Fuzzy Dynamic Programming

Seiichi IWAMOTO

Department of Economic Engineering
Graduate School of Economics, Kyushu University 27
Hakozaki 6–19–1, Higashiku, Fukuoka 812–8581, JAPAN
tel&fax: +81(92)642–2488, e-mail: iwamoto@en.kyushu-u.ac.jp

Abstract: We make a survey of fuzzy dynamic programming on the basis of Bellman and Zadeh's seminal paper "Decision-making in a fuzzy environment". Our principle is that any dynamic programming must generate an optimal solution to a given problem. From this viewpoint, we propose dynamic programming method on determinstic, stochastic, and fuzzy systems. On the stochastic system, it is formulated as a maximization problem of expected value of minimum criterion not over the conventional *Markov* policy class but over three new *broad* policy classes. We present three dynamic programming approaches — (1) membership-parametric method, (2) history-parametric method, and (3) decision tree-table method —, which yield a common optimal policy in a broad (*general* policy) class. A complete set of optimal solutions for Bellman and Zadeh's model is illustrated. Two conditional decision processes are introduced, one of which turns out to be the Bellman and Zadeh's decision process on stochastic system. Further, a threshold-membership criterion problem is solved through fuzzy dynamic programming.

Key words: dynamic programming, decision-making, minimum criterion, fuzzy environment, expanded Markov policy, general policy, primitive policy, invariant imbedding, membership accumulation, state augmentation

1 Introduction

Since Bellman and Zadeh have published "Decision-making in a fuzzy environment" [4], there has been a wide-ranging study on fuzzy decision theory and its applications [1,5,24–28,34]. Bellman and Zadeh have originated *deterministic, stochastic* and *fuzzy* systems on multistage decision processes in the fuzzy environment [4, §4,5]. We focus on the three systems.

It is well known that there exists an optimal policy which is *Markov* for the *additive* criterion [7,32]. However, as for the *minimum* criterion, there does not always exist an optimal policy in Markov policy class [11–14,16–23]. This fact raises a question. What is fuzzy dynamic programming?

In this paper, we discuss *fuzzy* dynamic programming on the three systems. Our viewpoint for dynamic programming is that *sequential* optimization assures *simultaneous* one [2,33]. Dynamic programming has to yield an optimal solution to the original problem whenever it is applied. Although we discuss dynamic programming on repective systems one by one, we direct our main attention to stochastic system. We develop the *deterministic* "final state model" [4,33] to the *stochastic* "final state model". The basic idea is imbedding by state augmentation, which incorporates the membership accumulation process into an additional state dynamics.

In section 2, we give a brief survey of fuzzy dynamic programming on deterministic system. Section 3 is devoted to fuzzy dynamic programming on stochastic system. Three new —*expanded Markov, general* and *primitive*— policies are defined. Introducing three equivalent methods — (1) membership-parameter method, (2) history-parameter method, and (3) multi-stage stochastic decision tree-table method —, we show that the three methods yield a common optimal policy in general policy (large) class. We also illustrares the optimal policy on three-state two-decision and two-stage model. Further, it is shown that there does not exist an optimal policy in Markov policy (small) class. In section 4, fuzzy dynamic programming on fuzzy system is discussed under the fuzzy (minimax) expectation. Section 5 introduces two (a posteriori and a priori) conditional decision processes on stochastic system. We show that the Bellman and Zadeh's decision process on stochastic system in [4] turns out to be the a posteriori conditional decision process. Finally, in section 6, we solve a threshold-membership criterion problem through fuzzy dynamic programming.

We use the notations and terminology in [4,13,15,30,31].

2 Deterministic System

A fuzzy decision process on *deterministic system* is the following multi-stage decision process with minimum criterion (Bellman and Zadeh [4]) :

$$\mathrm{D}_0(x_0) \qquad \begin{array}{lll} \text{Max} & \mu_0(x_0,u_0)\wedge\mu_1(x_1,u_1)\wedge\cdots\wedge\mu_{N-1}(x_{N-1},u_{N-1})\wedge\mu_G(x_N) & \\ \text{s.t.} & \text{(i)} \ \ x_{n+1}=f(x_n,u_n) & n=0,1,\ldots,N-1 \\ & \text{(ii)} \ \ u_n\in U & \end{array} \tag{1}$$

On the deterministic system, an *action* $u_n \in U$ at *state* $x_n \in X$ determines uniquely a next state $x_{n+1} = f(x_n, u_n)$, where $f : X \times U \to X$ is a *state dynamics*. $\mu_n : X \times U \to [0,1]$ is a membership function of *fuzzy set* R_n on $X \times U$:

$$\mu_n(x,u)=\mu_{R_n}(x,u).$$

$\mu_G : X \to [0,1]$ is a membership function of *fuzzy goal* (fuzzy set) G on X. Then the multi-stage decision process maximizes the *membership function*

$$\begin{aligned} & \mu_{R_0\cap R_1\cap\cdots\cap R_{N-1}\cap G}(x_0,u_0,x_1,u_1,\cdots,x_{N-1},u_{N-1},x_N) \\ = \ & \mu_0(x_0,u_0)\wedge\mu_1(x_1,u_1)\wedge\cdots\wedge\mu_{N-1}(x_{N-1},u_{N-1})\wedge\mu_G(x_N) \end{aligned}$$

of intersection $R_0 \cap R_1 \cap \cdots \cap R_{N-1} \cap G$.

We consider a class of policies. A mapping $\pi_n : X \to U$ is called *n-th Markov decision function.* A sequence of Markov decision functions $\pi = \{\pi_0, \pi_1, \ldots, \pi_{N-1}\}$ is called *Markov policy.* We denote the set of all Markov policies by Π, which is called *Markov class.*

Then we have the backward recursive formula ([33]) :

Theorem 2.1 *(Bellman and Zadeh [4])*

$$\begin{aligned} & v_n(x)=\operatorname*{Max}_{u\in U}[\mu_n(x,u)\wedge v_{n+1}(f(x,u))] \qquad x\in X,\ 0\le n\le N-1 \\ & v_N(x)=\mu_G(x) \qquad x\in X \end{aligned} \tag{2}$$

Now we solve the recursive equation. Let $\pi_n^*(x)$ be a maximizer in (2). Then we have both *maximum value functions* $\{v_n\}_0^N$ and an *optimal policy* $\pi^* = \{\pi_n^*\}_0^{N-1}$ in Markov class Π. The desired *maximum value* of $\mathrm{D}_0(x_0)$ is $v_0(x_0)$. Its *maximum point* $u^* = \{u_0^*, u_1^*, \ldots, u_{N-1}^*\}$ is given by use of the optimal policy π^*, the state dynamics f and the initial state x_0 through the *conventional* dynamic programming :

$$\begin{aligned} & \pi_0^*(x_0)=u_0^*\to f(x_0,u_0^*)=x_1^*\to\pi_1^*(x_1^*)=u_1^*\to f(x_1^*,u_1^*)=x_2^*\to \\ & \pi_2^*(x_2^*)=u_2^*\to f(x_2^*,u_2^*)=x_3^*\to\ \cdots\ \to f(x_{N-2}^*,u_{N-2}^*)=x_{N-1}^*\to \\ & \pi_{N-1}^*(x_{N-1}^*)=u_{N-1}^*\to f(x_{N-1}^*,u_{N-1}^*)=x_N^*, \end{aligned} \tag{3}$$

where the corresponding sequence of states $x^* = \{x_0, x_1^*, \ldots, x_N^*\}$ is also uniquely determined by the triplet (π^*, f, x_0). This is a fuzzy dynamic programming for the deterministic system. Here we note that in minimum criterion the function $g_n : X \times U \times R^1 \to R^1$ defined by $g_n(x,u;h) := \mu_n(x,u) \wedge h$ is nondecreasing in the third argument. Thus the minimum criterion enjoys the *monotonicity*, and it connotes the *separability* in itself [2,8–10,33].

Now let us solve the Bellman and Zadeh's three-state, two-decision, two-stage model [4, pp.B153] :

$$\begin{array}{ll} \text{Max} & \mu_0(u_0)\wedge\mu_1(u_1)\wedge\mu_G(x_2) \\ \text{s.t.} & x_{n+1}=f(x_n,u_n),\ \ u_n\in U \quad n=0,1 \end{array}$$

where the data is Table 1 :

$\mu_t(u_t)\backslash u_t$	a_1	a_2
$\mu_0(u_0)$	0.7	1.0
$\mu_1(u_1)$	1.0	0.6

x_2	$\mu_G(x_2)$
s_1	0.3
s_2	1.0
s_3	0.8

$x_t\backslash u_t$	a_1	a_2
s_1	s_1	s_2
s_2	s_3	s_1
s_3	s_1	s_3

Table 1: $\{\mu_0, \mu_1\}$ μ_G $f(x_t, u_t)$

We solve the recursive equation :

$$
\begin{aligned}
v_2(x_2) &= \mu_G(x_2) \quad x_2 \in X \\
v_1(x_1) &= \operatorname*{Max}_{u_1 \in U} [\mu_1(u_1) \wedge v_2(f(x_1, u_1))] \quad x_1 \in X \\
v_0(x_0) &= \operatorname*{Max}_{u_0 \in U} [\mu_0(u_0) \wedge v_1(f(x_0, u_0))] \quad x_0 \in X.
\end{aligned}
$$

Then the dynamic programming solution, which consists of a pair of sequence of optimal membership functions $\{v_0, v_1, v_2\}$ and optimal policy $\pi^* = \{\pi_0^*, \pi_1^*\}$, is shown in Table 2:

x_n	$v_0(x_0)$	$\pi_0^*(x_0)$	$v_1(x_1)$	$\pi_1^*(x_1)$	$v_2(x_2)$
s_1	0.8	a_2	0.6	a_2	0.3
s_2	0.6	a_1, a_2	0.8	a_1	1.0
s_3	0.6	a_1, a_2	0.6	a_2	0.8

Table 2: $\{v_0, v_1, v_2\}$ $\pi^* = \{\pi_0^*, \pi_1^*\}$

Thus the original fuzzy decision problem has the optimal solution as follows:

(i) $x_0 = s_1 \to u_0^* = a_2 \to x_1^* = s_2 \to u_1^* = a_1 \to x_2^* = s_3 \Longrightarrow v_0(s_1) = 1.0 \wedge 1.0 \wedge 0.8 = 0.8$

(ii) $x_0 = s_2 \to u_0^* = a_1 \to x_1^* = s_3 \to u_1^* = a_2 \to x_2^* = s_3 \Longrightarrow v_0(s_2) = 0.7 \wedge 0.6 \wedge 0.8 = 0.6$

or $u_0^* = a_2 \to x_1^* = s_1 \to u_1^* = a_2 \to x_2^* = s_3 \Longrightarrow v_0(s_2) = 1.0 \wedge 0.6 \wedge 0.8 = 0.6$

(iii) $x_0 = s_3 \to u_0^* = a_1 \to x_1^* = s_1 \to u_1^* = a_2 \to x_2^* = s_2 \Longrightarrow v_0(s_3) = 0.7 \wedge 0.6 \wedge 1.0 = 0.6$

or $u_0^* = a_2 \to x_1^* = s_3 \to u_1^* = a_2 \to x_2^* = s_3 \Longrightarrow v_0(s_3) = 1.0 \wedge 0.6 \wedge 0.8 = 0.6$

3 Stochastic System

Now let us consider a fuzzy dynamic programming on *stochastic system*. This enables us to discuss both monotonicity and separability for minimum criterion in stochastic sense. The following results provide a stochastic dynamic programming with nonadditive criterion.

We begin to introduce a large class of policies, which depend not only on today's state but also on state-to-date. Let $X^n := X \times X \times \cdots \times X$ be direct product of n state spaces X. A mapping $\sigma_n : X^{n+1} \to U$ is called *n-th general decision function*, whose sequence $\sigma = \{\sigma_0, \sigma_1, \ldots, \sigma_{N-1}\}$ constitutes a *general policy*. The set of all general policies Π_g is called *general class*. When each general decision function σ_n depends only on the last (= current) state, the general policy reduces to a *Markov policy* $\pi = \{\pi_0, \pi_1, \ldots, \pi_{N-1}\}$. Thus we have an inclusion relation : $\Pi \subset \Pi_g$.

The fuzzy decision process Bellman and Zadeh [4] have proposed is the maximization problem of expected value of *minimum* criterion over a Markov decision process $\{X_n\}_0^N$ with a transition law $p = \{p(\cdot|\cdot,\cdot)\}$ on state space X and decision space U [14, 16–19, 22, 23] :

$$
\mathrm{S}_0(x_0) \quad
\begin{array}{ll}
\text{Max} & E_{x_0}^{\sigma}[\mu_0 \wedge \mu_1 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] \\
\text{s.t.} & (\text{i})_n \ X_{n+1} \sim p(\cdot\,|x_n, u_n) \\
 & (\text{ii})_n \ u_n \in U \qquad\qquad 0 \le n \le N\text{–}1
\end{array}
$$

where $E_{x_0}^{\sigma}$ is the expectation operator on history space

$$H_N := X \times U \times X \times U \times \cdots \times U \times X \quad (2N+1)\text{-factors}$$

induced through an initial state x_0, the Markov transition law p and a general policy $\sigma(\in \Pi_g)$.

In general, Markov class Π is not enough for *nonadditive* criteria [22]. So we maximize the expected value over general class Π_g. Any general policy $\sigma(\in \Pi_g)$ determines the expected value in $\mathrm{S}_0(x_0)$, which is the multiple sum :

$$E^{\sigma}_{x_0}[\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] = \sum\sum\cdots\sum_{(x_1,x_2,\ldots,x_N)\in X\times X\times\cdots\times X} [\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] p_0 p_1 \cdots p_{N-1}. \tag{4}$$

where

$$\mu_n = \mu_n(x_n, u_n), \quad \mu_G = \mu_G(x_N), \quad p_n = p(x_{n+1}|x_n, u_n).$$

Further the sequence of decisions $\{u_0,\ u_1,\ \ldots,\ u_{N-1}\}$ is uniquely determined through general policy $\sigma = \{\sigma_0, \ldots, \sigma_{N-1}\}$ as follows:

$$u_0 = \sigma_0(x_0),\ u_1 = \sigma_1(x_0, x_1),\ \ldots,\ u_{N-1} = \sigma_{N-1}(x_0, x_1, \ldots, x_{N-1}). \tag{5}$$

Thus our problem $\mathrm{S}_0(x_0)$ is to find the *maximum value function* $v_0 = v_0(x_0)$ and an *optimal policy* $\sigma^*(\in \Pi_g)$ which attains the maximum :

$$v_0(x_0) = E^{\sigma^*}_{x_0}[\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] = \mathop{\mathrm{Max}}_{\sigma\in\Pi_g} E^{\sigma}_{x_0}[\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] \qquad x_0 \in X. \tag{6}$$

This is called *general problem.*

3.1 Membership-parameteric method

Now we imbed $\mathrm{S}_0(x_0)$ into a new class of additional parametric subproblems [3, 29]. First we define the *past-valued (cumulative) random variables* $\{\tilde{\Lambda}_n\}$ up to n-th stage and the *past-value sets* $\{\Lambda_n\}$ they take :

$$\begin{aligned} \tilde{\Lambda}_0 &\triangleq 1 \\ \tilde{\Lambda}_n &\triangleq \mu_0(X_0, U_0) \wedge \cdots \wedge \mu_{n-1}(X_{n-1}, U_{n-1}) \quad 1 \le n \le N, \end{aligned} \tag{7}$$

$$\begin{aligned} \Lambda_0 &\triangleq \{1\} \\ \Lambda_n &\triangleq \{\lambda_n \,|\, \lambda_n = \mu_0(x_0, u_0) \wedge \cdots \wedge \mu_{n-1}(x_{n-1}, u_{n-1}), \\ &\qquad (x_0, u_0, \ldots, x_{n-1}, u_{n-1}) \in X \times U \times \cdots \times X \times U\} \qquad 1 \le n \le N. \end{aligned} \tag{8}$$

The minimum criterion is terminal now :

$$\mu_0(X_0, U_0) \wedge \cdots \wedge \mu_{N-1}(X_{n-1}, U_{N-1}) \wedge \mu_G(X_N) = \tilde{\Lambda}_N \wedge \mu_G(X_N). \tag{9}$$

We have

Lemma 3.1 *(Forward recursive formulae)*

$$\begin{aligned} \tilde{\Lambda}_0 &= 1 \\ \tilde{\Lambda}_{n+1} &= \tilde{\Lambda}_n \wedge \mu_n(X_n, U_n) \quad 0 \le n \le N{-}1. \end{aligned} \tag{10}$$

$$\begin{aligned} \Lambda_0 &= \{1\} \\ \Lambda_{n+1} &= \{\lambda \wedge \mu_n(x, u) \,|\, \lambda \in \Lambda_n,\ u \in U\} \quad 0 \le n \le N{-}1. \end{aligned} \tag{11}$$

Further, we expand the original state space X to a direct product space :

$$Y_n \triangleq X \times \Lambda_n \qquad 0 \le n \le N. \tag{12}$$

Any Markov policy $\gamma = \{\gamma_0, \gamma_1, \ldots, \gamma_{N-1}\}$ on the expanded state spaces $\{Y_n\}$ is a sequence of Markov decision functions

$$\gamma_n : Y_n \to U \quad 0 \le n \le N-1.$$

We call γ an *expanded Markov* policy. The set of all expanded Markov polices is denoted by $\tilde{\Pi}$. We define a *terminal membership function* T by

$$T(x;\lambda) \overset{\triangle}{=} \lambda \wedge \mu_G(x) \qquad (x;\lambda) \in Y_N \tag{13}$$

and a *nonstationary Markov transition law* $q = \{q_n\}$ by

$$q_n((y;\mu)\,|\,(x;\lambda),\,u) \quad \overset{\triangle}{=} \quad \begin{cases} p(y|x,\,u) & \text{if } \lambda \wedge \mu_n(x,u) = \mu \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Then $\{(X_n, \tilde{\Lambda}_n)\}_0^N$ is a Markov decision process on state spaces $\{Y_n\}$ and decision space U with transition law q. We consider a *terminal* criterion on the new decision process :

$$\mathrm{T}_0(y_0) \qquad \begin{array}{ll} \text{Max} & \tilde{E}^{\gamma}_{y_0}[\tilde{\Lambda}_N \wedge \mu_G(X_N)] \\ \text{s.t.} & (\mathrm{i})_\mathrm{n} \quad X_{n+1} \sim p(\,\cdot\,|x_n, u_n) \\ & (\mathrm{i})'_\mathrm{n} \quad \tilde{\Lambda}_{n+1} = \tilde{\Lambda}_n \wedge \mu_n(X_n, U_n) \qquad 0 \le n \le N\text{–}1 \\ & (\mathrm{ii})_\mathrm{n} \quad u_n \in U \end{array} \tag{15}$$

where $y_0 = (x_0;1)$ is an initial state. The expectation operator $\tilde{E}^{\gamma}_{y_0}$ is based upon the probability measure $\tilde{P}^{\gamma}_{y_0}$, which is uniquely determined through the initial state y_0, an expanded Markov policy γ and the Markov transition law q [23].

Now we take a subprocess which starts at state $y_n = (x_n;\lambda_n)(\in Y_n)$ on n-th stage and terminates on the final N-th stage:

$$\mathrm{T}_n(y_n) \qquad \begin{array}{ll} \text{Max} & \tilde{E}^{\gamma}_{y_n}[\tilde{\Lambda}_N \wedge \mu_G(X_N)] \\ \text{s.t.} & (\mathrm{i})_\mathrm{m},\ (\mathrm{i})'_\mathrm{m},\ (\mathrm{ii})_\mathrm{m} \qquad n \le m \le N\text{–}1. \end{array}$$

Let $u^n(x_n;\lambda_n)$ be the maximum value of $\mathrm{T}_n(y_n)$ for $0 \le n \le N\text{–}1$, where

$$u^N(x_N;\lambda_N) \overset{\triangle}{=} \lambda_N \wedge \mu_G(x_N) \qquad (x_N;\lambda_N) \in Y_N.$$

Then we have the backward recursive relation :

Theorem 3.1 *(Expanded Markov class)*

$$u^N(x;\lambda) = \lambda \wedge \mu_G(x) \qquad x \in X,\ \ \lambda \in \Lambda_N$$
$$u^n(x;\lambda) = \operatorname*{Max}_{u\in U} \sum_{y \in X} u^{n+1}(y;\lambda \wedge \mu_n(x,u))p(y|x,u) \qquad x \in X,\ \ \lambda \in \Lambda_n,\ \ 0 \le n \le N\text{–}1. \tag{16}$$

Now let $\gamma^*_n(x;\lambda)$ be the set of all maximizers in (16). Then we have an optimal policy $\gamma^* = \{\gamma^*_0, \gamma^*_1, \ldots, \gamma^*_{N-1}\}$ in expanded Markov class $\tilde{\Pi}$ [23, Thm 4.2]. The optimal policy γ^* leads to a general policy $\sigma^* = \{\sigma^*_0, \sigma^*_1, \ldots, \sigma^*_{N-1}\}$ through the transformation [13, §3]. (An illustrative transformation will be given in Subsection §§3.3.) Then the policy σ^* is optimal in general class Π_g. Further, the maximum value over expanded Markov class $\tilde{\Pi}$ is equal to the maximum value over general class Π_g [23, Thm 6.1] :

$$u^0(x_0;1) = v_0(x_0). \tag{17}$$

3.2 History-parameteric method

In this section we maximize the expected value over a larger (primitive) class Π_p. We show that an optimal policy is obtained in the largre (general) class.

Now the expectation problem over Π_p is :

$$\mathrm{H}_0(x_0) \qquad \begin{array}{ll} \text{Max} & E^{\nu}_{x_0}[\mu_0 \wedge \mu_1 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] \\ \text{s.t.} & (\mathrm{i})_\mathrm{n}\ X_{n+1} \sim p(\,\cdot\,|x_n, u_n) \qquad 0 \le n \le N\text{–}1. \\ & (\mathrm{ii})_\mathrm{n}\ u_n \in U \end{array}$$

This is called *primitive problem.* We note that the sequence of decisions $u_0,\ u_1,\ldots,\ u_N$ is determined by the preceding decisions :

$$u_0 = \nu_0(x_0),\ u_1 = \nu_1(x_0,u_0,x_1),\ \ldots,\ u_{N-1} = \nu_{N-1}(x_0,u_0,x_1,u_1,\ldots,u_{N-2},x_{N-1}). \tag{18}$$

We consider the subprocess starting at "history" $h_n = (x_0,u_0,\ldots,x_{n-1},u_{n-1},x_n)(\in H_n)$ on and after n-th stage:

$$\mathrm{H}_n(h_n) \qquad \begin{array}{ll} \text{Max} & E^{\nu}_{h_n}[\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G] \\ \text{s.t.} & (\mathrm{i})_\mathrm{m},\ (\mathrm{ii})_\mathrm{m} \quad n \le m \le N\text{–}1, \end{array}$$

where $\nu = \{\nu_n,\nu_{n+1},\ldots,\nu_{N-1}\}$ is a primitive policy during the stage-interval $[n,N]$. The set of all such primitive policies is denoted by $\Pi_p(n)$. Let $w_n(h_n)$ be the maximum value over $\Pi_p(n)$ for $0 \le n \le N\text{–}1$, where

$$w_N(h_N) \ \stackrel{\triangle}{=}\ \mu_0(x_0,u_0) \wedge \cdots \wedge \mu_{N-1}(x_{n-1},u_{N-1}) \wedge \mu_G(x_N). \tag{19}$$

Then we have the backward equation :

Theorem 3.2 *(Primitive class)*

$$w_n(h) = \operatorname*{Max}_{u\in U} \sum_{y\in X} w_{n+1}(h,u,y)p(y|x,u) \qquad h \in H_n,\quad 0 \le n \le N-1 \tag{20}$$

$$w_N(h) = \mu_0(x_0,u_0) \wedge \cdots \wedge \mu_{N-1}(x_{n-1},u_{N-1}) \wedge \mu_G(x_N) \qquad h \in H_N. \tag{21}$$

Let $\hat{\nu}_n(h)$ denote the set of all maximizers in (20). Then we have an optimal policy $\hat{\nu} = \{\hat{\nu}_0,\hat{\nu}_1,\ldots,\hat{\nu}_{N-1}\}$ in primitive class Π_g [23, Thm 4.4]. Furthermore, the optimal policy $\hat{\nu}$ generates a general policy $\hat{\sigma} = \{\hat{\sigma}_0,\hat{\sigma}_1,\ldots,\hat{\sigma}_{N-1}\}$ through the transformation [13, §4]. (This transformation is also illustrated in next Subsection §§3.3.) Then the policy $\hat{\sigma}$ is optimal in general class Π_g. Finally, both the optimal solution of $\mathrm{H}_0(x_0)$ and the optimal solution of $\mathrm{T}_0(y_0)$ coincide [23, Cor 5.2] :

$$w_0(x_0) = v_0(x_0) = u^0(x_0;1), \qquad \hat{\sigma} = \sigma^*. \tag{22}$$

3.3 Bellman and Zadeh's Model

Let us reconsider the Bellman and Zadeh's three-state, two-decision, two-stage model [4, pp.B154] :

$$\begin{array}{ll} \text{Max} & E^{\sigma}_{x_0}[\mu_0(U_0) \wedge \mu_1(U_1) \wedge \mu_G(X_2)] \\ \text{s.t.} & (\mathrm{i})_\mathrm{n}\ X_{n+1} \sim p(\cdot\,|x_n,u_n) \\ & (\mathrm{ii})_\mathrm{n}\ u_n \in U \qquad\qquad n = 0,1 \end{array} \tag{23}$$

where the numerical data is :

$$\mu_0(a_1) = 0.7 \quad \mu_0(a_2) = 1.0$$
$$\mu_1(a_1) = 1.0 \quad \mu_1(a_2) = 0.6$$

$$\mu_G(s_1) = 0.3 \quad \mu_G(s_2) = 1.0 \quad \mu_G(s_3) = 0.8$$

	$p(x_{n+1}\|x_n,a_1)$			$p(x_{n+1}\|x_n,a_2)$		
$x_n \backslash x_{n+1}$	s_1	s_2	s_3	s_1	s_2	s_3
s_1	*0.8*	*0.1*	*0.1*	*0.1*	*0.9*	*0.0*
s_2	*0.0*	*0.1*	*0.9*	*0.8*	*0.1*	*0.1*
s_3	*0.8*	*0.1*	*0.1*	*0.1*	*0.0*	*0.9*

In this subsection, we solve this model by both the membership-parametric method and the history-parametric method. First the backward recursion (16) yields an optimal solution in expanded Markov class $\tilde{\Pi}$ — a pair of a sequence of optimal value fuctions :

$$u^0 = u^0(x_0; \lambda_0),\ u^1 = u^1(x_1; \lambda_1),\ u^2 = u^2(x_2; \lambda_2)$$

and an optimal policy :

$$\gamma^* = \{\gamma_0^*(x_0; \lambda_0),\ \gamma_1^*(x_1; \lambda_1)\}.$$

For instance, when $u^1 = u^1(x_1; \lambda_1)$ is calculated, $u^0(s_1; 1.0)$ and $\gamma_0^*(s_1; 1.0)$ are obtained as follows :

$$\begin{aligned} u^0(s_1; 1.0) &= [u^1(s_1; 0.7)0.8 + u^1(s_2; 0.7)0.1 + u^1(s_3; 0.7)0.1] \vee [u^1(s_1; 1.0)0.1 + u^1(s_2; 1.0)0.9 + u^1(s_3; 1.0)0.0] \\ &= [0.57 \times 0.8 + 0.7 \times 0.1 + 0.57 \times 0.1] \vee [0.57 \times 0.1 + 0.82 \times 0.9 + 0.57 \times 0.0] \\ &= 0.583 \vee 0.795 \\ &= 0.795 \qquad \gamma_0^*(s_1; 1.0) = a_2. \end{aligned} \tag{24}$$

The similar calculation yields the first optimal value function $u^0 = u^0(x_0; \lambda_0)$ together with the first optimal decision function $\gamma_0^* = \gamma_0^*(x_0; \lambda_0)$. Thus we obtain the optimal solution in Table 3 :

	$u^2(x_2;\lambda_2)$			$u^1(x_1;\lambda_1)$ $\gamma_1^*(x_1;\lambda_1)$				$u^0(x_0;\lambda_0)$	$\gamma_0^*(x_0;\lambda_0)$
$x_n \backslash \lambda_n$	0.6	0.7	1.0	0.7		1.0		1.0	
s_1	0.3	0.3	0.3	0.57	a_2	0.57	a_2	0.795	a_2
s_2	0.6	0.7	1.0	0.7	a_1	0.82	a_1	0.595	a_2
s_3	0.6	0.7	0.8	0.57	a_2	0.57	a_2	0.583	a_1

Table 3 : optimal value functions $\{u^0,\ u^1,\ u^2\}$ and optimal policy $\gamma^* = \{\gamma_0^*,\ \gamma_1^*\}$

Now we derive an optimal (general) policy $\sigma^* = \{\sigma_0^*, \sigma_1^*\}$ from the optimal (expanded Markov) policy $\gamma^* = \{\gamma_0^*, \gamma_1^*\}$ as follows :

$$\begin{aligned} \sigma_0^*(x_0) &:= \gamma_0^*(x_0; 1.0) \\ & \quad u_0 := \gamma_0^*(x_0; 1.0), \quad \lambda_1 := \mu_0(u_0) \\ \sigma_1^*(x_0, x_1) &:= \gamma_1^*(x_1; \lambda_1). \end{aligned}$$

The first decision function $\sigma_0^*(\cdot)$ becomes

$$\sigma_0^*(s_1) = \gamma_0^*(s_1; 1.0) = a_2, \qquad \sigma_0^*(s_2) = \gamma_0^*(s_2; 1.0) = a_2, \qquad \sigma_0^*(s_3) = \gamma_0^*(s_3; 1.0) = a_1.$$

The second decision function $\sigma_1^*(\cdot)$ is obtained as follows :

$$\begin{array}{lll} \sigma_1^*(s_1, s_1) = \gamma_1^*(s_1; 1.0) = a_2, & \sigma_1^*(s_1, s_2) = \gamma_1^*(s_2; 1.0) = a_1, & \sigma_1^*(s_1, s_3) = \gamma_1^*(s_3; 1.0) = a_2 \\ \sigma_1^*(s_2, s_1) = \gamma_1^*(s_1; 0.7) = a_2, & \sigma_1^*(s_2, s_2) = \gamma_1^*(s_2; 0.7) = a_1, & \sigma_1^*(s_2, s_3) = \gamma_1^*(s_3; 0.7) = a_2 \\ \sigma_1^*(s_3, s_1) = \gamma_1^*(s_1; 0.7) = a_2, & \sigma_1^*(s_3, s_2) = \gamma_1^*(s_2; 0.7) = a_1, & \sigma_1^*(s_3, s_3) = \gamma_1^*(s_3; 0.7) = a_2. \end{array}$$

We see that the optimal policy σ^* happens to be Markov. In general, the reduced σ^* is *not necessarily Markov.* It is *general* [22], as is shown in§§3.5.

Second, the recursive equation (20) for primitive class Π_p gives optimal value functions $\{w_0(x_0),\ w_1(x_0, u_0, x_1),\ w_2(x_0, u_0, x_1, u_1, x_2)\}$ and an optimal policy $\hat{\nu} = \{\ \hat{\nu}_0(x_0),\ \ \hat{\nu}_1(x_0, u_0, x_1))\}$. For instance, a couple of $w_0(s_1)$ and $\hat{\nu}_0(s_1)$ is calculated by use of $w_1 = w_1(x_0, u_0, x_1)$ as follows :

$$\begin{aligned} w_0(s_1) &= [w_1(s_1, a_1, s_1)p(s_1|s_1, a_1) + w_1(s_1, a_1, s_2)p(s_2|s_1, a_1) + w_1(s_1, a_1, s_3)p(s_3|s_1, a_1))] \\ & \quad \vee[w_1(s_1, a_2, s_1)p(s_1|s_1, a_2) + w_1(s_1, a_2, s_2)p(s_2|s_1, a_2) + w_1(s_1, a_2, s_3)p(s_3|s_1, a_2))] \\ &= [0.57 \times 0.8 + 0.7 \times 0.1 + 0.57 \times 0.1] \vee [0.57 \times 0.1 + 0.82 \times 0.9 + 0.57 \times 0.0] \\ &= 0.583 \vee 0.795 \\ &= 0.795 \qquad \hat{\nu}_0(s_1) = a_2. \end{aligned} \tag{25}$$

Similary we get the first optimal function $w_0 = w_0(x_0)$ together with the first optimal decision function $\hat{\nu}_0 = \hat{\nu}_0(x_0)$. Thus the optimal solution is shown in Tables 4,5 and 6.

u_0	a_1		a_2	
$x_2 \backslash u_1$	a_1	a_2	a_1	a_2
s_1	0.3	0.3	0.3	0.3
s_2	0.7	0.6	1.0	0.6
s_3	0.7	0.6	0.8	0.6

Table 4 : $w_2(\underline{x_0}, u_0, \underline{x_1}, u_1, x_2)$

x_1	$w_1(\underline{x_0}, a_1, x_1)$	$\hat{\nu}_1(\underline{x_0}, a_1, x_1)$	$w_1(\underline{x_0}, a_2, x_1)$	$\hat{\nu}_1(\underline{x_0}, a_2, x_1)$
s_1	0.57	a_2	0.57	a_2
s_2	0.7	a_1	0.82	a_1
s_3	0.57	a_2	0.57	a_2

Table 5 : $w_1(x_0, u_0, x_1)$ $\hat{\nu}_1(x_0, u_0, x_1)$

where the underline denotes an independence. For instance, $w_1(\underline{x_0}, a_1, x_1) = 0.57$ for $\forall x_0 = s_1, s_2, s_3$.

x_0	$w_0(x_0)$	$\hat{\nu}_0(x_0)$
s_1	0.795	a_2
s_2	0.595	a_2
s_3	0.583	a_1

Table 6 : $w_0(x_0)$ $\hat{\nu}_0(x_0)$

Now the optimal (primitive) policy $\hat{\nu} = \{\hat{\nu}_0, \hat{\nu}_1\}$ generates a general policy $\hat{\sigma} = \{\hat{\sigma}_0, \hat{\sigma}_1\}$ through the transformation

$$\begin{aligned} \hat{\sigma}_0(x_0) &:= \hat{\nu}_0(x_0) \\ \hat{\sigma}_1(x_0, x_1) &:= \hat{\nu}_1(x_0, u_0, x_1) \quad \text{where} \quad u_0 = \hat{\nu}_0(x_0). \end{aligned}$$

In fact, $\hat{\sigma}$ becomes as follows :

$$\begin{array}{lll} \hat{\sigma}_0(s_1) = a_2, & \hat{\sigma}_0(s_2) = a_2, & \hat{\sigma}_0(s_3) = a_1 \\ \hat{\sigma}_1(s_1, s_1) = a_2, & \hat{\sigma}_1(s_1, s_2) = a_1, & \hat{\sigma}_1(s_1, s_3) = a_2 \\ \hat{\sigma}_1(s_2, s_1) = a_2, & \hat{\sigma}_1(s_2, s_2) = a_1, & \hat{\sigma}_1(s_2, s_3) = a_2 \\ \hat{\sigma}_1(s_3, s_1) = a_2, & \hat{\sigma}_1(s_3, s_2) = a_1, & \hat{\sigma}_1(s_3, s_3) = a_2. \end{array}$$

We see that both the general σ^* and $\hat{\sigma}$ coincide. This identification is also verified through the following *two-stage stochastic decision tree-tables* (Figures 1, 2 and 3), which is the third method.

3.4 Decision Tree-Table Method

A *multistage stochastic decision tree-table* consists of a pair of decision-tree and decision-table. It contains in a sheet a problem-statement through solution-specification via backward-calculation process, which is based upon a total enumeration method.

According to progression of processes, it allocates the problem-data. It also tabulates the intermediate recursive computational results. Thus all possible histories to the final stage are evaluated by cumulative membership together with its probability. It gives an optimal decision on each subhistory. In this sense, the tree-table method is totally enumerative. From the tree-table we see that how optimal primitive policy is constructed and that in some situation it happens to be general or Markov. The tree-table method applies to any stochastic optimization problem—it is objective function free.

In the following three tree-tables, we have the simplified notations as follows :

$$\text{history} = x_0 \;\; \mu_0/u_0 \;\; p_0 \;\; x_1 \;\; \mu_1/u_1 \;\; p_1 \;\; x_2 \;\; \mu_G$$

where $\mu_0 = \mu_0(u_0)$, $p_0 = p(x_1|x_0, u_0)$, $\mu_1 = \mu_1(u_1)$, $p_1 = p(x_2|x_1, u_1)$, $\mu_G = \mu_G(x_2)$

$$\text{mini} = \text{minimum value} = \mu_0(u_0) \wedge \mu_1(u_1) \wedge \mu_G(x_2)$$

$$\textit{path} = \textit{path probability} = p(x_1|x_0, u_0)p(x_2|x_1, u_1)$$

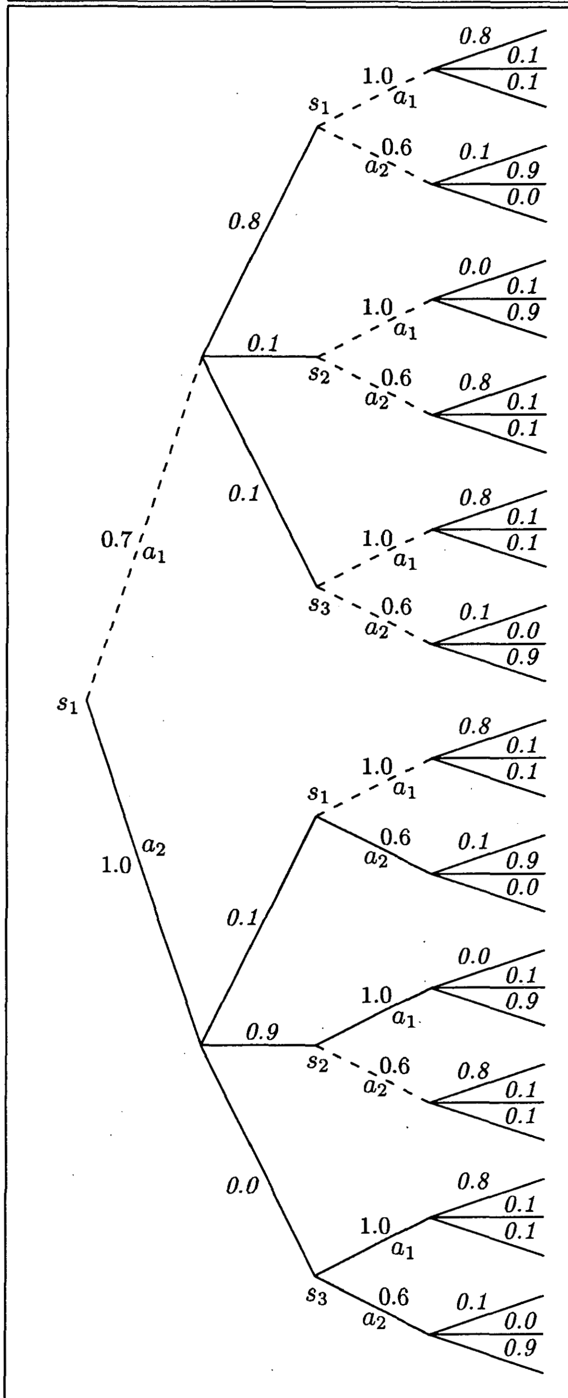
$$\textit{sub-expec.} = \textit{sub-expected value} = \sum_{x_2} [\mu_0(u_0) \wedge \mu_1(u_1) \wedge \mu_G(x_2)] p(x_1|x_0, u_0)p(x_2|x_1, u_1)$$

$$\textit{total-expec.} = \textit{total expected value} = \sum_{x_1} \sum_{x_2} [\mu_0(u_0) \wedge \mu_1(u_1) \wedge \mu_G(x_2)] p(x_1|x_0, u_0)p(x_2|x_1, u_1).$$

The *italic* means probabilities, and **bold** denotes a selection of the larger of the two; the upper expected value and the lower. We see that the decision-tree-table method realizes the computational process for "primitive" recursive equation. This applies to *any* criterion.

$$w_0(s_1) = \underset{\nu \in \Pi_p}{\mathrm{Max}}\, E^{\nu}_{s_1}[\,\mu_0(U_0) \wedge \mu_1(U_1) \wedge \mu_G(X_2)\,]$$

Figure 1 : Two-stage stochastic decision tree-table from state s_1



x_2	μ_G	minimum	*path prob.*	mini × *path*	sub-expec.	total-expec.
s_1	0.3	0.3	*0.64*	0.192		
s_2	1.0	0.7	*0.08*	0.056	0.304	
s_3	0.8	0.7	*0.08*	0.056		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.72*	0.432	**0.456**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.7	*0.01*	0.007	**0.070**	
s_3	0.8	0.7	*0.09*	0.063		0.583
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.01*	0.006	0.036	
s_3	0.8	0.6	*0.01*	0.006		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.7	*0.01*	0.007	0.038	
s_3	0.8	0.7	*0.01*	0.007		
s_1	0.3	0.3	*0.01*	0.003		
s_2	1.0	0.6	*0.0*	0	**0.057**	
s_3	0.8	0.6	*0.09*	0.054		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	1.0	*0.01*	0.01	0.042	
s_3	0.8	0.8	*0.01*	0.008		
s_1	0.3	0.3	*0.01*	0.003		
s_2	1.0	0.6	*0.09*	0.054	**0.057**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	1.0	*0.09*	0.09	**0.738**	
s_3	0.8	0.8	*0.81*	0.648		**0.795**
s_1	0.3	0.3	*0.72*	0.216		
s_2	1.0	0.6	*0.09*	0.054	0.324	
s_3	0.8	0.6	*0.09*	0.054		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	1.0	*0.0*	0	**0**	
s_3	0.8	0.8	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.6	*0.0*	0	**0**	
s_3	0.8	0.6	*0.0*	0		

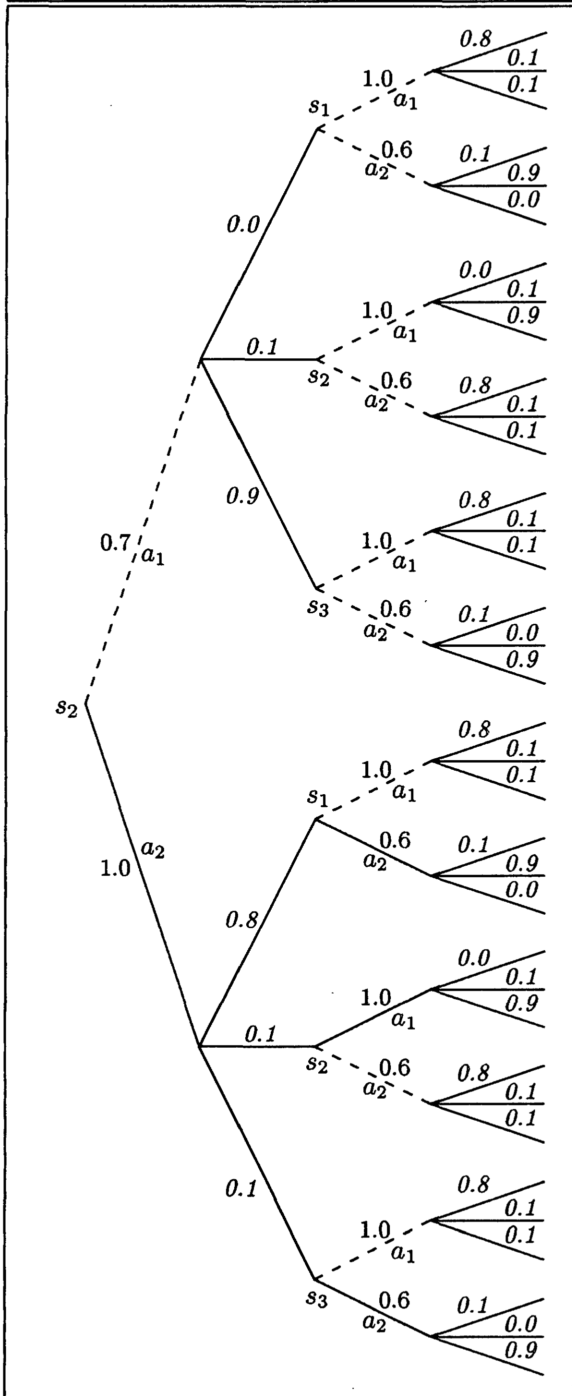
The tree-table (Figure 1) shows how the maximum values $w_0(s_1)$, $w^1(s_1, u_0, x_1)$ together with optimal decisions $\hat{\nu}_0(s_1)$, $\hat{\nu}_1(s_1, u_0, x_1)$ are backward-recursively calculated:

$$w^1(s_1, a_1, s_1) = \mathbf{0.456}, \quad w^1(s_1, a_1, s_2) = \mathbf{0.070}, \quad \ldots, \quad w^1(s_1, a_2, s_3) = \mathbf{0}, \quad w^0(s_1) = \mathbf{0.795}$$

$$\hat{\nu}_1(s_1, a_1, s_1) = a_2, \quad \hat{\nu}_1(s_1, a_1, s_2) = a_1, \quad \ldots, \quad \hat{\nu}_1(s_1, a_2, s_3) = a_1 \text{ or } a_2, \quad \hat{\nu}_0(s_1) = a_2.$$

$$w_0(s_2) = \operatorname*{Max}_{\nu \in \Pi_p} E^{\nu}_{s_2}[\,\mu_0(U_0) \wedge \mu_1(U_1) \wedge \mu_G(X_2)\,]$$

Figure 2 : Two-stage stochastic decision tree-table from state s_2



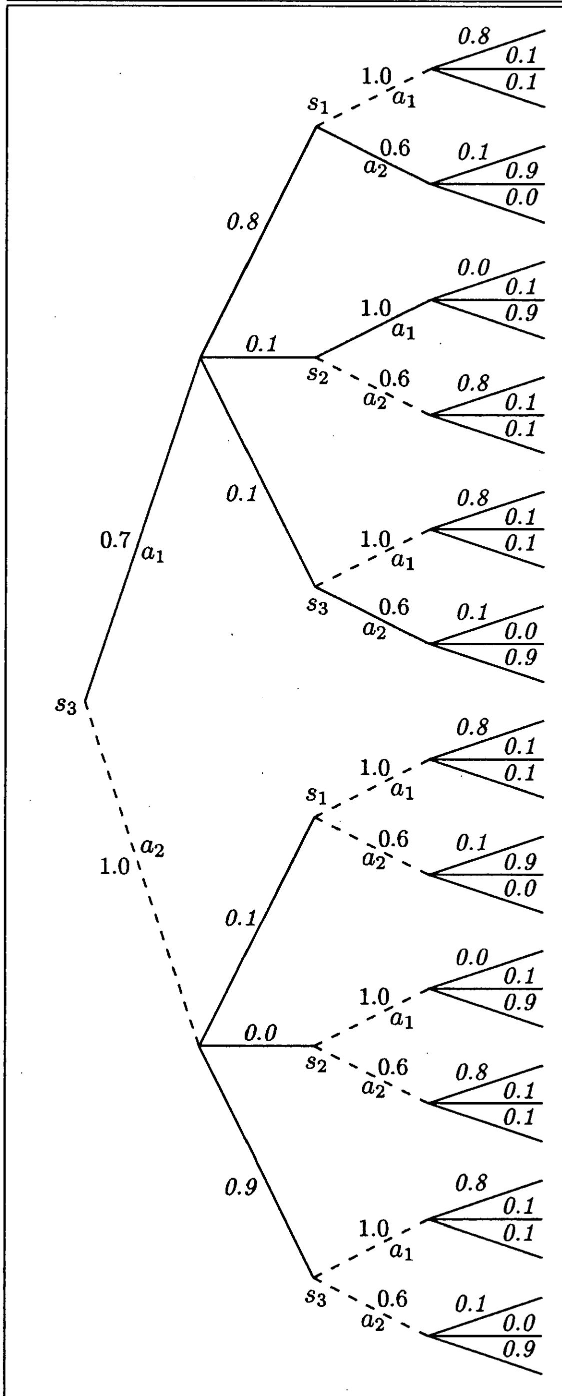
x_2	μ_G	mini mum	*path prob.*	mini × *path*	sub-expec.	total-expec.
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.7	*0.0*	0	**0**	
s_3	0.8	0.7	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.6	*0.0*	0	**0**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.7	*0.01*	0.007	**0.070**	
s_3	0.8	0.7	*0.09*	0.063		0.583
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.01*	0.006	0.036	
s_3	0.8	0.6	*0.01*	0.006		
s_1	0.3	0.3	*0.72*	0.216		
s_2	1.0	0.7	*0.09*	0.063	0.342	
s_3	0.8	0.7	*0.09*	0.063		
s_1	0.3	0.3	*0.09*	0.027		
s_2	1.0	0.6	*0.0*	0	**0.513**	
s_3	0.8	0.6	*0.81*	0.486		
s_1	0.3	0.3	*0.64*	0.192		
s_2	1.0	1.0	*0.08*	0.08	0.336	
s_3	0.8	0.8	*0.08*	0.064		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.72*	0.432	**0.456**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	1.0	*0.01*	0.01	**0.082**	
s_3	0.8	0.8	*0.09*	0.072		**0.595**
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.01*	0.006	0.036	
s_3	0.8	0.6	*0.01*	0.006		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	1.0	*0.01*	0.01	0.042	
s_3	0.8	0.8	*0.01*	0.008		
s_1	0.3	0.3	*0.01*	0.003		
s_2	1.0	0.6	*0.0*	0	**0.057**	
s_3	0.8	0.6	*0.09*	0.054		

The calculation proceeds in decision table-tree as follows : minimum → *path probability* → mini × *path* → sub-expected value → total-expected value. Thus the tree-table method is nothing but the the two-stage recursive computation (20).

$$w_0(s_3) = \underset{\nu \in \Pi_p}{\mathrm{Max}}\, E^{\nu}_{s_3}[\,\mu_0(U_0) \wedge \mu_1(U_1) \wedge \mu_G(X_2)\,]$$

Figure 3 : Two-stage stochastic decision tree-table from state s_3

history: x_0 μ_0/u_0 p_0 x_1 μ_1/u_1 p_1 x_2 μ_G



x_2	μ_G	minimum	*path prob.*	mini × path	sub-expec.	total-expec.
s_1	0.3	0.3	*0.64*	0.192		
s_2	1.0	0.7	*0.08*	0.056	0.304	
s_3	0.8	0.7	*0.08*	0.056		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.72*	0.432	**0.456**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.7	*0.01*	0.007	**0.070**	
s_3	0.8	0.7	*0.09*	0.063		**0.583**
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.6	*0.01*	0.006	0.036	
s_3	0.8	0.6	*0.01*	0.006		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	0.7	*0.01*	0.007	0.038	
s_3	0.8	0.7	*0.01*	0.007		
s_1	0.3	0.3	*0.01*	0.003		
s_2	1.0	0.6	*0.0*	0	**0.057**	
s_3	0.8	0.6	*0.09*	0.054		
s_1	0.3	0.3	*0.08*	0.024		
s_2	1.0	1.0	*0.01*	0.01	0.042	
s_3	0.8	0.8	*0.01*	0.008		
s_1	0.3	0.3	*0.01*	0.003		
s_2	1.0	0.6	*0.09*	0.054	**0.057**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	1.0	*0.0*	0	**0**	
s_3	0.8	0.8	*0.0*	0		0.570
s_1	0.3	0.3	*0.0*	0		
s_2	1.0	0.6	*0.0*	0	**0**	
s_3	0.8	0.6	*0.0*	0		
s_1	0.3	0.3	*0.72*	0.216		
s_2	1.0	1.0	*0.09*	0.09	0.378	
s_3	0.8	0.8	*0.09*	0.072		
s_1	0.3	0.3	*0.09*	0.027		
s_2	1.0	0.6	*0.0*	0	**0.513**	
s_3	0.8	0.6	*0.81*	0.486		

To conclude, we have shown that the three methods – (1) membership-parametric method, (2) history-parametric method and (3) multi-stage stochastic decision tree-table method – yield a common optimal solution. This new triplet is a very powerful tool for optimizing a broard class of stochastic decision processes where the criterion is not necessarily additive but associative through utility function (see [4,5,24]).

3.5 Markov policy is not enough

Let us consider the following Iwamoto, Tsurusaki and Fujita's data [22] :

$$\mu_0(a_1) = 0.9 \quad \mu_0(a_2) = 0.6 \ ; \quad \mu_1(a_1) = 1.0 \quad \mu_1(a_2) = 0.8$$

$$\mu_G(s_1) = 0.5 \quad \mu_G(s_2) = 0.2 \quad \mu_G(s_3) = 0.8$$

	$p(x_{n+1} \mid x_n, a_1)$			$p(x_{n+1} \mid x_n, a_2)$		
$x_n \backslash x_{n+1}$	s_1	s_2	s_3	s_1	s_2	s_3
s_1	0.4	0.5	0.1	0.1	0.6	0.3
s_2	0.2	0.6	0.2	0.7	0.2	0.1
s_3	0.3	0.1	0.6	0.3	0.3	0.4

$$J^0(x_0;\pi) = \sum_{(x_1,x_2)\in X\times X} [\mu_0(u_0) \wedge \mu_1(u_1) \wedge \mu_G(x_2)]p(x_1|x_0,u_0)p(x_2|x_1,u_1)$$

Table 7 : all expected value vectors $J^0(\pi) = \begin{pmatrix} J^0(s_1;\pi) \\ J^0(s_2;\pi) \\ J^0(s_3;\pi) \end{pmatrix}$, where $\pi = \{\pi_0, \pi_1\}$ is Markov

$\pi_0 \backslash \pi_1$	(a_1, a_1, a_1)	(a_1, a_1, a_2)	(a_1, a_2, a_1)	(a_1, a_2, a_2)	(a_2, a_1, a_1)	(a_2, a_1, a_2)	(a_2, a_2, a_1)	(a_2, a_2, a_2)
(a_1, a_1, a_1)	(0.407, 0.434, 0.542)	(0.395, 0.410, 0.470)	(0.452, 0.488, 0.551)	(0.440, 0.464, 0.479)	(0.419, 0.440, 0.551)	(0.407, 0.416, 0.479)	(0.464, 0.494, 0.560)	(0.452, 0.470, 0.488)
(a_1, a_1, a_2)	(0.407, 0.434, 0.422)	(0.395, 0.410, 0.390)	(0.452, 0.488, 0.455)	(0.440, 0.464, 0.423)	(0.419, 0.440, 0.419)	(0.407, 0.416, 0.387)	(0.464, 0.494, 0.452)	(0.452, 0.470, 0.420)
(a_1, a_2, a_1)	(0.407, 0.373, 0.542)	(0.395, 0.365, 0.470)	(0.452, 0.395, 0.551)	(0.440, 0.387, 0.479)	(0.419, 0.366, 0.551)	(0.407, 0.358, 0.479)	(0.464, 0.388, 0.560)	(0.452, 0.380, 0.488)
(a_1, a_2, a_2)	(0.407, 0.373, 0.422)	(0.395, 0.365, 0.390)	(0.452, 0.395, 0.455)	(0.440, 0.387, 0.423)	(0.419, 0.366, 0.419)	(0.407, 0.358, 0.387)	(0.464, 0.388, 0.452)	(0.452, 0.380, 0.420)
(a_2, a_1, a_1)	(0.399, 0.434, 0.542)	(0.375, 0.410, 0.470)	(0.465, 0.488, 0.551)	(0.441, 0.464, 0.479)	(0.398, 0.440, 0.551)	(0.374, 0.416, 0.479)	(0.464, 0.494, 0.560)	(0.440, 0.470, 0.488)
(a_2, a_1, a_2)	(0.399, 0.434, 0.422)	(0.375, 0.410, 0.390)	(0.465, 0.488, 0.455)	(0.441, 0.464, 0.423)	(0.398, 0.440, 0.419)	(0.374, 0.416, 0.387)	(0.464, 0.494, 0.452)	(0.440, 0.470, 0.420)
(a_2, a_2, a_1)	(0.399, 0.373, 0.542)	(0.375, 0.365, 0.470)	(0.465, 0.395, 0.551)	(0.441, 0.387, 0.479)	(0.398, 0.366, 0.551)	(0.374, 0.358, 0.479)	(0.464, 0.388, 0.560)	(0.440, 0.380, 0.488)
(a_2, a_2, a_2)	(0.399, 0.373, 0.422)	(0.375, 0.365, 0.390)	(0.465, 0.395, 0.455)	(0.441, 0.387, 0.423)	(0.398, 0.366, 0.419)	(0.373, 0.358, 0.387)	(0.464, 0.388, 0.452)	(0.440, 0.380, 0.420)

Then each preceding method yields an optimal solution. The optimal value vector is

$$V^0 = \begin{pmatrix} V^0(s_1) \\ V^0(s_2) \\ V^0(s_3) \end{pmatrix} = \begin{pmatrix} 0.465 \\ 0.494 \\ 0.56 \end{pmatrix}.$$

The optimal policy is $\sigma^* = \{\sigma_0^*, \sigma_1^*\}$, where

$$\sigma_0^*(s_1) = a_2, \quad \sigma_0^*(s_2) = a_1, \quad \sigma_0^*(s_3) = a_1 \tag{26}$$

$$\begin{array}{lll} \sigma_1^*(s_1, s_1) = a_1, & \sigma_1^*(s_2, s_1) = a_2, & \sigma_1^*(s_3, s_1) = a_2 \\ \sigma_1^*(s_1, s_2) = a_2, & \sigma_1^*(s_2, s_2) = a_2, & \sigma_1^*(s_3, s_2) = a_2 \\ \sigma_1^*(s_1, s_3) = a_1, & \sigma_1^*(s_2, s_3) = a_1, & \sigma_1^*(s_3, s_3) = a_1. \end{array} \tag{27}$$

Note that

$$\sigma_1^*(s_1, s_1) \neq \sigma_1^*(s_2, s_1).$$

Thus the optimal policy σ^* is not Markov but general. As Table 7 shows, any Markov policy can not attain the optimal value vector V^0. Hence Markov policy is not optimal in fuzzy dynamic programming.

4 Fuzzy System

We propose a dynamic programming method on Bellman and Zadeh's *fuzzy system.* Let us consider a maximization problem of fuzzy expected value of minimum criterion :

$$\mathrm{F}_0(x_0) \qquad \begin{array}{ll} \text{Max} & F_{x_0}^{\pi}[\,\mu_0 \wedge \mu_1 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G\,] \\ \text{s.t.} & \text{(i)} \;\; X_{n+1} \simeq \mu(\cdot|x_n, u_n) \qquad n = 0, 1, \ldots, N-1 \\ & \text{(ii)} \;\; u_n \in U \end{array} \tag{28}$$

Here $\mu = \{\mu(y|x,u)\}$ is a fuzzy transition law which controls a fuzzy state on X for a pair of current realized state x and applied control u. That is, for any given $(x,u) \in X \times U$, $\mu(\cdot|x,u)$ is a membership function of fuzzy set $X(x,u)$ on X. A membership degree of moving to next state x_{n+1} from state x_n under decision u_n on the n-th stage is $\mu(x_{n+1}|x_n,u_n)$ $(0 \leq \mu(x_{n+1}|x_n,u_n) \leq 1)$. Thus the degree that the fuzzy state variable X_{n+1} belongs to the fuzzy set $X(x_n,u_n)$ is represented by $\mu(x_{n+1}|x_n,u_n)$. We write this fuzzy transition as $X_{n+1} \simeq \mu(\cdot|x_n,u_n)$. Further, instead of the so called expectation $E[\cdots]$ by multiplication-addition operation through probability measure, we take a fuzzy expectation $F[\cdots]$ by minimization-maximization operation through fuzzy measure [4,6,16]. Let $f_0(x_0)$ denote the maximum of the fuzzy expected value over Markov class Π. Then we have a backward recursive formula – similar to one in stochastic system — as follows :

Theorem 4.1 *(Iwamoto and Sniedovich [20])*

$$\begin{array}{ll} f_n(x) = \underset{u \in U}{\mathrm{Max}} \bigvee_{y \in X} [\mu_n(x,u) \wedge (f_{n+1}(y) \wedge \mu(y|x,u))] & x \in X, \; 0 \leq n \leq N-1 \\ f_N(x) = \mu_G(x) & x \in X. \end{array} \tag{29}$$

5 Conditional Decision Processes

In this section we propose two types of "conditional" decision process for the "unconditional" decision process discussed in Section 3. 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.

5.1 A posterior conditional process

First we take, at each stage, backward conditional expectation of remaining process *after* performing take-action for the regular stochastic decision process $S_0(x_0)$ (Figure 4). This generates an *a posteriori conditional decision process* (cdp) as follows :

$$\begin{array}{ll}\text{Max} & \mu_0(x_0,u_0)\wedge E^{u_0}_{x_0}[\mu_1(x_1,u_1)\wedge\cdots\wedge E^{u_{N-2}}_{x_{N-2}}[\mu_{N-1}(x_{N-1},u_{N-1})\wedge E^{u_{N-1}}_{x_{N-1}}\mu_G]\cdots] \\ \text{s.t.} & \text{(i)}\ \ X_{n+1}\sim p(\cdot|x_n,u_n) \qquad n=0,1,\ldots,N-1 \\ & \text{(ii)}\ \ u_n\in U \end{array} \tag{30}$$

where $E^u_x\mu$ is a conditional expected value of function $\mu=\mu(\cdot)$ through probability distribution $p(\cdot|x,u)$ together with n-th Markov decision function π_n :

$$E^u_x\mu=\sum_{y\in X}\mu(y)p(y|x,u) \qquad u=\pi_n(x). \tag{31}$$

Let $w_0(x_0)$ denote the maximum value in (30) over Markov class Π. Then we have the backward recursive equation :

Theorem 5.1 *(Bellman and Zadeh [4], Iwamoto, Tsurusaki and Fujita [21])*

$$\begin{aligned} w_n(x) &= \underset{u\in U}{\text{Max}}\left[\mu_n(x,u)\wedge\left(\sum_{y\in X}w_{n+1}(y)p(y|x,u)\right)\right] \qquad x\in X,\ 0\le n\le N-1 \\ w_N(x) &= \mu_G(x) \qquad x\in X. \end{aligned} \tag{32}$$

This is *the* recursive formula Bellman and Zadeh have proposed on stochastic system [4].

Let us now solve the Bellman and Zadeh's multistage decision on stochastic system for the three-state, two-decision, two-stage data [4, pp.B154]. The corresponding recursive equation is solved as follows. First, we have

$$w_2(s_1)=0.3,\quad w_2(s_2)=1.0,\quad w_2(s_3)=0.8.$$

While Bellman and Zadeh [4, pp. B154] give

$$w_1(s_1)=0.6,\quad w_1(s_2)=0.82,\quad w_1(s_3)=0.6$$

$$\pi^*_1(s_1)=a_1,\quad \pi^*_1(s_2)=a_1,\quad \pi^*_1(s_3)=a_2,$$

$$w_0(s_1)=0.8,\quad w_0(s_2)=0.62,\quad w_0(s_3)=0.62$$

$$\pi^*_0(s_1)=a_1,\quad \pi^*_0(s_2)=a_1\ \text{or}\ a_2,\quad \pi^*_0(s_3)=a_1$$

Iwamoto and Fujita [19] specify the exact value of $w_0(x_0)$, $\pi^*_0(x_0)$ as follows:

$$w_0(s_1)=0.798,\quad w_0(s_2)=0.622,\quad w_0(s_3)=0.622$$

$$\pi^*_0(s_1)=a_2,\quad \pi^*_0(s_2)=a_1\ \text{or}\ a_2,\quad \pi^*_0(s_3)=a_1.$$

In fact, the exact value is also verified through the multi-stage stochastic decision tree-table method [15, pp.146-149]. The true optimal solution is tabulated in Table 8:

x_n	$w_0(x_0)$	$\pi^*_0(x_0)$	$w_1(x_1)$	$\pi^*_1(x_1)$	$w_2(x_2)$
s_1	0.798	a_2	0.6	a_1	0.3
s_2	0.622	a_1, a_2	0.82	a_1	1.0
s_3	0.622	a_1	0.6	a_2	0.8

Table 8 : $\{w_0,w_1,w_2\}$ $\pi=\{\pi^*_0,\pi^*_1\}$

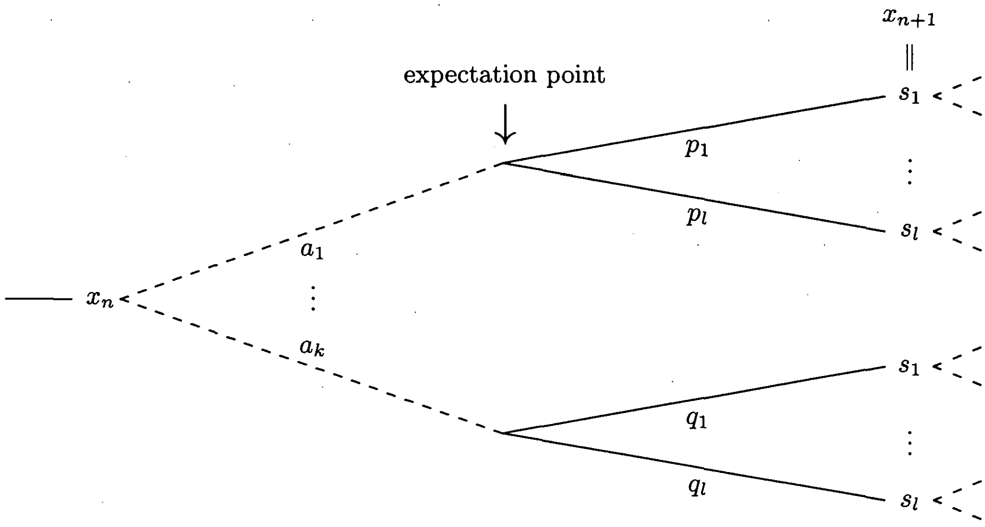


Figure 4 : Conditional expectation after take-action

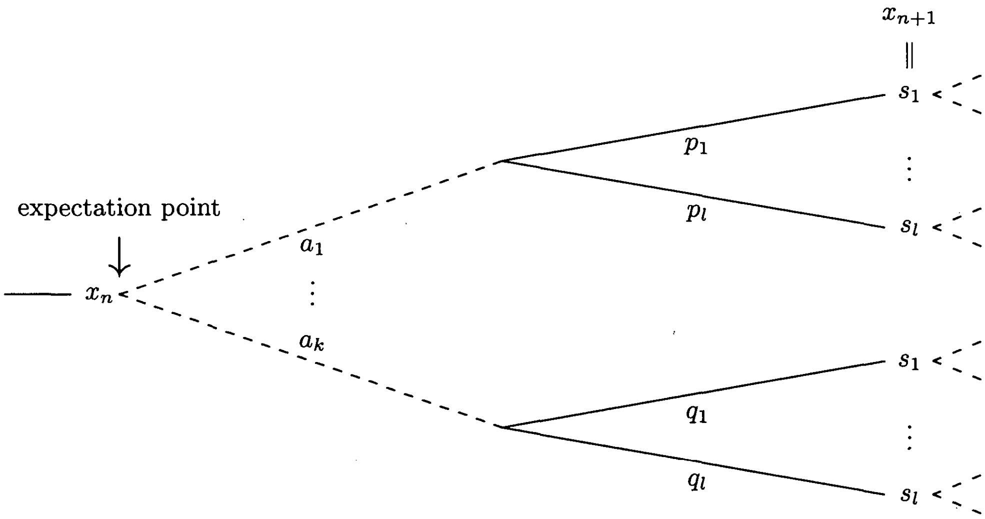


Figure 5 : Conditional expectation before take-action

5.2 A prior conditional process

Second, *before* in turn performing take-action for regular decision process, we take at each stage backward conditional expectation of remaining process (Figure 5). This generates the following *a priori* cdp :

$$\begin{array}{lll} \text{Max} & E^{u_0}_{x_0}[\mu_0(x_0,u_0) \wedge E^{u_1}_{x_1}[\mu_1(x_1,u_1) \wedge \cdots \wedge E^{u_{N-1}}_{x_{N-1}}[\mu_{N-1}(x_{N-1},u_{N-1}) \wedge \mu_G\,]\cdots]\,] & \\ \text{s.t.} & \text{(i)} \quad X_{n+1} \sim p(\cdot|x_n,u_n) & \\ & \text{(ii)} \quad u_n \in U & n = 0,1,\ldots,N-1 \end{array} \tag{33}$$

where the conditional expected value $E^u_x[\mu_n(x,u) \wedge \mu]$ is a priori :

$$E^u_x[\mu_n(x,u) \wedge \mu] = \sum_{y \in X}[\mu_n(x,u) \wedge \mu(y)]p(y|x,u), \qquad u = \pi_n(x) \qquad 0 \le n \le N-1. \tag{34}$$

Let $W_0(x_0)$ denote the maximum value in (33) over Markov class Π. Then we have the backward recursive equation :

Theorem 5.2 *(Iwamoto, Tsurusaki and Fujita [21])*

$$W_n(x) = \underset{u \in U}{\mathrm{Max}} [\mu_n(x,u) \wedge W_{n+1}(y)] p(y|x,u) \qquad x \in X,\ 0 \le n \le N-1$$
$$W_N(x) = \mu_G(x) \qquad x \in X. \tag{35}$$

6 Threshold Probability Criterion

Now we consider the problem of maximizing a threshold probability that *total membership* is greater than or equal to a given *lower grade* $\alpha \in [0, 1]$:

$$\mathrm{P}_0(x_0) \qquad \begin{array}{ll} \mathrm{Max} & P^{\pi}_{x_0}(\mu_0 \wedge \mu_1 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G \ge \alpha) \\ \mathrm{s.t.} & (\mathrm{i})_{\mathrm{n}}\ X_{n+1} \sim p(\cdot|x_n, u_n) \\ & (\mathrm{ii})_{\mathrm{n}}\ u_n \in U \qquad\qquad 0 \le n \le N{-}1 \end{array}$$

where $P^{\pi}_{x_0}$ is the (discrete) probability measure on history space H_N induced through an initial state x_0, the Markov transition law p and a Markov policy $\pi (\in \Pi)$.

We maximize the threshold probability over Markov class Π. Any Markov policy $\pi(\in \Pi)$ determines the threshold probability in $\mathrm{P}_0(x_0)$, which is a "partial" multiple sum :

$$P^{\pi}_{x_0}(\mu_0 \wedge \mu_1 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G \ge \alpha) \;=\; \sum\sum_{(x_1,x_2,\ldots,x_N)\in(*)}\cdots\sum p_0 p_1 \cdots p_{N-1} \quad (p_n = p(x_{n+1}|x_n, u_n)) \tag{36}$$

where the domain $(*)$ is the set of all $(x_1, x_2, \ldots, x_N) \in X^N$ satisfying

$$\mu_0(x_0, u_0) \wedge \mu_1(x_1, u_1) \wedge \cdots \wedge \mu_{N-1}(x_{N-1}, u_{N-1}) \wedge \mu_G(x_N) \quad \ge \quad \alpha. \tag{37}$$

Here the sequence of decisions $\{u_0,\ u_1,\ \ldots,\ u_{N-1}\}$ in (36),(37) is uniquely determined through Markov policy $\pi = \{\pi_0,\ \ldots,\ \pi_{N-1}\}$:

$$u_0 = \pi_0(x_0),\ u_1 = \pi_1(x_1),\ \ \ldots,\ u_{N-1} = \pi_{N-1}(x_{N-1}). \tag{38}$$

As for controlling threshold probability, Markov class Π is not enough for *additive* criteria but general class Π_g is enough [13]. However, in this section, we dare to maximize the threshold probability for *minimum* criteria over Markov class.

Thus our problem $\mathrm{P}_0(x_0)$ is to find the *maximum value function* $v_0 = v_0(x_0)$ and an *optimal policy* $\pi^*(\in \Pi)$ which attains the maximum :

$$v_0(x_0) \;=\; P^{\pi^*}_{x_0}(\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G \ge \alpha) = \underset{\pi \in \Pi}{\mathrm{Max}}\, P^{\pi}_{x_0}(\mu_0 \wedge \cdots \wedge \mu_{N-1} \wedge \mu_G \ge \alpha) \quad x_0 \in X. \tag{39}$$

Then we have the backward recursive relation :

Theorem 6.1 *(Recursive Equation)*

$$v_N(x) \;=\; \begin{cases} 1 & \text{if } \mu_G(x) \ge \alpha \\ 0 & \text{otherwise} \end{cases} \qquad x \in X$$

$$v_n(x) \;=\; \begin{cases} \underset{u;\mu_n(x,u)\ge\alpha}{\mathrm{Max}} \sum_{y\in X} v_{n+1}(y) p(y|x,u) & \text{if } \exists\ u\ ;\ \mu_n(x,u) \ge \alpha \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

$$x \in X, \quad 0 \le n \le N{-}1.$$

Now let us take any pair (n, x). If it satisfies $\mu_n(x,u) \ge \alpha$, then let $\pi^*_n(x)$ denote a $u^* \in U$ which attains the maximum in (40). Otherwise, let $\pi^*_n(x)$ denote any $u \in U$. Then we have an optimal n-th decision function $\pi^*_n : X \to U$. Thus we construct an optimal policy $\pi^* = \{\pi^*_0, \ldots, \pi^*_{N-1}\}$ in Markov class Π ([18]).

References

[1] J.F. Baldwin and B.W. Pilsworth, "Dynamic programming for fuzzy systems with fuzzy environment," *J. Math. Anal. Appl.*, vol.85,no.1, pp.1-23, 1982.

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

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

[4] R.E. Bellman and L.A. Zadeh, "Decision-making in a fuzzy environment," *Management Science*, vol.17, no.4, pp.B141-B164, 1970.

[5] A.O. Esogbue and R.E. Bellman, "Fuzzy dynamic programming and its extensions," *TIMS/Studies in the Management Sciences*, vol.20, pp.147-167, 1984.

[6] T. Fujita and S. Iwamoto, "An optimistic decision-making in fuzzy environment," *Proc. Intl Workshop on Computation, Optimization and Control (The Seventh Bellman Continuum)*, The Santa Fe Institute, 1999; *Applied Mathematics and Computation* **120**(2001), no.1/3, pp.123–137.

[7] R.A. Howard, *Dynamic Programming and Markov Processes*, Mass.: MIT Press, 1960.

[8] 岩本誠一, 動的計画の理論と応用, 数学 **31** 巻 4 号, 1979 年, 331-348.

[9] 岩本誠一, 動的計画論, 九大出版会, 1987.

[10] 岩本誠一, 動的計画の最近の進歩, 第２回ＲＡＭＰシンポジウム論文集, 1990 年, 129-140.

[11] S. Iwamoto, "Decison-making in fuzzy environment: a survey from stochastic decision process," *Proc. 2^{nd} Intl Conf. Knowledge-based Intelligent Electronics Systems (KES '98)*, Adelaide, AUS, 1998, pp.542-546.

[12] S. Iwamoto, "Conditional decision processes with recursive reward function," *J. Math. Anal. Appl.*, vol.230, no.2, pp.193-210, 1999.

[13] S. Iwamoto, "Maximizing threshold probability through invariant imbedding," *Proc. Intl Workshop on Intelligent Systems Resolutions*; The 8-th Bellman Continuum, Hsinchu, ROC, 2000, pp.17-22.

[14] S. Iwamoto, "Fuzzy decision-making through three dynamic programming approaches," *Proc. Intl Workshop on Intelligent Systems Resolutions*; The 8-th Bellman Continuum, Hsinchu, ROC, 2000, pp.23-27.

[15] 岩本誠一, 「第４章：ファジィ動的計画法」, 石井・坂和・岩本編『ファジィＯＲ』、2001 年、朝倉書店、112–157.

[16] S. Iwamoto, "A class of dual fuzzy dynamic programs," *Proc. Intl Workshop on Computation, Optimization and Control (The Seventh Bellman Continuum)*, The Santa Fe Institute, 1999; *Applied Mathematics and Computation* **120**(2001), no.1/3, pp.91–108.

[17] S. Iwamoto, "Fuzzy dynamic programming in stochastic environment," Ed. Y. Yoshida, "Dynamic Aspects in Fuzzy Decision Making" (Studies in Fuzziness and Soft Computiong; Vol.73), Physica-Verlarg, Heidelberg, 2001, pp.27–51.

[18] S. Iwamoto, "Fuzzy decision-making under threshold membership criterion," *Proc. 5^{th} Intl Conf. Knowledge-based Intelligent Information Engineering Systems & Allies Tecnologies (KES' 01)*, Osaka, Japan, 2001, to appear.

[19] S. Iwamoto and T. Fujita, "Stochastic decision-making in a fuzzy environment," *J. Oper. Res. Soc. Japan*, vol.38, no.4, pp.467-482, 1995.

[20] S. Iwamoto and M. Sniedovich, "Sequential decision making in fuzzy environment," *J. Math. Anal. Appl.*, vol.222, no.2, pp.208-224, 1998.

[21] S. Iwamoto, K. Tsurusaki and T. Fujita, "Conditional decision-making in a fuzzy environment," *J. Oper. Res. Soc. Japan*, vol.42, no.2, pp.198-218, 1999.

[22] S. Iwamoto, K. Tsurusaki and T. Fujita, "On Markov policies for minimax decision processes," *J. Math. Anal. Appl.*, vol.253, no.1, pp.58-78, 2001.

[23] S. Iwamoto, T. Ueno and T. Fujita, "Controlled Markov chains with utility functions," *Proc. of Intl Workshop on Markov Processes and Controlled Markov Chains*, Changsha, China, 1999, in press.

[24] J. Kacprzyk, "Decision-making in a fuzzy environment with fuzzy termination time," *Fuzzy Sets and Systems*, vol.1, no.2, pp.169-179, 1978.

[25] J. Kacprzyk, "Fuzzy optimal control revisited: Toward a new generation of fuzzy control?" *Proc. of* 2^{nd} *Intl Conf. Fuzzy Logic & Neural Networks*, Iizuka, Japan, 1992, pp.429-432.

[26] J. Kacprzyk, "Multistage control of a stochastic system under fuzzy goals and constraints using a generic algorithm," 7^{th} *IFSA World Congress*, Prague 1997, vol.2, pp.306-311.

[27] J. Kacprzyk and A.O. Esogbue, "Fuzzy dynamic programming: Main developments and applications," *Fuzzy Sets and Systems*, vol.81, no.1, pp.31-45, 1996.

[28] J. Kacprzyk and P. Staniewski, "A new approach to the control of stochastic systems in a fuzzy environment," *Archiwum Automatyki i Telemechaniki*, vol.25, no.4, pp.443-444, 1980.

[29] E.S. Lee, *Quasilinearization and Invariant Imbedding*, New York: Academic Press, 1968.

[30] 日本 OR 学会編、「OR 用語辞典」、日科技連出版社、2000.

[31] 日本 OR 学会編、「OR 事典 2000」CD-ROM 版、日本 OR 学会、2000.

[32] M.L. Puterman, *Markov Decision Processes : discrete stochastic dynamic programming*, New York: Wiley & Sons, 1994.

[33] M. Sniedovich, *Dynamic Programming*, New York: Marcel Dekker, 1992.

[34] W.E. Stein, "Optimal stopping in a fuzzy environment," *Fuzzy Sets and Systems*, vol.3, no.2, pp.253-259, 1980.