}{2}\vee\left(s_2-\frac{12}{7}\right)}\right],\quad A_3(s_3)=\left[\frac{1}{1\wedge\left(s_3-\frac{2}{3}\right)}, \frac{1}{\frac{1}{2}\vee\left(s_3-\frac{4}{3}\right)}\right],\quad A_4(s_4)=\left[\frac{1}{1\wedge(s_4-1)}, \frac{1}{\frac{1}{2}\vee(s_4-2)}\right],$$

$$g_n(s_n, a_n, s_{n+1}, g_{n+1})=\frac{a_n}{g_{n+1}}\quad 1\leqq n\leqq 4,\quad k(s_5)=1/s_5,$$

$$T_n(s_n, a_n)=\frac{2}{s_n-\frac{1}{a_n}}\quad 1\leqq n\leqq 4,\ \text{where } a\vee b=\max(a,b),\ a\wedge b=\min(a,b).$$

Then we have

$$u^0(s_5)=1/s_5\quad s_5\in[1,2]$$

$$u^1(s_4)=\underset{\frac{1}{1\vee(s_4-1)}\leqq a_4\leqq\frac{1}{\frac{1}{2}\vee(s_4-2)}}{\text{Max}}\left[a_4\times\frac{2}{\left(s_4-\frac{1}{a_4}\right)}\right]\quad s_4\in\left[\frac{3}{2}, 3\right]$$

$$u^2(s_3)=\underset{\frac{1}{1\vee\left(s_3-\frac{2}{3}\right)}\leqq a_3\leqq\frac{1}{\frac{1}{2}\vee\left(s_3-\frac{4}{3}\right)}}{\text{Min}}\left[a_3\times\frac{1}{u^1\left(\frac{2}{s_3-\frac{1}{a_3}}\right)}\right]\quad s_3\in\left[\frac{7}{6}, \frac{7}{3}\right]$$

$$u^3(s_2)=\max_{\frac{1}{1\wedge\left(s_2-\frac{6}{7}\right)}\leqq a_2\leqq\frac{1}{\frac{1}{2}\vee\left(s_2-\frac{12}{7}\right)}}\left[a_2\times\frac{1}{u^2\left(\frac{2}{s_2-\frac{1}{a_2}}\right)}\right]\quad s_2\in\left[\frac{19}{14},\frac{19}{7}\right]$$

$$u^4(s_1)=\min_{\frac{1}{1\wedge\left(s_1-\frac{14}{19}\right)}\leqq a_1\leqq\frac{1}{\frac{1}{2}\vee\left(s_1-\frac{28}{19}\right)}}\left[a_1\times\frac{1}{u^3\left(\frac{2}{s_1-\frac{1}{a_1}}\right)}\right]\quad s_1\in\left[\frac{47}{38},\frac{47}{19}\right].$$

It is difficult to calculate analytically the optimal value functions $\{u^0,u^1,\cdots,u^4\}$ and optimal policy $\{\pi_1^*,\pi_2^*,\pi_3^*,\pi_4^*\}$. However we obtain the following numerical ones:

s_5	$u^0(s_5)$
1.0000000	1.0000000
1.0500000	0.9523809
1.1000000	0.9090909
1.1500000	0.8695652
1.2000000	0.8333333
1.2500000	0.8000000
1.3000000	0.7692308
1.3500000	0.7407407
1.4000000	0.7142857
1.4500000	0.6896552
1.5000000	0.6666667
1.5500000	0.6451613
1.6000000	0.6250000
1.6500000	0.6060606
1.7000000	0.5882353
1.7500000	0.5714286
1.8000000	0.5555556
1.8500000	0.5405405
1.9000000	0.5263158
1.9500000	0.5128205
2.0000000	0.5000000

s_4	$u^1(s_4)$	$\pi_4^*(s_4)$
1.5000000	4.0000000	2.0000000
1.5500000	3.8000000	2.0000000
1.6000000	3.6383334	1.9666668
1.6500000	3.5000000	2.0000000
1.7000000	3.3000000	2.0000000
1.7500000	3.2000000	2.0000000
1.8000000	3.1000002	2.0000001
1.8500000	3.0000000	2.0000000
1.9000000	2.9000000	2.0000000
1.9500000	2.8000000	2.0000000
2.0000000	2.7000003	2.0000002
2.0500000	2.6000003	2.0000002
2.1000000	2.5000003	2.0000002
2.1500000	2.4000003	2.0000002
2.2000000	2.4000003	2.0000002
2.2500000	2.3000003	2.0000002
2.3000000	2.2000002	2.0000002
2.3500000	2.2000002	2.0000002
2.4000000	2.1000003	2.0000002
2.4500000	2.1000003	2.0000002
2.5000000	2.0000002	2.0000002
2.5500000	1.8181818	1.8181818
2.6000000	1.6666666	1.6666666
2.6500000	1.5384616	1.5384616
2.7000000	1.4285715	1.4285715
2.7500000	1.3333333	1.3333333
2.8000000	1.2500001	1.2500001
2.8500000	1.1764705	1.1764705
2.9000000	1.1111111	1.1111111
2.9500000	1.0526316	1.0526316
3.0000000	1.0000000	1.0000000

s_3	$u^2(s_3)$	$\pi_3^*(s_3)$
1.1666667	2.0000001	2.0000001
1.2166667	1.5745454	1.9681819
1.2666667	1.2000000	2.0000000
1.3166667	0.9468864	1.9884617
1.3666666	0.8707483	1.8285716
1.4166667	0.8015873	1.6833335
1.4666667	0.7380952	1.5500001
1.5166667	0.6776960	1.6264707
1.5666667	0.6296296	1.5111112
1.6166666	0.5844298	1.4026317
1.6666667	0.5517242	1.6000002
1.7166667	0.5166667	1.5500001
1.7666667	0.4807692	1.2500001
1.8166667	0.4464286	1.2500001
1.8666666	0.4166667	1.2500001
1.9166667	0.3965517	1.1500000
1.9666667	0.3709677	1.1500000
2.0166667	0.3548387	1.1000000
2.0666667	0.3285714	1.1500000
2.1166667	0.3125000	1.2500001
2.1666666	0.3000000	1.0500000
2.2166666	0.2830189	1.1320755
2.2666667	0.2625000	1.0500000
2.3166667	0.2500000	1.0000000
2.3333333	0.2500000	1.0000000

s_1	$u^4(s_1)$	$\pi_1^*(s_1)$
1.2368421	4.0000002	2.0000001
1.2868421	2.2493508	1.9681819
1.3368421	1.5764550	1.9666668
1.3868421	1.2519943	1.9884617
1.4368421	0.9964288	1.9285716
1.4868421	0.7419355	2.0000000
1.5368421	0.5660378	2.0000002
1.5868421	0.4816177	1.9264708
1.6368421	0.4402778	1.7611113
1.6868421	0.4006579	1.6026317
1.7368421	0.3681250	1.5500002
1.7868421	0.3375000	1.8000002
1.8368421	0.3062500	1.7500002
1.8868421	0.2823114	1.9000002
1.9368421	0.2542579	1.9000002
1.9868421	0.2274939	1.7000002
2.0368421	0.2125000	1.7000002
2.0868421	0.1940390	1.4500001
2.1368421	0.1812500	1.4500001
2.1868421	0.1739660	1.3000001
2.2368421	0.1605840	1.2000000
2.2868421	0.1500000	1.2000000
2.3368421	0.1437500	1.1500000
2.3868421	0.1368876	1.0951009
2.4368421	0.1250000	1.0382514
2.4736842	0.1250000	1.0000000

s_2	$u^3(s_2)$	$\pi_2^*(s_2)$
1.3571429	7.9999998	1.9999999
1.4071429	7.4727269	1.8681817
1.4571429	6.7301587	1.7666667
1.5071429	6.1538459	1.5384615
1.5571429	5.7142857	1.4285714
1.6071429	5.3333333	1.3333333
1.6571429	4.9999999	1.2500000
1.7071429	4.7058823	1.1764706
1.7571429	4.4444443	1.1111111
1.8071429	4.2105263	1.5526316
1.8571429	4.0000000	1.0000000
1.9071429	3.5333332	1.0000000
1.9571429	3.0434782	1.0000000
2.0071429	2.6956522	1.0000000
2.0571429	2.5217390	1.0000000
2.1071429	2.2399999	1.0000000
2.1571429	1.9354837	1.0000000
2.2071429	1.8124998	1.0000000
2.2571429	1.5882353	1.0000000
2.3071429	1.4755878	1.0000000
2.3571429	1.3548388	1.0000000
2.4071429	1.2475248	1.0000000
2.4571429	1.1617022	1.0000000
2.5071429	1.0560929	1.0000000
2.5571429	0.8750000	1.0500000
2.6071429	0.8333333	1.0000000
2.6571429	0.6351039	1.0000000
2.7071429	0.5035971	1.0071942
2.7142857	0.5000000	1.0000000

d	$\hat{y}_1(d)$	$\hat{y}_2(d)$	$\hat{y}_3(d)$	$\hat{y}_4(d)$	$\hat{y}_5(d)$	$V(d)$	$F(d)$
1.24	2.000	1.000	2.000	1.000	1.000	4.0000002	0.1236842
1.29	1.968	1.050	1.968	1.333	1.000	2.2493508	0.1285885
1.34	1.967	1.000	1.683	2.000	1.050	1.5764550	0.1340844
1.39	1.988	1.000	1.511	2.000	1.200	1.2519943	0.1387092
1.44	1.929	1.000	1.550	2.000	1.450	0.9964288	0.1440172
1.49	2.000	1.000	1.150	2.000	1.500	0.7419355	0.1490020
1.54	2.000	1.000	1.050	2.000	1.700	0.5660378	0.1535066
1.59	1.926	1.000	1.050	2.000	2.000	0.4816177	0.1585751
1.64	1.761	1.000	1.000	2.000	1.900	0.4402778	0.1635017
1.69	1.603	1.000	1.050	2.000	1.950	0.4006579	0.1685735
1.74	1.550	1.053	1.050	2.000	1.950	0.3681250	0.1735874
1.79	1.800	1.333	1.050	2.000	2.000	0.3375000	0.1786325
1.84	1.750	1.429	1.050	2.000	1.950	0.3062500	0.1834323
1.89	1.900	1.767	1.132	2.000	2.000	0.2823114	0.1888441
1.94	1.900	1.868	1.050	2.000	1.950	0.2542579	0.1935815
1.99	1.700	1.868	1.132	2.000	2.000	0.2274939	0.1979505
2.04	1.700	2.000	1.050	2.000	1.950	0.2125000	0.2033674
2.09	1.450	1.868	1.132	2.000	2.000	0.1940390	0.2080925
2.14	1.450	2.000	1.050	2.000	1.950	0.1812500	0.2135093
2.19	1.300	1.868	1.050	2.000	2.000	0.1739660	0.2187389
2.24	1.200	1.868	1.050	2.000	1.900	0.1605840	0.2233961
2.29	1.200	2.000	1.050	2.000	2.000	0.1500000	0.2287879
2.34	1.150	2.000	1.000	2.000	1.950	0.1437500	0.2334279
2.39	1.095	2.000	1.000	2.000	2.000	0.1368876	0.2386842
2.44	1.038	2.000	1.000	2.000	2.000	0.1297814	0.2436842
2.47	1.000	2.000	1.000	2.000	2.000	0.1250000	0.2473684

where $F(d)=\dfrac{1}{\hat{y}_1(d)+\dfrac{2}{\dfrac{1}{\hat{y}_2(d)}+\dfrac{2}{\dfrac{1}{\hat{y}_3(d)}+\dfrac{2}{\dfrac{1}{\hat{y}_4(d)}+\dfrac{2}{\hat{y}_5(d)}}}}}$ and $V(d)$ is the minimum value.

EXAMPLE 5. Maximize $\dfrac{r(s_1,a_1,s_2)}{1+\dfrac{r(s_2,a_2,s_3)}{1+\dfrac{r(s_3,a_3,s_4)}{\ddots\\ 1+\dfrac{r(s_{99},a_{99},s_{100})}{1+k(s_{100})}}}}$

subject to the stationary state transformation T

$T(s,a)$

s \ a	1	2	3
1	3	1	2
2	1	3	2
3	2	1	2

$S_n=\{1,2,3\}\quad 1\leqq n\leqq 100,$

$A_n(s)=A_n=\{1,2,3\}\quad 1\leqq n\leqq 99,\ s=1,2,3.$

where $k(s)=0$ for $s=1,2,3$ and $r(s,a,s')$ is

s	a	$r(s,a,1)$	$r(s,a,2)$	$r(s,a,3)$
1	1	1/2	1/3	1
	2	2	1	1/2
	3	1	1	1/3
2	1	1	2	1/3
	2	1/2	1/2	1/3
	3	1/4	1/2	1
3	1	1/2	1/3	1/3
	2	1	1/2	1/2
	3	1/3	1	1

This is just a 99-stage process. The recursive formula becomes

$$u^0(s_{100})=0\quad s_{100}=1,2,3$$

$$u^{100-n}(s_n)=\max_{a_n=1,2,3}\frac{r(s_n,a_n,T(s_n,a_n))}{1+u^{99-n}(T(s_n,a_n))}\qquad n=1,3,5,\cdots,99,\quad s_n=1,2,3$$

$$u^{100-n}(s_n)=\min_{a_n=1,2,3}\frac{r(s_n,a_n,T(s_n,a_n))}{1+u^{99-n}(T(s_n,a_n))}\qquad n=2,4,6,\cdots,98,\quad s_n=1,2,3.$$

Then we find the optimal policy $\{\pi_1^*,\pi_2^*,\cdots,\pi_{99}^*\}$ and optimal value functions $u^0\to u^1\to u^2\to\cdots\to u^{99}$, where

$$\pi_1^*=\begin{bmatrix}2\\1\\3\end{bmatrix},\pi_2^*=\begin{bmatrix}\overset{*}{1}\\2\\1\end{bmatrix},\pi_3^*=\begin{bmatrix}\overset{**}{2}\\1\\3\end{bmatrix},\pi_4^*=\begin{bmatrix}\overset{*}{1}\\2\\1\end{bmatrix},\cdots\pi_{97}^*=\begin{bmatrix}\overset{**}{2}\\1\\3\end{bmatrix},\pi_{98}^*=\begin{bmatrix}\overset{*}{1}\\2\\1\end{bmatrix},\pi_{99}^*=\begin{bmatrix}\overset{**}{2}\\1\\2\end{bmatrix},$$

$$\begin{bmatrix}0.0\\0.0\\0.0\end{bmatrix}\to\begin{bmatrix}2.0\\1.0\\1.0\end{bmatrix}\to\begin{bmatrix}0,50000\\0.16667\\0.16667\end{bmatrix}\to\begin{bmatrix}1.33333\\0.66667\\0.85714\end{bmatrix}\to\begin{bmatrix}0.53846\\0,17949\\0.20000\end{bmatrix}\to\begin{bmatrix}1.30000\\0.65000\\0.84783\end{bmatrix}\to$$

$$\begin{bmatrix}0.54118\\0.18039\\0.20202\end{bmatrix}\to\begin{bmatrix}1.29771\\0.64886\\0.84718\end{bmatrix}\to\begin{bmatrix}0.54137\\0.18046\\0.20216\end{bmatrix}\to\begin{bmatrix}1.\overset{*}{2}9754\\0.64877\\0.84713\end{bmatrix}\to\begin{bmatrix}0.\overset{**}{5}4138\\0.18046\\0.20217\end{bmatrix}\to$$

$$\begin{bmatrix}1.\overset{*}{2}9754\\0.64877\\0.84713\end{bmatrix}\to\begin{bmatrix}0.\overset{**}{5}4138\\0.18046\\0.20217\end{bmatrix}\to\begin{bmatrix}1.\overset{*}{2}9754\\0.64877\\0.84713\end{bmatrix}\to\begin{bmatrix}0.\overset{**}{5}4138\\0.18046\\0.20217\end{bmatrix}\to\cdots\to$$

$$\begin{bmatrix}0.\overset{**}{5}4138\\0.18046\\0.18046\end{bmatrix}\to\begin{bmatrix}1.\overset{*}{2}9754\\0.64877\\0.84713\end{bmatrix}.$$

Here we have used the conventional notations $\pi_n^*=\begin{bmatrix}\pi_n^*(1)\\\pi_n^*(2)\\\pi_n^*(3)\end{bmatrix}$ and $u^n=\begin{bmatrix}u^n(1)\\u^n(2)\\u^n(3)\end{bmatrix}$.

For instance, $\pi_1^*(2)=1$, $\pi_2^*(1)=1$, $u^0(1)=0.0$ and $u^{99}(3)=0.84713$.

4. A further comment.

In conclusion with the paper, it is noted that the author has used "Second Principle of Optimality" in two senses. Here we should distinguish two meanings —one, like as Theorems 1, 2 and 3, denotes usually the recursive formula between $(N-n+1)$- and $(N-n)$-subprocesses, and the other, like as Theorem 4, claims that any subpolicy of the optimal policy is optimal. The former was used in [6, 7] and the latter in [2].

Referenecs

[1] BELLMAN, R.: Dynamic Programming, Princeton Univ. Press, Princeton, New Jersey, 1957.

[2] FURUKAWA, N. and IWAMOTO, S.: *Dynamic programming on recursive reward systems*, Bull. Math. Statist., **17** (1976), 103-126.

[3] IWAMOTO, S.: *Inverse dynamic programming*, Mem. Fac. Sci. Kyushu Univ., Ser. A, Math. **30** (1976), 24-42.

[4] IWAMOTO, S.: *Inverse dynamic programming II*, Mem. Fac. Sci. Kyushu Univ., Ser. A, Math. **31** (1977), 25-44.

[5] IWAMOTO, S.: "Applications of Recursive Programming with Monotonicity", unpublished preprint (in Japanese), 1976, p. 128.

[6] IWAMOTO, S.: *A class of inverse theorems on recursive programming with monotonicity*, J. Operations Res. Soc. Japan, 20 (1977), in press.

[7] IWAMOTO, S.: *Inverse theorem in dynamic programming I, II,* J. Math. Anal. Appl., **57**, **58** (1977), in press.
[8] IWAMOTO, S.: Recursive programming approach to inequalities, to appear.
[9] MITTEN, L.G.: *Composition principles for synthesis of optimal multistage process,* Operations Res., **12** (1964), 601-619.
[10] NEMHAUSER, G.L.: Introduction to Dynamic Programming, John Wiley, 1966.