**Proceedings of the 11th**
**KACC, October 1996**

# Automatic GA Fuzzy Modeling with Fine Tuning Method

[0]You Seok Son[a], Wook Chang[a], Jin Bae Park[a], and Young Hoon Joo[aa]

[a] Dept. of Electrical Eng., Yonsei Univ., Seodaemoon Ku, Seoul 120-749, KOREA
Tel: +82-02-361-2773; Fax: +82-02-392-4230; E-mail: ysson@control.yonsei.ac.kr

[aa] Dept.of Control & Instrumentation Eng. Kunsan Univ.
Tel: +82-0654-60-1626; Fax: +82-0654-466-2086; E-mail: yhjoo@knusunl.kunsan.ac.kr

**Abstract** This paper presents a systematic approach to identify a linguistic fuzzy model for a multi-input and single-output complex system. Such a model is composed of fuzzy rules, and its output is inferred by the simplified reasoning. The structure and membership function parameters for a fuzzy model are automatically and simultaneously identified by GA (Genetic Algorithm). After GA search, optimal parameters for the fuzzy model are finely tuned by a gradient method. A numerical example is provided to evaluate the feasibility of the proposed approach. Comparison shows that the suggested approach can produce the linguistic fuzzy model with higher accuracy and a smaller number of rules than the ones achieved previously in other methods.

## 1. INTRODUCTION

In recent years, fuzzy modeling. as a compliment to the conventional modeling techniques, has been studied to deal with complex, ill-defined and uncertain systems. The studies on the fuzzy system modeling have largely been devoted to two approaches. One is based on composite relational equations [1]. The approach is theoretically clear, but may suffer difficulties since the solution of a fuzzy relational equation is usually not unique, and sometimes it even does not exist at all. The other is termed linguistic model [2-4], in which a fuzzy model is composed of a set of fuzzy implications, and they are identified by optimization techniques from sample data. The model has been popular in industrial applications.

As a new means to determine linguistic fuzzy models, Genetic Algorithm(GA) has been frequently used. Karr [5] adapted successfully GA for fuzzy controllers to define fuzzy membership functions. However the determination of rule sets is also important on designing linguistic fuzzy models, so Joo [6] determined the structure of rules using fuzzy c-means clustering(FCM) and identified simultaneously the parameters in the premise and the consequence of the fuzzy model. Recently, GA design methods for fuzzy models/controllers have been developed to generate the whole parts of fuzzy models simultaneously [7-8].

In this paper we develop a coding format to determine a fuzzy model by a chromosome in GA and present a systematic approach in the identification procedure of a fuzzy system through the proposed coding format. The proposed GA modeling method determine both rule sets and membership functions simultaneously and automatically. After GA searching in order to determine an optimal fuzzy model for a complex system, a gradient descent method is used to tune the fuzzy membership function parameters in premise part and the real number in consequent part for better performance.

We provide a numerical example to evaluate the advantages and the effectiveness of the proposed approach.

## 2. AUTOMATIC GA FUZZY MODELLING

### 2.1 fuzzy model and reasoning

We consider the following format of the fuzzy model for a multi-input and single output system.

$$\text{Rule } i: \text{ if } x_1 \text{ is } A_{i1} \ldots x_n \text{ is } A_{in}, \text{ then } y \text{ is } w_i \quad (1)$$

where Rule $i$ is $i$th rule $(1 \le i \le c)$, $x_j (1 \le j \le n)$ is input variable and y is output. $A_{ij}$ is the fuzzy membership function in premise part defined by (2) and $w_i$ is a real number of consequence part.

$$A_{ij}(x_j) = \begin{cases} \text{if } (a_{ij} - b_{ij}/2) \le x_j \le (a_{ij} + b_{ij}/2) \\ \text{then } \quad 1 - 2|x_j - a_{ij}|/b_{ij} \\ \\ \text{if } x_j < (a_{ij} - b_{ij}/2) \text{ or } x_j > (a_{ij} + b_{ij}/2) \\ \text{then } \quad 0 \end{cases} \quad (2)$$

where $a_{ij}$ and $b_{ij}$ are the center point and the width of an isosceles triangle.

As for reasoning, we consider the following simplified procedure.

1) Given the $l$th input/output data $\{x_{l1}, x_{l2}, \ldots, x_{ln}, y_l\}$, calculate the degree of the fulfillment $\mu_i$ in the premise for the $i$th rule as :

$$\mu_i = A_{i1}(x_{l1})\times A_{i2}(x_{l2})\times\cdots\times A_{in}(x_{ln}) \qquad (3)$$

2) Calculate the inferred value $y_l^*$ by taking the weighted average of $w_i$ with respect to $\mu_i$ as :

$$y_l^* = \frac{\sum_{i=1}^{c}\mu_i\, w_i}{\sum_{i=1}^{c}\mu_i} \qquad (4)$$

where c is the number of fuzzy rules.

### 2.2 Coding the Genetic Algorithm

In a GA, parameters for a give problem are encoded into a string, analogous to a chromosome in nature. Each string, therefore, contains a possible solution to the problem. To determine how well a chromosome solves the problem, it is first broken down into the individual substrings which represent each variable and these values are then used to evaluate the cost function, yielding a "fitness".

Two types of coding used in this research are "real number coding" and "integer coding". Parameters and the structure of a fuzzy model are encoded to substrings of a chromosome by one of the two. A chromosome is composed of two substrings(candidate strings and decision strings) and these substrings are divided into two parts(premise part and consequent part). It can be represented as Fig. 1.

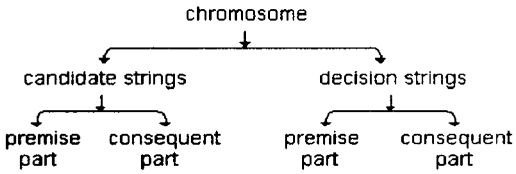


**Fig. 1. Substrings in a chromosome**

Candidate strings are encoded by real numbers, that is, each code of strings is a real number. Parameters for a membership function in premise part, $a_{ij}$ and $b_{ij}$, are determined by a certain code of the candidate string for premise part. Fuzzy singleton, $w_i$, is determined by a certain code of candidate string for consequent part. Fig. 2. illustrates the coding format for candidate strings in a chromosome, where $n$ represents the number of input variables, $r$ the number of candidates for premise parameters and $s$ the number of candidates for consequent real numbers.

Decision strings are encoded by integers and determine the structure of rules and the number of rules by choosing one of parameters in candidate strings.

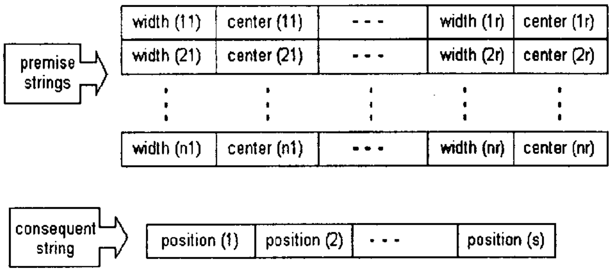


**Fig. 2. Candidate strings in a chromosome**

The premise structure of a rule is determined by decision strings for premise part, which are composed of $n$ codes from $0$ to $r$ per each rule. This integer code chooses one parameter from $r$ parameters in the candidate string. $0$ code means that the related input is not included in the rule. If all codes are $0$'s in the decision string for a rule , the rule will be deleted from the whole rule sets. The determination of the consequent structure directly means the determination of the number of rules. The decision string for consequent part is composed of $c$(the maximum rule number) codes from $0$ to $s$, which chooses one real number from $s$ candidates in the candidate string for consequent part and $0$ code deletes the related rule. Fig. 3. illustrates the coding format for decision string in a chromosome.

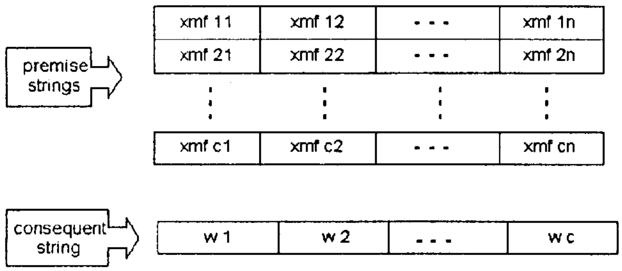


**Fig. 3. Decision strings in a chromosome**

By codes in candidate stings and decision strings, parameters are determined as (5).

$$a_{ij} = \text{center}(\text{xmf ij}),\quad b_{ij} = \text{width}(\text{xmf ij}) \qquad (5)$$
$$w_i = \text{position}(\text{w i})$$

We determine a fuzzy model by the above coding formats, so the fuzzy rule sets are not by every possible combination of input fuzzy sets but by a chromosome itself. The proposed determination method for a fuzzy linguistic model can efficiently consider problems of redundant rules and narrow range for choosing structures and parameters for fuzzy rule sets.

### 2.3 Algorithm Description

GA is an iterative adaptive general purpose search strategy based on the principle of natural selection. GAs explore a population of solutions in parallel. Each solution in the population is encoded as a chromosome, and a collection of chromosomes forms a generation. A new generation evolves by performing genetic operations, such as reproduction, crossover and mutation on strings in the current population and then placing the products into the new generation. Reproduction is a process in which individual strings are copied according to their fitness. After the members of the newly reproduced strings in the mating pool are mated at random, offsprings are constructed by copying the portion of parent strings designated by random crossover points with a crossover probability. As each bit is copied from parent to offspring, the bit has the probability of mutation, another GA operation for changing populations.

In this paper, GA is used to determine and optimize a fuzzy linguistic model to identify complex systems. During the GA search, fitness value is determined by the inverse of mean square errors between real data and inferred output. When the nullset in input variables exists, the fitness function of the related population is multiplied by a fixed penalty. We check the existence of nullset for inputs not on the individual input fuzzy set but on the whole rule sets. By evolving a generation repeatedly, we obtain a satisfactory fuzzy model. For guaranteeing the convergence of GA search, we adopt the elitist reproduction. The procedure for the automatic fuzzy modeling by GA is summarized as follows :

**Step 1** Set the maximum generation number (*max_gen*) and population size. Fix crossover rate and mutation rate. Set the maximum rule number($c$) and candidate strings' lengths for premise and consequent part($r$, $s$).

**Step 2** Normalize input/output data pairs to be identified. Generate initial populations composed of randomly generated codes.

**Step 3** Decode the chromosome of each population and determine the fuzzy model. Evaluate the mean square error of fuzzy model by (6) and give fitness value to each population by (7).

$$E = \frac{1}{N}\sum_{l=1}^{N}(y_l - y_l^*)^2 \tag{6}$$

$$fitness(a) = \begin{cases} \text{if no null set exists :} \\ \quad \dfrac{1}{E} = \dfrac{N}{\sum_{l=1}^{N}(y_l - y_l^*)^2} \\ \text{if null set exists :} \\ \quad \dfrac{1}{E} \times penalty = \dfrac{N}{\sum_{l=1}^{N}(y_l - y_l^*)^2} \times penalty \end{cases} \tag{7}$$

**Step 4** Evolve all populations by reproduction, crossover and mutation. Increase generation number by replacing old generation with new generation. During the replacement, preserve the population which has the maximum fitness value by the elitist reproduction.

**Step 5** Repeat **Step 3**~**Step 4** until the satisfactory population appears or the generation number is over *max_gen*.

## 3. FINE TUNING OF PARAMETERS

By adopting GA to fuzzy modelling, we can use its full advantage of global search. If GA was left searching for enough time, it would be eventually converge to an optimal solution. However, GA does not perform very well at fine-tuning when the optimal solution is nearby, therefore we adopt a gradient method to the last stage for automatic fuzzy modelling.

After finding near optimal fuzzy model by GA, we finely tune the membership function parameters of premise part and the real numbers of consequent part by a gradient method. This process is performed so as to minimize the error function $E$ given by equation (6). Learning rules for optimizing parameters are then obtained as follows :

$$a_{ij}(k+1) = a_{ij}(k) - K_a \cdot \partial E/\partial a_{ij} \tag{8}$$

$$b_{ij}(k+1) = b_{ij}(k) - K_b \cdot \partial E/\partial b_{ij} \tag{9}$$

$$w_{ij}(k+1) = w_{ij}(k) - K_w \cdot \partial E/\partial w_{ij} \tag{10}$$

## 4. NUMERICAL EXAMPLE

We consider the fuzzy model of a dynamic process using a famous example of the system identification by Box and Jenkins[9]. The process is a gas furnace with an input $u(k)$ and output $y(k)$ : gas flow rate and $CO_2$ concentration, respectively. 296 I/O data pairs $\{y(k), u(k), k=1,2,\ldots,296\}$ are available in this example. Since the process is dynamic, we take $u(k-4)$ and $y(k-1)$ as input variables to affect the present output $y(k)$.

Initial parameters for running the GA modeling procedure are as follows. Maximum generation number is 10000, population size is 100, maximum rule number is 6, candidate string lengths(r, s) are 12 and 12 respectively. During the population evolving, crossover rate is 0.8, mutation rate is 0.5. Nullset penalty on determination of fitness value is 0.0001. On the fine tuning stage the learning rates $K_a, K_b$ and $k_w$ are 0.0001, 0.0001 and 0.001, respectively.

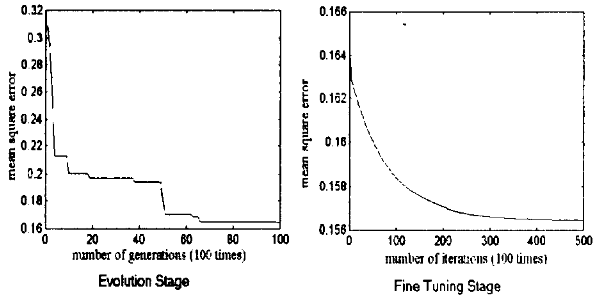


**Fig. 4. Change of the mean square error**

Fig. 4. shows the change of the mean square error during the evolution and fine tuning stage.

The fuzzy model identified by the GA and gradient method is shown in Fig. 5.

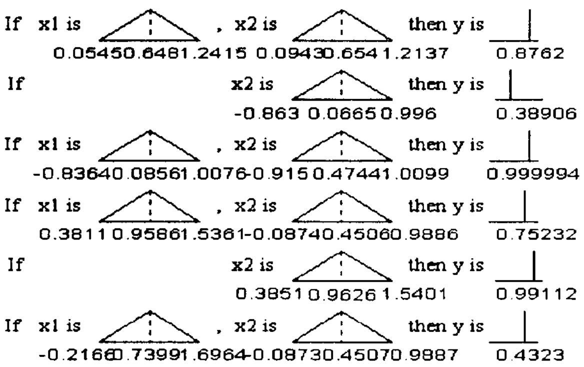


**Fig. 5. Identified fuzzy model for gas furnace.**

In TABLE 1, we compare our linguistic fuzzy model's performance with other linguistic fuzzy models and fuzzy relational models, in which our model has the best performance with the smallest number of rules.

**TABLE 1. COMPARISON OF OUR MODEL WITH OTHER MODELS**

| Model Name | Inputs | Number of Rules | Mean Square Error |
|---|---|---|---|
| Tong's [2] | $y_{k-1}$ $u_{k-4}$ | 19 | 0.469 |
| Pedrycz's [1] | $y_{k-1}$ $u_{k-4}$ | 81 | 0.320 |
| Xu's [3] | $y_{k-1}$ $u_{k-4}$ | 25 | 0.328 |
| Sugeno's [4] | $y_{k-1}$ $u_{k-3}$ $u_{k-4}$ | 6 | 0.190 |
| Ours | $y_{k-1}$ $u_{k-1}$ | 6 | 0.156 |

## 5. CONCLUSION

The objective of this paper was to develop a new GA scheme as a tool for the automatic fuzzy modeling. To do this, we proposed a coding format of chromosomes in GA and a complete algorithm by the coding format. We also adopted a gradient descent method to obtain the better result. A great advantage of the presented approach is that the only thing we have to do to design a fuzzy model for a complex system is determining the maximum number of rule sets and any previous knowledge about the system is not needed. The simulation result showed that GA and fine tuning method were successfully used to form a fuzzy model automatically. Future work will focus on developing fitness functions which are aimed at multi-object performance.

## 6. REFERENCES


[1] Pedrycz W., "An Identification Algorithm in Fuzzy Relational Systems", *Fuzzy Sets and Systems*, vol. 13, pp. 153-167, 1984.

[2] Tong R. M., "The Evaluation of Fuzzy Models Derived from Experimental Data", *Fuzzy Sets and Systems*, vol. 4, pp. 1-12, 1980.

[3] Xu C. W., "Fuzzy Systems Identification and Self-learning for Dynamic Systems", *IEEE Trans. Syst. Man. Cybern.*, vol. 17, no. 4, pp. 683-689, 1987.

[4] Sugeno M. and Yasukawa T., "A Fuzzy-Logic-Based Approach to Qualitative Modeling", *IEEE Trans. Fuzzy sys.*, vol. 1, no. 1, 1993.

[5] Karr C., "Applying Genetics to Fuzzy Logic". *AI Expert*, vol. 6, pp. 26-33, 1991.

[6] Y. H. Joo, H. S. Hwang, K. B. Kim and K. B. Woo, "Fuzzy System Modeling by Fuzzy Partition and GA Hybrid Schemes", *Fuzzy Sets and Systems*, 1996, (To be printed)

[7] Liska J. and Melsheimer S. S., "Complete Design of Fuzzy Logic Systems using Genetic Algorithms", *Proc. of the third IEEE confer. Fuzzy systems*, pp. 1377-1382, 1994.

[8] A. Homaifar and V. E. McCormick, "Simultaneous Design of Membership Functions and Rule Sets for Fuzzy Controllers Using Genetic Algorithms," *IEEE Trans. Fuzzy sys.*, vol. 3, no. 2, pp. 129-139, MAY 1995.

[9] Box G. E. P and Jenkins G. M., "Time Series Analysis, Forecasting and Control", (Holden Day, San Francisco, 1970).