Computers Math. Applic. Vol. 21, No. 11/12, pp. 117–127, 1991
Printed in Great Britain.
0097-4943/91 $3.00 + 0.00

COMPUTATIONAL ASPECTS AND APPLICATIONS OF A BRANCH AND BOUND ALGORITHM FOR FUZZY MULTISTAGE DECISION PROCESSES

AUGUSTINE O. ESOGBUE
School of Industrial and Systems Engineering, Georgia Institute of Technology
Atlanta, Georgia 30332

Abstract—Fuzzy multistage decision processes are normally modelled and solved via fuzzy dynamic programming algorithms. We first review the field and present a branch and bound type alternative due to Kacprzyk. We next rectify some computational errors in Kacprzyk's example and show some examples from environmental damage reduction planning studies where the algorithm is applicable and efficient.

1. INTRODUCTION

What is now known as a fuzzy decision process with the system under control, the goals, decisions, and constraints defined over fuzzy sets, may be formally stated as follows:

Given a set of $X = \{x\}$ of alternatives; a fuzzy goal G and a fuzzy constraint C, all defined over X, i.e., $G \subset X$ and $C \subset X$, then the fuzzy decision D defined also over the space X is simply the intersection of goals and constraints, i.e.,

$$D = G \cap C. \tag{1}$$

Another way to represent (1) in terms of its membership function $\mu_D(x)$ is

$$\mu_D(x) = \mu_G(x) \wedge \mu_C(x) = \min\{\mu_G(x), \mu_C(x)\}. \tag{2}$$

An optimal policy is a sequence of controls which optimizes the value of the membership function.

In a completely fuzzy system operating in a fuzzy environment, we may assume that the usual system descriptors of state, decision, transformation and return functions as well as the termination time are fuzzified. For such a system then, we may expect the usual issues and questions normally discussed in their non fuzzy analogs to be of concern. Indeed, they have been raised by various authors such as Esogbue and Ramesh [1], Kacprzyk [2–3], Stein [4], Esogbue and Bellman [5], Baldwin *et al.* [6], etc. The seminal work by Bellman and Zadeh [7] provides the foundation for all work in this area.

2. FUZZY MULTISTAGE DECISION PROCESSES

A review of processes of this genre is provided by Esogbue and Bellman [5] with an update emphasizing applications by Kacprzyk and Esogbue [8]. Briefly and for simplicity, let us for the moment focus attention on the following time-invariant, finite-state deterministic automaton $A = \{U, X, f\}$, where $U = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}, X = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ are finite sets known as the input (control), and state spaces respectively, and $f : X \times U \to X$. The temporal evolution of A is described by the state equation

$$x_{t+1} = f((x_t, u_t)), \qquad t = 0, 1, \ldots, N-1, \tag{3}$$

This work was made possible by funds from the U.S. Department of the Interior, U.S. Geological Survey, Water Resources Division, through Research Grant No.14-08-0001-G1629. The contributions of Kejiao Guo are gratefully acknowledged.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX where $x_0 \in X$ is the initial state and N is the final or termination time which we assume to be fixed.

Let us assume that $\forall t, \exists$ i) a fuzzy constraint $C^t \subseteq U$ characterized by a membership function $\mu_t(u_t)$, and ii) a fuzzy goal $G^N \subseteq X$. Given an initial state X_0, we are interested in finding a maximizing decision via dynamic programming.

We can at once express the decision, a decomposable fuzzy set in $U \times U \times \ldots \times U$ as

$$R = C^0 \cap C^1 \cap \ldots \cap C^{N-1} \cap \bar{G}^N, \tag{4}$$

where $\bar{G}^N$ is the fuzzy set in $U \times U \times \ldots \times U$ which induces G^N in X. In terms of membership functions, we have

$$\mu_D(u_0, u_1, \ldots, u_{N-1}) = \min(\mu_{C^0}(u_0), \mu_{C^1}(u_1), \ldots, \mu_{C^{N-1}}(u_{N-1}), \mu_{G^N}(x_N)), \tag{5}$$

where X_N is expressible as a function of x_0 and $u_0, \ldots, u_{N-1}$.

We may rephrase the problem as: find the sequence of inputs $u_0, \ldots, u_{N-1}$ which maximizes μ_D of (5). The solution may be conveniently expressed in terms of π_t, the policy function with

$$U_t = \pi_t(x_t), \qquad t = 0, 1, 2, \ldots, N-1.$$

Dynamic programming may then be employed to obtain both the π_t and the maximizing decisions $U_0^M, \ldots, U_{N-1}^M$. More specifically, this reduces to

$$\begin{aligned}\mu_D(U_0^M, \ldots, U_{n-1}^M) = \max_{u_0, \ldots, u_{N-2}} \max_{u_{N-1}} &(\mu_0(u_0) \wedge \ldots \wedge \mu_{N-2}(u_{N-2}) \\ &\wedge \mu_{N-1}(u_{N-1}) \wedge \mu_{G^N}(f(x_{N-1}, u_{N-1})).\end{aligned} \tag{6}$$

Now, if γ is a constant and g is any function of u_{N-1}, we have the identity

$$\max_{u_{N-1}}(\gamma \wedge g(u_{N-1})) = \gamma \wedge \max_{u_{N-1}} g(u_{N-1}).$$

Consequently, (6) may be rewritten as

$$\mu_D(u_0^M, \ldots, u_{N-1}^M) = \max_{u_0, \ldots, u_{N-1}} (\mu_0(u_0) \wedge \ldots \wedge \mu_{N-2}(u_{N-2}) \wedge \mu_{G^{N-1}}(x_{N-1})), \tag{7}$$

where

$$\mu_{G^{N-1}}(x_{N-1}) = \max_{u_{N-1}}(\mu_{N-1}(u_{N-1}) \wedge \mu_{G^N}(f(x_{N-1}, u_{N-1}))) \tag{8}$$

may be regarded as the membership function of a fuzzy goal at time $t = N-1$ which is induced by the given goal G^N at time $t = N$.

On repeating this backward iteration, which is a simple instance of dynamic programming, we obtain the set of recurrence equations

$$\begin{aligned}\mu_{G^{N-v}}(x_{N-v}) &= \max_{u_{N-v}}(\mu(U_{N-v}) \wedge \mu_{G^{N-v}} + \neg(x_{N-v+1}), \\ x_{N-v+1} &= f(x_{N-v}, U_{N-v}), \qquad v = 1, \ldots, N,\end{aligned} \tag{9}$$

which yield the solution to the problem. Thus, a maximizing decision $u_0^M, \ldots, u_{N-1}^M$ is given by the succesive maximizing values of u_{N-v} in (9), with u_{N-v}^M defined as a function of $X_{N-v}, v = 1, \ldots, N$.

3. A BRANCH AND BOUND ALGORITHM FOR THE FUZZY DECISION PROBLEM

The fuzzy dynamic program presented in the foregoing, as well as its various variants, has applications in many real life situations. For example, its use in resource allocation and scheduling are well documented in [5] and recently in [8]. The solution approaches proposed for such models include variations of dynamic programming algorithms, branch and bound procedures, and hybrid dynamic programming-branch and bound algorithms. In the sequel, we sketch aspects of one such branch and bound algorithm proposed by Kacprzyk [2] for the multistage fuzzy decision problem.

Consider a fuzzy multistage decision problem such as was described in Section 2. The system under control may be represented as a conditioned fuzzy set whose membership function is given by

$$\mu_{X_{t+1}}(x_{t+1} \mid x_t, u_t).$$

The system's dynamics is then governed by

$$\mu_{X_{t+1}}(x_{t+1}) = \max_{x_t}\{\mu_{X_t}(x_t) \wedge \mu_{X_{t+1}}(x_{t+1} \mid x_t, u_t)\}, \tag{10}$$

$$\mu_{X_{t+2}}(x_{t+2}) = \max_{x_{t+1}}\{\max_{x_t}\{\mu_{X_t}(x_t) \wedge \mu_{X_{t+1}}(x_{t+1} \mid x_t, u_t)\} \wedge \mu_{X_{t+2}}(x_{t+2} \mid x_{t+1}, u_{t+1})\}, \tag{11}$$

and, in general,

$$\begin{aligned}\mu_{X_{t+n}}(x_{t+n}) = \max_{x_{t+n-1}} \big(\max_{x_{t+n-2}} (\dots\{\max_{x_t} \mu_{X_t}(x_t) \wedge \mu_{X_{t+1}}(x_{t+1} \mid x_t, u_t) \\ \wedge\, \mu_{X_{t+2}}(x_{t+2} \mid x_{t+1}, u_{t+1}) \wedge \dots\}) \wedge \mu_{X_{t+n}}(x_{t+n} \mid x_{t+n-1}, u_{t+n-1})\big).\end{aligned} \tag{12}$$

If both the state and control spaces are finite, then (10)–(12) can be written more compactly. Let $M(u_t)$ denote a matrix whose (i,j) element is given by

$$M_{ij}(u_t) = \mu_{X_i}(x_i \mid x_j, u_t), \qquad u_t \in U, \tag{13}$$

and $\tilde{x}_{t+1}$ and $\tilde{x}_t$ denote the column vectors whose i-th elements are $\mu_{X_{t+1}}(x_{t+1})$ and $\mu_{X_t}(x_t)$, respectively, evaluated at x_{t+1} and x_t equal to x_i, for $i = 1, 2, \dots,$ max number of states, say n.

Rewriting equation (13) in matrix terms results in

$$\tilde{x}_{t+1} = M(u_t)\,\tilde{x}_t, \tag{14}$$

with $M(u_t)\,\tilde{x}_t$, the max-min matrix product of $M(u_t)$ and $\tilde{x}_t$. In general then,

$$\tilde{x}_{t+n} = M(u_{t+n-1})\,M(u_{t+n-2})\dots M(u_t)\,\tilde{x}_t. \tag{15}$$

We will make use of these operations when illustrating the hybrid dynamic programming branch-and-bound technique with an example.

Recall that the objective of the decision making problem is to seek the sequence of inputs $u_1^*, u_2^*, \dots, u_N^*$ that will yield the maximal membership functions. Thus, we need to find

$$\begin{aligned}&\mu_D(u_1^*, u_2^*, \dots, u_N^*) = \max_{u_i, t_i}\{\min\{\mu_{C^1}(u_1)\,\mu_{G^1}(x_1), \dots, \mu_{C^N}(u_N)\,\mu_{\bar{G}^N}(x_N)\}\}, \\ &\text{for } i = 1, 2, \dots, N.\end{aligned} \tag{16}$$

It is assumed that at each stage i a fuzzy goal G^i with membership function $\mu_{G^i}(x_i)$ is set, and the aim of the control u_i is to return the state of the system x_i as close as possible to a predetermined one given by G^i. As a measure of the closeness between X_N and G^N we may use the relative distance $d(X_N, G^N)$ between the two fuzzy sets:

$$d(X_N, G^N) = \left(\frac{1}{n}\right) \left(\sum_{i=1}^{n} \mid \mu_{X^i}(x_i) - \mu_{G^i}(x_i) \mid\right), \tag{17}$$

where n is the number of all possible states that the system can be in. Note further that the $\mu_{\bar{G}^N}(x)$ in equation (16) is given by $\mu_{\bar{G}^N}(x) = 1 - d(X_N, G^N)$.

Let the set of controls be $U = \{a_1, a_2, \dots, a_m\}$. The decision process can conveniently be represented by a decision tree whose root is the initial state of the system X_0. The edges are associated with the particular values of the controls applied while the nodes are associated with subsequent states attained. Let $X_{klm\dots w}$ denote the state of the system attained at stage k from state X_0 through the sequence of controls $a_l, a_m, \dots, a_w$.

Now consider a general case where we have N goals and N constraints. Let the sequence $u_1, u_2, \dots, u_N$ be called a decision, while the subsequence $u_1, u_2, \dots, u_i, i \leq N$, the partial decision at stage i, be denoted by d_i. Correspondingly, let the value of equation (16), which is also its grade of membership in the fuzzy decision D, be called the value of the decision $u_1, u_2, \dots, u_N$. Similarly, let the membership function value of the partial decision be the following equation

$$v_i = v_i(d_i) = \mu_{C^1}(u_1)\,\mu_{G^1}(x_1) \wedge \dots \wedge \mu_{C^i}(u_i)\,\mu_{G^i}(x_i). \tag{18}$$

For the value of the partial decision at stage i, but without considering the fuzzy goal G^i at this stage, the value v_i' is given by

$$v_i' = v_i'(d_i) = \mu_{C^1}(u_1)\,\mu_{G^1}(x_1) \wedge \dots \wedge \mu_{C^i}(u_i). \tag{19}$$

The problem is to determine a maximizing decision, i.e., the partial decision d_N with the best membership function value in equation (16).

The principial idea of the method is based on the following property:

$$\text{For } k \leq m \qquad \begin{aligned} &\min\{\mu_{C^1}(u_1)\,\mu_{G^1}(x_1), \dots, \mu_{C^k}(u_k)\,\mu_{G^k}(x_k)\} \\ &\quad \geq \min\{\mu_{C^1}(u_1)\,\mu_{G^1}(x_1), \dots, \mu_{C^m}(u_m)\,\mu_{G^m}(x_m)\}. \end{aligned} \tag{20}$$

We branch via the controls applied at particular control stages and we bound as follows:

At the k-th control stage, we add that control that will maximize the fuzzy decision function at that stage.

If we consider consecutively partial decisions at successive stages $i = 1, 2, \dots, N$, we should take into account only those found so far that have the highest value. We note that both v_i and v_i' are monotone nonincreasing functions of increasing i. Thus, we apply only to the best partial decision a further control and proceed to a future state, obtain a new partial decision, compute its value and compare it with the existing one, choosing only for further considerations, the one with the highest value. The process is terminated when we obtain a complete decision d with value greater than all those considered so far. Evidently, it need not be unique.

4. COMPUTATIONAL ASPECTS

Kacprzyk considers two versions of this problem. The first version considers N fuzzy constraints with the fuzzy goal applied only at the N-th stage. The second one considers N fuzzy goals. In the first example, the maximizing decision was unique. In the second example with three goals, two decisions, i.e., (a_2, a_3, a_1) and (a_2, a_3, a_2) were obtained. Note that in each example, the same fuzzy matrix is applied to all stage transitions. Although this illustrates the nonuniqueness of this solution, the wrong solution was obtained. We will show that computational errors in Kacprzyk's example can be avoided by a correct application of the algorithm.

Suppose we have a multistage decision process with N fuzzy constraints as well as N fuzzy goals. Following the foregoing model, let the state of the system be given by $X = \{\sigma_1, \sigma_2, \dots, \sigma_5\}$, while the controls are $U = \{a_1, a_2, a_3\}$. Let the system under control be equated with a conditioned fuzzy set: $\mu_{X_{i+1}}(x_{i+1} \mid x_i, u_i)$. Thus, we have at each stage five possible states and three possible controls that can be applied. Consider the following three matrices $M(a_1), M(a_2)$ and $M(a_3)$ as required by equation (13) which show, for each of the three controls $U(a_1, a_2, a_3)$, the membership functions for possible limitations from x_i to x_{i+1} for each of the various stages.

$u = a_1$

$M^T(a_1)$

x_i \ x_{i+1}	σ_1	σ_2	σ_3	σ_4	σ_5
σ_1	1	0.1	0.9	0.1	0.2
σ_2	0.8	0.5	0.7	0.3	0.5
σ_3	0.7	0.9	0.5	0.5	0.7
σ_4	0.5	0.7	0.7	0.3	0.4
σ_5	0.2	0.3	0.9	0.7	0.3

$u = a_2$

$M^T(a_2)$

x_i \ x_{i+1}	σ_1	σ_2	σ_3	σ_4	σ_5
σ_1	0.3	0.9	1	0.4	0.6
σ_2	0.5	0.7	0.5	0.2	0.3
σ_3	0.8	0.5	0.3	0.5	0.2
σ_4	0.9	0.7	0.7	0.9	0.5
σ_5	0.7	0.9	0.7	1	0.7

$u = a_3$

$M^T(a_3)$

x_i \ x_{i+1}	σ_1	σ_2	σ_3	σ_4	σ_5
σ_1	0.5	0.7	0.7	1	0.7
σ_2	0.7	0.8	0.1	0.5	0.9
σ_3	0.8	0.1	0.2	0.3	1
σ_4	0.9	0.2	0.3	0.5	0.8
σ_5	1	0.5	0.4	0.7	0.4

In addition to the foregoing, we are provided the following data on the system

i) fuzzy initial state

$$X_0 = 0.1/\sigma_1 + 0.2/\sigma_2 + 0.3/\sigma_3 + 0.7/\sigma_4 + 1/\sigma_5,$$

ii) the fuzzy constraints

$$C^1 = 0.3/a_1 + 0.7/a_2 + 1/a_3,$$
$$C^2 = 0.5/a_1 + 1/a_2 + 0.7/a_3,$$
$$C^3 = 1/a_1 + 0.8/a_2 + 0.6/a_3,$$

and iii) the fuzzy goals

$$G^1 = 0.7/\sigma_1 + 1/\sigma_2 + 0.7/\sigma_3 + 0.4/\sigma_4 + 0.1/\sigma_5,$$
$$G^2 = 0.2/\sigma_1 + 0.5/\sigma_2 + 0.7/\sigma_3 + 0.8/\sigma_4 + 1/\sigma_5,$$
$$G^3 = 0.4/\sigma_1 + 0.7/\sigma_2 + 1/\sigma_3 + 0.7/\sigma_4 + 0.4/\sigma_5,$$

We can now perform our computations to determine the optimal control policy. Starting from X_0 and applying controls a_1, a_2, a_3 we obtain, using equations (18) and (19),

$$v_1'(a_1) = 0.3,$$
$$v_1'(a_2) = 0.7,$$
$$v_1'(a_3) = 1.$$

Thus, working backwards we consider a_3 and proceed to calculate $X_{13}, \mu_{\tilde{G}^1}$ and $v_1(a_3)$. The result is

$$X_{13} = 1/\sigma_1 + 0.5/\sigma_2 + 0.4/\sigma_3 + 0.7/\sigma_4 + 0.7/\sigma_5,$$
$$\mu_{\tilde{G}^1} = 1 - d(X_{13}, G^1) = 1 - \frac{1}{5}(0.3 + 0.5 + 0.3 + 0.6) = 0.6,$$

and $v_1(a_3) = 1 \wedge 0.6 = 0.6$ (from equation 18).

Next, we consider a_2 and proceed to X_{12} given by

$$X_{12} = 0.7/\sigma_1 + 0.9/\sigma_2 + 0.7/\sigma_3 + 1/\sigma_4 + 0.7/\sigma_5.$$

As before, $\mu_{\tilde{G}^1}$ and $v_1(a_3)$ are computed as

$$\mu_{\tilde{G}^1} = 1 - d(X_{12}, G^1) = 1 - \frac{1}{5}(0.1 + 0.6 + 0.6) = 0.74,$$

and $v_1(a_2) = 0.7 \wedge 0.74 = 0.7$.

Thus, we start from X_{12} and applying a_1, a_2, a_3 we obtain the values of the partial decisions.

$$\begin{aligned} v_2'(a_2, a_1) &= 0.7 \wedge 0.5 = 0.5, \\ v_2'(a_2, a_2) &= 0.7 \wedge 7 = 0.7, \\ v_2'(a_2, a_3) &= 0.7 \wedge 0.7 = 0.7. \end{aligned}$$

We next proceed to compute X_{222} and X_{223}. These are given, respectively, by

$$X_{222} = 0.9/\sigma_1 + 0.7/\sigma_2 + 0.7/\sigma_3 + 0.9/\sigma_4 + 0.7/\sigma_5,$$

$$X_{223} = 0.9/\sigma_1 + 0.8/\sigma_2 + 0.7/\sigma_3 + 0.7/\sigma_4 + 0.9/\sigma_5,$$

Now for X_{222},

$$\mu_{\tilde{G}^2} = 1 - d(X_{222}, G^2) = 1 - \frac{1}{5}(0.7 + 0.2 + 0 + 0.1 + 0.3) = 0.74,$$

and for X_{223},

$$\mu_{\tilde{G}^2} = 1 - d(X_{223}, G^2) = 1 - \frac{1}{5}(0.7 + 0.3 + 0 + 0.1 + 0.1) = 0.76,$$

while $v_2(a_2, a_2) = 0.7 \wedge 0.74 = 0.7$ and $v_2(a_2, a_3) = 0.7 \wedge 0.76 = 0.7$.

We may now compute the values of the partial decisions as done previously. Thus we start from X_{223} and applying a_1, a_2, a_3 we obtain

$$\begin{aligned} v_3'(a_2, a_2, a_1) &= 0.7 \wedge 1 = 0.7, & v_3'(a_2, a_3, a_1) &= 0.7 \wedge 1 = 0.7, \\ v_3'(a_2, a_2, a_2) &= 0.7 \wedge 0.8 = 0.7, & v_3'(a_2, a_3, a_2) &= 0.7 \wedge 0.8 = 0.7, \\ v_3'(a_2, a_2, a_3) &= 0.7 \wedge 0.6 = 0.6, & v_3'(a_2, a_3, a_3) &= 0.7 \wedge 0.6 = 0.6. \end{aligned}$$

Finally, we proceed to compute $X_{3231}, X_{3222}, X_{3231}$ and X_{3232} respectively as

$$\begin{aligned} X_{3221} &= 0.9/\sigma_1 + 0.7/\sigma_2 + 0.9/\sigma_3 + 0.7/\sigma_4 + 0.7/\sigma_5, \\ X_{3222} &= 0.9/\sigma_1 + 0.9/\sigma_2 + 0.9/\sigma_3 + 0.9/\sigma_4 + 0.7/\sigma_5, \\ X_{3231} &= 0.9/\sigma_1 + 0.7/\sigma_2 + 0.9/\sigma_3 + 0.7/\sigma_4 + 0.7/\sigma_5, \\ X_{3232} &= 0.7/\sigma_1 + 0.9/\sigma_2 + 0.9/\sigma_3 + 0.9/\sigma_4 + 0.7/\sigma_5. \end{aligned}$$

At this stage, we need to find μ_{G^3} and v_3 for $X_{3221}, X_{3222}, X_{3231}$ and X_{3232}

$$\begin{aligned} &\text{for } X_{3221}, && \mu_{\tilde{G}^3} = 1 - d(X_{3221}, G^3) = 0.82 && \text{and } v_3(a_2, a_2, a_1) = 0.7 \wedge 0.82 = 0.7, \\ &\text{for } X_{3222}, && \mu_{\tilde{G}^3} = 1 - d(X_{3222}, G^3) = 0.74 && \text{and } v_3(a_2, a_2, a_2) = 0.7 \wedge 0.74 = 0.7, \\ &\text{for } X_{3231}, && \mu_{\tilde{G}^3} = 1 - d(X_{3231}, G^3) = 0.82 && \text{and } v_3(a_2, a_3, a_1) = 0.7 \wedge 0.82 = 0.7, \\ &\text{for } X_{3232}, && \mu_{\tilde{G}^3} = 1 - d(X_{3232}, G^3) = 0.78 && \text{and } v_3(a_2, a_3, a_2) = 0.7 \wedge 0.78 = 0.7. \end{aligned}$$

Since there is no other partial decision with higher value, these four (a_2, a_2, a_1), (a_2, a_2, a_2), (a_2, a_3, a_1) and (a_2, a_3, a_2) are the maximizing ones.

We note that in this example the four values are equal in contrast to the two obtained by Kacprzyk. As correctly pointed out by Kacprzyk, however, the solutions need not be unique.

5. APPLICATIONS TO DISASTER CONTROL PLANNING

The algorithm proposed by Kacprzyk, which has been illustrated in the foregoing, can be modified and used in conjuction with a model for optimal disaster control planning. Two examples are instructive—one from earthquake damage and the other from flood damage control planning studies. We present two examples from flood control illustrating scenarios where the policy is unique and non unique, respectively.

In both examples, we have a fuzzy state of flood damage representing five levels: no damage, slight damage, moderate damage, severe damage and disastrous damage. The decision space concerns three investment levels for each of the three flood control measures (structural and/or non-structural). These measures represent the three stages of the model. There are three fuzzy goals, different for each control measure, and expressed in terms of membership functions. Similarly, we have three fuzzy constraints, expressed in terms of membership functions, for each measure. Additionally, we are given the membership function for the fuzzy initial state. The problem is to determine the optimal combination of controls or measures together with the associated funding levels to put in place, so as to minimize the damage levels due to incipient floods. We state parenthetically, that fuzzy set theory is used to model these systems because usually the damage levels and goals can not be stated precisely in such disaster control systems.

Note that we have the same fuzzy initial state and the same goal for the first measure, in the two examples but different goals and constraints for the other measures in the two examples. The first example led to a single unique optimal decision solution while the second generated two optimal solutions. The examples and computations are given below.

Example 1: A problem with only one optimal solution.

****

STATE SPACE: the flood damage level
{no, slight, moderate, severe, disastrous}
DECISION SPACE: the investment level for the measures
{low, medium, high}

****

(1) The Membership Function of Initial State:

X0 = 0.1/no + 0.4/slight + 0.7/moderate + 1.0/severe + 0.8/disastrous

(2) The Membership Function of Goal State:

G1 = 0.4/no + 0.6/slight + 0.6/moderate + 0.7/severe + 0.5/disastrous
G2 = 0.7/no + 0.8/slight + 0.5/moderate + 0.4/severe + 0.2/disastrous
G3 = 1.0/no + 0.7/slight + 0.4/moderate + 0.1/severe + 0.0/disastrous

(3) The Membership Function of Constraint For Measures:

C1 = 0.35/low + 0.85/medium + 0.60/high
C2 = 0.25/low + 0.50/medium + 0.75/high
C3 = 1.00/low + 0.70/medium + 0.40/high

(4) The Fuzzy Transform Matrix:

$$T1\ (\text{low}) = \begin{vmatrix} 0.3 & 0.8 & 0.5 & 0.3 & 0.1 \\ 0.2 & 0.3 & 0.8 & 0.5 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.8 & 0.5 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.8 \\ 0.0 & 0.0 & 0.1 & 0.2 & 0.3 \end{vmatrix} \quad T1\ (\text{medium}) = \begin{vmatrix} 0.6 & 0.9 & 0.4 & 0.1 & 0.0 \\ 0.1 & 0.6 & 0.9 & 0.4 & 0.1 \\ 0.0 & 0.1 & 0.6 & 0.9 & 0.4 \\ 0.0 & 0.0 & 0.1 & 0.6 & 0.9 \\ 0.0 & 0.0 & 0.0 & 0.1 & 0.6 \end{vmatrix} \quad T1\ (\text{high}) = \begin{vmatrix} 0.4 & 0.6 & 0.8 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.6 & 0.8 & 0.2 \\ 0.2 & 0.3 & 0.4 & 0.6 & 0.8 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.6 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.4 \end{vmatrix}$$

T2 (low) =

0.4	0.7	0.5	0.3	0.1
0.3	0.4	0.7	0.5	0.3
0.2	0.3	0.4	0.7	0.5
0.1	0.2	0.3	0.4	0.7
0.0	0.1	0.2	0.3	0.4

T2 (medium) =

0.5	0.8	0.4	0.2	0.1
0.3	0.5	0.8	0.4	0.2
0.1	0.3	0.5	0.8	0.4
0.0	0.1	0.3	0.5	0.8
0.0	0.0	0.1	0.3	0.5

T2 (high) =

0.5	0.6	0.8	0.4	0.1
0.3	0.5	0.6	0.8	0.4
0.2	0.3	0.5	0.6	0.8
0.1	0.2	0.3	0.5	0.6
0.0	0.1	0.2	0.3	0.5

T3 (low) =

0.3	0.7	0.6	0.4	0.2
0.2	0.3	0.7	0.6	0.4
0.1	0.2	0.3	0.7	0.6
0.0	0.1	0.2	0.3	0.7
0.0	0.0	0.1	0.2	0.3

T3 (medium) =

0.7	0.9	0.3	0.2	0.0
0.4	0.7	0.9	0.3	0.2
0.1	0.4	0.7	0.9	0.3
0.0	0.1	0.4	0.7	0.9
0.0	0.0	0.1	0.4	0.7

T3 (high) =

0.3	0.6	0.9	0.2	0.1
0.2	0.3	0.6	0.9	0.2
0.2	0.2	0.3	0.6	0.9
0.1	0.2	0.2	0.3	0.6
0.0	0.1	0.2	0.2	0.3

****

SOLUTION:

stage 1~~~~~~

$v1'$ (low) = 0.35

$v1'$ (medium) = 0.85 ***

$v1'$ (high) = 0.60

X1m = 0.4/no + 0.7/slight + 0.9/moderate + 0.8/severe + 0.6/disastrous

1.0 - D[X1m,G1] = 1.0 - [0.0 + 0.1 + 0.3 + 0.1 + 0.1] / 5 = 0.88

$v1$ (medium) = 0.85 $\wedge$ 0.88 = 0.85

stage 2~~~~~~

$v2'$ (low) = 0.25 $\wedge$ 0.85 = 0.25

$v2'$ (medium) = 0.50 $\wedge$ 0.85 = 0.50

$v2'$ (high) = 0.75 $\wedge$ 0.85 = 0.75 ***

X2mh = 0.8/no + 0.8/slight + 0.6/moderate + 0.6/severe + 0.5/disastrous

1.0 - D[X2mh,G2] = 1.0 - [0.1 + 0.0 + 0.1 + 0.2 + 0.3] / 5 = 0.86

$v2$ (high) = 0.75 $\wedge$ 0.86 = 0.75

stage3 ~~~~~~

$v3'$ (low) = 1.00 $\wedge$ 0.75 = 0.75 ***

$v3'$ (medium) = 0.70 $\wedge$ 0.75 = 0.70

$v3'$ (high) = 0.40 $\wedge$ 0.75 = 0.40

X3mhl = 0.7/no + 0.6/slight + 0.6/moderate + 0.5/severe + 0.3/disastrous

1.0 - D[X3mhl,G3] = 1.0 - [0.3 + 0.1 + 0.2 + 0.4 + 0.3] / 5 = 0.74

$v3$ (low) = 0.75 $\wedge$ 0.74 = 0.74

~~~~~~

The optimal solution is thus [medium, high, low].

~~~~~~

Example 2: A problem with two optimal solutions.

****

STATE SPACE: the flood damage level
{no, slight, moderate, severe, disastrous}
DECISION SPACE: the investment level for the measures
{low, medium, high}

****

(1) The Membership Function of Initial State:

X0 = 0.1/no + 0.4/slight + 0.7/moderate + 1.0/severe + 0.8/disastrous

(2) The Membership Function of Goal State:

G1 = 0.4/no + 0.6/slight + 0.6/moderate + 0.7/severe + 0.5/disastrous
G2 = 0.9/no + 0.7/slight + 0.5/moderate + 0.3/severe + 0.1/disastrous
G3 = 1.0/no + 0.8/slight + 0.4/moderate + 0.1/severe + 0.0/disastrous

(3) The Membership Function of Constraint for Measures:

C1 = 0.45/low + 0.85/medium + 0.65/high
C2 = 1.00/low + 0.80/medium + 0.60/high
C3 = 0.50/low + 0.70/medium + 0.90/high

(4) The Fuzzy Transform Matrix:

$$\text{T1 (low)} = \begin{vmatrix} 0.3 & 0.8 & 0.5 & 0.3 & 0.1 \\ 0.2 & 0.3 & 0.8 & 0.5 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.8 & 0.5 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.8 \\ 0.0 & 0.0 & 0.1 & 0.2 & 0.3 \end{vmatrix} \quad \text{T1 (medium)} = \begin{vmatrix} 0.6 & 0.9 & 0.4 & 0.1 & 0.0 \\ 0.1 & 0.6 & 0.9 & 0.4 & 0.1 \\ 0.0 & 0.1 & 0.6 & 0.9 & 0.4 \\ 0.0 & 0.0 & 0.1 & 0.6 & 0.9 \\ 0.0 & 0.0 & 0.0 & 0.1 & 0.6 \end{vmatrix} \quad \text{T1 (high)} = \begin{vmatrix} 0.4 & 0.6 & 0.8 & 0.2 & 0.1 \\ 0.3 & 0.4 & 0.6 & 0.8 & 0.2 \\ 0.2 & 0.3 & 0.4 & 0.6 & 0.8 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.6 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.4 \end{vmatrix}$$

$$\text{T2 (low)} = \begin{vmatrix} 0.4 & 0.7 & 0.5 & 0.3 & 0.1 \\ 0.3 & 0.4 & 0.7 & 0.5 & 0.3 \\ 0.2 & 0.3 & 0.4 & 0.7 & 0.5 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.7 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.4 \end{vmatrix} \quad \text{T2 (medium)} = \begin{vmatrix} 0.5 & 0.8 & 0.4 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.8 & 0.4 & 0.2 \\ 0.1 & 0.3 & 0.5 & 0.8 & 0.4 \\ 0.0 & 0.1 & 0.3 & 0.5 & 0.8 \\ 0.0 & 0.0 & 0.1 & 0.3 & 0.5 \end{vmatrix} \quad \text{T2 (high)} = \begin{vmatrix} 0.5 & 0.6 & 0.8 & 0.4 & 0.1 \\ 0.3 & 0.5 & 0.6 & 0.8 & 0.4 \\ 0.2 & 0.3 & 0.5 & 0.6 & 0.8 \\ 0.1 & 0.2 & 0.3 & 0.5 & 0.6 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.5 \end{vmatrix}$$

$$\text{T3 (low)} = \begin{vmatrix} 0.3 & 0.7 & 0.6 & 0.4 & 0.2 \\ 0.2 & 0.3 & 0.7 & 0.6 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.7 & 0.6 \\ 0.0 & 0.1 & 0.2 & 0.3 & 0.7 \\ 0.0 & 0.0 & 0.1 & 0.2 & 0.3 \end{vmatrix} \quad \text{T3 (medium)} = \begin{vmatrix} 0.7 & 0.9 & 0.3 & 0.2 & 0.0 \\ 0.4 & 0.7 & 0.9 & 0.3 & 0.2 \\ 0.1 & 0.4 & 0.7 & 0.9 & 0.3 \\ 0.0 & 0.1 & 0.4 & 0.7 & 0.9 \\ 0.0 & 0.0 & 0.1 & 0.4 & 0.7 \end{vmatrix} \quad \text{T3 (high)} = \begin{vmatrix} 0.3 & 0.6 & 0.9 & 0.2 & 0.1 \\ 0.2 & 0.3 & 0.6 & 0.9 & 0.2 \\ 0.2 & 0.2 & 0.3 & 0.6 & 0.9 \\ 0.1 & 0.2 & 0.2 & 0.3 & 0.6 \\ 0.0 & 0.1 & 0.2 & 0.2 & 0.3 \end{vmatrix}$$

****

SOLUTION:
stage 1~~~~~~
$v1'$ (low) = 0.45
$v1'$ (medium) = 0.85 ***
$v1'$ (high) = 0.65

X1m = 0.4/no + 0.7/slight + 0.9/moderate + 0.8/severe + 0.6/disastrous

1.0 - D[X1m,G1] = 1.0 - [0.0 + 0.1 + 0.3 + 0.1 + 0.1] / 5 = 0.88

$v1$ (medium) = 0.85 $\wedge$ 0.88 = 0.85

stage 2~~~~~~

$v2'$ (low) = 1.00 $\wedge$ 0.85 = 0.85 ***
$v2'$ (medium) = 0.80 $\wedge$ 0.85 = 0.80 **
$v2'$ (high) = 0.60 $\wedge$ 0.85 = 0.60

X2ml = 0.7/no + 0.7/slight + 0.7/moderate + 0.6/severe + 0.4/disastrous

1.0 - D[X2ml,G2] = 1.0 - [0.2 + 0.0 + 0.2 + 0.3 + 0.3] / 5 = 0.80

X2mm = 0.7/no + 0.8/slight + 0.8/moderate + 0.6/severe + 0.5/disastrous

1.0 - D[X2mm,G2] = 1.0 - [0.2 + 0.1 + 0.3 + 0.3 + 0.4] / 5 = 0.74

$v2$ (low) = 0.85 $\wedge$ 0.80 = 0.80 ***
$v2$ (medium) = 0.80 $\wedge$ 0.74 = 0.74 **

stage 3~~~~~~

$v3'$ (low) = 0.50 $\wedge$ 0.80 = 0.50
$v3'$ (medium) = 0.70 $\wedge$ 0.80 = 0.70
$v3'$ (high) = 0.90 $\wedge$ 0.80 = 0.80 ***

X3mlh = 0.7/no + 0.6/slight + 0.6/moderate + 0.4/severe + 0.3/disastrous

1.0 - D[X3mlh,G3] = 1.0 - [0.3 + 0.2 + 0.2 + 0.3 + 0.3] / 5 = 0.74

$v3$ (high) = 0.80 $\wedge$ 0.74 = 0.74
$v3'$ (low) = 0.50 $\wedge$ 0.74 = 0.50
$v3'$ (medium) = 0.70 $\wedge$ 0.74 = 0.70
$v3'$ (high) = 0.90 $\wedge$ 0.74 = 0.74 ***

X3mmh = 0.8/no + 0.6/slight + 0.6/moderate + 0.5/severe + 0.3/disastrous

1.0 - D[X3mmh,G3] = 1.0 - [0.2 + 0.2 + 0.2 + 0.4 + 0.3] / 5 = 0.74
$v3$ (high) = 0.74 $\wedge$ 0.74 = 0.74

~~~~~~

The optimal policies are thus both [medium, low, high,] and [medium, medium, high].

~~~~~~

6. DISCUSSION

Fuzzy decision processes, especially those of the multistage variety, abound in numerous areas of real life. Thus, there is considerable motivation and interest to study them. The classical modeling and solution procedure is fuzzy dynamic programming. However, in certain situations alternate solution approaches, such as the modified branch and bound procedure introduced by Kacprzyk may be instructive. The algorithm, when correctly applied, is not only efficient in terms of computational complexity but generates optimal solutions. The possibility of the existence of alternate optima in the algorithm as illustrated in the preceding examples may be both a curse and a blessing. The advantage lies in the flexibility afforded the decision maker, while the curse may arise in the added selection problem especially when several optima result.

REFERENCES

1. A.O. Esogbue and V. Ramesh, Dynamic programming and fuzzy allocation processes, Technical Memorandum No. 202, Department of Operations Research, Case Western Reserve University, Cleveland, Ohio, (1970).
2. J. Kacprzyk, A branch-and-bound algorithm for the multistage control of a fuzzy system in fuzzy environment, *Kybernetes* **8**, 139–147 (1979).
3. J. Kacprzyk and A. Straszak, Application of fuzzy decision-making models for determining optimal policies in stable regional development, In *Fuzzy Sets* (Edited by P.P. Wang and S.K. Chang), Plenum Press, New York, (1980).
4. W.E. Stein, Optimal stopping in a fuzzy environment, *Fuzzy Sets and Systems* **3**, 193–197 (1980).
5. A.O. Esogbue and R.E. Bellman, Fuzzy dynamic programming and its extensions, *Fuzzy Sets and Decision Analysis, TIMS Studies in Management Science* **20** (1984).
6. J.P. Baldwin and B.W. Pilsworth, Dynamic programming, fuzzy systems with fuzzy environment, *Journal of Mathematical Analysis and Applications* **85**, 1–23 (1982).
7. R.E. Bellman and L.A. Zadeh, Decision making in a fuzzy environment, *Management Science* **17B** (4.), 141–164 (1970).
8. J. Kacprzyk and A.O. Esogbue, Fuzzy dynamic programming and its applications, a survey, To appear.
9. A.O. Esogbue, M. Fedrizzi and J. Kacprzyk, Fuzzy dynamic programming with stochastic systems, In *Combining Fuzzy Imprecision with Stochastic Uncertainty in Decision Making*, Springer-Verlag, Rheinhold, Cologne, (1988).
10. L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. on Syst. Man and Cyber.*, 28–44 (1973).