CONDITIONAL DECISION-MAKING IN FUZZY ENVIRONMENT

Seiichi Iwamoto  Kazuyoshi Tsurusaki  Toshiharu Fujita

Kyushu University  Kyushu Institute of Technology

(Received May 12, 1998; Revised November 6, 1998)

Abstract  Recently a regular (unconditional) decision process has been mathematically formulated from the multistage decision process in Bellman and Zadeh's paper "Decision-making in a fuzzy environment". According to the available information on total fuzziness, we propose two types of conditional decision process for regular decision process. One is an "a posteriori conditional decision process" and the other is an "a priori conditional decision process." The a posteriori process is formulated through taking at each stage backward conditional expectation of remaining process after performing take-action for the regular decision process. The a priori is through taking at each stage backward conditional expectation before take-action. We derive recursive equations for both a posteriori and a priori processes with numerical illustrations.

1. Introduction

Since Bellman and Zadeh have published their seminal paper [5], a large amount of efforts has been devoted to the study of fuzzy theory of mathematical programming. Of course, fuzzy theory of dynamic programming has been studied ([1], [6], [7], [15], [16], [17] and others). Bellman and Zadeh [5] have proposed an essentially same recursive formula for both deterministic process and stochastic process. Their recursive formula for deterministic process is valid. However, their derivation of recursive formula for stochastic process lacks a mathematical consistency: their dynamic programming solution does not coincide with an enumerative solution. Recently pointing out this inconsistency, Iwamoto and Fujita [12] have proposed an invariant imbedding method, whose solution assures the enumerative one (see also Iwamoto and Sniedovich [13]). Bellman and Zadeh [5], Iwamoto and Fujita [12] and Iwamoto and Sniedovich [13] are, of course, dynamic programming methods. Any dynamic programming - whatever its style may be - should yield a solution of the original problem.

A motivation of this paper is to consider an inverse problem to [5, §5], that is, to derive an optimization problem whose recursive formula yields Bellman and Zadeh's stochastic recursive formula. (There are several kinds of inverse problem. See [2],[3] for inverse problems in this Bellman's sense).

In Section 2, we consider a regular (unconditional) decision process with associative binary relation. We show two approaches to the regular process. One is a direct approach. The other is an invariant imbedding approach ([4],[8],[9],[10],[18],[19],[20],[21]). In Section 3, we propose two types of conditional decision process for regular process. One is an "a posteriori conditional decision process" and the other is an "a priori conditional decision process."

It makes a difference to the decision-maker whether or not the information on associating a current membership with the remaining total fuzziness is available to him/her. The a posteriori process is formulated through taking at each stage backward conditional
Conditional Decision-making

expectation of remaining process after performing take-action for the regular decision process. The a priori is through taking at each stage backward conditional expectation before take-action. We derive recursive equations for both a posteriori and a priori processes. The recursive equation for a posteriori process is identical with the desired Bellman and Zadeh's stochastic recursive formula, which at the same time results in having given a solution to the inverse problem. In the last section, we illustrate numerical examples of a posteriori and of a priori processes. The example for a posteriori process is nothing but Bellman and Zadeh's stochastic example.

2. Regular Decision Process

Throughout the paper, the following data is given:

- \( N \geq 2 \) is an integer; the total number of stages
- \( X = \{s_1, s_2, \ldots, s_l\} \) is a finite state space
- \( U = \{a_1, a_2, \ldots, a_k\} \) is a finite action space
- \( \mu_n : X \times U \to [0, 1] \) is an \( n \)-th membership function (1 \( \leq n \leq N \))
- \( \mu_{N+1} : X \to [0, 1] \) is a goal membership function
- \( \circ : [0, 1] \times [0, 1] \to [0, 1] \) is an associative binary relation
- \( \lambda, \mu, \nu \) is a left-identity element
- \( p \) is a Markov transition law
- \( y \sim p(\cdot \mid x, u) \) denotes that next state \( y \) conditioned on state \( x \) and action \( u \) appears with probability \( p(y|x, u) \).

First, we consider as a regular decision process the following optimization problem subject to a successive constraint:

\[
\begin{align*}
\text{Maximize } & E_{x_1}^\sigma[\mu_1 \circ \mu_2 \circ \cdots \circ \mu_N \circ \mu_{N+1}] \\
\text{subject to } & (i)_n \ x_{n+1} \sim p(\cdot \mid x_n, u_n), \ u_n \in U \quad 1 \leq n \leq N
\end{align*}
\]

where \( E^\sigma \) denotes the expectation (summation) operator on \( X \times X \cdots \times X \) (\( N \)-times) induced from the conditional probability functions \( p(x_{n+1}\mid x_n, u_n) \), a general policy \( \sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_N\} \) and an initial state \( x_1 \).

2.1. Direct approach

In this section, we use the following notation:

\[
H_n := X \times U \times X \times U \times \cdots \times X \quad (2n-1\text{-times}).
\]

First, we derive directly a recursive formula for the process (2.2). 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:

\[
\begin{align*}
v_n(h_n) &= \max_{\mu} E_{h_n}^\nu[\mu_1 \circ \cdots \circ \mu_N \circ \mu_{N+1} \mid (i)_m \ n \leq m \leq N] \\
&\quad h_n \in H_n, \ 1 \leq n \leq N
\end{align*}
\]

\[
v_{N+1}(h_{N+1}) = \mu_1(x_1, u_1) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \quad h_{N+1} \in H_{N+1}
\]
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 \textit{primitive} policy \(\nu = \{\nu_n, \nu_{n+1}, \ldots, \nu_N\}\) consisting of decision functions

\[
\nu_m : H_m \to U \quad n \leq m \leq N
\]

as follows:

\[
\begin{align*}
\nu_n(h_n) &= u_n \to p(\cdot | x_n, u_n) \sim x_{n+1} \\
\nu_{n+1}(h_{n+1}) &= u_{n+1} \to p(\cdot | x_{n+1}, u_{n+1}) \sim x_{n+2} \\
\vdots & \to \nu_N(h_N) = u_N \to p(\cdot | x_N, u_N) \sim x_{N+1}.
\end{align*}
\]

(2.6)

The maximization is taken for all \textit{primitive} policies \(\nu\) for a 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 \(\nu = \{\nu_n, \nu_{n+1}, \ldots, \nu_N\}\) for the subprocess yields the expected value in (2.3) defined by the multiple summation:

\[
E_{h_n}^{\nu} \left[ \mu_1 \circ \cdots \circ \mu_{N+1} \right](i, m) \quad n \leq m \leq N
\]

\[=
\sum_{(x_{n+1}, \ldots, x_{N+1}) \in X \times \cdots \times X} \mu_1(x_1, u_1) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \times p(x_{n+1}|x_n, u_n) \cdots p(x_{N+1}|x_N, u_N).
\]

(2.7)

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

\textbf{Theorem 1}

\[
v_n(h) = \max_{u \in U} \sum_{y \in X} v_{n+1}(h, u, y)p(y|x, u) \quad h \in H_n, \quad n = 1, 2, \ldots, N
\]

(2.8)

\[
v_{N+1}(h) = \mu_1(x_1, u_1) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \quad h \in H_{N+1}.
\]

(2.9)

\textit{Proof} The addition \(a + b : R^l \times R^l \to R^l\) is commutative, associative, and monotone. These properties imply the validity of recursive formula (2.8).

Solving the recursive equation (2.8), we have a \textit{primitive} optimal policy

\[
\nu^* = \{\nu_1^*, \nu_2^*, \ldots, \nu_N^*\}.
\]

By successively projecting the optimal decision function \(\nu_n^* : H_n \to U\) onto the original state space \(X \times \cdots \times X\) \((n\text{-times})\), we obtain a \textit{general} optimal policy

\[
\sigma^* = \{\sigma_1^*, \sigma_2^*, \ldots, \sigma_N^*\}
\]

as follows:

\[
\begin{align*}
\sigma_1^*(x_1) :&= \nu_1^*(h_1) \quad (h_1 = x_1) \\
\sigma_2^*(x_1, x_2) :&= \nu_2^*(h_2) \quad (h_2 = (x_1, u_1, x_2), \ u_1 = \nu_1^*(h_1)) \\
\sigma_3^*(x_1, x_2, x_3) :&= \nu_3^*(h_3) \quad (h_3 = (h_2, u_2, x_3), \ u_2 = \nu_2^*(h_2)) \\
\vdots \\
\sigma_N^*(x_1, x_2, \ldots, x_N) :&= \nu_N^*(h_N) \quad (h_N = (h_{N-1}, u_{N-1}, x_N), \ u_{N-1} = \nu_{N-1}^*(h_{N-1})).
\end{align*}
\]

(2.10)
2.2. Invariant imbedding approach

Second, we derive an important recursive formula for this process by imbedding the problem (2.2) into the following relatively large family of parameterized problems. Let us consider for any given $n$ ($1 \leq n \leq N + 1$), $x_n \in X$ and $\lambda_n \in [0, 1]$ the maximization problem:

$$v_n(x_n; \lambda_n) = \max_{\mu} E_{x_n, \lambda_n}^{\pi_n} \left[ \lambda_n \circ \mu_n \circ \cdots \circ \mu_N \circ \mu_{N+1} \right]_{(i)_{m\in [n,m \leq N]}} \tag{2.11}$$

$$v_{N+1}(x_{N+1}; \lambda_{N+1}) = \lambda_{N+1} \circ \mu_{N+1}(x_{N+1}) \quad 0 \leq \lambda_{N+1} \leq 1. \tag{2.12}$$

Here the maximization is taken for all Markov policies $\pi$ for a subprocess starting from one-dimensionally augmented state $(x_n, \lambda_n) \in X \times [0, 1]$ at stage $n$ and terminating at state $(x_{N+1}, \lambda_{N+1})$. Note that any Markov policy $\pi = \{\pi_0, \pi_1, \ldots, \pi_N\}$ on the augmented state space $X \times [0, 1]$ is specified by a sequence of decision functions:

$$\pi_n : X \times [0, 1] \to U \quad n \leq m \leq N. \tag{2.13}$$

Further we remark that the expected value in (2.11) is defined by the multiple summation:

$$E_{x_n, \lambda_n}^{\pi_n} \left[ \lambda_n \circ \mu_n \circ \cdots \circ \mu_N \circ \mu_{N+1} \right]_{(i)_{m\in [n,m \leq N]}} = \sum_{(x_{n+1}, \ldots, x_{N+1}) \in X \times \cdots \times X} \left\{ \lambda_n \circ \mu_n(x_n, u_n) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \right\} \times p(x_{n+1}|x_n, u_n) \cdots p(x_{N+1}|x_N, u_N) \tag{2.14}$$

where the alternating sequence of action and one-dimensionally augmented state

$$\{u_n, (x_{n+1}, \lambda_{n+1}), u_{n+1}, (x_{n+2}, \lambda_{n+2}), \ldots, u_N, (x_{N+1}, \lambda_{N+1})\}$$

is stochastically generated through the Markov policy $\pi$ and the starting state $(x_n, \lambda_n)$ as follows:

$$\pi_n(x_n, \lambda_n) = u_n \rightarrow \left\{ p^*(x_n, u_n) \sim x_{n+1} \lambda_n \circ \mu_n(x_n, u_n) = \lambda_{n+1} \right\}$$

$$\rightarrow \pi_{n+1}(x_{n+1}, \lambda_{n+1}) = u_{n+1} \rightarrow \left\{ p^*(x_{n+1}, u_{n+1}) \sim x_{n+2} \lambda_{n+1} \circ \mu_{n+1}(x_{n+1}, u_{n+1}) = \lambda_{n+2} \right\} \tag{2.15}$$

$$\rightarrow \ldots$$

$$\rightarrow \pi_N(x_N, \lambda_N) = u_N \rightarrow \left\{ p^*(x_N, u_N) \sim x_{N+1} \lambda_N \circ \mu_N(x_N, u_N) = \lambda_{N+1} \right\}.$$ 

However, note that the sequence of the latter halves of the states $\{\lambda_{n+1}, \lambda_{n+2}, \ldots, \lambda_{N+1}\}$ behaves deterministically in (2.15).

Then we have the recursive equation between value $v_n(x; \lambda)$ and two-variable function $v_{n+1}(\cdot; \cdot)$:

**Theorem 2**

$$v_n(x; \lambda) = \max_{u \in U} \sum_{y \in X} v_{n+1}(y; \lambda \circ \mu_n(x, u)) p(y|x, u) \tag{2.16}$$

$$x \in X, \quad 0 \leq \lambda \leq 1 \quad n = 1, 2, \ldots, N,$$

$$v_{N+1}(x; \lambda) = \lambda \circ \mu_{N+1}(x) \quad x \in X, \quad 0 \leq \lambda \leq 1. \tag{2.17}$$
By replacing the binary relation \(\land\) in proof of Theorem 6.2 of ([14]) with the associative relation \(\circ\), we can derive the recursive formula (2.16) in a same way as in ([14]).

Solving the recursive equation (2.16) yields an \(n\)-th optimal decision function \(\pi_n^* : X \times [0, 1] \to U\). Hence, as a whole, we have a Markov optimal policy

\[
\pi^* = \{\pi_1^*, \pi_2^*, \ldots, \pi_N^*\}
\]
on the one-dimensionally extended state space \(X \times [0, 1]\). By projecting the optimal policy \(\pi^*\) onto the original history space

\[
X \times X \times \cdots \times X \quad (N + 1)\text{-times}
\]
with starting state \((x_1, \iota)\), we obtain a general optimal policy

\[
\sigma^* = \{\sigma_1^*, \sigma_2^*, \ldots, \sigma_N^*\}
\]
on the state space \(X\). At the same time, the desired optimal value is given by \(v_1(x_1; \iota)\), which is attained by the policy \(\sigma^*\) ([14]).

3. Conditional Decision Processes

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

Throughout this section, we consider the class of all Markov policies on the original state space \(X\). Note that any Markov policy \(\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}\) is specified by a sequence of Markov decision functions :

\[
\pi_n : X \to U \quad 1 \leq n \leq N.
\] (3.1)

We assume that the binary relation \(\circ\) is monotone :

\[
\mu < \nu \implies \lambda \circ \mu \leq \lambda \circ \nu.
\] (3.2)

However, we do not assume the associativity of the relation \(\circ\).

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

\[
E_x^n[\mu_1 \circ [\mu_2 \circ \cdots \circ [\mu_N \circ \mu_{N+1}] \cdots]]
= \sum_{(x_2, \ldots, x_{N+1}) \in X \times \cdots \times X} \sum_{x_1} \{[\mu_1(x_1, u_1) \circ [\mu_2(x_2, u_2) \circ \cdots \circ [\mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1})] \cdots]]
\times p(x_2|x_1) p(x_3|x_2, u_2) \cdots p(x_{N+1}|x_N, u_N)\}
\] (3.3)

where the sequence of controls is determined through Markov policy \(\pi\) :

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

The multiple summation (3.3) is not necessarily decomposed into an iterative (or repeated) summation. We show two types of decomposition by taking backward conditional expectation. In the following subsections, we optimize such decomposed forms in the class of Markov policies.
3.1. A posteriori conditional decision process

First, at each stage we take backward conditional expectation of remaining process after performing take-action for regular decision process (Figure 1). This generates an a posteriori cdp as follows:

Maximize $\mu_1(x_1, u_1) \circ E^u_{x_1} \circ \cdots \circ E^u_{x_{N-1}} \circ [\mu_N(x_N, u_N) \circ E^u_{x_N} \mu_{N+1}] \cdots$

subject to $(i)_{x_{n+1}} \sim p(\cdot \mid x_n, u_n), \; u_n \in U \; 1 \leq n \leq N$ (3.4)

Here we note that

$$E^n_x \mu = \sum_{y \in X} \mu(y)p(y \mid x, u) \quad \text{for} \; \mu = \mu(\cdot).$$ (3.5)

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

$$E^n_x \mu := E^u_{x_n} \mu$$ (3.6)

$$\mu_n \circ E^n_x \mu := \mu_n(x_n, u_n) \circ E^n_x \mu \; 1 \leq n \leq N.$$ (3.7)

Thus the objective function in (3.4) is written as follows:

$$\mu_1 \circ E^1_{x_1} \circ \cdots \circ E^{N-1}_{x_{N-1}} \circ [\mu_N \circ E^N \mu_{N+1}] \cdots$$

$$:= \mu_1(x_1, u_1) \circ E^u_{x_1} \circ \cdots \circ E^u_{x_{N-1}} \circ [\mu_N(x_N, u_N) \circ E^u_{x_N} \mu_{N+1}] \cdots].$$ (3.8)

We should remark that Markov policy $\pi$ is implicit in the notation $E^n$ in (3.8). That is,

$$E^n_x \mu = E^u_{x_n} \mu, \; u_n = \pi_n(x_n) \; 1 \leq n \leq N.$$ (3.9)

Thus the resulting a posteriori conditional expected value from Markov policy $\pi$ is one backward iterative summation:

$$\mu_1 \circ E^1_{x_1} \circ \cdots \circ E^{N-1}_{x_{N-1}} \circ [\mu_N \circ E^N \mu_{N+1}] \cdots$$

$$= \mu_1(x_1, u_1) \circ \sum_{x_2 \in X} \mu_2(x_2, u_2) \circ \cdots \circ \mu_{N-1}(x_{N-1}, u_{N-1}) \circ$$

$$\sum_{x_N \in X} \mu_N(x_N, u_N) \circ \sum_{x_{N+1} \in X} \mu_{N+1}(x_{N+1}) p(x_{N+1} \mid x_N, u_N)$$

$$p(x_N \mid x_{N-1}, u_{N-1}) \cdots p(x_2 \mid x_1, u_1)$$

$$(u_n = \pi_n(x_n) \; 1 \leq n \leq N).$$ (3.10)

On the other hand, the so-called expected value is the multiple summation:

$$E^n_{x_1} \circ \cdots \circ [\mu_N \circ \mu_{N+1}] \cdots$$

$$= \sum_{(x_2, \ldots, x_{N+1}) \in X \times \cdots \times X} \prod_{(x_2, \ldots, x_{N+1}) \in X \times \cdots \times X} [\mu_1(x_1, u_1) \circ \mu_2(x_2, u_2) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \cdots]$$

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

$$(u_n = \pi_n(x_n) \; 1 \leq n \leq N).$$ (3.11)

We note that in general the equality

$$E^n_{x_1} \circ \cdots \circ [\mu_N \circ \mu_{N+1}] \cdots$$

$$= \mu_1 \circ E^1_{x_1} \circ \cdots \circ E^{N-1}_{x_{N-1}} \circ [\mu_N \circ E^N \mu_{N+1}] \cdots.$$ (3.12)
does not hold. However, two typical processes admit the equality (3.12). One is the additive process $\circ = +$. The other is the multiplicative process $\circ = \times$. Throughout the remainder, we are mainly concerned with the class of processes which do not admit the equality (3.12).

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

$$
\begin{align*}
W_n(x_n) &= \text{Max}_\pi [\mu_n \circ E^n[\mu_{n+1} \circ \cdots \circ E^{N-1}[\mu_N \circ E^N[\mu_{N+1} \cdots ]](i)_m \ | \ n \leq m \leq N ]} \\
W_{n+1}(x_{n+1}) &= \mu_{N+1}(x_{n+1})
\end{align*}
$$

(3.13)

(3.14)

where maximization is taken for all Markov policies $\pi = \{\pi_n, \ldots, \pi_N\}$. Then we have the recursive equation between value $w_n(x)$ and one-variable function $w_{n+1}(\cdot)$:

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
Theorem 3

\[ w_n(x) = \max_{u \in U} \left[ \mu_n(x, u) \circ \sum_{y \in X} w_{n+1}(y) p(y|x, u) \right] \quad (3.15) \]

\[ w_{N+1}(x) = \mu_{N+1}(x) \quad x \in X. \quad (3.16) \]

Proof. This is the recursive formula for a deterministic dynamic program under monotone relation \( \circ \).

The validity of recursive formula (3.15) is equivalent to the validity of equality

\[ \max_\pi \left[ \mu_1 \circ E^1[\mu_2 \circ \cdots \circ E^{N-1}[\mu_N \circ E^N \mu_{N+1}] \cdots] \right] \]

\[ = \max_\pi \left[ \mu_1 \circ E^1 \left[ \max_\pi_2 \left[ \mu_2 \circ \cdots \circ E^{N-1} \left[ \max_\pi_N \left[ \mu_N \circ E^N \mu_{N+1} \right] \cdots \right] \right] \right] \]  

\[ (u_n = \pi_n(x_n) \quad 1 \leq n \leq N). \quad (3.17) \]

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 (Figure 2). This generates the following a priori cdp:

Maximize \( E^u_{x_1} [\mu_1(x_1, u_1) \circ E^u_{x_2} [\mu_2(x_2, u_2) \circ \cdots \circ E^u_{x_N} [\mu_N(x_N, u_N) \circ \mu_{N+1}] \cdots]] \)

subject to \( (i)_n \ x_{n+1} \sim p(\cdot|x_n, u_n), \ u_n \in U \quad 1 \leq n \leq N. \quad (3.18) \)

Here we note that

\[ E^u_x [\mu_n(x, u) \circ \mu] = \sum_{y \in X} [\mu_n(x, u) \circ \mu(y)] p(y|x, u) \quad \text{for} \ \mu = \mu(\cdot). \quad (3.19) \]

We use the following short notations:

\[ E^n[\mu_n \circ \mu] := E^u_{x_n} [\mu_n(x_n, u_n) \circ \mu] \quad 1 \leq n \leq N. \quad (3.20) \]

Henceforth, the objective function in (3.18) is written as follows:

\[ E^1[\mu_1 \circ E^2[\mu_2 \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}] \cdots]] \]

\[ := E^u_{x_1} [\mu_1(x_1, u_1) \circ E^u_{x_2} [\mu_2(x_2, u_2) \circ \cdots \circ E^u_{x_N} [\mu_N(x_N, u_N) \circ \mu_{N+1}] \cdots]]. \quad (3.21) \]

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

\[ E^n[\mu_n \circ \mu] = E^u_{x_n} [\mu_n(x_n, u_n) \circ \mu], \quad u_n = \pi_n(x_n) \quad 1 \leq n \leq N. \quad (3.22) \]

Thus the a priori conditional expected value is the other backward iterative summation:

\[ E^1[\mu_1 \circ E^2[\mu_2 \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}] \cdots]] \]

\[ = \sum_{x_1 \in X} [\mu_1(x_1, u_1)] \circ \sum_{x_2 \in X} [\mu_2(x_2, u_2)] \circ \cdots \circ \sum_{x_N \in X} [\mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1})] p(x_{N+1}|x_N, u_N) \]

\[ \circ [p(x_N|x_{N-1}, u_{N-1}) \cdots p(x_2|x_1, u_1)] p(x_2|x_1, u_1)]. \quad (3.23) \]
We remark that the a priori conditional expected value (3.23) is not always identical with the a posteriori (3.11). It may also different from the so-called expected value (3.3). However, three expected values (3.3),(3.10),(3.23) 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:

\[
W_n(x_n) = \max \left[ E^n[N \circ E^{n+1}[\mu_{n} \circ \cdots \circ E^N[\mu_N \\
\circ \mu_{N+1}]]] | (i)_m, \ (ii)_m \ n \leq m \leq N \right] \tag{3.24}
\]

\[
W_{N+1}(x_{N+1}) = \mu_{N+1}(x_{N+1}) \tag{3.25}
\]

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

Theorem 4

\[
W_n(x) = \max_{u \in U} \sum_{y \in X} [\mu_n(x, u) \circ W_{n+1}(y)]p(y|x, u) \tag{3.26}
\]

\[
x \in X, \ n = 1, 2, \ldots, N \\
W_{N+1}(x) = \mu_{N+1}(x) \quad x \in X. \tag{3.27}
\]

Proof This is also the recursive formula for a deterministic dynamic program under monotone relation $\circ$. \hfill \Box

The recursive formula (3.26) with (3.27) states the equality

\[
\max_{\pi_1} E^1[\mu_1 \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}]] = \max_{\pi_2} E^2[\mu_2 \circ \cdots \circ E^{N}[\mu_N \circ \mu_{N+1}]]. \tag{3.28}
\]

Now let us consider the difference between the two cdps from a practical viewpoint. Throughout the a priori cdp, the conditional expectation is taken prior to take-action. Thus the a priori cdp is available when the decision-maker knows the remaining total fuzziness before take-action (possesses an a priori information on associating the current membership with the remaining total fuzziness). For instance, when the decision-maker gets an advance notice that the total fuzziness is associatively evaluated through the immediate membership and the remaining fuzziness, he/she chooses the a priori cdp. Otherwise, he/she has to draw a lottery for the remaining future, which enables him/her to choose the a posteriori cdp.

4. Examples

In this section, we illustrate two approaches and two cdps on a three-state, two-action and two-stage process with Bellman and Zadeh’s data [5, pp. B154]:

\[
\begin{align*}
\mu_3(s_1) &= 0.3 & \mu_3(s_2) &= 1.0 & \mu_3(s_3) &= 0.8 \\
\mu_2(a_1) &= 1.0 & \mu_2(a_2) &= 0.6 \\
\mu_1(a_1) &= 0.7 & \mu_1(a_2) &= 1.0 \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>$x_t \setminus x_{t+1}$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$u_t = a_1$</th>
<th>$u_t = a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.9</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0.0</td>
<td>0.1</td>
<td>0.9</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
Since the invariant imbedding approach is discussed in [12], we give only the direct approach.

4.1. Direct recursive equation

Then the resulting optimal equation (2.8) reduces to the recursive equations:

\[
\begin{align*}
v_3(h_3) &= \mu_1(u_1) \land \mu_2(u_2) \land \mu_3(x_3) \\
v_2(h_2) &= \max_{u_2} \sum_{x_3} v_3(h_2, u_2, x_3)p(x_3|x_2, u_2) \\
v_1(x_1) &= \max_{u_1} \sum_{x_2} v_2(x_1, u_1, x_2)p(x_2|x_1, u_1)
\end{align*}
\]

where

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

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

\[
\begin{array}{c|cccc}
u_3(\cdot, a_1, \cdot, u_2, x_3) & s_1 & s_2 & s_3 \\ \hline
u_2 & 0.3 & 0.7 & 0.7 \\
a_1 & 0.3 & 0.6 & 0.6 \\
a_2 & 0.3 & 0.6 & 0.6 \\
\end{array}
\]

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

\[
\begin{array}{c|cccc}
u_2(\cdot, \cdot, u_1, x_2) & s_1 & s_2 & s_3 \\ \hline
u_1 & 0.57, a_2, 0.7, a_1, 0.57, a_2 \\
a_1 & 0.57, a_2, 0.82, a_1, 0.57, a_2 \\
\end{array}
\]

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

\[
v_1(s_1) = 0.795, \quad v_1(s_2) = 0.595, \quad v_1(s_3) = 0.583.
\]

\[
\nu_1^*(s_1) = a_2, \quad \nu_1^*(s_2) = a_2, \quad \nu_1^*(s_3) = a_1.
\]

This result is also verified in Figures 3, 4 and 5. The optimal primitive policy \(\nu^* = \{\nu_1^*, \nu_2^*\}\) yields an optimal general policy \(\sigma^* = \{\sigma_1^*, \sigma_2^*\}\):

\[
\sigma_1^*(s_1) = a_2, \quad \sigma_1^*(s_2) = a_2, \quad \sigma_1^*(s_3) = a_1
\]

\[
\sigma_2^*(s_1, s_1) = a_2, \quad \sigma_2^*(s_2, s_1) = a_2, \quad \sigma_2^*(s_3, s_1) = a_2
\]

\[
\sigma_2^*(s_1, s_2) = a_1, \quad \sigma_2^*(s_3, s_2) = a_1, \quad \sigma_2^*(s_3, s_2) = a_1
\]

\[
\sigma_2^*(s_1, s_3) = a_1, a_2, \quad \sigma_2^*(s_2, s_3) = a_2, \quad \sigma_2^*(s_3, s_3) = a_2.
\]

Note that this optimal general policy \(\sigma^*\) is Markov.
4.2. Bellman and Zadeh’s process

As an example of a posteriori cdp, we cite Bellman and Zadeh’s stochastic model [5, pp. B154]. They have got a head start on decision-making in a fuzzy environment. Afterwards, some related models [6], [7], [15], [12] have followed [5]. We consider Bellman and Zadeh’s model as the following a posteriori cdp:

Maximize \[ \left[ \mu_1(u_1) \land E_{x_1}^{u_1} \left[ \mu_2(u_2) \land E_{x_2}^{u_2} \mu_3 \right] \right] \]

subject to \( (i)_n \quad x_{n+1} \sim p(\cdot \mid x_n, u_n), \quad u_n \in \{a_1, a_2\} \quad n = 1, 2 \)  \( (4.8) \)
Figure 4: All two-stage behaviors from $s_2$ and selection of maximum branch

Then, (3.17) reduces to

$$\text{Max}_{\pi_1} \left[ \mu_1(u_1) \land E^{u_1}_{\pi_2} \left[ \mu_2(u_2) \land E^{u_2}_{\pi_3} \mu_3 \right] \right]$$

$$= \text{Max}_{\pi_1} \left[ \mu_1(u_1) \land E^{u_1}_{\pi_2} \text{Max}_{\pi_2} \left[ \mu_2(u_2) \land E^{u_2}_{\pi_3} \mu_3 \right] \right]$$

$$= \text{Max}_{\pi_1} \left[ \mu_1(u_1) \land E^{u_1}_{\pi_2} \mu_2 \right]$$

$$= \text{Max}_{\pi_1} \left[ \mu_1(u_1) \land E^{u_1}_{\pi_2} \mu_2 \right]$$

(4.9)

This is equivalent to the recurrence equations:

$$u_n = \pi_n(x_n) \quad n = 1, 2.$$
Figure 5 : All two-stage behaviors from $s_3$ and selection of maximum branch

$$w_3(x_3) = \mu_3(x_3)$$
$$w_2(x_2) = \max_{a_2} [\mu_2(u_2) \land \sum_{x_3} w_3(x_3)p(x_3|x_2, u_2)] \quad (4.10)$$
$$w_1(x_1) = \max_{a_1} [\mu_1(u_1) \land \sum_{x_2} w_2(x_2)p(x_2|x_1, u_1)].$$

Bellman and Zadeh [5, pp. B154] give the following optimal solution through the backward equations:

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
However, Iwamoto and Fujita [12] have given an exact expression of \( w_1(x_1) \), \( \pi_1(x_1) \) as follows:

\[
\begin{align*}
  w_1(s_1) &= 0.798, \\
  w_1(s_2) &= 0.622, \\
  w_1(s_3) &= 0.622 \\
  \pi_1(s_1) &= a_2, \\
  \pi_1(s_2) &= a_1 \text{ or } a_2, \\
  \pi_1(s_3) &= a_1.
\end{align*}
\] (4.15)

This fact is also verified in Figures 6, 7, 8 and 9.

\[
\begin{align*}
\text{Max}_{u_2} \left[ \mu_2(u_2) \wedge \sum_{x_3} \mu_3(x_3)p(x_3|x_2,u_2) \right]
\end{align*}
\]

<table>
<thead>
<tr>
<th>history</th>
<th>ter.</th>
<th>( \times p_1 )</th>
<th>( E^2 )</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.42</td>
<td>0.42</td>
</tr>
<tr>
<td>s2</td>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1</td>
<td>0.3</td>
<td>0.03</td>
<td>0.93</td>
<td>0.6</td>
</tr>
<tr>
<td>s2</td>
<td>1.0</td>
<td>0.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td>0.8</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>s2</td>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td>0.8</td>
<td>0.72</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.42</td>
<td>0.42</td>
</tr>
<tr>
<td>s2</td>
<td>1.0</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1</td>
<td>0.3</td>
<td>0.03</td>
<td>0.75</td>
<td>0.6</td>
</tr>
<tr>
<td>s2</td>
<td>1.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td>0.8</td>
<td>0.72</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: One-stage a posteriori conditional decision tree from \( s_1, s_2 \) and \( s_3 \)

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
\[
\text{Max}_{u_1} \left[ \mu_1(u_1) \land \sum_{x_2} \text{Max}_{u_2} \left\{ \mu_2(u_2) \land \sum_{x_3} \mu_3(x_3)p(x_3|x_2, u_2) \right\} p(x_2|s_1, u_1) \right]
\]

<table>
<thead>
<tr>
<th>history</th>
<th>ter.</th>
<th>(\times p_2)</th>
<th>(E^2)</th>
<th>min</th>
<th>(\times p_1)</th>
<th>(E^1)</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.03</td>
<td>0.93</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.9</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.0</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.82</td>
<td>0.82</td>
<td>0.082</td>
<td>0.622</td>
<td>0.622</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.75</td>
<td>0.6</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.0</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.72</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.82</td>
<td>0.82</td>
<td>0.738</td>
<td>0.798</td>
<td>0.798</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td>0.42</td>
<td>0.42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.75</td>
<td>0.6</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.0</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.72</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Two-stage a posteriori conditional decision tree from \(s_1\)
\[
\max_{u_1} \left[ \mu_1(u_1) \land \max_{x_2} \left\{ \mu_2(u_2) \land \max_{x_2} \mu_3(x_3)p(x_3|x_2, u_2) \right\} p(x_2|s_2, u_1) \right]
\]

<table>
<thead>
<tr>
<th>history</th>
<th>ter.</th>
<th>\times p_2</th>
<th>E^2</th>
<th>min</th>
<th>\times p_1</th>
<th>E^1</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.42</td>
<td>0.42</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td>0.82</td>
<td>0.622</td>
<td>0.622</td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.03</td>
<td>0.82</td>
<td>0.82</td>
<td>0.082</td>
<td>0.75</td>
<td>0.6</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td>0.082</td>
<td></td>
<td>0.622</td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.03</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.0</td>
<td>0.75</td>
<td>0.6</td>
<td>0.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.72</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td>0.082</td>
<td>0.24</td>
<td>0.42</td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td>0.82</td>
<td>0.82</td>
<td>0.082</td>
<td>0.75</td>
<td>0.6</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td>0.082</td>
<td></td>
<td>0.622</td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_1</td>
<td>0.3</td>
<td>0.24</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>s_2</td>
<td>1.0</td>
<td>0.1</td>
<td>0.42</td>
<td>0.42</td>
<td>0.082</td>
<td>0.75</td>
<td>0.6</td>
</tr>
<tr>
<td>s_3</td>
<td>0.8</td>
<td>0.08</td>
<td></td>
<td></td>
<td>0.093</td>
<td></td>
<td>0.6</td>
</tr>
</tbody>
</table>

Figure 8: Two-stage a posteriori conditional decision tree from \(s_2\)
\[
\text{Max}_{a_1} \left[ \mu_1(u_1) \land \sum_{x_2} \text{Max}_{u_2} \left\{ \mu_2(u_2) \land \sum_{x_3} \mu_3(x_3)p(x_3|x_2, u_2) \right\} p(x_2|s_3, u_1) \right]
\]

Figure 9: Two-stage a posteriori conditional decision tree from \(s_3\)

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
4.3. A priori process

As an a priori cdp for Bellman and Zadeh's process (4.8), we consider the following problem:

Maximize \( E_{x_1}^{u_1}[\mu_1(u_1) \land E_{x_2}^{u_2}[\mu_2(u_2) \land \mu_3]] \)

subject to \( (i)_{n} \quad x_{n+1} \sim p(\cdot|x_n, u_n), \quad u_n \in \{a_1, a_2\} \quad n = 1, 2. \) \hspace{1cm} (4.17)

For the preceding data, the corresponding recursive equations

\[
W_3(x_3) = \mu_3(x_3) \\
W_2(x_2) = \text{Max}_{u_2} \sum_{x_3} [\mu_2(u_2) \land W_3(x_3)]p(x_3|x_2, u_2) \\
W_1(x_1) = \text{Max}_{u_1} \sum_{x_2} [\mu_1(u_1) \land W_2(x_2)]p(x_2|x_1, u_1)
\]

have the solution

\[
W_3(s_1) = 0.3, \quad W_3(s_2) = 1.0, \quad W_3(s_3) = 0.8, \hspace{1cm} (4.19)
\]

\[
W_2(s_1) = 0.57, \quad W_2(s_2) = 0.82, \quad W_2(s_3) = 0.57 \hspace{1cm} (4.20)
\]

\[
\pi^*_2(s_1) = a_2, \quad \pi^*_2(s_2) = a_1, \quad \pi^*_2(s_3) = a_2, \hspace{1cm} (4.21)
\]

\[
W_1(s_1) = 0.795, \quad W_1(s_2) = 0.595, \quad W_1(s_3) = 0.583 \hspace{1cm} (4.22)
\]

\[
\pi^*_1(s_1) = a_2, \quad \pi^*_1(s_2) = a_2, \quad \pi^*_1(s_3) = a_1. \hspace{1cm} (4.23)
\]

This solution is also illustrated in Figures 10, 11, 12 and 13. (Figures 12 and 13 are omitted.)

\[
\text{Max}_{u_2} \left[ \sum_{x_3} [\mu_2(u_2) \land \mu_3(x_3)]p(x_3|x_2, u_2) \right]
\]

<table>
<thead>
<tr>
<th>history</th>
<th>ter.</th>
<th>min</th>
<th>( \times p )</th>
<th>( E^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_1 )</td>
<td>0.3</td>
<td>0.3</td>
<td>0.24</td>
<td>0.42</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.57</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>0.8</td>
<td>0.8</td>
<td>0.08</td>
<td>0.82</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0.3</td>
<td>0.3</td>
<td>0.24</td>
<td>0.36</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1.0</td>
<td>1.0</td>
<td>0.06</td>
<td>0.42</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>0.8</td>
<td>0.8</td>
<td>0.06</td>
<td>0.57</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0.3</td>
<td>0.3</td>
<td>0.24</td>
<td>0.03</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.54</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>0.8</td>
<td>0.8</td>
<td>0.08</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Figure 10: One-stage a priori conditional decision tree from \( s_1, s_2 \) and \( s_3 \)

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
\[
\max_{u_1} \left[ \sum_{x_2} \left\{ \mu_1(u_1) \land \max_{u_2} \sum_{x_3} \left\{ \mu_2(u_2) \land \mu_3(x_3) \right\} p(x_3|x_2, u_2) \right\} p(x_2|s_1, u_1) \right]
\]

$s_1$  

<table>
<thead>
<tr>
<th>history</th>
<th>ter.</th>
<th>min</th>
<th>$\times p_2$</th>
<th>$E^2$</th>
<th>min</th>
<th>$\times p_1$</th>
<th>$E^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1.0</td>
<td>0.3</td>
<td>0.24</td>
<td>0.42</td>
<td>0.57</td>
<td>0.456</td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td>0.8</td>
<td>0.3</td>
<td>0.03</td>
<td>0.57</td>
<td>0.7</td>
<td>0.07</td>
<td>0.583</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0.7</td>
<td>0.3</td>
<td>0.24</td>
<td>0.36</td>
<td>0.57</td>
<td>0.057</td>
<td></td>
</tr>
</tbody>
</table>

Figure 11: Two-stage a priori conditional decision tree from $s_1$
5. Concluding Remarks

We remark that for the case \( \circ = \land \)

\[
W_n(x) \leq w_n(x) \quad 1 \leq n \leq N
\]

(5.1)

and that for the case \( \circ = \lor \)

\[
W_n(x) \geq w_n(x) \quad 1 \leq n \leq N.
\]

(5.2)

We note that the equality

\[
\sum_{x \in X} [\lambda \circ g(x)] p(x) = \lambda \circ \sum_{x \in X} g(x) p(x) \quad \circ = +, \times
\]

holds for a real constant \( \lambda \), a function \( g : X \to \mathbb{R} \) and a probability function \( p \). However, for any binary relation \( \circ \), the equality

\[
\sum_{x \in X} [\lambda \circ g(x)] p(x) = \lambda \circ \sum_{x \in X} g(x) p(x)
\]

does not always hold. (For further details, see [11].)

Acknowledgements

The authors would like to thank two anonymous referees for their careful reading and useful comments on the original manuscript.

References


S. Iwamoto, K. Tsurusaki and T. Fujita: On Markov policies for minimax decision processes. under consideration.


Seiichi IWAMOTO
Department of Economic Engineering
Faculty of Economics
Kyushu University 27
Fukuoka 812-8581, Japan
E-mail: iwamoto@en.kyushu-u.ac.jp

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.