

# T-S Fuzzy Modelling Using Advanced Genetic Algorithms

PAVEL ŠIŠPERA, M.Sc.<sup>1</sup>  
Prof. MIROSLAV POKORNÝ, Ph.D.<sup>1</sup>  
JAN ROUPEC, Ph.D.<sup>2</sup>

<sup>1</sup>Department of Measurement and Control  
Faculty of Electrical Engineering and Computer Science, VSB Technical University Ostrava  
17. listopadu 15, 708 33 Ostrava  
CZECH REPUBLIC

<sup>2</sup> Department of Automatic Control and Computer Science  
Faculty of Mechanical Engineering, Technical University of Brno  
Technická 2, 616 69 Brno  
CZECH REPUBLIC

## *Abstract:*

This paper introduces a soft-computing oriented approach to Takagi-Sugeno fuzzy modelling using the evolutionary principles. Genetic algorithms are applied to optimize fuzzy input variables space through genetic fuzzy clustering procedure and to identify the fuzzy model. Some advanced procedures e.g. individuals lifetime limitation and redundant genes application are used. The presented algorithm allows also the determination of the relevant inputs variables of fuzzy model from their potential candidates. To clarify the advantages of the proposed approaches the numerical example of modelling of fuzzy non-linear system is also introduced.

## *Key-words:*

Takagi-Sugeno Fuzzy Model, Input Variables Selection, Fuzzy Clustering, Advanced Genetic Algorithm, Numerical Example

## **1. Introduction**

New research works focused on soft-computing methods namely on exploitation of evolution approaches in fuzzy modelling and control were started at the Department of Measurement and Control FEI VSB Technical University Ostrava. The aim of this contribution is to provide information concerning results of investigation effort aimed on genetic algorithms application and fuzzy model identification improvement.

In fuzzy modelling technology the suitable fuzzy diversifications of input variable space as well as the number of linguistic rules determination and their structure and/or parameters identification are of crucial importance. To investigate soft-computing methods in tasks of Takagi-Sugeno (T-S) predictive model [8], [9] identification of the genetic algorithms (GA) was applied to find out the optimal distribution of cluster

centres in data processed and proper T-S fuzzy model identification.

## **2. Genetic Algorithm in T-S Fuzzy Model Identification**

### **2.1 Advanced Genetic Algorithm (A-GA)**

The basic idea of a genetic algorithm is quite simple [5]. GA works not only with one solution in time but with the whole population of solutions. The population contains many (ordinary several hundreds) individuals – bit strings representing solutions. The mechanism of GA involves only elementary operations like strings copying, partially bit swapping or bit value changing. GA starts with a population of strings and thereafter generates successive populations using the following three basic operations: reproduction, crossover, and mutation. Reproduction is the process by which

individual strings are copied according to an objective function value (fitness). Copying of strings according to their fitness value means that strings with a higher value have a higher probability of contributing one or more offspring to next generation. This is an artificial version of natural selection. Mutation is an occasional (with a small probability) random alteration of the string position value. Mutation is needed since, in spite of reproduction and crossover effectively searching and recombining the existing representations, they occasionally become overzealous and lose some potentially useful genetic material. The mutation operator prevents such an irrecoverable loss. The recombination mechanism allows mixing of parental information while passing it to their descendants, and mutation introduces innovation into the population.

In spite of simple principles, the design of GA for successful practical using is surprisingly complicated. GA has many parameters that depend on the problem to be solved. In the first, it is the size of population. Larger populations usually decrease the number of iterations needed, but dramatically increase the computer time for any iteration. The factors increasing demands on the size of population are the complexity of the problem being solved and the length of the individuals. Every individual contains one or more chromosomes containing value of potential solution. Chromosomes consist of genes. The gene in our version of Advanced GA (A-GA) is a structure representing one bit of solution value. It is usually advantageous to use some redundancy in genes and so the physical length of our genes is greater than one bit. This type of redundancy was introduced by Ryan [2]. The structure of gene we use is in Figure 1.

gene content	meaning
1 1 1	value 1
1 1 0	
1 0 1	
1 0 0	shade zone (value set randomly to 0 or 1)
0 1 1	
0 1 0	value 0
0 0 1	
0 0 0	

Fig.1 - The Structure of Gene

To prevent degeneration and the deadlock in local extreme the limited lifetime of individual in A-GA can be used. Limited lifetime is realized by the “death” operator [3], which represents something like continual restart of GA. This operator enables decreasing of

population size as well as increasing the speed of convergence. It is necessary to store the best solution obtained separately – the corresponding individual need not to be always present in the population because of the limited lifetime.

Many GAs are implemented on a population consisting of haploid individuals (each individual contains one chromosome). However, in nature, many living organisms have more than one chromosome and there are mechanisms used to determine dominant genes. Sexual recombination generates an endless variety of genotype combination that increases the evolutionary potential of the population. Because it increases the variation among the offspring produced by an individual, it improves the change that some of them will be successful in varying and often-unpredictable environments they will encounter. Using diploid or “multiploid” individuals can often decrease demands on the population size. However the use of multiploid GA with sexual reproduction brings some complications, the advantage of multiploidy can be often substitute by the “death” operator and redundant genes coding.

New individuals are created by operation called crossover. In the simplest case crossover means swapping of two parts of two chromosomes split in randomly selected point (so called one point crossover). In A-GA we use the uniform crossover on the bit level is used.

The strategy of selection individuals for crossover is very important. It strongly determines the behaviour of GA. For genetic clustering the ranking selection with elite brings satisfactory results.

We must consider the mechanisms linking genetic algorithm to the solved problem. It is very important to find a good encoding of solutions to the bit strings in chromosomes. The Gray code is usually used to eliminate the Hamming barrier. The second mechanism is represented by an objective function. The evaluation of this function is the link between the genetic algorithm and the problem to be solved. Most of the computer time in GAs is spent by evaluating objective functions.

Genetic algorithms commonly use heuristic and stochastic approaches. From the theoretical viewpoint, the convergence of heuristic algorithms is not guaranteed for the most of application cases. That is why the definition of the stopping rule of the GA brings a new problem. It can be shown [4] that while using a proper version of GA the typical number of iterations can be determine.

A-GA we use has the following scheme:

1. *Generation of the initial population:* At the beginning the whole population is generated randomly, the members are sorted by the fitness (in descendent order).
2. *Mutation:* The mutation is applied to each gene with the same probability, all GAs described here use  $p_{mut} = 0.05$ . The mutation of the gene means the inversion of one randomly selected bit in the gene.
3. *Death:* Classical GA uses two main operations – crossover and mutation (the other operation should be migration). In A-GA described in this paper, we use the third operation – *death*. Every individual has the additional information – age. A simple counter that is incremented in each of GA iterations represents the age. If the age of any member reaches the preset lifetime limit  $LT$ , this member "dies" and is immediately replaced by a new randomly generated member. The age is not mutated nor crossed over. The age of new individuals (incl. individuals created by crossover) is set to zero. Lifetime limit in our application is set to 5.
4. *Sorting by the fitness.*
5. *Crossover:* Uniform crossover is used for all genes (each bit of the offspring gene is selected separately from corresponding bits of both parent's genes).
6. Go to step 2.

In crossover, we do not replace all members of the population. The crossover generates the number of individuals corresponding to the quarter of the population only. Created individuals are sorted into the corresponding places in the population according to their fitness in such a way that the size of the population remains the same. Newly created offspring of low fitness do not have to be involved in the population.

## 2.2 A-GA in Task of Fuzzy Model Identification

The method of fuzzy genetic clustering is inspired by [5]. In the beginning the centre of clusters are generated. We do not use randomly generated centres, but the centres equidistantly cover the space of input data. The number of centres is the parameter of the task and is entered manually. The second parameter is a shape of fuzzy sets. The value bits in chromosomes represents if the corresponding cluster is used or not (one bit for each cluster) and the parameters of fuzzy

sets (eight bits for each fuzzy set, the total number of bits depends on the dimension of the space of input variables and on the number of clusters, in our case sixteen bits for each cluster). In our example we use the triangular fuzzy sets and the parameters means the shape of fuzzy set. Fuzzy sets parameters use the Gray code.

We denote the input data points,  $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ ,  $n$  is the number of data points, the number of clusters is  $k$ . The objective function we use has the form

$$f(\bar{x}) = \sum_{i=1}^n \left( 1 - \max_{j=1, \dots, k} (\mu_j(\bar{x}_i)) \right)^2 \quad (1)$$

$\mu_j(\bar{x}_i)$  is the membership value of the point  $\bar{x}_i$  to the  $j$ -cluster. We search for minimum of  $f(\bar{x})$ .

The objective function described above does not affect all disadvantageous situations possible and we must use some penalization. The unacceptable case occurs, when we can observe point that does not belong to any cluster. The penalty value is 100 for each point. The extreme situation occurs when no point belong to any cluster, the objective function value for this improbable case is  $10^{99}$ . The penalization is also used, when we have cluster that does not contain any data point (or when any cluster contains very low number of points – the value depends on input data). In this case, the objective function is multiplied by 4 for each "empty" cluster.

Because we have relatively short chromosomes in our example, we can use small population (200 individuals) and the convergence is very fast. The typical course of convergence is shown in Figure 2.

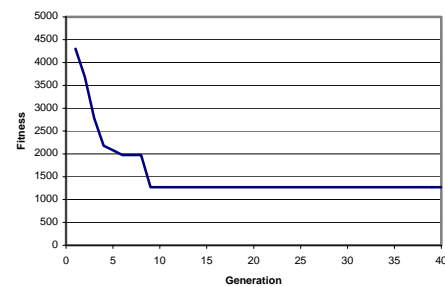


Fig. 2 - The Convergence of A-GA

Next, the above-mentioned A-GA genetic algorithm we use in task of non-linear fuzzy system modelling [7].

### 3. Case Study

The A-GA procedures were tested through modelling and identification of simple non-linear fuzzy system with three inputs and one output variable. The eight rules of initial T-S fuzzy system were determined (Figure 3) and input/output data set was generated.

IF ( $x_1$  is S) AND ( $x_2$  is S) AND ( $x_3$  is S) THEN  
 $y_1 = 3.2x_1 + 2.2x_2 + 0.05x_3 + 1$   
 IF ( $x_1$  is S) AND ( $x_2$  is H) AND ( $x_3$  is S) THEN  
 $y_2 = 2.5x_1 + 2.3x_2 + 0.06x_3 + 4$   
 IF ( $x_1$  is H) AND ( $x_2$  is S) AND ( $x_3$  is S) THEN  
 $y_3 = 0.3x_1 + 3.0x_2 + 0.05x_3 + 5$   
 IF ( $x_1$  is H) AND ( $x_2$  is H) AND ( $x_3$  is S) THEN  
 $y_4 = 0.9x_1 + 0.9x_2 + 0.06x_3 + 7$   
 IF ( $x_1$  is S) AND ( $x_2$  is S) AND ( $x_3$  is H) THEN  
 $y_5 = 3.2x_1 + 2.2x_2 + 0.06x_3 + 1$   
 IF ( $x_1$  is S) AND ( $x_2$  is H) AND ( $x_3$  is H) THEN  
 $y_6 = 2.5x_1 + 2.3x_2 + 0.05x_3 + 4$   
 IF ( $x_1$  is H) AND ( $x_2$  is S) AND ( $x_3$  is H) THEN  
 $y_7 = 0.3x_1 + 3.0x_2 + 0.06x_3 + 5$   
 IF ( $x_1$  is H) AND ( $x_2$  is H) AND ( $x_3$  is H) THEN  
 $y_8 = 0.9x_1 + 0.9x_2 + 0.05x_3 + 7$

Fig. 3 – Initial Three-Inputs Fuzzy System

Input 1 ( $x_1$ ): S (0, 0, 8), H (4, 10, 10). Input 2 ( $x_2$ ): S (0, 0, 7), H (3, 12, 12). Input 3 ( $x_3$ ): S (0, 0, 6), H (5, 11, 11).

Firstly the procedure of relevant system inputs selection was applied.

#### 3.1 A Relevant Input Variables Selection

If more input variable candidates exist for the inputs to the system, we can determine relevant pre-defined number of them through procedure using a combinatorial approach. We take a heuristic method to select some inputs and increase the number of combinatorial steps watching a regularity criterion.

Let us consider a fuzzy system with  $n$  inputs and one output (which is representing a nonlinear function) and a set of output values of this system for randomly chosen inputs. In general, let  $X$  be a set of possible input candidates  $x_1, x_2, \dots, x_n$  then the total number of cases is the number of subsets except an empty subset of  $X$ , i.e.,  $2^n - 1$ . Here we take a heuristic method to select some inputs from among the candidates. We increase the number of inputs one by one, watching a suitable criterion [6]. First, we divide the data into two groups: A (first half of the data) and B (second half of

the data). As a criterion to this purpose, we use the so-called regularity criterion  $RC$ , in GMDH (group method of data handling), which is defined as follows [6]:

$$RC = \left[ \sum_{i=1}^{k_A} (y_i^A - y_i^{AB})^2 / k_A + \sum_{i=1}^{k_B} (y_i^B - y_i^{BA})^2 / k_B \right] / 2 \quad (2)$$

where  $k^A$  and  $k^B$  are the number of data of the groups A and B,  $y_i^A$  and  $y_i^B$  are the output data of the groups A and B. Next the  $y^{AB}$  represents the model output for the group A input estimated by the model identified using the group B data and  $y^{BA}$  the model output for the group B input estimated by the model identified using the group A data. Therefore, we build two models for two data groups at each stage of the identification. First, we begin with a fuzzy model with one input. We make  $n$ -models: one model for one particular input. After the identification of the models using the data groups A and B, we calculate  $RC$  of each model and select one model to minimize  $RC$  from among the one-input models. Next, we fix the one input selected above and add another input to our fuzzy model from among the remaining three candidates. Our fuzzy model has two inputs at this stage. We select the second input as we do at the first step, according to the value of  $RC$ . We continue the process until the value of  $RC$  increases. If the value of  $RC$  becomes bigger in all cases in some step then in the previous one, the process is terminated and the combination of inputs with the smallest  $RC$  value is the result of our selection.

In our case, we find out 2 relevant input variables from 3 possible candidates. The minimum of  $RC$  criterion was calculated ( $RC=1.9303$ ) and relevant inputs  $x_1$  and  $x_2$  were determined for future model identification (Figure 4).

IF ( $x_1$  is S) AND ( $x_2$  is S) THEN  
 $y_1 = 3.2x_1 + 2.2x_2 + 1$   
 IF ( $x_1$  is S) AND ( $x_2$  is H) THEN  
 $y_2 = 2.5x_1 + 2.3x_2 + 4$   
 IF ( $x_1$  is H) AND ( $x_2$  is S) THEN  
 $y_3 = 0.3x_1 + 3.0x_2 + 5$   
 IF ( $x_1$  is H) AND ( $x_2$  is H) THEN  
 $y_4 = 0.9x_1 + 0.9x_2 + 7$

Fig. 4 – Rules of Initial Reduced Fuzzy System  
 Input 1 ( $x_1$ ): S (0, 0, 8), H (4, 10, 10). Input 2 ( $x_2$ ): S (0, 0, 7): H (3, 12, 12).

The shape of input/output dependence of determined two-dimensional non-linear system is shown in Figure 5.

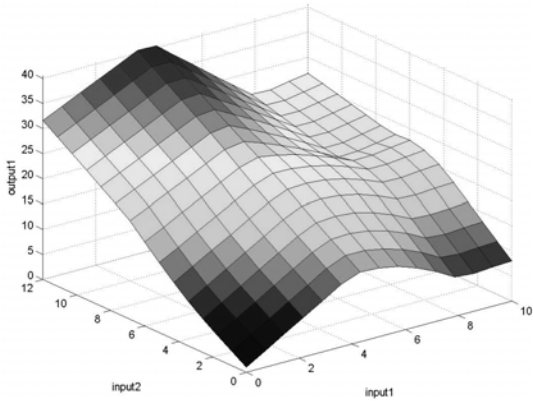


Fig. 5 – Shape of Two-Dimensional Function

Using values of two inputs  $x_1$ ,  $x_2$  the initial two-dimensional fuzzy input space was defined using data-points in Figure 6.

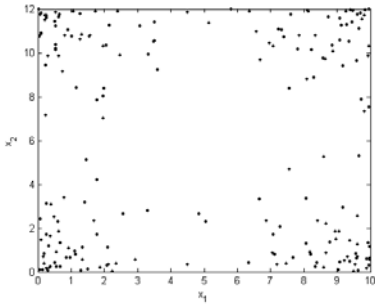


Fig. 6 – Two-Dimensional Input Space

### 3.2 Data Fuzzy Clustering and Rule Premise Part Determination

The data set of initial input space diversification (Figure 4) was investigated using fuzzy clustering A-GA algorithm to find out the suitable number and fuzzy approximation of input variable terms. The results of this procedure are shown in Figure 7 and Figure 8. Both of two inputs variables  $x_1$ ,  $x_2$  are expressed using two linguistic terms – namely SMALL (S) and HIGH (H).

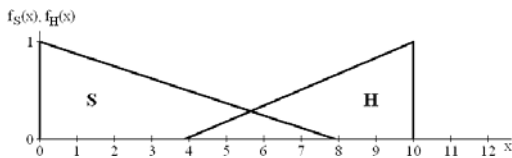


Fig. 7 – Linguistic Terms of  $x_1$

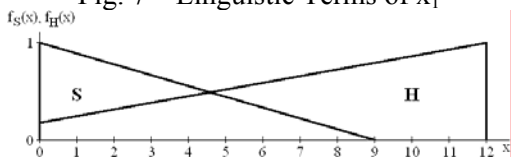


Fig. 8 – Linguistic Terms of  $x_2$

Using results of genetic fuzzy clustering the premises structure and parameters of 4 rules was done and antecedent part of identified fuzzy T-S model was obtained.

### 3.3 Rule Consequent Parts Determination

Next, the consequent part of T-S model was identified using GA procedure of genetic toolbox of MATLAB and final fuzzy T-S model was obtained (Figure 9).

$$\begin{aligned} &\text{IF } (x_1 \text{ is S}) \text{ AND } (x_2 \text{ is S}) \text{ THEN} \\ &y_1 = 3.507x_1 + 1.736x_2 + 0.1082 \\ &\text{IF } (x_1 \text{ is S}) \text{ AND } (x_2 \text{ is H}) \text{ THEN} \\ &y_2 = 2.487x_1 + 2.163x_2 + 5.497 \\ &\text{IF } (x_1 \text{ is H}) \text{ AND } (x_2 \text{ is S}) \text{ THEN} \\ &y_3 = 0.7395x_1 + 4.142x_2 + 1.019 \\ &\text{IF } (x_1 \text{ is H}) \text{ AND } (x_2 \text{ is H}) \text{ THEN} \\ &y_4 = 0.1535x_1 + 1.766x_2 + 4.076 \end{aligned}$$

Fig. 9 – Rules of Final Fuzzy Model

Input 1 ( $x_1$ ): S (-7.808, 0.0362, 7.881), H (3.888, 9.999, 16.11). Input 2 ( $x_2$ ): S (-8.985, 0.0063, 8.997), H (-2.763, 11.98, 26.72).

The GA algorithm of MATLAB we used started with next parameters: size of population  $N = 800$  individuals, generation number limit  $t = 1500$  generations, information was coded into chromosomes as a chain of real numbers with 16 genes. As a selection method the tournament selection between 2 individuals was used, as a crossover method the arithmetic crossover and as a mutation method the uniform mutation were used. As a termination condition the generation limit reached is used. Fitness function has the following formula

$$Fitness = - \sum_{i=1}^N |y_i^{GA} - y_i| \quad (3)$$

where  $N$  is the number of samples. The value  $y$  is an output of the reference fuzzy model.  $y^{GA}$  is an output value of the fuzzy model being designed by the genetic algorithm.

The course of fitness function convergence we can see in Figure 10.

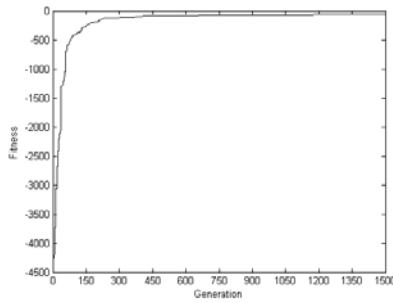


Fig. 10 – Course of Fitness Convergence

The shape of final model input/output non-linear function is shown in Figure 11.

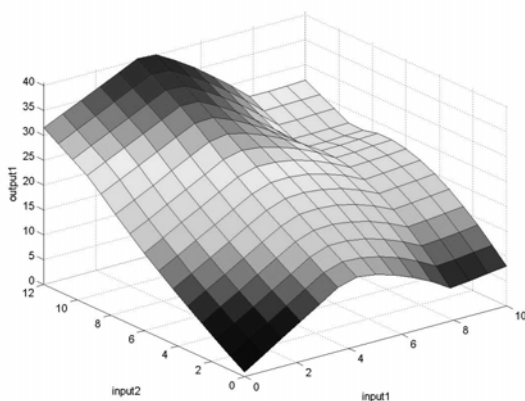


Fig. 11 – Shape of Final Function

### 3.4 Final Fuzzy Model Testing

The final results of identification process can be judged subjectively to compare both of the function shapes in the Figures 5 and Figure 11. To calculate the error criterion

$$FCR = \frac{100}{N} \cdot \sum_{i=1}^N \frac{|y^{GA} - y|}{y} \quad (4)$$

we can obtain the relative error of final model of value 1.659%.

## 4. Conclusion

The main problem of fuzzy data clustering is to find out a suitable number and shape of data clusters to achieve satisfying results of next fuzzy modelling.

In approach presented the input data fuzzy partition is modified in such a way that the suitable fuzzy clusters are determined using the advanced genetic algorithm. Some effective procedures namely individual lifetime limitation and the redundant genes application are used. In this method, the computational effort is bigger when conventional iterative methods rarely fall in local extreme. The method delivers

sufficient convergence speed with initial equidistant data clusters centre distribution.

The results of genetic fuzzy clustering procedure and the suitable diversification of fuzzy model input space is done including the number of input variable linguistic values and appropriate approximation of their membership functions as well. The numerical example proved effectiveness of proposed methods. The model identification algorithms include the procedure of pre-defined relevant number of input variables selection.

The future research works will be focused on hierarchical and parallel GAs application to increase the necessary computational effort.

### Acknowledgments:

This work has been supported by the GACR project 102/05/H525: “The Postgradual Study Rationalization at Faculty of Electrical Engineering and Computer Science VSB-TU Ostrava”.

### References:

- [1] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Pub. Comp., INC, 1989, ISBN 0-201-15767-5
- [2] Ryan, C.: Shades. Polygenic Inheritance Scheme. In Proceedings of MENDEL '97, Brno, 1997, pp.140-147.
- [3] Roupec, J.: Design of Genetic Algorithm for Optimisation of Fuzzy Controllers Parameters. (In Czech) Ph.D. Thesis, Brno University of Technology, Brno, 2001.
- [4] Roupec, J. – Popela, P. – Ošmera, P.: The Additional Stopping Rule for Heuristic Algorithms. In Proceedings of Mendel '97, Brno, 1997. pp. 135–139
- [5] Pistauer, M.: Clustering Algorithms for process data Analysis, Proc.15-th IASTED, Innsbruck, 1996, pp.137-139
- [6] Sugeno, M., Yasukawa, T.: A Fuzzy-Logic Based Approach to Qualitative Modelling, IEEE Trans. on Fuzzy Systems, 1, No.1, 1993, pp.7-31
- [7] Pokorný, M.: Umělá inteligence v modelování a řízení, BEN Praha, 1996, ISBN 80-901984-4-9
- [8] Sugeno, M., Kang, G.T.: Structure Identification of Fuzzy Model, Fuzzy Sets and Systems, 28, 1988
- [9] Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and Its Application to Modelling and Control, IEEE Trans. on Systems, MAC, Vol.15, No.1, 1985