Mathware & Soft Computing 4 (1997) 219-232

A Genetic Algorithm for the Multistage Control of a Fuzzy System in a Fuzzy Environment

Janusz Kacprzyk
Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01–447 Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

Abstract

We discuss a prescriptive approach to multistage optimal fuzzy control of a fuzzy system, given by a fuzzy state transition equation. Fuzzy constraints and fuzzy goals at consecutive control stages are given, and their confluence, Bellman and Zadeh's fuzzy decision, is an explicit performance function to be optimized. First, we briefly survey previous basic solution methods of dynamic programming (Baldwin and Pilsworth, 1982) and branch-and-bound (Kacprzyk, 1979), which are plagued by low numerical efficiency, and then sketch Kacprzyk's (1993a–e, 1994a) approach based on possibilistic interpolative reasoning aimed at enhancing the numerical efficiency but requiring a solution of a simplified auxiliary problem, and then some "readjustment" of the solution obtained. We propose a genetic algorithm for solving the problem considered. Real coding and specially defined operations of crossover, mutation, etc. are employed. The approach yields good results, and is quite efficient numerically.

Keywords: multistage fuzzy control, fuzzy dynamic system, fuzzy dynamic programming, branch-and-bound, interpolative reasoning, genetic algorithm.

1 Introduction

A huge interest in fuzzy (logic) control that we encounter all over the world, both in the academia and among practitioners, is amplified by successful applications ranging from "big" specialized equipment and technological processes, as, e.g., cranes, cement kilns, subway trains, etc. to "small" everyday products as, e.g., washing machines or cameras. Clearly, the latter ones are more visible to the general public, media, and even authorities responsible for research financing, and are therefore an important factor.

In those applications control rules are used that encode a control policy (control to be applied in a specific situation). These rules, forming the knowledge base, are known from experience by the human operator, are stated in process manuals, etc., and need to be elicited by some knowledge elicitation or acquisition methods. Then, these rules are used to infer control for a specific situation (not necessarily one of those taken into account while constructing the knowledge base). Thus, the rules just describe how to control the process, hence this is a **descriptive approach** to fuzzy control.

However, though a traditional descriptive approach to fuzzy control is practically efficient, it has some inherent limitations. First of all, it lacks an explicit performance function, and – what is clearly related – makes it not possible to simply change the performance function, i.e. to make possible the control of the process in a different way, for instance not aiming at a highest possible fuel efficiency but at a possibly fast action. Such a simple and often encountered change of control strategy would require in a traditional fuzzy control system the existence of a new set of control rules covering such a different situation. Needless to say that since at the extreme the number of such possible different control strategies (equivalent to different performance functions) might be very high (theoretically infinite), then the knowledge base of a fuzzy controller would have to contain a very high number of rules that would often be prohibitive.

Fortunately enough, there also exists an earlier approach to fuzzy control which does not exhibit the drawbacks of the conventional fuzzy (logic) control models mentioned above. This approach is from the late 1960s and early 1970s (cf. Bellman and Zadeh, 1970), and has been further developed by, e.g., Fung and Fu (1977), Kacprzyk (1977–1994a), etc., and is presented in detail in Kacprzyk's (1983a,1997) books.

Its essence is the assumptions that: dynamics of the system under control is known, a performance function is explicitly specified, and (optimal) controls are to be found by an algorithm. So, we do not describe how the system should be controlled but prescribe how to control it. This is therefore a **prescriptive approach**.

Such a prescriptive approach to fuzzy control is considered here. We assume a fuzzy system under control given by a fuzzy state transition equation, and operating under fuzzy constraints on controls and fuzzy goals on states (which, when aggregated, constitute in fact an explicit performance function). We seek optimal fuzzy controls over some planning horizon.

First, to provide a point of departure, show the essence of the problem considered and its inherent difficulties, and give motivation for our work, we sketch the basic traditional solution techniques: dynamic programming (Baldwin and Pilsworth, 1982), branch-and-bound (Kacprzyk, 1979), and Kacprzyk's (1993a–e, 1994a) attempts to enhance efficiency by using interpolative reasoning.

As an alternative solution method, we propose a real coded genetic algorithm, with specially devised genetic operations of crossover and mutation. We advocate it as a viable alternative.

2 Control of a fuzzy system in a fuzzy environment: problem formulation and traditional solution techniques

Dynamics of the fuzzy system under control is given by a fuzzy state transition equation

$$X_{t+1} = \mathrm{F}(X_t, U_t); t = 0, 1, \dots \tag{1}$$

where X_t, X_{t+1} are fuzzy states at control stage t and $t+1$, respectively, defined as fuzzy sets in the state space $X = \{x\} = \{s_1, \dots, s_n\}$, and U_t is a fuzzy control at t, $t = 0, 1, \dots$, defined as a fuzzy set in the control space $U = \{u\} = \{c_1, \dots, c_m\}$.

At each t, the control applied U_t is subjected to a fuzzy constraint $\mu_{C^t}(u_t)$, and on the state attained X_{t+1} a fuzzy goal $\mu_{G^{t+1}}(x_{t+1})$ is imposed, $t = 0, 1, \dots, N-1$; N is a control horizon (termination time) assumed to be fixed and specified, and finite (cf. Kacprzyk, 1983a, 1997 for other type of finite and infinite termination times).

Both the fuzzy controls U_t's and fuzzy states X_{t+1}'s are now fuzzy, hence their grades of membership in the fuzzy constraints and goals cannot be directly determined as the values of $\mu_{C^t}(u_t)$ and $\mu_{G^{t+1}}(x_{t+1})$, and some "trickery" is needed. For instance, one can use the following redefinition of the fuzzy constraint and fuzzy goal

$$\begin{cases} \mu_{\overline{C}^t}(U_t) = 1 - \mathrm{d}(C^t, U_t) \\ \mu_{\overline{G}^{t+1}}(X_{t+1}) = 1 - \mathrm{d}(G^{t+1}, X_{t+1}) \\ t = 0, 1, \dots, N-1 \end{cases} \tag{2}$$

where $\mathrm{d} : [0,1] \times [0,1] \longrightarrow [0,1]$ is some measure of dissemblance (as, e.g., (12)), a normalized (e.g. Hamming's or Euclidean) distance between fuzzy sets (as, e.g., in Kacprzyk, 1979), etc.

Now, employing Bellman and Zadeh's (1970) general framework for decision making under fuzziness, the fuzzy decision is

$$\begin{aligned} &\mu_D(U_0, \dots, U_{N-1} \mid X_0) = \\ &= \quad \mu_{\overline{C}^0}(U_0) \wedge \mu_{\overline{G}^1}(X_1) \wedge \dots \wedge \mu_{\overline{C}^{N-1}}(U_{N-1}) \wedge \mu_{\overline{G}^N}(X_N) \end{aligned} \tag{3}$$

It is easy to see that the fuzzy decision serves the purpose of an explicit performance function which is a prerequisite for the approach considered. Namely, it specifies the degree to which the fuzzy constraints and fuzzy goals are satisfied at all the control stages.

Notice that although the minimum operation, "$\wedge$" is used above, the aggregation of the fuzzy constraints and fuzzy goals may proceed using defferent operations as, e.g., some t-norm, the weighted average (cf. Kacprzyk, 1983a), using a fuzzy linguistic quantifier (cf. Kacprzyk, 1983b or Kacprzyk and Iwański, 1987); for simplicity, but without loss of generality, the most commonly used $\wedge$-based aggregation will be employed in this paper.

We wish to best satisfy the fuzzy constraints and fuzzy goals at all the control stages, so we seek an optimal sequence of fuzzy controls $U_0^*, \ldots, U_{N-1}^*$ such that

$$\begin{aligned} &\mu_D(U_0^*, \ldots, U_{N-1}^* \mid X_0) = \\ &= \max_{U_0, \ldots, U_{N-1}} \mu_D(U_0, \ldots, U_{N-1} \mid X_0) = \\ &= \max_{U_0, \ldots, U_{N-1}} [\mu_{\overline{C}^0}(U_0) \wedge \mu_{\overline{G}^1}(X_1) \wedge \ldots \wedge \mu_{\overline{C}^{N-1}}(U_{N-1}) \wedge \mu_{\overline{G}^N}(X_N)] \end{aligned} \tag{4}$$

Hence, the fuzzy control problem is now of an optimization type. The main solution techniques are:

- dynamic programming,
- dynamic programming with interpolative reasoning, and
- branch-and-bound,

and these approaches will now be briefly discussed.

2.1 Solution by dynamic programming

This approach, which may be viewed as a reformulation of the classic Bellman and Zadeh's (1970) dynamic programming solution for the deterministic and stochastic system under control, now to be able to handle the fuzzy system under control, was proposed by Baldwin and Pilsworth (1982).

Basically, after some "trickery" we arrive at the following set of dynamic programming recurrence equations

$$\begin{cases} \mu_{\overline{G}^N}(X_N) = \max_{x_N}(\mu_{X_N}(x_N) \wedge \mu_{G^N}(x_N)) \\ \mu_{\overline{G}^{N-i}}(X_{N-i}) = \max_{U_{N-i}}(\max_{u_{N-i}}(\mu_{U_{N-i}}(u_{N-i}) \wedge \\ \qquad \wedge \mu_{C^{N-i}}(u_{N-i})) \wedge \mu_{\overline{G}^{N-i+1}}(x_{N-i+1})) \\ \mu_{X_{N-i+1}}(x_{N-i+1}) = \max_{x_{N-i}}(\max_{u_{N-i}}(\mu_{U_{N-i}}(u_{N-i}) \wedge \\ \qquad \wedge \mu_{X_{N-i+1}}(x_{N-i+1} \mid x_{N-i}, u_{N-i})) \wedge \mu_{X_{N-i}}(x_{N-i})) \\ i = 1, \ldots N \end{cases} \tag{5}$$

In principle, this set of equations is solvable. However, first, the $\mu_{\overline{G}^{N-i}}(X_{N-i})$ is to be specified for each possible fuzzy state X_{N-i}, and second, the maximization is to proceed over all possible fuzzy controls U_{N-i}. The numbers of possible U_{N-i} 's and X_{N-i} 's may be very high, hence the solution of (5) is practically very often impossible.

The so-called *fuzzy interpolation* (Baldwin and Pilsworth, 1982; Kacprzyk and Staniewski, 1982) is therefore used. Basically, a relatively small number, r, of *reference fuzzy states* $\overline{S}^1, \ldots, \overline{S}^r$, and a relatively small number of *reference fuzzy controls* $\overline{C}^1, \ldots, \overline{C}^p$ are introduced. Then, an optimal control policy, $\overline{U}_t^* = \overline{a}_t^*(\overline{X}_t)$, $t = 0, 1, \ldots, N-1$, is specified for the reference fuzzy states and controls only. Such an optimal policy is represented as a fuzzy relation $\mu_{\overline{R}_t^*}(x_t, u_t)$, and for a current fuzzy state X_t, not necessarily reference, an optimal fuzzy control U_t^*, not necessarily reference, is given by the compositional rule of inference $U_t^* = x_t \circ \overline{R}_t^*$ where "$\circ$" is, e.g., the max-min composition of the respective fuzzy relations.

However, by the very nature of the compositional rule of inference, there should be relatively many "overlapping" reference fuzzy states and controls to yield significant results. This may clearly imply too high a problem dimensionality that may lead to numerical problems, and hence Kacprzyk's (1993a–e, 1994a) interpolative reasoning approach was proposed, and it will be presented later.

In general, the fuzzy dynamic programming approach presented above is not practically tractable, and the two approaches to be presented below were devised to alleviate this difficulty.

More details on fuzzy dynamic programming can be found, e.g. in Kacprzyk's (1983a,1997) books or Kacprzyk (1994b).

2.2 Solution by dynamic programming with interpolative reasoning

Since for the compositional rule of inference to work properly in the above dynamic programming approach there should be "overlapping" reference fuzzy states and fuzzy controls, whose number may be too high for many non-trivial problems, in recent Kacprzyk's (1993a–e, 1994) papers another approach was proposed in which (very) small numbers of (non-overlapping) fuzzy reference fuzzy controls and states are assumed, an (auxiliary) control problem is formulated in their terms, and then quickly solved by dynamic programming (in fact, also by another method as, say, branch-and-bound to be discussed later). An "auxiliary" optimal solution is obtained which is then readjusted to obtain an optimal solution of a "real" control problem, i.e. formulated in terms of the source fuzzy states and fuzzy controls.

Suppose that we obtain in such a case an optimal (auxiliary) control policy a_t^* stating that

$$\begin{cases} \text{IF } \overline{X}_t = \overline{S}_1 \text{ THEN } \overline{U}_t^* = \overline{C}_{t1} \\ \text{ELSE} \ldots \text{ELSE} \\ \text{IF } \overline{X}_t = \overline{S}_i \text{ THEN } \overline{U}_t^* = \overline{C}_{ti} \\ \text{ELSE} \\ \text{IF } \overline{X}_t = \overline{S}_{i+1} \text{ THEN } \overline{U}_t^* = \overline{C}_{t(i+1)} \\ \text{ELSE} \ldots \text{ELSE} \\ \text{IF } \overline{X}_t = \overline{S}_r \text{ THEN } \overline{U}_t^* = \overline{C}_{tr} \end{cases} \tag{6}$$

To implement such a control policy, we need to find a fuzzy control U_t^* (not necessarily reference) for a current X_t (not necessarily reference). If X_t is a fuzzy number between $\overline{S}_i$ and $\overline{S}_{i+1}$, we seek an U_t^* corresponding to X_t via $\overline{a}_t^*$; since X_t is not reference, U_t^* need not be reference either.

Kacprzyk's (1993a–e, 1994) idea is now as follows. First, we assume – for simplicity, but which is fully sufficient in our context – that the fuzzy controls and fuzzy states are triangular fuzzy numbers. We seek therefore their mean values and widths (left and right spreads).

First we determine the mean value of U_t^* using the following simple and intuitively appealing relation

$$\frac{\mathrm{d}(\overline{S}_i, X_t)}{\mathrm{d}(X_t, \overline{S}_{i+1})} = \frac{\mathrm{d}(\overline{C}_i, U_t^*)}{\mathrm{d}(U_t^*, \overline{C}_{t(i+1)})} \tag{7}$$

where d(.,.) is a distance, dissemblance, etc. of two fuzzy sets [cf. (2)] . That is, the relative position of fuzzy control U_t^* with respect to its closest reference counterparts, i.e. $\overline{C}_i$ and $\overline{C}_{t(i+1)}$ should be the same as that of X_t and its closest reference counterparts, i.e. $\overline{S}_i$ and $\overline{S}_{i+1}$ (this approach is close to Kóczy and Hirota, 1992).

The above interpolation type mechanism is clearly the simplest one, and in Kacprzyk (1994a) some enhancement was proposed. Namely, notice that we employ above information on the two neighboring reference fuzzy states and controls only, i.e. a limited one. As this may be insufficient, we may use the three neighboring fuzzy states (except for the lower and upper most) to better capture the dynamics of change of the respective $\overline{U}_t^*$'s. The idea is: first, determine the mean value of U_t^* as before. Second, find the value of a composite measure of change (e.g., rate of increase, decrease, ...) of $\overline{C}_{i-1}$, $\overline{C}_i$, $\overline{C}_{i+1}$ that correspond via the auxiliary optimal policy to $\overline{S}_{i-1}$, $\overline{S}_i$ and $\overline{S}_{i+1}$, respectively. Third, correct the mean value of U_t^* using this measure: if it indicates an increase, move toward "higher values", otherwise – toward "lower" values. This may be generalized for more than three values too. Notice that the general idea outlined above has much to do with the so-called gradual rules.

The second aspect is the width of U_t^*. The reasoning is that the lower the number of reference fuzzy states and controls, i.e., the more sparse the data set, the less precise is the available information. Hence, the fuzzier (of a larger width) U_t^* should be. For instance, we can use a formula

$$\overline{w}(U_t^*) = \frac{1}{5}[\overline{w}(\overline{S}_i) + \overline{w}(\overline{S}_{i+1}) + \overline{w}(\overline{X}_t) + \overline{w}(\overline{C}_{ti}) + \overline{w}(\overline{C}_{t(i+1)})] \tag{8}$$

where $\overline{w}(.)$ is a relative width (related to the universe of discourse of the fuzzy states and controls), and the simplest arithmetics mean (8) can be replaced by another formula expressing the above rationale as, e.g., a weighted average.

Moreover, it may often be expedient to include in (8) some term(s) accounting for the fuzziness of the control problem as, e.g., of the fuzzy constraints, fuzzy goals, fuzzy system under control, etc. As a first, somehow *ad hoc* attempt, we can use a degree of specificity of a fuzzy set and calculate the mean fuzziness (non-specificity) of fuzzy constraints, fuzzy goals, etc. For the degree of fuzziness (non-specificity) of the fuzzy system under control (given as a set of IF – THEN rules), we can use Kacprzyk's (1994) approach.

Thus, the relative width of U_t^* defined initially by (8), which involves the fuzziness (non-specificity) of the fuzzy states and fuzzy controls only, may be further modified to involve the fuzziness of fuzzy constraints, fuzzy goals and fuzzy system under control yielding, e.g.,

$$\overline{w}(U_t^*) = \overline{w}(U_t^*) \cdot \text{corr}[\sum_{i=1}^{N}(1 - \text{spec}(C^{i-1})), \sum_{i=1}^{N}(1 - \text{spec}(G^i)), 1 - \text{spec}(S)] \quad (9)$$

where $\overline{w}(U_t)$ is the relative width of the fuzzy control given by (8), spec(C^{i-1}) is a degree of specificity (e.g., in Yager's, 1988 sense – cf. Kacprzyk, 1994) of a fuzzy constraint C^{i-1}, spec(G^i) is the degree of specificity of a fuzzy goal G^i, spec(S) is the degree of specificity of a fuzzy system under control (given as a set of IF – THEN rules – cf. Kacprzyk, 1994), and

$$\text{corr} : [0,1] \times [0,1] \times [0,1] \longrightarrow (1, \infty) \quad (10)$$

is a correcting non-decreasing function which increases the width of an optimal fuzzy control to be determined in relation to the fuzziness (non-specificity) of the fuzzy constraints, fuzzy goals and fuzzy system under control.

A detailed description of this approach is beyond the scope of this paper, and will appear elsewhere. In general, it works very well, and helps attain more realistic results.

2.3 Solution by branch-and-bound

The earlier Kacprzyk's (1979) branch-and-bound approach is conceptually simpler and more efficient. It needs (for a decision tree it works on) a finite number of controls, i.e. nonfuzzy controls (whose set is here finite by definition) or some predefined reference fuzzy controls.

The branch-and-bound procedure starts from the initial state x_0. We apply control u_0 and proceed to x_1. Next, we apply u_1 and proceed to x_2, etc. Finally, in x_{N-1}, we apply u_{N-1} and attain x_N. This may be presented as a decision tree whose nodes are the states attained, and with whose edges the controls applied are associated.

First, denote $v_k = \mu_{C^0}(u_0) \wedge \ldots \wedge \mu_{C^k}(u_k)$. Then, due to "$\wedge$": for any $u_0, \ldots, u_k$, $0 < k < N-1$, for each $k < w \leq N-1$ there holds

$$v_k \geq v_w = v(k) \wedge \mu_{C^{k+1}}(u_{k+1}) \wedge \ldots \wedge \mu_{C^w}(u_w) \geq v_N = \mu_D(u_0, \ldots, u_{N-1} \mid X_0) \quad (11)$$

So, if we are at $t = k$, and have traversed some path from X_0 to X_k, the most rational continuation of tree traversal is to apply controls to the node which corresponds to the greatest value of v_i, $i - 1, \ldots, k$.

Notice that in case of (reference) fuzzy states and fuzzy controls this scheme works as well, with obvious replacements of u_t by $\overline{U}_t$, x_{t+1} by X_{t+1}, C^t by $\overline{C}^t$, G^{t+1} by $\overline{G}^t$, and using the fuzzy decision (3).

The above facts make it possible to devise a branch-and-bound algorithm with the branching through the controls applied at the consecutive t's, and the bounding via the values of the v_k's, $k = 0, \ldots, N$.

And again, in case of the reference fuzzy controls, optimal control policies define optimal fuzzy controls as functions of the reference fuzzy states. So, to find optimal fuzzy controls for non-reference fuzzy states, one needs to go as before through the representation of an optimal policy via a fuzzy relation, and then using the compositional rule of inference. Then, for larger, non-trivial problems, if one encounters numerical efficiency problems, one may resort to an interpolative reasoning algorithm analogous to that discuss in Section 2.2 in the context of dynamic programming.

The above three approaches are satisfactory for many problems, even practical ones (in particular the one based on interpolative reasoning). However, for excessively large problems they may be not efficient enough. One may therefore try to use some other tools as, e.g., a genetic algorithm which is proposed in this paper.

3 Solution by a genetic algorithm

For solving the problem considered (4), we will use now a genetic algorithm (cf. Michalewicz, 1995) which for our purposes may be sketched as:

```
begin
    t=: 0
    set the initial population P(t)
    evaluate strings in P(t)
    while termination condition not fulfilled do
    begin
        t := t + 1
        select current population P(t) from P(t − 1)
        perform recombination on elements of P(t)
        calculate the evaluation function for each element of P(t)
    end
end
```

The basic elements of the above general algorithm description are to be generally meant as:

- the problem is represented by strings of fuzzy controls $U_0, \ldots, U_{N-1}$ (assumed for simplicity to be fuzzy numbers defined in $[0,1]$) so we use real coding; in fact, we further simplify the analysis, and assume triangular fuzzy numbers as the representation of fuzzy controls; moreover, some reference fuzzy controls, $\overline{U}_0, \ldots, \overline{U}_{N-1}$, are also used, and they are also assumed to be triangular fuzzy numbers in $[0,1]$;

- the evaluation (objective) function is (3), i.e. the fuzzy decision given as

$$\begin{aligned} &\mu_D(U_0, \ldots, U_{N-1} \mid X_0) = \\ &= \mu_{\overline{C}^0}(U_0) \wedge \mu_{G^1}(X_1) \wedge \ldots \wedge \mu_{\overline{C}^{N-1}}(U_{N-1}) \wedge \mu_{\overline{G}^N}(X_N) \end{aligned}$$

  and for the calculation of $\mu_{\overline{C}^t}(U_t)$ and $\mu_{\overline{G}^{t+1}}(X_{t+1})$ [notice that the controls and states are fuzzy, and cannot be directly accounted for in the fuzzy constraint and fuzzy goal – cf. (2)] we use the dissemblance index (Kaufmann and Gupta, 1985) defined, for triangular fuzzy numbers assumed, as: if A and B are triangular fuzzy numbers, then the *degree of dissemblance* of A and B is

$$\text{diss}(A, B) = \int_{\alpha=0}^{1} \frac{1}{2}(\mid \underline{a}^{\alpha} - \underline{b}^{\alpha} \mid + \mid \overline{a}^{\alpha} - \overline{b}^{\alpha} \mid) \, d\alpha \tag{12}$$

where $[\underline{a}^{\alpha}, \overline{a}^{\alpha}]$ and $[\underline{b}^{\alpha}, \overline{b}^{\alpha}]$ are the α-cuts (intervals) of A and B, $\forall \alpha \in (0, 1]$. Therefore, if we denote

$$f_t(U_t, C^t, X_{t+1}, G^{t+1}) = [1 - \text{diss}(U_t, C^t)] \wedge [1 - \text{diss}(X_{t+1}, G^{t+1})] \tag{13}$$

for $t = 0, 1, \ldots N - 1$, the evaluation function (fuzzy decision) becomes

$$\begin{aligned} & f(U_0, X_1, \ldots U_{N-1}, X_N) = \\ & = \mu_D(U_0, \ldots, U_{N-1} \mid X_0) = \\ & = f_0(U_0, C^0, X_1, G^1) \wedge \ldots \wedge f_{N-1}(U_{N-1}, C^{N-1}, X_N, G^N) \end{aligned} \tag{14}$$

- standard random selections of elements from the consecutive populations, standard concepts of crossover and mutation (evidently, applied to real coded strings), and a standard termination condition, mainly a predefined number of iterations, or iteration-to-iteration improvement lower than a threshhold) are used;

We assume further that:

- fuzzy controls are fuzzy sets in $[0, 1]$ defined as triangular fuzzy numbers in $[0, 1]$, i.e. as the triples (a, b, c), $0 \leq a \leq b \leq c \leq 1$; the left and right spreads (widths) are assumed equal to 5% each, for simplicity, hence only the mean value (b) is practically generated; moreover, 10 ("equally spaced" in [0,1]) reference fuzzy controls are introduced;
- fuzzy states are defined as fuzzy sets in the state space $X = \{s_1, \ldots, s_{10}\}$,
- fuzzy constraints and fuzzy goals are defined as trapezoid fuzzy numbers in [0,1];
- dynamics of the fuzzy system under control (1), i.e. the state transition equation, is defined as a set of fuzzy relations R_S in $X \times X$, for each of the reference fuzzy control (notice that here we do need reference fuzzy controls as otherwise we would need to specify infinitely many fuzzy relations, for each possible fuzzy control); so, to choose an appropriate table (relation) to determine the state transition, first we find a reference fuzzy control that is the closest (in the sense of, say, the dissemblance index used) to the current control, and then we take its corresponding fuzzy relation, and employ the compositional rule of inference to find the resulting fuzzy state X_{t+1}.

The genetic algorithm works therefore as follows:

```
begin
    t=: 0
    set the initial population P(t) which constitutes of
        randomly generated strings of triangular fuzzy controls
        (i.e. of randomly generated mean values,
        with 5% left and right spreads);
    for each U_0, ..., U_{N-1} in each string in the population P(t),
        find the resulting X_{t+1} (by finding first the closest
        reference fuzzy control to choose an appropriate relation,
        and then employing the compositional rule of inference),
        and use the evaluation function (14) to evaluate each string in P(t)
    while t < the maximum number of iterations
        (or, alternatively, a predefined step-to-step improvement) do
    begin
        t := t + 1
        assign the probabilities to each string in P(t - 1),
            which are propotional to the value of the evaluation function
            for each string, and randomly (using those probabilities)
            generate the new population P(t)
        perform crossover and mutation on the strings in P(t)
        calculate the evaluation function (14) for each string in P(t)
    end
end
```

The algorithm was tested on the following example.

Example. Suppose that the number of control stages is $N = 10$, the state space is $X = \{s_1, \ldots, s_{10}\}$, the controls are triangular fuzzy numbers in $[0, 1]$, and there are 10 "equally-spaced" (with the mean values at 0.1, ..., 0.9, 1) reference fuzzy controls defined as the trapezoid fuzzy numbers in $[0, 1]$ as follows:

$$
\begin{array}{ll}
\overline{C}_1 = (0.0, 0.1, 0.1, 0.2) & \overline{C}_2 = (0.1, 0.2, 0.2, 0.3) \\
\overline{C}_3 = (0.2, 0.3, 0.3, 0.4) & \overline{C}_4 = (0.3, 0.4, 0.4, 0.5) \\
\overline{C}_5 = (0.4, 0.5, 0.5, 0.6) & \overline{C}_6 = (0.5, 0.6, 0.6, 0.7) \\
\overline{C}_7 = (0.6, 0.7, 0.7, 0.8) & \overline{C}_8 = (0.7, 0.8, 0.8, 0.9) \\
\overline{C}_9 = (0.8, 0.9, 0.9, 1.0) & \overline{C}_{10} = (0.9, 1.0, 1.0, 1.0)
\end{array}
$$

The initial fuzzy state is

$$X_0 = 1.0/s_1 + 0.7/s_2 + 0.4/s_3 + 0.1/s_4$$

The fuzzy constraints at the particular control stages are also given as the following trapezoid fuzzy numbers:

$$
\begin{array}{ll}
\overline{C}^0 = (0.0, 0.0, 0.5, 0.8) & \overline{C}^1 = (0.0, 0.0, 0.5, 0.8) \\
\overline{C}^2 = (0.0, 0.0, 0.5, 0.8) & \overline{C}^3 = (0.0, 0.0, 0.5, 0.8) \\
\overline{C}^4 = (0.0, 0.0, 0.5, 0.8) & \overline{C}^5 = (0.0, 0.0, 0.5, 0.8) \\
\overline{C}^6 = (0.0, 0.0, 0.5, 0.8) & \overline{C}^7 = (0.0, 0.0, 0.5, 0.8) \\
\overline{C}^8 = (0.0, 0.0, 0.5, 0.8) & \overline{C}^9 = (0.0, 0.0, 0.5, 0.8)
\end{array}
$$

The fuzzy goals at the particular control stages are:

$$\begin{aligned}
\overline{G}^1 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^2 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^3 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^4 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + 1.0/s_5 + 0.6/s_6 + \\
&\quad + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^5 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^6 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^7 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^8 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^9 &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10} \\
\overline{G}^{10} &= 0.1/s_1 + 0.2/s_2 + 0.3/s_3 + 0.6/s_4 + \\
&\quad + 1.0/s_5 + 0.6/s_6 + 0.3/s_7 + 0.2/s_8 + 0.1/s_9 + 0.0/s_{10}
\end{aligned}$$

The fuzzy state transitions (1) are specified as conditioned fuzzy sets for each particular reference fuzzy control, $\overline{C}_1, \ldots, \overline{C}_{10}$. Due to lack of space we will only present below the state transtion equations for the first and last reference fuzzy control, i.e. $\overline{C}_1$ and $\overline{C}_{10}$, and these are:

- for $\overline{C}_1$

	$x_{t+1} = s_1$	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
$x_t = s_1$	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_2	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_3	0.0	0.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_4	0.0	0.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_5	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_6	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_7	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_8	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_9	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1
s_{10}	0.0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.1

- ...

- for $\overline{C}_{10}$

	$x_{t+1} = s_1$	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
$x_t = s_1$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
s_{10}	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Suppose that the main parameters are:

- the population size is 50,
- the maximum number of iterations (termination condition) is 1000,
- the crossover rate is 0.6, and
- the mutation rate is 0.001.

The ten best results obtained may be summarized as follows:

- the optimal fuzzy controls (triangular fuzzy numbers $U_t = (a, b, c)$) at the particular control stages $t = 0, 1, \ldots, 10$, i.e. the best result obtained, are:

$$\begin{array}{ll} U_0^* = (0.4885, 0.5142, 0.5399) & U_1^* = (0.5031, 0.5296, 0.5561) \\ U_2^* = (0.4236, 0.4459, 0.4682) & U_3^* = (0.4842, 0.5097, 0.5352) \\ U_4^* = (0.4651, 0.4895, 0.5140) & U_5^* = (0.4916, 0.5175, 0.5434) \\ U_6^* = (0.3218, 0.3387, 0.3556) & U_7^* = (0.5225, 0.5500, 0.5775) \\ U_8^* = (0.3451, 0.3633, 0.3815) & U_9^* = (0.2615, 0.2752, 0.2890) \end{array}$$

  and the value of the fuzzy decision (14) is

$$\mu_D(U_0^*, \ldots, U_9^* \mid X_0) = 0.681881$$

- the second best result is

$$\begin{array}{ll} U_0 = (0.4885, 0.5142, 0.5399) & U_1 = (0.5031, 0.5296, 0.5561) \\ U_2 = (0.4236, 0.4459, 0.4682) & U_3 = (0.4842, 0.5097, 0.5352) \\ U_4 = (0.4651, 0.4895, 0.5140) & U_5 = (0.4916, 0.5175, 0.5434) \\ U_6 = (0.3218, 0.3387, 0.3556) & U_7 = (0.5225, 0.5500, 0.5775) \\ U_8 = (0.3451, 0.3633, 0.3815) & U_9 = (0.2615, 0.2752, 0.2890) \end{array}$$

  and the value of the fuzzy decision (14) is

$$\mu_D(U_0^*, \ldots, U_9^* \mid X_0) = 0.681881$$

- while the tenth best result is:

$$
\begin{array}{ll}
U_0^* = (0.2510, 0.2642, 0.2774) & U_1^* = (0.4758, 0.5008, 0.5259) \\
U_2^* = (0.4855, 0.5111, 0.5366) & U_3^* = (0.5432, 0.5718, 0.6004) \\
U_4^* = (0.4780, 0.5032, 0.5284) & U_5^* = (0.5182, 0.5455, 0.5728) \\
U_6^* = (0.5100, 0.5368, 0.5637) & U_7^* = (0.3316, 0.3491, 0.3665) \\
U_8^* = (0.4639, 0.4883, 0.5127) & U_9^* = (0.3816, 0.4016, 0.4217)
\end{array}
$$

and the value of the fuzzy decision (14) is

$$\mu_D(U_0^*, \ldots, U_9^* \mid X_0) = 0.679795$$

As to the computational efficiency, the best values of the fuzzy decision (14) have been obtained pretty early, well before the 1,000 iterations assumed. In general, also for many different problems solved, the algorithm has proven to be efficient (cf. Kacprzyk, 1997).

4 Concluding remarks

We proposed the use of a genetic algorithm for solving multistage (optimal) control of a fuzzy system in a fuzzy environment (under fuzzy constraints and goals). Real coding and standard crossover and mutation operations were used. The method is simple and the results are promising. It seems that the use of a genetic algorithm can provide a viable alternative for solving this class of prescriptive fuzzy control problems that have been plagued by inherent numerical inefficiency. Moreover, we think that a more sophisticated evolutionary type strategy might be even more adequate for the problem considered.

References

Baldwin J.F. and B.W. Pilsworth (1982) Dynamic programming for fuzzy systems with fuzzy environment. *J. of Mathematical Analysis and Applications* **85** 1–23.

Bellman R.E. and L.A. Zadeh (1970) Decision-making in a fuzzy environment *Management Science* **17**, 141–164.

Fung L.W. and K.S. Fu (1977) Characterization of a class of fuzzy optimal control problems. In M.M. Gupta, G.N. Saridis and B.R. Gaines (Eds.): *Fuzzy Automata and Decision Processes*, New York: North-Holland, pp. 209–219.

Kacprzyk J. (1977) Control of a non-fuzzy system in a fuzzy environment with fuzzy termination time. *Systems Science* **3**, 320–331.

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

Kacprzyk J. (1979) A branch-and-bound algorithm for the multistage control of a fuzzy system in a fuzzy environment. *Kybernetes* **8**, 139–147.

Kacprzyk J. (1983a) *Multistage Decision-Making under Fuzziness.* Cologne: Verlag TÜV Rheinland.

Kacprzyk J. (1983b) A generalization of fuzzy multistage decision making and control via linguistic quantifiers, *Int. Journal of Control* **38**, 1249–1270.

Kacprzyk J. (1992) Fuzzy optimal control revisited: toward a new generation of fuzzy control? *Proceedings of Second International Conference on Fuzzy Logic and Neural Networks – Iizuka'92* (Iizuka, Japan), Vol. 2, pp. 429–432.

Kacprzyk J. (1993a) Interpolative reasoning in optimal fuzzy control. *Proceedings of Second IEEE International Conference on Fuzzy Systems*, (San Francisco, USA), Vol.2, pp. 1259–1263.

Kacprzyk J. (1993b) Interpolative reasoning for computationally efficient optimal fuzzy control. *Proceedings of Fifth IFSA World Congress* (Seoul, Korea), Vol. I, 624–626..

Kacprzyk J. (1993c) Fuzzy control with an explicit performance function using dynamic programming and interpolative reasoning, *Proceedings of EUFIT'93* (Aachen, Germany), Vol.3, pp. 1459–1463.

Kacprzyk J. (1993e) In search for a new generation of fuzzy control: can a prescriptive approach based on interpolative reasoning and neural networks help?, *Proceedings of ANZIIS '93* (Perth, Australia), pp. 402–406.

Kacprzyk J. (1994a) Fuzzy dynamic programming – battling against the curse of dimensionality via interpolative reasoning, *Procedings of Third International Conference on Fuzzy Logic, Neural Nets and Soft Computing - IIZUKA'94* (Iizuka, Japan), pp. 245–246.

Kacprzyk J. (1994b) Fuzzy dynamic programming – basic issues, In: M. Delgado, J. Kacprzyk, J.-L. Verdegay and M.A. Vila (Eds.): *Fuzzy Optimization: Recent Advances*, Physica-Verlag, Heidelberg (A Springer-Verlag Company), pp. 321–331.

Kacprzyk J. (1994c) On measuring the specificity of IF–THEN rules, *Int. Journal of Approximate Reasoning* **11**, 29–53.

Kacprzyk J. (1997) *Multistage Fuzzy Control.* Wiley, Chichester.

Kacprzyk J. and C. Iwański (1987) A generalization of discounted multistage decision making and control through fuzzy linguistic quantifiers: an attempt to introduce commonsense knowledge, *Int. Journal of Control* **45**, 1909–1930.

Kacprzyk J. and P. Staniewski (1983) Control of a deterministic system in a fuzzy environment over infinite planning horizon. *Fuzzy Sets and Systems* **10**, 291–298.

Kaufmann A. and M.M. Gupta (1985) *Introduction to Fuzzy Arithmetic: Theory and Applications.* Van Nostrand – Reinhold, New York.

Kóczy L. and K. Hirota (1992) Analogical fuzzy reasoning and gradual inference rules. *Proceedings of Second Int. Conference on Fuzzy Logic and Neural Networks – Iizuka '92* (Iizuka, Japan), Vol. 1, pp. 329-332.

Michalewicz Z. (1995) *Genetic Algorithms + Data Structures = Evolution Programs*, Third Edition. Springer-Verlag, Berlin.

Yager R.R. (1983) Entropy and specificity in a mathematical theory of evidence. *Int. Journal of General Systems* **9**, 249–260