<table>
<thead>
<tr>
<th>Title</th>
<th>A Priori and a Posteriori Conditional Decision Processes with Nonadditively Recursive Utility (Mathematical Economics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Iwamoto, Seiichi</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (1999), 1108: 29-43</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1999-08</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/63289">http://hdl.handle.net/2433/63289</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>

Kyoto University
A Priori and a Posteriori Conditional Decision Processes
with
Nonadditively Recursive Utility

Seiichi Iwamoto
Department of Economic Engineering, Faculty of Economics
Kyushu University 27, Fukuoka 812-8581, Japan
Tel&Fax. +81(92)642-2488 Email: iwamoto@en.kyushu-u.ac.jp

Abstract

We consider stochastic optimization of not necessarily additive but recursive utilities over multi-stage decision processes. Without assuming any monotonicity, we optimize a regular process through a direct dynamic programming approach. On the regular decision process, we propose two related conditional decision processes: an a posteriori conditional decision process and an a priori. When the Markov transition law degenerates into a deterministic dynamics, the two conditional processes reduce to the same deterministic decision process. The conditional processes with monotonicity are optimized by the usual backward dynamic programming. We show that under additional convexity the regular process dominates the a priori in maximum value function and the prior does the a posteriori. We show that the a posteriori process illustrates Kreps and Porteus’s dynamic choice problem. The numerical example also verifies the dominance relation in three optimal value functions.

1 Introduction

In this paper we are concerned with a broad class of multi-stage stochastic decision processes with recursive utility system. It is well known that a dynamic programming recursive equation is valid under both separability and monotonicity in criterion function ([1],[5],[6],[22],[23],[26],[27]). The criterion for stochastic optimization problem is the expected value, which is a multiple summation in discrete process ([4],[7],[10],[11],[18],[19],[25]). The expected value of additive or multiplicative utility is easily decomposed into the current (immediate) return and the resulting remaining. Thus the expected value is separable, because of lineality of expectation operator. Of course, it is monotone. We focus on the decomposition (separability and monotonicity) of the expected value of nonadditively recursive utility, which is generated by the recursive system.

In Section 2, we optimize a regular process without an explicit monotonicity through a direct dynamic programming approach.

In Section 3, in contrast to the regular decision process, we propose two related conditional decision processes, which admit separability and monotonicity. One is an a posteriori
conditional decision process. The other is an a priori. The conditional processes are optimized by the usual backward dynamic programming. We show that for a deterministic dynamics the two conditional decision processes reduce to a common deterministic decision process.

In Section 4, we compare three decision processes under convexity or concavity. It is shown that, for convex utility system, the maximum value function of regular process dominates that of a priori, which in turn dominates that of a posteriori and that the dominance is reversed for concave system.

In Section 5, we illustrate three decision processes through a three-state, two-decision and two-stage model. We show that Kreps and Porteus's dynamic choice problem is nothing but an a posteriori decision process.

2 Regular Decision Process

Throughout the paper, the following data is given:

\( N \geq 2 \) is an integer; the total number of stages

\( X = \{s_1, s_2, \ldots, s_p\} \) is a finite state space

\( U = \{a_1, a_2, \ldots, a_q\} \) is a finite action space

\( g_n : X \times U \times R^1 \to R^1 \) is an \( n \)-th utility function \( (1 \leq n \leq N) \)

\( k : X \to R^1 \) is a terminal utility function (1)

\( p \) is a Markov transition law

\( p(y|x, u) \geq 0 \quad \forall(x, u, y) \in X \times U \times X, \quad \sum_{y \in X} p(y|x, u) = 1 \quad \forall(x, u) \in X \times U \)

\( y \sim p(\cdot|x, u) \) denotes that next state \( y \) conditioned on state \( x \) and action \( u \) appears with probability \( p(y|x, u) \).

We use the following notations:

\[ X^n := X \times X \cdots \times X \) (n-times) \]

\[ H_n := X \times U \times X \times U \times \cdots \times X \) ((2n - 1)-factors) \]

\[ h_n := (x_1, u_1, x_2, u_2, \ldots, x_n) \]

\[ g_n(g_{n+1}(\cdots g_N(k)(\cdots)) \]

\[ := g_n(x_n, u_n; g_{n+1}(x_{n+1}, u_{n+1}; \cdots; g_N(x_N, u_N; k(x_{N+1}))(\cdots)) \]

\[ E_{x_1}^u l := \sum_{y \in X} l(y)p(y|x, u) \quad \text{for } l = l(\cdot) \]
As a *regular* decision process, we consider the following optimization problem subject to a successive constraint:

Maximize \( E_{x_{1}}^{\sigma_{1}} g_{1}(g_{2}(\cdots g_{N}(k)\cdots)) \) \hspace{1cm} (2)

subject to \( (i)_{n} x_{n+1} \sim p(\cdot|x_{n}, u_{n}), \quad u_{n} \in U \quad 1 \leq n \leq N \)

where \( E_{x_{1}}^{\sigma} \) denotes the (regular) expectation operator on \( X^{N} \) induced from the conditional probability functions \( p(x_{n+1}|x_{n}, u_{n}) \), a *general* policy \( \sigma = \{\sigma_{1}, \sigma_{2}, \ldots, \sigma_{N}\} \) and an initial state \( x_{1} \).

We derive directly a recursive formula for this process. Let us consider for any given \( n (1 \leq n \leq N+1), \) \( h_{n} = (x_{1}, u_{1}, x_{2}, u_{2}, \ldots, x_{n}) \in H_{n} \) the maximization problem:

\[
v_{n}(h_{n}) = \operatorname{Max}_{\mu} E_{h_{n}}^{\mu}[g_{1}(\cdots g_{N}(k)\cdots)|(i)_{m} n \leq m \leq N] \quad h_{n} \in H_{n}, \quad 1 \leq n \leq N \hspace{1cm} (3)
\]

\[
v_{N+1}(h_{N+1}) = g_{1}(x_{1}, u_{1}; \cdots ; g_{N}(x_{N}, u_{N}; k(x_{N+1}))\cdots) \quad h_{N+1} \in H_{N+1} \hspace{1cm} (4)
\]

where the sequence of action and state \( (u_{n}, x_{n+1}, u_{n+1}, \ldots, u_{N}, x_{N+1}) \) after starting state \( h_{n} \) is governed stochastically by a *primitive* policy \( \mu = \{\mu_{n}, \mu_{n+1}, \ldots, \mu_{N}\} \) consisting of decision functions

\[
\mu_{m} : H_{m} \rightarrow U \quad n \leq m \leq N
\]

as follows:

\[
\mu_{n}(h_{n}) = u_{n} \rightarrow p(\cdot|x_{n}, u_{n}) \sim x_{n+1}
\]

\[
\rightarrow \mu_{n+1}(h_{n+1}) = u_{n+1} \rightarrow p(\cdot|x_{n+1}, u_{n+1}) \sim x_{n+2}
\]

\[
\rightarrow \cdots \rightarrow \mu_{N}(h_{N}) = u_{N} \rightarrow p(\cdot|x_{N}, u_{N}) \sim x_{N+1}.
\]

The maximization is taken for all *primitive* policies \( \mu \) for the subprocess starting from state \( h_{n} \in H_{n} \) at stage \( n \) and terminating at state \( h_{N+1} \in H_{N+1} \). Note that any primitive policy \( \mu = \{\mu_{n}, \mu_{n+1}, \ldots, \mu_{N}\} \) for the subprocess yields the expected value in (3) defined by the multiple summation:

\[
E_{h_{n}}^{\mu}[g_{1}(\cdots g_{N}(k)\cdots)|(i)_{m} n \leq m \leq N] = \sum_{(x_{n+1}, \ldots, x_{N+1}) \in X_{N-n+1}} \cdots \sum g_{1}(x_{1}, u_{1}; \cdots ; g_{N}(x_{N}, u_{N}; k(x_{N+1}))\cdots) \times p(x_{n+1}|x_{n}, u_{n}) \cdots p(x_{N+1}|x_{N}, u_{N}).
\]

Then we have the recursive equation between value \( v_{n}(h) \) and two-variable function \( v_{n+1}(h, \cdot, \cdot) \):

**Theorem 1**

\[
v_{n}(h) = \max_{u \in U} E_{x_{1}}^{u} v_{n+1}(h, u, \cdot) \quad h \in H_{n}, \quad n = 1, 2, \ldots, N \hspace{1cm} (8)
\]

\[
v_{N+1}(h) = g_{1}(x_{1}, u_{1}; \cdots ; g_{N}(x_{N}, u_{N}; k(x_{N+1}))\cdots) \quad h \in H_{N+1}.
\]

**Proof** The addition \( a + b : R^{1} \times R^{1} \rightarrow R^{1} \) is commutative, associative, and distributive over multiplication \( \times \). These properties imply the validity of recursive formula (8). \( \square \)
Solving the recursive equation (8) yields an $n$-th optimal decision function $\mu_n^* : H_n \rightarrow U$. As a whole, we have a \textit{primitive} optimal policy

$$\mu^* = \{\mu_1^*, \mu_2^*, ..., \mu_N^*\}.$$  

By successively projecting the optimal decision function $\mu_n^* : H_n \rightarrow U$ onto the original state space $X^n$, we obtain a \textit{general} optimal policy

$$\sigma^* = \{\sigma_1^*, \sigma_2^*, ..., \sigma_N^*\}$$

as follows:

$$\sigma_1^*(x_1) := \mu_1^*(h_1) \ (h_1 = x_1)$$

$$\sigma_2^*(x_1, x_2) := \mu_2^*(h_2) \ (h_2 = (x_1, u_1, x_2), \ u_1 = \mu_1^*(h_1))$$

$$\sigma_3^*(x_1, x_2, x_3) := \mu_3^*(h_3) \ (h_3 = (h_2, u_2, x_3), \ u_2 = \mu_2^*(h_2))$$

$$\vdots$$

$$\sigma_N^*(x_1, x_2, ..., x_N) := \mu_N^*(h_N) \ (h_N = (h_{N-1}, u_{N-1}, x_N), \ u_{N-1} = \mu_{N-1}^*(h_{N-1})).$$  

\section{Conditional Decision Processes}

In this section, we propose two conditional optimization problems subject to the successive constraint; one is an \textit{a posteriori} conditional decision process and the other an \textit{a priori}.

Throughout this section, we consider the class of all Markov policies on the original state space $X$. Note that any Markov policy $\pi = \{\pi_1, \tau_{12}, ..., \tau_{1N}\}$ is specified by a sequence of Markov decision functions:

$$\pi_n : X \rightarrow U \quad 1 \leq n \leq N.$$  

We assume that the utility system $\{g_n(x, u; \cdot)\}_1^N$ is \textit{monotone} with respect to the third-variable:

$$a < b \implies g_n(x, u; a) \leq g_n(x, u; b).$$  

Then we are concerned with optimization of expected value of the backward accumulated returns:

$$E_{x_1}^{\pi_1} g_1(g_2(\cdots g_N(k)\cdots))$$

$$= \sum_{(x_2, \ldots, x_{N+1}) \in X^N} \sum_{x_1} \cdots \sum_{x_{N+1}} g_1(x_1, u_1; g_2(x_2, u_2; \cdots ; g_N(x_N, u_N; k(x_{N+1}))) \cdots))$$

$$\times p(x_2|x_1, u_1)p(x_3|x_2, u_2) \cdots p(x_{N+1}|x_N, u_N)$$  

where the sequence of actions are determined through Markov policy $\pi$:

$$u_n = \pi_n(x_n) \quad 1 \leq n \leq N.$$  

The multiple summation (13) is not necessarily decomposed into iterative (or repeated) summation. We present two types of decomposition by taking backward expectations. In the following subsections, we optimize such decomposed forms in the class of Markov policies.
3.1 A Posteriori Conditional Decision Process

First, we take at each stage backward conditional expectation of remaining process after performing take-action for regular decision process. This generates an a posteriori conditional decision process as follows:

Maximize \( g_1(x_1, u_1; E_{x_2}^{u_2}g_2(x_2, u_2; \cdots E_{x_{N-1}}^{u_{N-1}}g_N(x_N; u_N; E_{x_N}^{u_N}k) \cdots)) \) \hspace{1cm} (14)
subject to \( (i_n x_{n+1} \sim p(x_n, u_n), \ u_n \in U \ 1 \leq n \leq N. \)

Here we note that

\[
E_{x}^{u}l = \sum_{y \in X} l(y) p(y| x, u)
\]
for \( l = l(\cdot) \). \hspace{1cm} (15)

For the sake of simplicity we use the following short notations:

\[
E^{n}l := E_{x_{n}}^{u_{n}}l \quad 1 \leq n \leq N. \hspace{1cm} (16)
\]

Thus the objective function in (14) is written as follows:

\[
g_1(E_1^{u_1}g_2(\cdots E_{x_{N-1}}^{u_{N-1}}g_N(x_N, u_N; E_{x_N}^{u_N}k) \cdots))
\]
\[
:= g_1(x_1, u_1; E_{x_2}^{u_2}g_2(x_2, u_2; \cdots E_{x_{N-1}}^{u_{N-1}}g_N(x_N, u_N; E_{x_N}^{u_N}k) \cdots)). \hspace{1cm} (18)
\]

We should remark that Markov policy \( \pi \) is implicit in the notation \( E^{n} \) in (18). That is,

\[
E^{n}l = E_{x_{n}}^{u_{n}}l, \quad u_n = \pi_n(x_n) 1 \leq n \leq N. \hspace{1cm} (19)
\]

Thus the a posteriori conditional expected value in (14) is not always equal to the called expected value (13). That is, in general, the equality

\[
E_{x_1}^{u_1}g_1(g_2(\cdots g_N(k) \cdots)) = g_1(E_1^{u_1}g_2(\cdots E_{x_{N-1}}^{u_{N-1}}g_N(E_{x_N}^{u_N}k) \cdots)) \hspace{1cm} (20)
\]
does not hold. However, two typical processes admit the equality (20). One is the additive process: \( g_n(x, u; h) = g_n(x, u) + h \). The other is the multiplicative process with nonnegative stage-wise return: \( g_n(x, u; h) = g_n(x, u) \times h \ (g_n(x, u) \geq 0) \). Throughout the remainder, we are mainly concerned with the class of processes which do not admit the equality (20).

Let us consider for any given \( n (1 \leq n \leq N + 1), \ x_n \in X \) the maximization problem:

\[
w_n(x_n) = \underset{\pi}{\text{Max}} \ [g_n(E^n g_{n+1}(\cdots E^{N-1} g_N(E^N k) \cdots)) \mid (i_m n \leq m \leq N)] \hspace{1cm} (21)
\]
\[
w_{N+1}(x_{N+1}) = k(x_{N+1}). \hspace{1cm} (22)
\]

Then we have the recursive equation between value \( w_n(x) \) and one-variable function \( w_{n+1} = w_{n+1}(\cdot) \):

**Theorem 2**

\[
w_n(x) = \underset{u \in U}{\text{Max}} g_n(x, u; E_{x}^{u}w_{n+1}) \quad x \in X, \ n = 1, 2, \ldots, N \hspace{1cm} (23)
\]
\[
w_{N+1}(x) = k(x) \quad x \in X. \hspace{1cm} (24)
\]
Proof The monotonicity in utility systems implies the validity of the recursive formula.

The validity of recursive formula (23), (24) is equivalent to the validity of equality

$$\max_{\pi_1} g_1(E^{1} g_2 \cdots E^{N-1} g_N(E^{N} k) \cdots)$$

$$= \max_{\pi_1} g_1(E^{1} \max_{\pi_2} g_2(\cdots E^{N-1} \max_{\pi_N} g_N(E^{N} k) \cdots))$$

$$\quad (u_n = \pi_n(x_n) \quad 1 \leq n \leq N).$$

(25)

We remark that the a posteriori cdp (14) is expressed in the following problem with backward aggregated return-variables \(\{m_n(\cdot)\}_{n=1}^{N+1}\):

Maximize \(m_1(x_1)\)

subject to \((i)_n \quad x_{n+1} \sim p(x_{n+1} | x_n, u_n), \quad u_n \in U \quad 1 \leq n \leq N\) (26)

\((ii) \quad m_{N+1}(x_{N+1}) = k(x_{N+1})\)

\((iii) \quad m_n = g_n(E^{n} m_{n+1}) \quad N \geq n \geq 1.\)

3.2 A Priori Conditional Decision Process

Second, before in turn performing take-action for regular decision process, we take at each stage backward conditional expectation of remaining process. This generates the following a priori conditional decision process:

Maximize \(E^u_0 g_1(x_1, u_1; E^u_1 g_2(x_2, u_2; \cdots; E^u_N g_N(x_N, u_N; k) \cdots))\) (27)

subject to \((i)_n \quad x_{n+1} \sim p(x_{n+1} | x_n, u_n), \quad u_n \in U \quad 1 \leq n \leq N.\)

Here we note that

\(E^u_0 g_0(x, u; l) = \sum_{y \in X} g_n(x, u; l(y))p(y|x, u)\) for \(l = l(\cdot).\) (28)

We use the following short notations:

\(E^n g_n(l) := E^n g_n(x_n, u_n; l) := E^n_{x_n, u_n} g_n(x_n, u_n; l) \quad 1 \leq n \leq N.\) (29)

Henceforth, the objective function in (27) is written as follows:

\(E^1 g_1(E^2 g_2(\cdots E^N g_N(k) \cdots))\)

\(= E^n_{x_1, u_1} g_1(x_1, u_1; E^n_{x_2, u_2} g_2(x_2, u_2; \cdots; E^n_{x_N, u_N} g_N(x_N, u_N; k) \cdots)).\) (30)

In the above notation \(E^n\), the relevant Markov policy \(\pi\) is also implicit:

\(E^n g_n(l) = E^n_{x_n, u_n} g_n(l), \quad u_n = \pi_n(x_n) \quad 1 \leq n \leq N.\) (31)

We remark that the a priori conditional expected value in (27) is not always identical with the a posteriori in (14). It may also different from the so-called expected value (13). However, the three expected values are identical both for the additive process and for the multiplicative process. The reason is nothing but the linearity of the expectation operator.
Let us consider for any given \( n \) \((1 \leq n \leq N+1)\), \( x_n \in X \) the maximization problem:

\[
W_n(x_n) = \operatorname{Max}_z \left[ E^n g_n(E^{n+1} g_{n+1}(\cdots E^N g_N(k) \cdots)) \right] \quad (i)_{m \leq n \leq N} \tag{32}
\]

\[
W_{N+1}(x_{N+1}) = k(x_{N+1}). \tag{33}
\]

Then we have the recursive equation between value \( W_n(x) \) and one-variable function \( W_{n+1} = W_{n+1}(\cdot) \):

**Theorem 3**

\[
W_n(x) = \operatorname{Max}_{u \in U} E^n g_n(x, u; W_{n+1}) \quad x \in X, \ n = 1, 2, \ldots, N \tag{34}
\]

\[
W_{N+1}(x) = k(x) \quad x \in X. \tag{35}
\]

**Proof** The monotonicity also implies the validity of the recursive formula. \( \Box \)

The recursive formula (34),(35) states the equality

\[
\operatorname{Max}_\pi E^1 g_1(E^2 g_2(\cdots E^N g_N(k) \cdots))
\]

\[
= \operatorname{Max}_\pi E^1 g_1(\operatorname{Max}_{\pi_2} E^2 g_2(\cdots \operatorname{Max}_{\pi_N} E^N g_N(k) \cdots)). \tag{36}
\]

We remark that the a priori cdp (27) is stated in the following problem with backward aggregated return-variables \( \{m_n(\cdot)\}_{n=1}^{N+1} \):

Maximize \( m_1(x_1) \)

subject to \( (i) \) \( x_{n+1} \sim p(\cdot| x_n, u_n), \ u_n \in U \ 1 \leq n \leq N \) \( (37) \)

\( (ii) \) \( m_{N+1}(x_{N+1}) = k(x_{N+1}) \)

\( (iii) \) \( m_n = E^m g_n(m_{n+1}) \ N \geq n \geq 1 \)

\[3.3 \text{ Deterministic Decision Process}\]

We consider the special dynamics where the Markov transition law \( p = p(y|x, u) \) degenerates into a deterministic dynamics:

\[ f = f(x, u) \text{ represents the successor state of } x \text{ for action } u. \tag{38} \]

(See also [5],[6],[8]). Then we have no difference between the a posterior conditional decision process and the a priori process:

Maximize \( g_1(x_1; u_1; g_2(x_2, u_2; \cdots; g_N(x_N, u_N; k(x_{N+1})) \cdots)) \)

subject to \( (i) \) \( f(x_n, u_n) = x_{n+1}, \ u_n \in U \ 1 \leq n \leq N \) \( (39) \)

Then the corresponding optimal value functions \( \{v_n(\cdot)\} \) satisfy the following:

**Corollary 1**

\[
(i) \quad W_n(x) = w_n(x) = v_n(x) \quad x \in X \tag{40}
\]

\[
(ii) \quad v_n(x) = \operatorname{Max}_{u \in U} g_n(x, u; v_{n+1}(f(x, u))) \quad x \in X, \ n = 1, 2, \ldots, N \tag{41}
\]

\[
v_{N+1}(x) = k(x) \quad x \in X. \tag{42}
\]
4 Convexity/Concavity

In this section, we compare the optimal value functions of regular, a priori and a posteriori decision processes under an additional convexity or concavity. We say that utility system \( \{g_n\} \) is convex (resp. concave) if \( g_n(x, u; \cdot): R^1 \rightarrow R^1 \) is convex (resp. concave) for \( (x, u) \in X \times U, \ 1 \leq n \leq N \).

**Theorem 4** Let utility system \( \{g_n\} \) be convex. Then we have

\[
E^\pi_{x_n} g_n(g_n+1(\cdots g_N(k)\cdots)) \\
\geq E^n g_n(E^{n+1} g_N(\cdots E^N g_N(k)\cdots)) \\
\geq g_n(E^n g_{n+1}(\cdots E^{N-1} g_N(E^N k)\cdots))
\]  

(43)

for any Markov policy \( \pi = \{\pi_n, \pi_{n+1}, \ldots, \pi_N\} \). The inequalities are reversed under concavity.

**Proof** We prove the inequalities for two-stage convex processes, because the inequalities for \( N \)-stage processes are similarly proved. First we note that the convexity implies

\[
E^u_{x_1} g_1(x_1, u_1; l) \geq g_1(x_1, u_1; E^u_{x_1} l)
\]

and

\[
E^u_{x_2} g_1(x_1, u_1; g_2(x_2, u_2; k)) \geq g_1(x_1, u_1; E^u_{x_2} g_2(x_2, u_2; k))
\]

where

\[
E^u_{x_2} g_1(x_1, u_1; g_2(x_2, u_2; k)) = \sum_{x_3 \in X} g_1(x_1, u_1; g_2(x_2, u_2; k(x_3)))p(x_3|x_2, u_2)
\]

\[
g_1(x_1, u_1; E^u_{x_2} g_2(x_2, u_2; k)) = g_1(x_1, u_1; \sum_{x_3 \in X} g_2(x_2, u_2; k(x_3))p(x_3|x_2, u_2)).
\]

Since the expectation operator \( E^u_{x_1} \) is monotone, we have

\[
E^u_{x_1} g_1(x_1, u_1; g_2(x_2, u_2; k)) = E^u_{x_1} E^u_{x_2} g_1(x_1, u_1; g_2(x_2, u_2; k)) \geq E^u_{x_1} g_1(x_1, u_1; E^u_{x_2} g_2(x_2, u_2; k)).
\]

This implies

\[
E^u_{x_1} g_1(g_2(k)) \geq E^1 g_1(E^2 g_2(k)).
\]  

(44)

Second, we have

\[
E^u_{x_2} g_2(x_2, u_2; k) \geq g_2(x_2, u_2; E^u_{x_2} k).
\]

This together with monotonicity of \( g_1(x_1, u_1; \cdot) \) implies

\[
g_1(x_1, u_1; E^u_{x_2} g_2(x_2, u_2; k)) \geq g_1(x_1, u_1; g_2(x_2, u_2; E^u_{x_2} k))
\]  

(45)

Thus by taking expectation operator \( E^u_{x_1} \) on both hand-sides, we get

\[
E^u_{x_1} g_1(x_1, u_1; E^u_{x_2} g_2(x_2, u_2; k)) \geq E^u_{x_1} g_1(x_1, u_1; g_2(x_2, u_2; E^u_{x_2} k)) \geq g_1(x_1, u_1; E^u_{x_1} g_2(x_2, u_2; E^u_{x_2} k))
\]
which implies

\[ E^1 g_1(E^2 g_2(k)) \geq g_1(E^1 g_2(E^2 k)). \]  \hfill (46)

This completes the proof. \qed

**Theorem 5** Let utility system \( \{g_n\} \) be convex. Then we have for any given \( n \) \( (1 \leq n \leq N) \)

(i) \[ v_n(h_n) \geq g_1(x_1, u_1; \ldots; g_{n-1}(x_{n-1}, u_{n-1}; W_n(x_n)) \ldots) \]
\[ \geq g_1(x_1, u_1; \ldots; g_{n-1}(x_{n-1}, u_{n-1}; w_n(x_n)) \ldots) \]  \hfill (47)

\[ h_n = (x_1, u_1, x_2, u_2, \ldots, x_n) \in H_n \]

(ii) \[ W_n(x_n) \geq w_n(x_n) \quad x_n \in S_n. \]

The inequalities are reversed for minimization problems under concavity.

**Proof** We show the inequalities for two-stage convex processes, because the inequalities for \( N \)-stage processes are similarly shown. First we note

\[ v_3(x_1, u_1, x_2, u_2, x_3) = g_1(x_1, u_1; g_2(x_2, u_2; W_3(x_3))), \quad W_3(x_3) = w_3(x_3) = k(x_3). \]

Let \( \pi^*_2 : X \rightarrow U \) be an optimal (Markov) decision function for the remaining one-stage a priori process. Then we have

\[ E_{x_2}^2 g_2(x_2, u_2; k) = W_2(x_2) \quad u_2^* = \pi^*_2(x_2). \]

From the definition of \( v_2(\cdot) \) and convexity of \( g_1(x_1, u_1; \cdot) \) we have

\[ v_2(x_1, u_1, x_2) \geq E_{x_2}^2 g_1(x_1, u_1; g_2(x_2, u_2^*; k)) \]
\[ \geq g_1(x_1, u_1; E_{x_2}^2 g_2(x_2, u_2^*; k)) \]
\[ = g_1(x_1, u_1; W_2(x_2)). \]

Further we get

\[ v_1(x_1) = \max_{\pi_1} v_2(x_1, u_1, x_2) \]
\[ \geq \max_{\pi_1} g_1(x_1, u_1; W_2) \]
\[ = W_1(x_1). \]

Second, taking maximum operator \( \max \) on both hand-sides

\[ E^2 g_2(k) \geq g_2(E^2 k) \]

we have

\[ W_2(x_2) \geq w_2(x_2). \]

Further by successive operation of \( \max \), \( \max \) for

\[ E^1 g_1(E^2 g_2(k)) \geq g_1(E^1 g_2(E^2 k)) \]
we get
\[ W_1(x_1) \geq w_1(x_1). \]
This completes the proof. \(\square\)

Example 1. (Non-additive/Additive Utility System [11]) Let
\[ g_n(x, u; h) = [g_n(x, u) + h^\alpha]^{1/\alpha} \quad g_n(x, u) \geq 0, \quad \alpha > 0. \] (48)
Then \(g_n(x, u; \cdot)\) is increasing on \([0, \infty)\). If \(0 < \alpha < 1\) (resp. \(\alpha > 1\)), it is concave (resp. convex). Then the utility function (48) generates the recursive utility:
\[ (g_1(x_1, u_1) + g_2(x_2, u_2) + \cdots + g_N(x_N, u_N) + k^\alpha(x_{N+1}))^{1/\alpha}. \]
If \(\alpha = 1\), it is linear. The resulting utility is additive:
\[ g_1(x_1, u_1) + g_2(x_2, u_2) + \cdots + g_N(x_N, u_N) + k(x_{N+1}). \]

Example 2. (Maximum/Minimum Utility System [9],[12],[13],[14],[15],[16],[17]) Let
\[ g_n(x, u; h) = g_n(x, u) \vee h \quad (\text{resp. } g_n(x, u) \wedge h) \quad -\infty < g_n(x, u) < \infty. \] (49)
Then \(g_n(x, u; \cdot)\) is nondecreasing and convex (resp. concave) on \(R^1\). The utility system yields the maximum (resp. minimum) utility:
\[ g_1(x_1, u_1) \vee g_2(x_2, u_2) \vee \cdots \vee g_N(x_N, u_N) \vee k(x_{N+1}) \] (resp. \(g_1(x_1, u_1) \wedge g_2(x_2, u_2) \wedge \cdots \wedge g_N(x_N, u_N) \wedge k(x_{N+1})\))

5 Examples

In this section, we illustrate three decision processes; a regular decision process and two conditional decision processes. One conditional decision process is the dynamic choice theory which has been originally introduced by Kreps and Porteus ([20],[21]). The other is its a priori process. In this section, we consider the utility system \(\{g_n(x, u; \cdot)\}_{1}^{N}\) as follows:
\[ g_n(x, u; h) = [g_n(u) + h^\alpha]^{1/\alpha} \quad (g_n(u) \geq 0, \quad \alpha > 0). \] (50)
(see also Epstein and Zin [3] and Ozaki and Streufert [24]). For the sake of simplicity we take the case \(N = 2, \quad \alpha = 2\) over Bellman and Zadeh's data [2, pp. B154]:

\[
\begin{array}{c|ccc}
    & s_1 & s_2 & s_3 \\
\hline
  s_1 & 0.8 & 0.1 & 0.1 \\
  s_2 & 0.0 & 0.1 & 0.9 \\
  s_3 & 0.8 & 0.1 & 0.1 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
    & s_1 & s_2 & s_3 \\
\hline
  s_1 & 0.1 & 0.9 & 0.0 \\
  s_2 & 0.8 & 0.1 & 0.1 \\
  s_3 & 0.1 & 0.0 & 0.9 \\
\end{array}
\]

\[ k(s_1) = 0.3 \quad k(s_2) = 1.0 \quad k(s_3) = 0.8 \] (51)
\[ g_2(a_1) = 1.0 \quad g_2(a_2) = 0.6 \] (52)
\[ g_1(a_1) = 0.7 \quad g_1(a_2) = 1.0 \] (53)
\[ u_t = a_1 \quad u_t = a_2 \]
5.1 Regular Decision Process

First we note that

\[
g_1(x_1, u_1; g(x_2, u_2; k(x_3))) = [g_1(u_1) + (g_2(u_2) + k^2(x_3))^{1/2}]^{1/2} = (g_1(u_1) + g_2(u_2) + k^2(x_3))^{1/2}.
\]

(54)

Then the resulting optimal equations (8),(9) reduce to the recursive equations:

\[
v_3(h_3) = (g_1(u_1) + g_2(u_2) + k^2(x_3))^{1/2}
v_2(h_2) = \max_{u_2} \sum_{x_3} v_3(h_2, u_2, x_3)p(x_3|x_2, u_2)
v_1(x_1) = \max_{u_1} \sum_{x_2} v_2(x_1, u_1, x_2)p(x_2|x_1, u_1)
\]

(55)

where

\[
\max_{u_n} = \max_{u_n \in \{a_1, a_2\}}, \quad \sum_{x_n} = \sum_{x_n \in \{s_1, s_2, s_3\}}.
\]

(56)

First, we have \(v_3(h_3); h_3 = (x_1, u_1, x_2, u_2, x_3)\):

\[
\begin{array}{c|ccc}
v_3(\cdot, a_1, \cdot, u_2, x_3) & s_1 & s_2 & s_3 \\
\hline
\text{a_1} & 1.3379 & 1.6432 & 1.5927 \\
a_2 & 1.1790 & 1.5166 & 1.3928 \\
\end{array}
\]

Second we calculate \(v_2(h_2); h_2 = (x_1, u_1, x_2)\):

\[
\begin{array}{c|ccc}
v_2(\cdot, u_1, x_2), \mu_2^*(\cdot, u_1, x_2) & s_1 & s_2 & s_3 \\
\hline
\text{a_1} & 1.4828, a_2 & 1.5411, a_1 & 1.3876, a_1 \\
a_2 & 1.5813, a_2 & 1.6355, a_1 & 1.4923, a_1 \\
\end{array}
\]

Here we note that

\[
v_3(x_1, u_1, x_2, u_2, x_3) = v_3(x'_1, u_1, x'_2, u_2, x_3) \quad \forall x_1, x_2, x'_1, x'_2 \in X
\]

\[
v_2(x_1, u_1, x_2) = v_2(x'_1, u_1, x_2) \quad \forall x_1, x'_1 \in X.
\]

Finally, we get

\[
v_1(s_1) = 1.6301, \quad v_1(s_2) = 1.5778, \quad v_1(s_3) = 1.5012
\]

\[
\mu_1^*(s_1) = a_2, \quad \mu_1^*(s_2) = a_2, \quad \mu_1^*(s_3) = a_2.
\]

The optimal primitive policy \(\mu^* = \{\mu_1^*, \mu_2^*\}\) yields an optimal general policy \(\sigma^* = \{\sigma_1^*, \sigma_2^*\}\):

\[
\begin{align*}
\sigma_1^*(s_1) &= a_2, & \sigma_1^*(s_2) &= a_2, & \sigma_1^*(s_3) &= a_2 \\
\sigma_2^*(s_1, s_1) &= a_2, & \sigma_2^*(s_2, s_1) &= a_2, & \sigma_2^*(s_3, s_1) &= a_2 \\
\sigma_1^*(s_1, s_2) &= a_1, & \sigma_1^*(s_2, s_2) &= a_1, & \sigma_1^*(s_3, s_2) &= a_1 \\
\sigma_2^*(s_1, s_3) &= a_1, & \sigma_2^*(s_2, s_3) &= a_1, & \sigma_2^*(s_3, s_3) &= a_1.
\end{align*}
\]

(57)

(58)

Note that this optimal general policy \(\sigma^*\) is Markov.
5.2 Dynamic Choice Process

We consider the following conditional optimization problem:

\[
\begin{align*}
\text{Maximize} & \quad [g_1(u_1) + (E_{x_1}^{u_1}[g_2(u_2) + (E_{x_2}^{u_2}k)^{2}]^{1/2})^{2}]^{1/2} \\
\text{subject to} & \quad (i)_n x_{n+1} \sim p(\cdot|x_n, u_n), \quad u_n \in \{a_1, a_2\} \quad n = 1, 2
\end{align*}
\]

(59)

Then, the "deterministic" dynamic programming technique, as we have already pointed in §3.1, yields the identity

\[
\begin{align*}
\operatorname{Max} \pi \left[ g_1(u_1) + (E_{x_1}^{u_1}[g_2(u_2) + (E_{x_2}^{u_2}k)^{2}]^{1/2})^{2} \right]^{1/2} \\
= \operatorname{Max} \left[ g_1(u_1) + (E_{x_1}^{u_1} \operatorname{Max} [g_2(u_2) + (E_{x_2}^{u_2}k)^{2}]^{1/2})^{2} \right]^{1/2}
\end{align*}
\]

(60)

\(u_n = \pi_n(x_n) \quad n = 1, 2\).

This reduces to the recurrence equations:

\[
\begin{align*}
w_3(x_3) &= k(x_3) \\
w_2(x_2) &= \operatorname{Max}_{u_2} \left[ g_2(u_2) + \left( \sum_{x_3} w_3(x_3)p(x_3|x_2, u_2) \right)^2 \right]^{1/2} \\
w_1(x_1) &= \operatorname{Max}_{u_1} \left[ g_1(u_1) + \left( \sum_{x_2} w_2(x_2)p(x_2|x_1, u_1) \right)^2 \right]^{1/2}.
\end{align*}
\]

(61)

We have the following optimal solution for the a posteriori conditional process:

\[
\begin{align*}
w_3(s_1) &= 0.3, \quad w_3(s_2) = 1.0, \quad w_3(s_3) = 0.8 \\
w_2(s_1) &= 1.2103, \quad w_2(s_2) = 1.2932, \quad w_2(s_3) = 1.0846 \\
\pi_2(s_1) &= a_2, \quad \pi_2(s_2) = a_1, \quad \pi_2(s_3) = a_1, \\
w_1(s_1) &= 1.6282, \quad w_1(s_2) = 1.5667, \quad w_1(s_3) = 1.4845 \\
\pi_1(s_1) &= a_2, \quad \pi_1(s_2) = a_2, \quad \pi_1(s_3) = a_2.
\end{align*}
\]

(62)

(63)

(64)

(65)

(66)

Now we evaluate \(g_1(x_1, u_1; w_2(x_2))\):

\[
\begin{array}{c|ccc}
\text{ } & s_1 & s_2 & s_3 \\
\hline
u_1 \\ a_1 & 1.4713 & 1.5403 & 1.3698 \\
a_2 & 1.5700 & 1.6347 & 1.4753
\end{array}
\]

where

\[
g_1(\cdot, u_1; w_2(x_2)) = [g_1(u_1) + w_2^2(x_2)]^{1/2}.
\]
5.3 A Priori Process

As an a priori process for the Kreps and Porteus's process ([20],[21]), we consider the following problem:

\[
\text{Maximize } E_{x_1}^{u_1}[g_1(u_1) + (E_{x_2}^{u_2}[g_2(u_2) + k^2]^{1/2})^2]^{1/2}
\]

subject to (i) \( x_{n+1} \sim p(\cdot|x_n, u_n) \), \( u_n \in \{a_1, a_2\} \) \( n = 1, 2 \). \hspace{1cm} (67)

Then, for the preceding data, the corresponding recursive equation

\[
W_3(x_3) = k(x_3)
\]
\[
W_2(x_2) = \max_{u_2} \sum_{x_3} [g_2(u_2) + W_3^2(x_3)]^{1/2} p(x_3|x_2, u_2)
\]
\[
W_1(x_1) = \max_{u_1} \sum_{x_2} [g_1(u_1) + W_2^2(x_2)]^{1/2} p(x_2|x_1, u_1)
\]

yields in turn

\[
W_3(s_1) = 0.3, \quad W_3(s_2) = 1.0, \quad W_3(s_3) = 0.8,
\]

\[
W_2(s_1) = 1.2215, \quad W_2(s_2) = 1.2940, \quad W_2(s_3) = 1.1047
\]
\[
\pi_2^*(s_1) = a_2, \quad \pi_2^*(s_2) = a_1, \quad \pi_2^*(s_3) = a_1,
\]

\[
W_1(s_1) = 1.6297, \quad W_1(s_2) = 1.5754, \quad W_1(s_3) = 1.4990
\]
\[
\pi_1^*(s_1) = a_2, \quad \pi_1^*(s_2) = a_2, \quad \pi_1^*(s_3) = a_2.
\]

Now we evaluate \( g_1(x_1, u_1; W_2(x_2)) \):

\[
g_1(\cdot, u_1; W_2(x_2))
\]

<table>
<thead>
<tr>
<th>( u_1 )</th>
<th>( x_2 )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>1.4806</td>
<td>1.5409</td>
<td>1.3858</td>
<td></td>
</tr>
<tr>
<td>( a_2 )</td>
<td>1.5786</td>
<td>1.6354</td>
<td>1.4901</td>
<td></td>
</tr>
</tbody>
</table>

where

\[
g_1(\cdot, u_1; W_2(x_2)) = [g_1(u_1) + W_2^2(x_2)]^{1/2}.
\]

Finally we observe that

\[
v_3(x_1, u_1, x_2, u_2, x_3) = g_1(x_1, u_1; g_2(x_2, u_2; W_3(x_3))) = g_1(x_1, u_1; g_2(x_2, u_2; w_3(x_3)))
\]
\[
v_2(x_1, u_1, x_2) \geq g_1(x_1, u_1; W_2(x_2)) \geq g_1(x_1, u_1; w_2(x_2)) \geq v_1(x_1) \geq W_1(x_1) \geq w_1(x_1)
\]

(74)

and

\[
k(x_3) = W_3(x_3) = w_3(x_3)
\]
\[
W_2(x_2) \geq w_2(x_2).
\]

(75)
References


