Journal of the Operations Research
Society of Japan
Vol. 42, No. 2, June 1999

CONDITIONAL DECISION-MAKING IN FUZZY ENVIRONMENT

Seiichi Iwamoto Kazuyoshi Tsurusaki
Kyushu University
Toshiharu Fujita
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 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 :

$$\begin{aligned}
&N \geq 2 \text{ is an integer; the } \textit{total number of stages}\\
&X = \{s_1, s_2, \ldots, s_l\} \text{ is a finite } \textit{state space}\\
&U = \{a_1, a_2, \ldots, a_k\} \text{ is a finite } \textit{action space}\\
&\mu_n : X \times U \to [0, 1] \text{ is an } \textit{n-th membership function} \quad (1 \leq n \leq N)\\
&\mu_{N+1} : X \to [0, 1] \text{ is a } \textit{goal membership function} \qquad (2.1)\\
&\circ : [0, 1] \times [0, 1] \to [0, 1] \text{ is a } \textit{associative} \text{ binary relation}\\
&\quad \text{with a } \textit{left-identity element } \iota\\
&\quad : \lambda \circ (\mu \circ \nu) = (\lambda \circ \mu) \circ \nu, \quad \iota \circ \lambda = \lambda \ \ \forall \lambda \in [0, 1]\\
&p \text{ is a } \textit{Markov transition law}\\
&\quad : p(y|x, u) \geq 0 \ \ \forall (x, u, y) \in X \times U \times X, \quad \sum_{y \in X} p(y|x, u) = 1 \ \ \forall (x, u) \in X \times U\\
&\quad y \sim p(\cdot\, |x, u) \text{ denotes that next state } y \text{ conditioned on state } x \text{ and action } u\\
&\quad \text{appears with probability } p(y|x, u).
\end{aligned}$$

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

$$\begin{aligned}
&\text{Maximize} \quad E^{\sigma}_{x_1}[\,\mu_1 \circ \mu_2 \circ \cdots \circ \mu_N \circ \mu_{N+1}\,]\\
&\text{subject to} \quad (\mathrm{i})_{\mathrm{n}} \ \ x_{n+1} \sim p(\cdot |\, x_n, u_n), \ \ u_n \in U \quad 1 \leq n \leq N
\end{aligned} \tag{2.2}$$

where E^{σ} denotes the expectation (summation) operator on $X \times X \cdots \times X$ (N-times) induced from the conditional probability functions $p(x_{n+1}|x_n, u_n)$, a *general* policy $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_N\}$ 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 :

$$v_n(h_n) = \operatorname*{Max}_{\nu} E^{\nu}_{h_n}[\,\mu_1 \circ \cdots \circ \mu_N \circ \mu_{N+1} \,|\, (\mathrm{i})_{\mathrm{m}} \ \ n \leq m \leq N\,] \tag{2.3}$$

$$h_n \in H_n, \quad 1 \leq n \leq N$$

$$v_{N+1}(h_{N+1}) = \mu_1(x_1, u_1) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \quad h_{N+1} \in H_{N+1} \tag{2.4}$$

where the sequence of action and state $(u_n, x_{n+1}, u_{n+1}, \dots, u_N, x_{N+1})$ after starting state h_n is governed stochastically by a *primitive* policy $\nu = \{\nu_n, \nu_{n+1}, ..., \nu_N\}$ consisting of decision functions

$$\nu_m : H_m \to U \quad n \leq m \leq N \tag{2.5}$$

as follows :

$$\begin{aligned} &\nu_n(h_n) = u_n \;\to\; p(\cdot\,|x_n, u_n) \sim x_{n+1} \\ &\to\; \nu_{n+1}(h_{n+1}) = u_{n+1} \;\to\; p(\cdot\,|x_{n+1}, u_{n+1}) \sim x_{n+2} \\ &\to\; \cdots \;\to\; \nu_N(h_N) = u_N \;\to\; p(\cdot\,|x_N, u_N) \sim x_{N+1}. \end{aligned} \tag{2.6}$$

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

$$\begin{aligned} &E^{\nu}_{h_n}[\,\mu_1 \circ \cdots \circ \mu_N \circ \mu_{N+1} |\,(\mathrm{i})_{\mathrm{m}} \;\; n \leq m \leq N\,] \\ =& \sum\sum\cdots\sum_{(x_{n+1},\dots,x_{N+1})\in X\times\cdots\times X} \mu_1(x_1, u_1) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \\ &\qquad\qquad \times p(x_{n+1}|x_n, u_n) \cdots p(x_{N+1}|x_N, u_N). \end{aligned} \tag{2.7}$$

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

Theorem 1

$$v_n(h) = \operatorname*{Max}_{u\in U} \sum_{y\in X} v_{n+1}(h, u, y) p(y|x, u) \qquad h \in H_n, \quad n = 1, 2, \dots, N \tag{2.8}$$

$$v_{N+1}(h) = \mu_1(x_1, u_1) \circ \cdots \circ \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1}) \quad h \in H_{N+1}. \tag{2.9}$$

Proof The addition $a + b : R^1 \times R^1 \to R^1$ is commutative, associative, and monotone. These properties imply the validity of recursive formula (2.8). □

Solving the recursive equation (2.8), we have a *primitive* optimal policy

$$\nu^* = \{\nu_1^*, \nu_2^*, ..., \nu_N^*\}.$$

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

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

as follows :

$$\begin{aligned} &\sigma_1^*(x_1) := \nu_1^*(h_1) \;\; (\, h_1 = x_1 \,) \\ &\sigma_2^*(x_1, x_2) := \nu_2^*(h_2) \;\; (\, h_2 = (x_1, u_1, x_2), \; u_1 = \nu_1^*(h_1)\,) \\ &\sigma_3^*(x_1, x_2, x_3) := \nu_3^*(h_3) \;\; (\, h_3 = (h_2, u_2, x_3), \; u_2 = \nu_2^*(h_2)\,) \\ &\quad\cdots \\ &\sigma_N^*(x_1, x_2, \dots, x_N) := \nu_N^*(h_N) \;\; (\, h_N = (h_{N-1}, u_{N-1}, x_N), \; u_{N-1} = \nu_{N-1}^*(h_{N-1})\,). \end{aligned} \tag{2.10}$$

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 $(1 \le n \le N+1)$, $x_n \in X$ and $\lambda_n \in [0,1]$ the maximization problem :

$$v_n(x_n;\lambda_n) = \operatorname*{Max}_{\pi} E^{\pi}_{x_n,\lambda_n}[\,\lambda_n \circ \mu_n \circ \cdots \circ \mu_N \circ \mu_{N+1} |\, (\mathrm{i})_{\mathrm{m}} \quad n \le m \le N\,] \tag{2.11}$$

$$1 \le n \le N$$

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

Here the maximization is taken for all *Markov* policies π 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}, λ_{N+1}). Note that any Markov policy $\pi = \{\pi_n, \pi_{n+1}, ..., \pi_N\}$ on the augmented state space $X \times [0,1]$ is specified by a sequence of decision functions :

$$\pi_m : X \times [0,1] \to U \qquad n \le m \le N. \tag{2.13}$$

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

$$\begin{aligned} &E^{\pi}_{x_n,\lambda_n}[\,\lambda_n \circ \mu_n \circ \cdots \circ \mu_N \circ \mu_{N+1} \,|\, (\mathrm{i})_{\mathrm{m}} \quad n \le m \le N\,] \\ &= \sum_{(x_{n+1},\ldots,x_{N+1}) \in X \times \cdots \times X} \sum \cdots \sum \{[\lambda_n \circ \mu_n(x_n,u_n) \circ \cdots \circ \mu_N(x_N,u_N) \circ \mu_{N+1}(x_{N+1})] \\ &\qquad \times p(x_{n+1}|x_n,u_n) \cdots p(x_{N+1}|x_N,u_N)\} \end{aligned} \tag{2.14}$$

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 π and the starting state (x_n, λ_n) as follows :

$$\begin{aligned} &\pi_n(x_n,\lambda_n) = u_n \ \to \left\{ \begin{array}{l} p(\cdot\,|x_n,u_n) \sim x_{n+1} \\ \lambda_n \circ \mu_n(x_n,u_n) = \lambda_{n+1} \end{array} \right. \\ &\to\ \pi_{n+1}(x_{n+1},\lambda_{n+1}) = u_{n+1} \ \to \left\{ \begin{array}{l} p(\cdot\,|x_{n+1},u_{n+1}) \sim x_{n+2} \\ \lambda_{n+1} \circ \mu_{n+1}(x_{n+1},u_{n+1}) = \lambda_{n+2} \end{array} \right. \\ &\to\ \quad \cdots \\ &\to\ \pi_N(x_N,\lambda_N) = u_N \ \to \left\{ \begin{array}{l} p(\cdot\,|x_N,u_N) \sim x_{N+1} \\ \lambda_N \circ \mu_N(x_N,u_N) = \lambda_{N+1}. \end{array} \right. \end{aligned} \tag{2.15}$$

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

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

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

$$v_{N+1}(x;\lambda) = \lambda \circ \mu_{N+1}(x) \qquad x \in X, \quad 0 \le \lambda \le 1. \tag{2.17}$$

Proof By replacing the binary relation $\wedge$ 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

$$\pi^* = \{\pi_1^*, \pi_2^*, ..., \pi_N^*\}$$

on the one-dimensionally extended state space $X \times [0,\ 1]$. By projecting the optimal policy π^* onto the original history space

$$X \times X \times \cdots \times X \quad (N+1)\text{-times}$$

with starting state (x_1, ι), we obtain a *general* optimal policy

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

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 σ^* ([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 $\pi = \{\pi_1, \pi_2, ..., \pi_N\}$ is specified by a sequence of Markov decision functions :

$$\pi_n : X \to U \quad 1 \le n \le N. \tag{3.1}$$

We assume that the binary relation $\circ$ is monotone :

$$\mu < \nu \implies \lambda \circ \mu \le \lambda \circ \nu. \tag{3.2}$$

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 :

$$\begin{aligned} & E_{x_1}^{\pi}[\mu_1 \circ [\mu_2 \circ \cdots \circ [\mu_N \circ \mu_{N+1}] \cdots]] \\ = & \sum_{(x_2,\ldots,x_{N+1}) \in X \times \cdots \times X} \sum \cdot \cdot \sum \{[\mu_1(x_1,u_1) \circ [\mu_2(x_2,u_2) \circ \cdots \circ [\mu_N(x_N,u_N) \circ \mu_{N+1}(x_{N+1})] \cdots]] \\ & \qquad \times p(x_2|x_1,u_1)p(x_3|x_2,u_2) \cdots p(x_{N+1}|x_N,u_N)\} \end{aligned} \tag{3.3}$$

where the sequence of controls is determined through Markov policy π :

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

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:

$$\begin{aligned}\text{Maximize} \quad & \mu_1(x_1,u_1)\circ E^{u_1}_{x_1}[\mu_2(x_2,u_2)\circ\cdots\circ E^{u_{N-1}}_{x_{N-1}}[\mu_N(x_N,u_N)\\ & \qquad\qquad\qquad\qquad\qquad\circ E^{u_N}_{x_N}\mu_{N+1}]\cdots]\\ \text{subject to} \quad & (\mathrm{i})_{\mathrm{n}} \quad x_{n+1}\sim p(\cdot|\,x_n,u_n),\ u_n\in U \quad 1\le n\le N \end{aligned} \tag{3.4}$$

Here we note that

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

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

$$E^n\mu \;:=\; E^{u_n}_{x_n}\mu \tag{3.6}$$

$$\mu_n\circ E^n\mu \;:=\; \mu_n(x_n,u_n)\circ E^n\mu \quad 1\le n\le N. \tag{3.7}$$

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

$$\begin{aligned}&\mu_1\circ E^1[\mu_2\circ\cdots\circ E^{N-1}[\mu_N\circ E^N\mu_{N+1}]\cdots]\\ :=\;& \mu_1(x_1,u_1)\circ E^{u_1}_{x_1}[\mu_2(x_2,u_2)\circ\cdots\circ E^{u_{N-1}}_{x_{N-1}}[\mu_N(x_N,u_N)\circ E^{u_N}_{x_N}\mu_{N+1}]\cdots].\end{aligned} \tag{3.8}$$

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

$$E^n\mu=E^{u_n}_{x_n}\mu, \qquad u_n=\pi_n(x_n) \quad 1\le n\le N. \tag{3.9}$$

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

$$\begin{aligned}&\mu_1\circ E^1[\mu_2\circ\cdots\circ E^{N-1}[\mu_N\circ E^N\mu_{N+1}]\cdots]\\ =\;& \mu_1(x_1,u_1)\circ\sum_{x_2\in X}[\mu_2(x_2,u_2)\circ\cdots[\mu_{N-1}(x_{N-1},u_{N-1})\circ\\ &\quad\sum_{x_N\in X}[\mu_N(x_N,u_N)\circ\sum_{x_{N+1}\in X}\mu_{N+1}(x_{N+1})p(x_{N+1}|x_N,u_N)]\\ &\qquad\qquad\qquad p(x_N|x_{N-1},u_{N-1})]\cdots]p(x_2|x_1,u_1)\\ &\qquad\qquad\qquad (u_n=\pi_n(x_n) \quad 1\le n\le N).\end{aligned} \tag{3.10}$$

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

$$\begin{aligned}&E^\pi_{x_1}[\mu_1\circ[\mu_2\circ\cdots\circ[\mu_N\circ\mu_{N+1}]\cdots]]\\ =\;& \sum\sum_{(x_2,\ldots,x_{N+1})\in X\times\cdots\times X}\cdot\cdot\sum\{[\mu_1(x_1,u_1)\circ[\mu_2(x_2,u_2)\circ\cdots\circ[\mu_N(x_N,u_N)\circ\mu_{N+1}(x_{N+1})]\cdots]]\\ &\qquad\times p(x_2|x_1,u_1)p(x_3|x_2,u_2)\cdots p(x_{N+1}|x_N,u_N)\}\\ &\qquad\qquad (u_n=\pi_n(x_n) \quad 1\le n\le N).\end{aligned} \tag{3.11}$$

We note that in general the equality

$$\begin{aligned}&E^\pi_{x_1}[\mu_1\circ[\mu_2\circ\cdots\circ[\mu_N\circ\mu_{N+1}]\cdots]]\\ =\;& \mu_1\circ E^1[\mu_2\circ\cdots\circ E^{N-1}[\mu_N\circ E^N\mu_{N+1}]\cdots]\end{aligned} \tag{3.12}$$

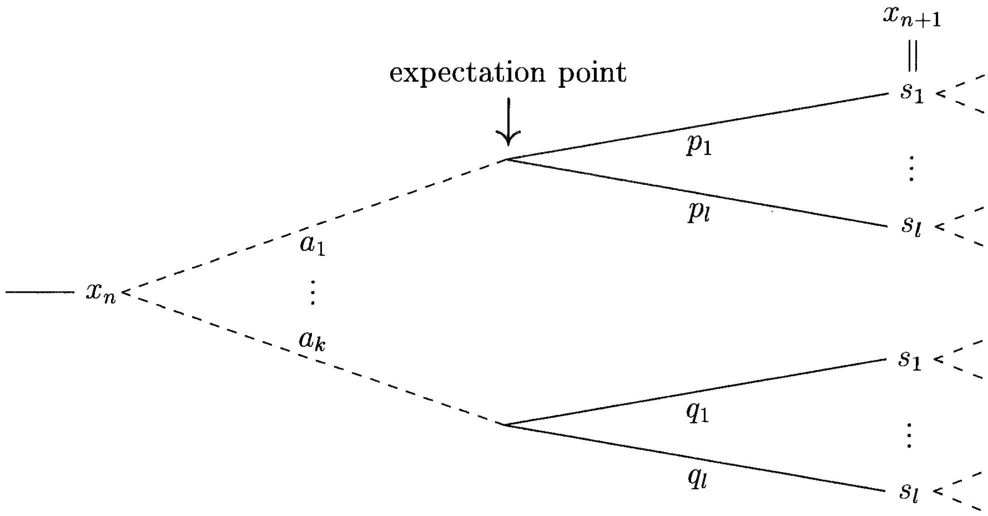


Figure 1 : Conditional expectation after take-action

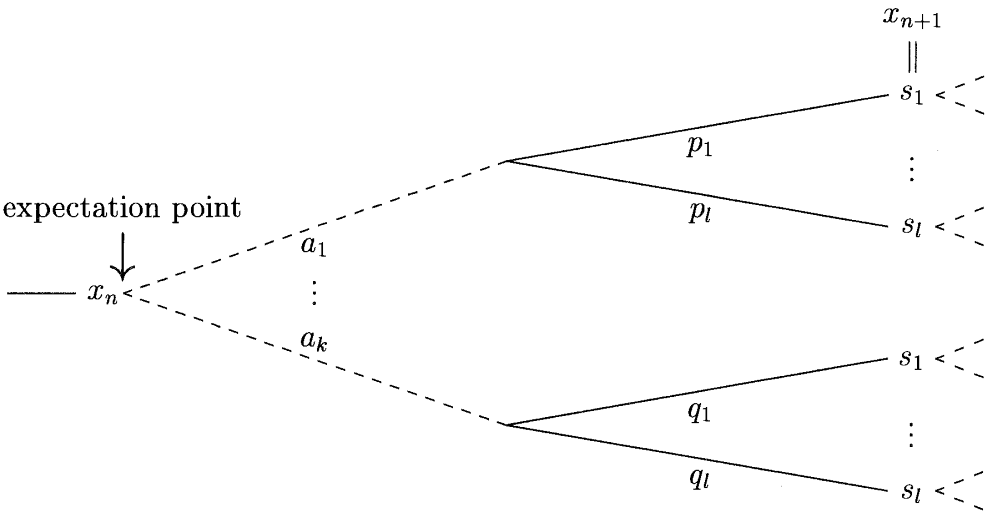


Figure 2 : Conditional expectation before take-action

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 :

$$w_n(x_n) = \operatorname*{Max}_{\pi} [\, \mu_n \circ E^n[\mu_{n+1} \circ \cdots \circ E^{N-1}[\mu_N \circ E^N \mu_{N+1}\,] \cdots] \mid (\mathrm{i})_{\mathrm{m}} \;\; n \leq m \leq N \,] \tag{3.13}$$

$$w_{N+1}(x_{N+1}) = \mu_{N+1}(x_{N+1}), \tag{3.14}$$

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

Theorem 3

$$w_n(x) = \operatorname*{Max}_{u \in U}[\mu_n(x,u) \circ \sum_{y \in X} w_{n+1}(y)p(y|x,u)] \tag{3.15}$$
$$x \in X, \quad n = 1, 2, \ldots, N$$
$$w_{N+1}(x) = \mu_{N+1}(x) \qquad x \in X. \tag{3.16}$$

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

$$\begin{aligned} & \operatorname*{Max}_{\pi} [\, \mu_1 \circ E^1[\mu_2 \circ \cdots \circ E^{N-1}[\mu_N \circ E^N \mu_{N+1}\,] \cdots]\,] \\ = \; & \operatorname*{Max}_{\pi_1} [\, \mu_1 \circ E^1 \operatorname*{Max}_{\pi_2} [\mu_2 \circ \cdots \circ E^{N-1} \operatorname*{Max}_{\pi_N} [\mu_N \circ E^N \mu_{N+1}\,] \cdots]\,] \\ & (u_n = \pi_n(x_n) \quad 1 \leq n \leq N). \end{aligned} \tag{3.17}$$

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 :

$$\begin{aligned} \text{Maximize} \quad & E_{x_1}^{u_1}[\mu_1(x_1, u_1) \circ E_{x_2}^{u_2}[\mu_2(x_2, u_2) \circ \cdots \\ & \circ E_{x_N}^{u_N}[\mu_N(x_N, u_N) \circ \mu_{N+1}\,] \cdots]\,] \\ \text{subject to} \quad & (\mathrm{i})_{\mathrm{n}} \quad x_{n+1} \sim p(\cdot|\, x_n, u_n), \; u_n \in U \quad 1 \leq n \leq N. \end{aligned} \tag{3.18}$$

Here we note that

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

We use the following short notations :

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

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

$$\begin{aligned} & E^1[\mu_1 \circ E^2[\mu_2 \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}] \cdots]\,] \\ := \; & E_{x_1}^{u_1}[\mu_1(x_1, u_1) \circ E_{x_2}^{u_2}[\mu_2(x_2, u_2) \circ \cdots \circ E_{x_N}^{u_N}[\mu_N(x_N, u_N) \circ \mu_{N+1}\,] \cdots]\,]. \end{aligned} \tag{3.21}$$

In the above, the relevant Markov policy π is compressed into the notation E^n :

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

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

$$\begin{aligned} & E^1[\mu_1 \circ E^2[\mu_2 \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}\,] \cdots]\,] \\ = \; & \sum_{x_2 \in X} [\, \mu_1(x_1, u_1) \circ \sum_{x_3 \in X} [\, \mu_2(x_2, u_2) \circ \cdots \circ \sum_{x_N \in X} [\, \mu_{N-1}(x_{N-1}, u_{N-1}) \\ & \quad \circ \sum_{x_{N+1} \in X} [\, \mu_N(x_N, u_N) \circ \mu_{N+1}(x_{N+1})\,] p(x_{N+1}|x_N, u_N) \\ & \quad] p(x_N|x_{N-1}, u_{N-1}) \cdots p(x_3|x_2, u_2)\,] p(x_2|x_1, u_1)\,]. \\ & (u_n = \pi_n(x_n) \quad 1 \leq n \leq N) \end{aligned} \tag{3.23}$$

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 :

$$W_n(x_n) = \underset{\pi}{\mathrm{Max}}[\, E^n[\,\mu_n \circ E^{n+1}[\mu_{n+1} \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}]\cdots]] \mid (\mathrm{i})_{\mathrm{m}},\ (\mathrm{ii})_{\mathrm{m}}\ \ n \leq m \leq N\,] \tag{3.24}$$

$$W_{N+1}(x_{N+1}) = \mu_{N+1}(x_{N+1}). \tag{3.25}$$

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

Theorem 4

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

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

$$W_{N+1}(x) = \mu_{N+1}(x) \qquad x \in X. \tag{3.27}$$

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

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

$$\begin{aligned} & \underset{\pi}{\mathrm{Max}}\, E^1[\mu_1 \circ E^2[\mu_2 \circ \cdots \circ E^N[\mu_N \circ \mu_{N+1}]\cdots]] \\ = \ & \underset{\pi_1}{\mathrm{Max}}\, E^1[\mu_1 \circ \underset{\pi_2}{\mathrm{Max}}\, E^2[\mu_2 \circ \cdots \circ \underset{\pi_N}{\mathrm{Max}}\, E^N[\mu_N \circ \mu_{N+1}]\cdots]]. \end{aligned} \tag{3.28}$$

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 cdp's on a three-state, two-action and two-stage process with Bellman and Zadeh's data [5, pp. B154]:

$$\mu_3(s_1) = 0.3 \qquad \mu_3(s_2) = 1.0 \qquad \mu_3(s_3) = 0.8 \tag{4.1}$$

$$\mu_2(a_1) = 1.0 \qquad \mu_2(a_2) = 0.6 \tag{4.2}$$

$$\mu_1(a_1) = 0.7 \qquad \mu_1(a_2) = 1.0 \tag{4.3}$$

$u_t = a_1$

$x_t \diagdown x_{t+1}$	s_1	s_2	s_3
s_1	0.8	0.1	0.1
s_2	0.0	0.1	0.9
s_3	0.8	0.1	0.1

$u_t = a_2$

$x_t \diagdown x_{t+1}$	s_1	s_2	s_3
s_1	0.1	0.9	0.0
s_2	0.8	0.1	0.1
s_3	0.1	0.0	0.9

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 :

$$\begin{aligned} v_3(h_3) &= \mu_1(u_1) \wedge \mu_2(u_2) \wedge \mu_3(x_3) \\ v_2(h_2) &= \operatorname*{Max}_{u_2} \sum_{x_3} v_3(h_2, u_2, x_3) p(x_3|x_2, u_2) \\ v_1(x_1) &= \operatorname*{Max}_{u_1} \sum_{x_2} v_2(x_1, u_1, x_2) p(x_2|x_1, u_1) \end{aligned} \tag{4.4}$$

where

$$\operatorname*{Max}_{u_n} = \operatorname*{Max}_{u_n \in \{a_1, a_2\}}, \qquad \sum_{x_n} = \sum_{x_n \in \{s_1, s_2, s_3\}}. \tag{4.5}$$

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

$v_3(\,\cdot\,, \underline{a_1}, \,\cdot\,, u_2, x_3)$

$u_2 \setminus x_3$	s_1	s_2	s_3
a_1	0.3	0.7	0.7
a_2	0.3	0.6	0.6

$v_3(\,\cdot\,, \underline{a_2}, \,\cdot\,, u_2, x_3)$

$u_2 \setminus x_3$	s_1	s_2	s_3
a_1	0.3	1.0	0.8
a_2	0.3	0.6	0.6

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

$v_2(\,\cdot\,, u_1, x_2),\ \nu_2^*(\,\cdot\,, u_1, x_2)$

$u_1 \setminus x_2$	s_1	s_2	s_3
a_1	0.57, a_2	0.7, a_1	0.57, a_2
a_2	0.57, a_2	0.82, a_1	0.57, a_2

Here we note that

$$v_3(x_1, u_1, x_2, u_2, x_3) = v_3(x_1', u_1, x_2', u_2, x_3) \qquad {}^{\forall}x_1, x_2, x_1', x_2' \in X$$

$$v_2(x_1, u_1, x_2) = v_2(x_1', u_1, x_2) \qquad {}^{\forall}x_1, x_1' \in X.$$

Finally, we get

$$v_1(s_1) = 0.795, \qquad v_1(s_2) = 0.595, \qquad v_1(s_3) = 0.583.$$

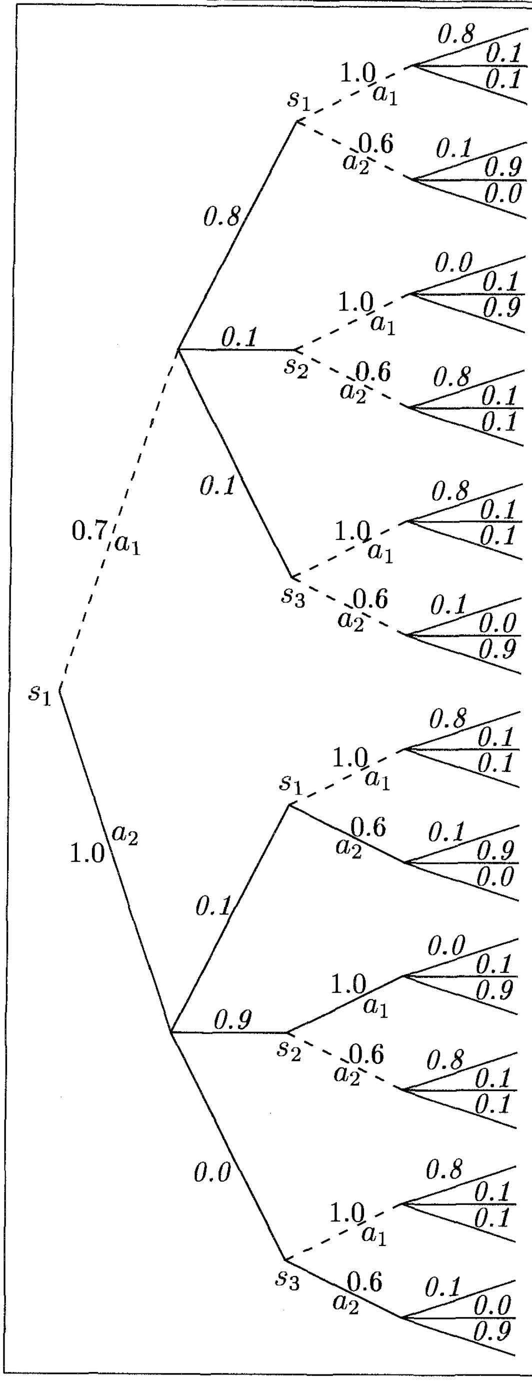
$$\nu_1^*(s_1) = a_2, \qquad \nu_1^*(s_2) = a_2, \qquad \nu_1^*(s_3) = a_1.$$

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 $\sigma^* = \{\sigma_1^*, \sigma_2^*\}$:

$$\sigma_1^*(s_1) = a_2, \qquad \sigma_1^*(s_2) = a_2, \qquad \sigma_1^*(s_3) = a_1 \tag{4.6}$$

$$\begin{aligned} &\sigma_2^*(s_1, s_1) = a_2, \quad \sigma_2^*(s_2, s_1) = a_2, \quad \sigma_2^*(s_3, s_1) = a_2 \\ &\sigma_2^*(s_1, s_2) = a_1, \quad \sigma_2^*(s_2, s_2) = a_1, \quad \sigma_2^*(s_3, s_2) = a_1 \\ &\sigma_2^*(s_1, s_3) = a_1, a_2 \quad \sigma_2^*(s_2, s_3) = a_2, \quad \sigma_2^*(s_3, s_3) = a_2. \end{aligned} \tag{4.7}$$

Note that this optimal general policy σ^* is Markov.



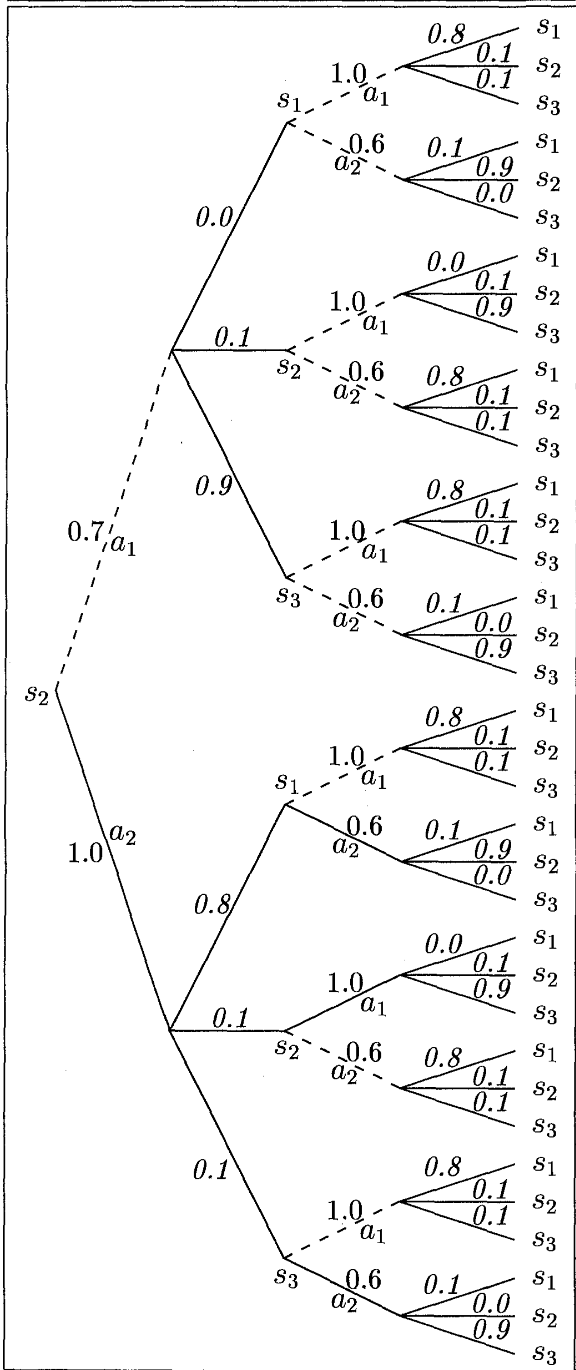
history	ter.	min	$\times p_2$	sub.	$\times p_1$	total
s_1	0.3	0.3	0.24			
s_2	1.0	0.7	0.07	0.38		
s_3	0.8	0.7	0.07		0.456	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.54	**0.57**		
s_3	0.8	0.6	0.0			
s_1	0.3	0.3	0.0			
s_2	1.0	0.7	0.07	**0.7**		
s_3	0.8	0.7	0.63		0.07	0.583
s_1	0.3	0.3	0.24			
s_2	1.0	0.6	0.06	0.36		
s_3	0.8	0.6	0.06			
s_1	0.3	0.3	0.24			
s_2	1.0	0.7	0.07	0.38		
s_3	0.8	0.7	0.07		0.057	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.0	**0.57**		
s_3	0.8	0.6	0.54			
s_1	0.3	0.3	0.24			
s_2	1.0	1.0	0.1	0.42		
s_3	0.8	0.8	0.08		0.057	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.54	**0.57**		
s_3	0.8	0.6	0.0			
s_1	0.3	0.3	0.0			
s_2	1.0	1.0	0.1	**0.82**		
s_3	0.8	0.8	0.72		0.738	**0.795**
s_1	0.3	0.3	0.24			
s_2	1.0	0.6	0.06	0.36		
s_3	0.8	0.6	0.06			
s_1	0.3	0.3	0.24			
s_2	1.0	1.0	0.1	0.42		
s_3	0.8	0.8	0.08		0.0	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.0	**0.57**		
s_3	0.8	0.6	0.54			

Figure 3 : All two-stage behaviors from s_1 and selection of maximum branch

4.2. Bellman and Zadeh's process

As an example of a posteriori cdp, we cite Bellman and Zadeh's stochastic model [5, p-p. 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 :

$$\begin{array}{lll} \text{Maximize} & [\,\mu_1(u_1) \wedge E_{x_1}^{u_1}\,[\mu_2(u_2) \wedge E_{x_2}^{u_2}\mu_3\,]\,] & \\ \text{subject to} & (\text{i})_n \;\; x_{n+1} \sim p(\cdot \mid x_n, u_n),\; u_n \in \{a_1, a_2\} \quad n = 1, 2 & (4.8) \end{array}$$



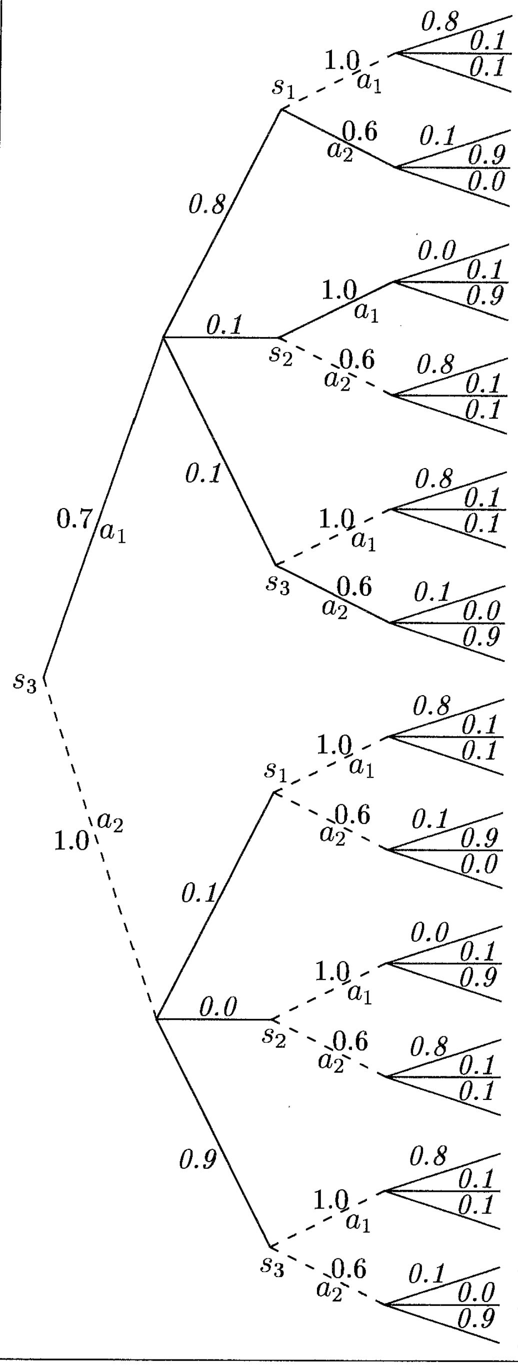
ter.	min	$\times p_2$	sub.	$\times p_1$	total
0.3	0.3	0.24			
1.0	0.7	0.07	0.38		
0.8	0.7	0.07		0.0	
0.3	0.3	0.03			
1.0	0.6	0.54	**0.57**		
0.8	0.6	0.0			
0.3	0.3	0.0			
1.0	0.7	0.07	**0.7**		
0.8	0.7	0.63		0.07	0.583
0.3	0.3	0.24			
1.0	0.6	0.06	0.36		
0.8	0.6	0.06			
0.3	0.3	0.24			
1.0	0.7	0.07	0.38		
0.8	0.7	0.07		0.513	
0.3	0.3	0.03			
1.0	0.6	0.0	**0.57**		
0.8	0.6	0.54			
0.3	0.3	0.24			
1.0	1.0	0.1	0.42		
0.8	0.8	0.08		0.456	
0.3	0.3	0.03			
1.0	0.6	0.54	**0.57**		
0.8	0.6	0.0			
0.3	0.3	0.0			
1.0	1.0	0.1	**0.82**		
0.8	0.8	0.72		0.082	**0.595**
0.3	0.3	0.24			
1.0	0.6	0.06	0.36		
0.8	0.6	0.06			
0.3	0.3	0.24			
1.0	1.0	0.1	0.42		
0.8	0.8	0.08		0.057	
0.3	0.3	0.03			
1.0	0.6	0.0	**0.57**		
0.8	0.6	0.54			

Figure 4 : All two-stage behaviors from s_2 and selection of maximum branch

Then, (3.17) reduces to

$$\begin{aligned} & \operatorname*{Max}_{\pi}[\,\mu_1(u_1) \wedge E^{u_1}_{x_1}[\mu_2(u_2) \wedge E^{u_2}_{x_2}\mu_3\,]\,] \\ = \; & \operatorname*{Max}_{\pi_1}[\,\mu_1(u_1) \wedge E^{u_1}_{x_1} \operatorname*{Max}_{\pi_2}[\mu_2(u_2) \wedge E^{u_2}_{x_2}\mu_3\,]\,] \qquad (4.9) \\ & (u_n = \pi_n(x_n) \quad n = 1, 2). \end{aligned}$$

This is equivalent to the recurrence equations :



history	ter.	min	$\times p_2$	sub.	$\times p_1$	total
s_1	0.3	0.3	0.24			
s_2	1.0	0.7	0.07	0.38		
s_3	0.8	0.7	0.07		0.456	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.54	**0.57**		
s_3	0.8	0.6	0.0			
s_1	0.3	0.3	0.0			
s_2	1.0	0.7	0.07	**0.7**		
s_3	0.8	0.7	0.63		0.07	**0.583**
s_1	0.3	0.3	0.24			
s_2	1.0	0.6	0.06	0.36		
s_3	0.8	0.6	0.06			
s_1	0.3	0.3	0.24			
s_2	1.0	0.7	0.07	0.38		
s_3	0.8	0.7	0.07		0.057	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.0	**0.57**		
s_3	0.8	0.6	0.54			
s_1	0.3	0.3	0.24			
s_2	1.0	1.0	0.1	0.42		
s_3	0.8	0.8	0.08		0.057	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.54	**0.57**		
s_3	0.8	0.6	0.0			
s_1	0.3	0.3	0.0			
s_2	1.0	1.0	0.1	**0.82**		
s_3	0.8	0.8	0.72		0.0	0.57
s_1	0.3	0.3	0.24			
s_2	1.0	0.6	0.06	0.36		
s_3	0.8	0.6	0.06			
s_1	0.3	0.3	0.24			
s_2	1.0	1.0	0.1	0.42		
s_3	0.8	0.8	0.08		0.513	
s_1	0.3	0.3	0.03			
s_2	1.0	0.6	0.0	**0.57**		
s_3	0.8	0.6	0.54			

Figure 5 : All two-stage behaviors from s_3 and selection of maximum branch

$$
\begin{aligned}
w_3(x_3) &= \mu_3(x_3) \\
w_2(x_2) &= \operatorname*{Max}_{u_2}[\,\mu_2(u_2) \wedge \sum_{x_3} w_3(x_3)p(x_3|x_2,u_2)\,] \qquad (4.10)\\
w_1(x_1) &= \operatorname*{Max}_{u_1}[\,\mu_1(u_1) \wedge \sum_{x_2} w_2(x_2)p(x_2|x_1,u_1)\,].
\end{aligned}
$$

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

$$w_2(s_1) = 0.6, \quad w_2(s_2) = 0.82, \quad w_2(s_3) = 0.6 \tag{4.11}$$

$$\pi_2(s_1) = a_1, \quad \pi_2(s_2) = a_1, \quad \pi_2(s_3) = a_2, \tag{4.12}$$

$$w_1(s_1) = 0.8, \quad w_1(s_2) = 0.62, \quad w_1(s_3) = 0.62 \tag{4.13}$$

$$\pi_1(s_1) = a_1, \quad \pi_1(s_2) = a_1 \text{ or } a_2, \quad \pi_1(s_3) = a_1. \tag{4.14}$$

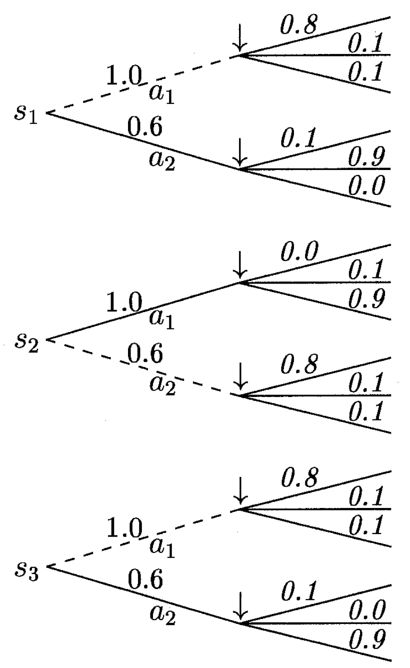
However, Iwamoto and Fujita [12] have given an exact expression of $w_1(x_1)$, $\pi_1(x_1)$ as follows :

$$w_1(s_1) = 0.798, \quad w_1(s_2) = 0.622, \quad w_1(s_3) = 0.622 \tag{4.15}$$

$$\pi_1(s_1) = a_2, \quad \pi_1(s_2) = a_1 \text{ or } a_2, \quad \pi_1(s_3) = a_1. \tag{4.16}$$

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

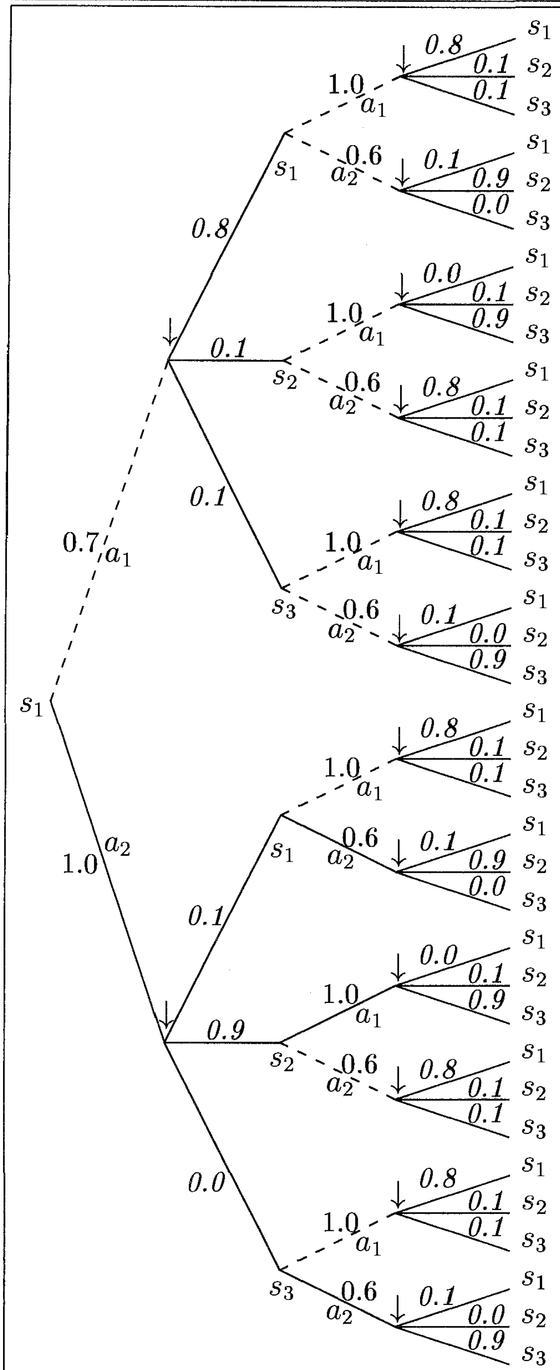
$$\underset{u_2}{\text{Max}}\left[\mu_2(u_2) \wedge \sum_{x_3} \mu_3(x_3)p(x_3|x_2, u_2)\right]$$



history	ter.	$\times p_1$	E^2	min
s_1	0.3	0.24		
s_2	1.0	0.1	0.42	0.42
s_3	0.8	0.08		
s_1	0.3	0.03		
s_2	1.0	0.9	0.93	**0.6**
s_3	0.8	0.0		
s_1	0.3	0.0		
s_2	1.0	0.1	0.82	**0.82**
s_3	0.8	0.72		
s_1	0.3	0.24		
s_2	1.0	0.1	0.42	0.42
s_3	0.8	0.08		
s_1	0.3	0.24		
s_2	1.0	0.1	0.42	0.42
s_3	0.8	0.08		
s_1	0.3	0.03		
s_2	1.0	0.0	0.75	**0.6**
s_3	0.8	0.72		

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

$$\underset{u_1}{\text{Max}}\left[\mu_1(u_1) \wedge \sum_{x_2} \underset{u_2}{\text{Max}}\left\{\mu_2(u_2) \wedge \sum_{x_3} \mu_3(x_3) p(x_3|x_2, u_2)\right\} p(x_2|s_1, u_1)\right]$$

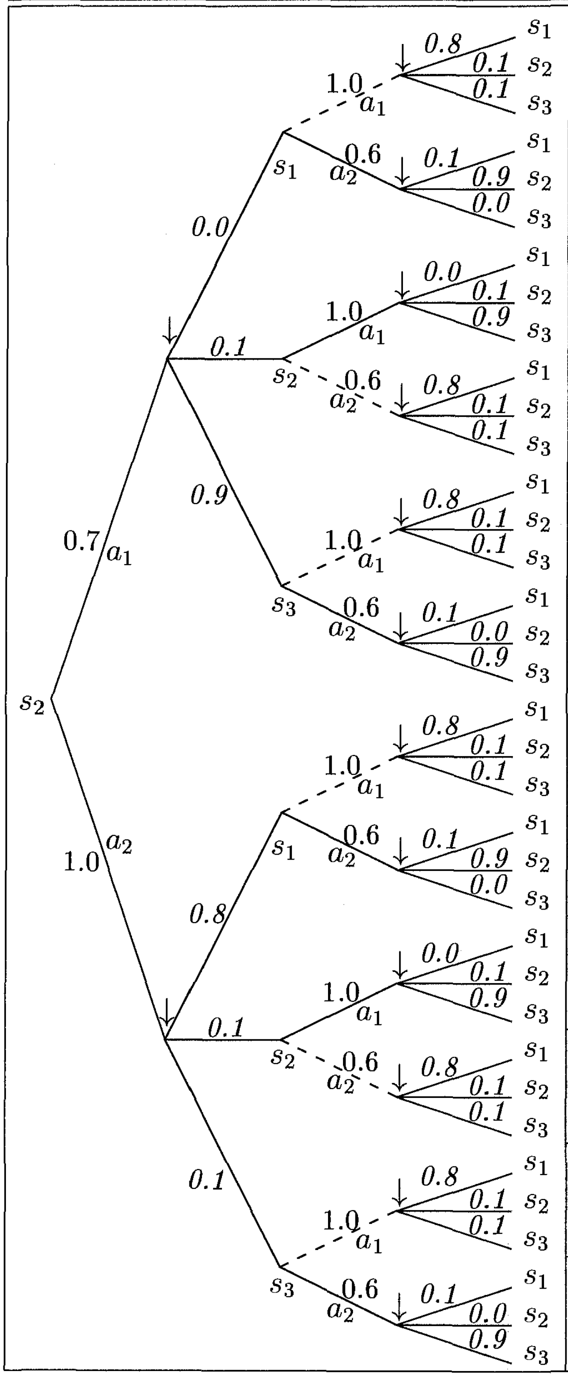


history	ter.	$\times p_2$	E^2	min	$\times p_1$	E^1	Min
s_1	0.3	0.24					
s_2	1.0	0.1	0.42	0.42			
s_3	0.8	0.08					
s_1	0.3	0.03			0.48		
s_2	1.0	0.9	0.93	**0.6**			
s_3	0.8	0.0					
s_1	0.3	0.0					
s_2	1.0	0.1	0.82	**0.82**			
s_3	0.8	0.72					
s_1	0.3	0.24			0.082	0.622	0.622
s_2	1.0	0.1	0.42	0.42			
s_3	0.8	0.08					
s_1	0.3	0.24					
s_2	1.0	0.1	0.42	0.42			
s_3	0.8	0.08					
s_1	0.3	0.03			0.06		
s_2	1.0	0.0	0.75	**0.6**			
s_3	0.8	0.72					
s_1	0.3	0.24					
s_2	1.0	0.1	0.42	0.42			
s_3	0.8	0.08					
s_1	0.3	0.03			0.06		
s_2	1.0	0.9	0.93	**0.6**			
s_3	0.8	0.0					
s_1	0.3	0.0					
s_2	1.0	0.1	0.82	**0.82**			
s_3	0.8	0.72					
s_1	0.3	0.24			0.738	0.798	**0.798**
s_2	1.0	0.1	0.42	0.42			
s_3	0.8	0.08					
s_1	0.3	0.24					
s_2	1.0	0.1	0.42	0.42			
s_3	0.8	0.08					
s_1	0.3	0.03			0.0		
s_2	1.0	0.0	0.75	**0.6**			
s_3	0.8	0.72					

Figure 7 : Two-stage a posteriori conditional decision tree from s_1

$$\operatorname*{Max}_{u_1}\left[\mu_1(u_1) \wedge \sum_{x_2} \operatorname*{Max}_{u_2}\left\{\mu_2(u_2) \wedge \sum_{x_3} \mu_3(x_3)p(x_3|x_2,u_2)\right\}p(x_2|s_2,u_1)\right]$$

history:



ter.	$\times p_2$	E^2	min	$\times p_1$	E^1	Min
0.3	0.24	0.42	0.42	0.0	0.622	**0.622**
1.0	0.1					
0.8	0.08					
0.3	0.03	0.93	**0.6**			
1.0	0.9					
0.8	0.0					
0.3	0.0	0.82	**0.82**	0.082		
1.0	0.1					
0.8	0.72					
0.3	0.24	0.42	0.42			
1.0	0.1					
0.8	0.08					
0.3	0.24	0.42	0.42	0.54		
1.0	0.1					
0.8	0.08					
0.3	0.03	0.75	**0.6**			
1.0	0.0					
0.8	0.72					
0.3	0.24	0.42	0.42	0.48	0.622	**0.622**
1.0	0.1					
0.8	0.08					
0.3	0.03	0.93	**0.6**			
1.0	0.9					
0.8	0.0					
0.3	0.0	0.82	**0.82**	0.082		
1.0	0.1					
0.8	0.72					
0.3	0.24	0.42	0.42			
1.0	0.1					
0.8	0.08					
0.3	0.24	0.42	0.42	0.06		
1.0	0.1					
0.8	0.08					
0.3	0.03	0.75	**0.6**			
1.0	0.0					
0.8	0.72					

Figure 8 : Two-stage a posteriori conditional decision tree from s_2

$$\underset{u_1}{\mathrm{Max}}\Big[\mu_1(u_1)\wedge\sum_{x_2}\underset{u_2}{\mathrm{Max}}\Big\{\mu_2(u_2)\wedge\sum_{x_3}\mu_3(x_3)p(x_3|x_2,u_2)\Big\}p(x_2|s_3,u_1)\Big]$$

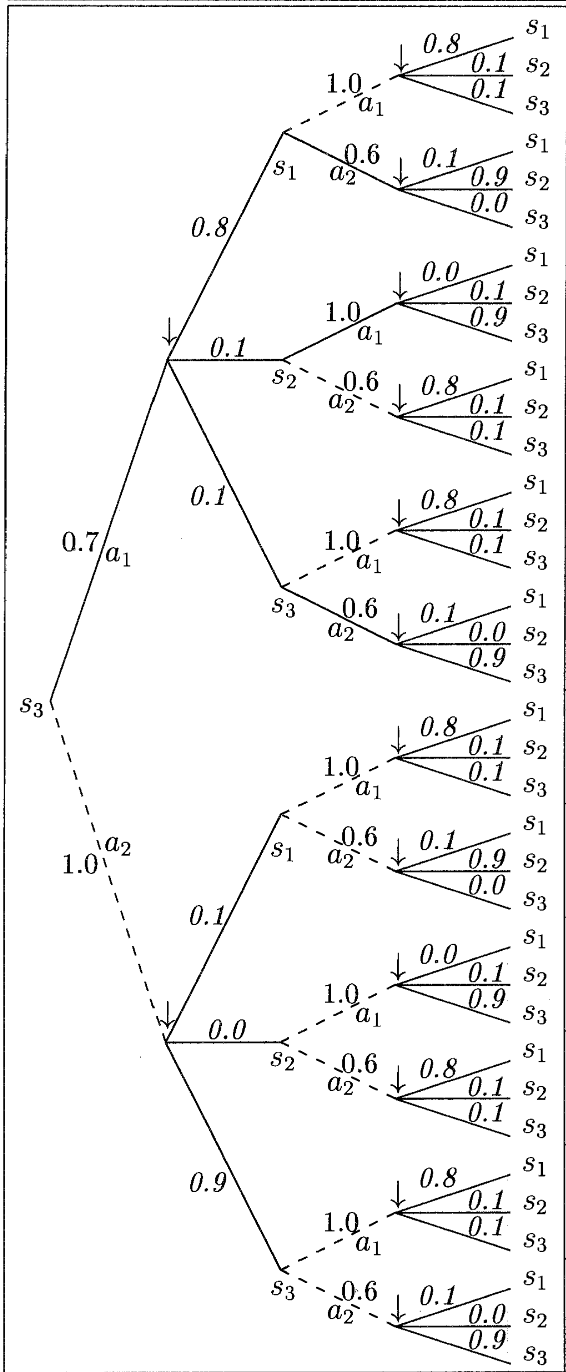
history	ter.	$\times p_2$	E^2	min	$\times p_1$	E^1	Min
	0.3	0.24					
	1.0	0.1	0.42	0.42			
	0.8	0.08					
	0.3	0.03			0.48		
	1.0	0.9	0.93	**0.6**			
	0.8	0.0					
	0.3	0.0					
	1.0	0.1	0.82	**0.82**			
	0.8	0.72					
	0.3	0.24			0.082	0.622	**0.622**
	1.0	0.1	0.42	0.42			
	0.8	0.08					
	0.3	0.24					
	1.0	0.1	0.42	0.42			
	0.8	0.08					
	0.3	0.03			0.06		
	1.0	0.0	0.75	**0.6**			
	0.8	0.72					
	0.3	0.24					
	1.0	0.1	0.42	0.42			
	0.8	0.08					
	0.3	0.03			0.06		
	1.0	0.9	0.93	**0.6**			
	0.8	0.0					
	0.3	0.0					
	1.0	0.1	0.82	**0.82**			
	0.8	0.72					
	0.3	0.24			0.0	0.6	0.6
	1.0	0.1	0.42	0.42			
	0.8	0.08					
	0.3	0.24					
	1.0	0.1	0.42	0.42			
	0.8	0.08					
	0.3	0.03			0.54		
	1.0	0.0	0.75	**0.6**			
	0.8	0.72					

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

4.3. A priori process

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

$$\begin{aligned} &\text{Maximize} \quad E^{u_1}_{x_1}[\mu_1(u_1) \wedge E^{u_2}_{x_2}[\mu_2(u_2) \wedge \mu_3\,]] \\ &\text{subject to} \quad (\mathrm{i})_{\mathrm{n}} \quad x_{n+1} \sim p(\cdot \mid x_n, u_n), \; u_n \in \{a_1, a_2\} \quad n = 1, 2. \end{aligned} \tag{4.17}$$

For the preceding data, the corresponding recursive equations

$$\begin{aligned} W_3(x_3) &= \mu_3(x_3) \\ W_2(x_2) &= \operatorname*{Max}_{u_2} \sum_{x_3} [\,\mu_2(u_2) \wedge W_3(x_3)\,] p(x_3|x_2, u_2) \\ W_1(x_1) &= \operatorname*{Max}_{u_1} \sum_{x_2} [\,\mu_1(u_1) \wedge W_2(x_2)\,] p(x_2|x_1, u_1) \end{aligned} \tag{4.18}$$

have the solution

$$W_3(s_1) = 0.3, \quad W_3(s_2) = 1.0, \quad W_3(s_3) = 0.8, \tag{4.19}$$

$$W_2(s_1) = 0.57, \quad W_2(s_2) = 0.82, \quad W_2(s_3) = 0.57 \tag{4.20}$$

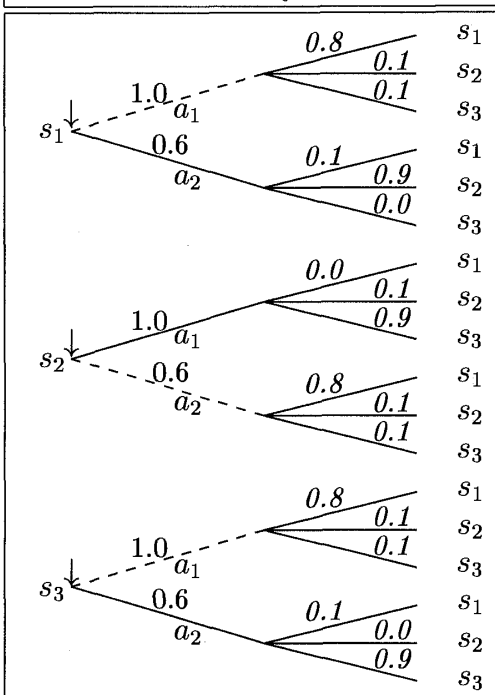
$$\pi^*_2(s_1) = a_2, \quad \pi^*_2(s_2) = a_1, \quad \pi^*_2(s_3) = a_2, \tag{4.21}$$

$$W_1(s_1) = 0.795, \quad W_1(s_2) = 0.595, \quad W_1(s_3) = 0.583 \tag{4.22}$$

$$\pi^*_1(s_1) = a_2, \quad \pi^*_1(s_2) = a_2, \quad \pi^*_1(s_3) = a_1. \tag{4.23}$$

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

$$\operatorname*{Max}_{u_2}\left[\sum_{x_3} \{\mu_2(u_2) \wedge \mu_3(x_3)\} p(x_3|x_2, u_2)\right]$$

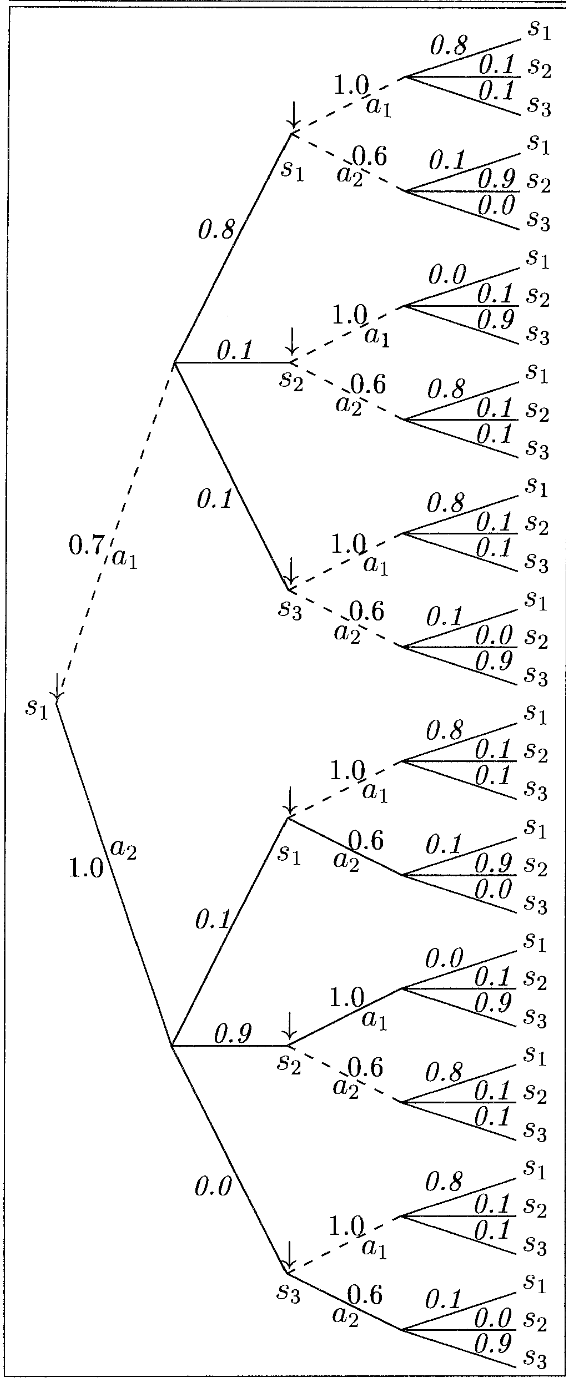


history	ter.	min	$\times p$	E^2
s_1	0.3	0.3	0.24	
s_2	1.0	1.0	0.1	0.42
s_3	0.8	0.8	0.08	
s_1	0.3	0.3	0.03	
s_2	1.0	0.6	0.54	**0.57**
s_3	0.8	0.6	0.0	
s_1	0.3	0.3	0.0	
s_2	1.0	1.0	0.1	**0.82**
s_3	0.8	0.8	0.72	
s_1	0.3	0.3	0.24	
s_2	1.0	0.6	0.06	0.36
s_3	0.8	0.6	0.06	
s_1	0.3	0.3	0.24	
s_2	1.0	1.0	0.1	0.42
s_3	0.8	0.8	0.08	
s_1	0.3	0.3	0.03	
s_2	1.0	0.6	0.0	**0.57**
s_3	0.8	0.6	0.54	

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

$$\operatorname*{Max}_{u_1}\left[\sum_{x_2}\left\{\mu_1(u_1)\wedge\operatorname*{Max}_{u_2}\sum_{x_3}\{\mu_2(u_2)\wedge\mu_3(x_3)\}p(x_3|x_2,u_2)\right\}p(x_2|s_1,u_1)\right]$$

history



ter.	min	$\times p_2$	E^2	min	$\times p_1$	E^1
0.3	0.3	0.24	0.42	0.57	0.456	0.583
1.0	1.0	0.1				
0.8	0.8	0.08				
0.3	0.3	0.03	**0.57**			
1.0	0.6	0.54				
0.8	0.6	0.0				
0.3	0.3	0.0	**0.82**	0.7	0.07	
1.0	1.0	0.1				
0.8	0.8	0.72				
0.3	0.3	0.24	0.36			
1.0	0.6	0.06				
0.8	0.6	0.06				
0.3	0.3	0.24	0.42	0.57	0.057	
1.0	1.0	0.1				
0.8	0.8	0.08				
0.3	0.3	0.03	**0.57**			
1.0	0.6	0.0				
0.8	0.6	0.54				
0.3	0.3	0.24	0.42	0.57	0.057	**0.795**
1.0	1.0	0.1				
0.8	0.8	0.08				
0.3	0.3	0.03	**0.57**			
1.0	0.6	0.54				
0.8	0.6	0.0				
0.3	0.3	0.0	**0.82**	0.82	0.738	
1.0	1.0	0.1				
0.8	0.8	0.72				
0.3	0.3	0.24	0.36			
1.0	0.6	0.06				
0.8	0.6	0.06				
0.3	0.3	0.24	0.42	0.57	0.0	
1.0	1.0	0.1				
0.8	0.8	0.08				
0.3	0.3	0.03	**0.57**			
1.0	0.6	0.0				
0.8	0.6	0.54				

Figure 11 : Two-stage a priori conditional decision tree from s_1

5. Concluding Remarks

We remark that for the case $\circ = \wedge$

$$W_n(x) \leq w_n(x) \quad 1 \leq n \leq N \tag{5.1}$$

and that for the case $\circ = \vee$

$$W_n(x) \geq w_n(x) \quad 1 \leq n \leq N. \tag{5.2}$$

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 λ, a function $g : X \to R^1$ 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

[1] R. E. Bellman: *Dynamic Programming* (Princeton Univ. Press, NJ, 1957).

[2] R. E. Bellman: Dynamic programming for inverse optimal problems in mathematical economics. *Journal of Mathematical Analysis and Applications*, **29** (1970) 424-428.

[3] R. E. Bellman: *Introduction to the Mathematical Control Processes, Vol.II; Nonlinear Processes* (Academin Press, NY, 1971)

[4] R. E. Bellman and E. D. Denman: *Invariant Imbedding,* Lect. Notes in Operation Research and Mathematical Systems. **52** (Springer-Verlag, Berlin, 1971).

[5] R. E. Bellman and L. A. Zadeh: Decision-making in a fuzzy environment. *Management Science*, **17** (1970) B141-B164.

[6] J. F. Baldwin and B. W. Pilsworth: Dynamic programming for fuzzy systems with fuzzy environment. *Journal of Mathematical Analysis and Applications*, **85** (1982) 1-23.

[7] A. O. Esogbue and R. E. Bellman: Fuzzy dynamic programming and its extensions. *TIMS/Studies in the Management Sciences*, **20** (1984) 147-167.

[8] T. Fujita and K. Tsurusaki: Stochastic optimization of multiplicative functions with negative value. *Journal of the Operations Research Society of Japan*, **41** (1998) 351-373.

[9] S. Iwamoto: Associative dynamic programs. *Journal of Mathematical Analysis and Applications*, **201** (1996) 195-211.

[10] S. Iwamoto: On expected values of Markov statistics. *Bulletin of Informatics and Cybernetics*, **30** (1998) 1-24.

[11] S. Iwamoto: Conditional decision processes with recursive function. *Journal of Mathematical Analysis and Applications*, to appear.

[12] S. Iwamoto and T. Fujita: Stochastic decision-making in a fuzzy environment. *Journal of the Operations Research Society of Japan*, **38** (1995) 467-482.

[13] S. Iwamoto and M. Sniedovich: Sequential decision making in fuzzy environment. *Journal of Mathematical Analysis and Applications*, **222** (1998) 208-224.

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

[15] J. Kacprzyk: Decision-making in a fuzzy environment with fuzzy termination time. *Fuzzy Sets and Systems*, **1** (1978) 169-179.

[16] J. Kacprzyk and A. O. Esogbue: Fuzzy dynamic programming: Main developments and applications. *Fuzzy Sets and Systems*, **81** (1996) 31-45.

[17] J. Kacprzyk and P. Staniewski: A new approach to the control of stochastic systems in a fuzzy environment. *Archiwum Automatyki i Telemechaniki*, **25** (1980) 443-444.

[18] E. S. Lee: *Quasilinearization and Invariant Imbedding* (Academic Press, NY, 1968).

[19] M. R. Scott: *Invariant Imbedding and its Applications to Ordinary Differential Equations : An introduction* (Addison-Wesly, London, 1973).

[20] M. Sniedovich: A class of nonseparable dynamic programming problems. *Journal of Optimization Theory and Applications*, **52** (1987) 111-121.

[21] M. Sniedovich: *Dynamic Programming* (Marcel Dekker, Inc. NY, 1992).

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