Pareto-optimality in Fuzzy Modeling*

Antonio F. Gómez-Skarmeta[1] Fernando Jiménez[2] Jesús Ibáñez[3]
Facultad de Informática. Dept. Informática, Inteligencia Artificial y Electrónica
Universidad de Murcia. Apartado 4021. 30001 Murcia.
Phone: +34 - 68 - 364607, Fax: +34 - 68 - 364151
e-mail:
[1]skarmeta@dif.um.es [2]fernan@dif.um.es [3]jesus@dif.um.es

ABSTRACT: Recent techniques in fuzzy identification and multiobjective optimization with Evolutionary Algorithms are combined in this paper. We propose a single step Evolutionary Algorithm to find multiple Pareto-optimal solutions to the problems of generation and/or tuning of fuzzy models which make it possible that a decision maker can choose the most appropiate solution according to the current decision environment.
Keywords: Fuzzy modeling; Pareto-optimality; Evolutionary Algorithms.

1 INTRODUCTION

Systems description with fuzzy quantities is known as *Fuzzy modeling* (Sugeno and Yasukawa, 1993). The identification of a model in fuzzy modeling usually contains two steps: structure identification and parameter identification. To identify the structure of the model, on the one hand it is necessary to find input variables by determining firstly the possible input candidates and subsequently a set of input variables which affect the output. On the other hand, we need to find input-output relations, i.e., number of fuzzy rules and partition of input space. Parameter identification in a fuzzy model consists of determining the appropiate parameter in the membership functions of the fuzzy sets. In this paper we deal with the problems of finding the input-output relations and the parameter identification.

Basically, the problem consists in to find a finite set of fuzzy rules being able to reproduce the input-output system behaviour. The only information about this behaviour is assumed to be a sample of input-output example pairs. We assume the system is a MISO (multi-input, single-output) one, i.e. described by rules with the form:

$$R_h : \text{ If } x_1 \text{ is } \tilde{A}_h^1 \text{ and } \dots \text{ and } x_p \text{ is } \tilde{A}_h^p \text{ then } y \text{ is } \tilde{B}_h \tag{1}$$

where R_h is the hth rule ($1 \leq h \leq k$), x_j ($1 \leq j \leq p$) are input variables (p input variables), y is the output (one output variable), and $\tilde{A}_h^j$ and $\tilde{B}_h$ are fuzzy values for these linguistic variables.

Thus, the first step is to generate a first approximation of the fuzzy model that describes the system. Implicit in this step is the determination of the necessary number of rules. The output of this step is a collection of rules which can be seen as rough estimates of the final rule base. The second step consists of a tuning of the initial rough rules to give us our final rule base.

Both generation and tuning steps have been recently solved with *Evolutionary Algorithms* (EA) by several authors. Herrera, Lozano and Verdegay (1998) describe different EA to generate and tune the fuzzy model in separate steps. Gómez-Skarmeta and Jiménez (1998) describe an EA which achieves generation and tuning simultaneously. One of the major difficulties which appear in fuzzy modeling with EA is finding the aggregation mechanism of the different criteria which should be taken into account to measure the fitness of a solution of the problem. Some criteria considered in fuzzy modeling are quadratic error, number of rules, number of samples covered, frequency value, width and symmetry of the membership funtion, niche interaction rate, etc. In all the above methods, multiple objectives are combined to form one objective by using some knowledge of the problem which is normally obtained by experimentation. A drawback of this methods is that a decision maker must have a prior knowledge of the priority of each objective before aggregating in a single objective the set of objectives. Thus, for different situations, different priorities need to be assigned and the same problem requires to be solved

*Research supported in part by CICyT under project TIC97-1343-C02-02.

a number of times. A more realistic method would be one that can find multiple Pareto-optimal (nondominated) solutions simultaneously so that a decision maker may be able to choose the most appropiate solution for the current situation. A pareto-optimum is a solution which offers the least objective conflict, i.e., a point in the search space that is optimally placed from the individual optimum of each objective and cannot be improved for one objective without making it worse for another. The knowledge of many Pareto-optimal solutions is also useful for later use, when the current situation has changed and a new solution is required. Since EA deal with a population of points, multiple Pareto-optimal solutions can be captured in the population in a single run. The idea of assigning equal probability of reproduction to all nondominated individuals in the population by means of a nondominated sorting procedure was first proposed by Goldberg (1989). Next, several authors ((Fonseca and Fleming, 1993), (Horn and Nafpliotis, 1993), (Srinivas and Deb, 1995)) developed variants based on this concept, and successfully applied it to some problems. An overview of EA in multiobjective optimization can be found in (Fonseca and Fleming, 1995).

In this paper we present an EA based approach to find multiple Pareto-optimal solutions to either of the following problems:

1. *Only tuning:* The EA is used to tune a rule base with k rules previously introduced using the fuzzy clustering process as suggested in (Delgado, Gómez-Skarmeta and Martín, 1995). The final rule base will contain exactly k rules.
2. *Generating and tuning:* EA is used in this case to generate and to tune a rule base directly from data. It is also possible to incorporate an initial rule base with k rules as in the previous method, that "helps" the search process. The final rule base will contain any number N of rules, with $1 \leq N \leq MAX$, where MAX is given by user.

In the following sections the main characteristics of the EA are described.

2 A PARETO-OPTIMALITY EVOLUTIONARY ALGORITHM FOR FUZZY IDENTIFICATION

Basically, we can say that an EA is based on the following components (Michalevicz, 1992): a *representation* of solutions to the problem; a way to create an *initial population* of solutions; an *evaluation function* to measure the fitness of any solution, and plays the role of the "environment", in which the better solutions (i.e., those that have a better objective value) may have greater probability of survival; *variation operators* (probabilistic transition rules) that effect the composition of children during reproduction; and values for the *parameters* that the algorithm uses to guide its evolution: population size, number of generations, crossing and mutation probabilities, etc.

In the following we show the main characteristics of the Pareto-Optimality EA to Fuzzy Identification by means of the components cited above, and some remarks of design such as selection and generational replacement mechanisms are made.

2.1 REPRESENTATION

In this problem, one solution is a collection of rules as in (1). The number of rules is known and constant if the proposal is to tune a previous rule base with a fixed number k of rules, but it is unknown and variable ($N \in [1..MAX]$) if we want to achieve processing completely, i.e., "generate and tune" task. Firstly, we show a possible representation of the solutions to generate and tune a rule base, and we will see how this representation can be used to tune a rule base introduced previously ("only tuning" task).

Populations in this EA are composed of individuals H_i, $i = 1, \ldots, popsize$, represented as follows:

$$Individual\ H_i :$$

$$\underbrace{R_1^i\ R_2^i\ \ldots\ R_{MAX}^i}_{Rule\ set}\ \ \underbrace{d_1^i\ d_2^i\ \ldots\ d_{MAX}^i}_{Control\ set}$$

where the rule $R_h^i = \{\tilde{A}_h^{i1} \tilde{A}_h^{i2} \ldots \tilde{A}_h^{ip} \tilde{B}_h^i\}$ $(1 \leq h \leq MAX,\ 1 \leq i \leq popsize)$ consist of fuzzy values for the p input variables and output variable, which are represented as:

$$\tilde{A}_h^{ij} = (a_{h1}^{ij}, a_{h2}^{ij}, a_{h3}^{ij}, a_{h4}^{ij}), \quad (1 \leq h \leq MAX,\ 1 \leq i \leq popsize,\ 1 \leq j \leq p)$$
$$\tilde{B}_h^{i} = (b_{h1}^{i}, b_{h2}^{i}, b_{h3}^{i}, b_{h4}^{i}), \quad (1 \leq h \leq MAX,\ 1 \leq i \leq popsize)$$

i.e., trapezoidal fuzzy numbers (Dubois and Prade, 1980) with $a_{hl}^{ij}, b_{hl}^{i} \in I\!R$, $(1 \leq h \leq MAX,\ 1 \leq l \leq 4,\ 1 \leq i \leq popsize,\ 1 \leq j \leq p)$. Values $d_h^i \in \{0,1\}$ $(1 \leq h \leq MAX,\ 1 \leq i \leq popsize)$ indicate if rule h of the individual i is active in the solution (1 = active, 0 = inactive). Thus, the number N of rules in the individual i is given by the number of values $d_h^i = 1$, $1 \leq h \leq MAX$.

Note that the control set is not necessary if we want to tune an existing rule base with a fixed number k of rules. Thus, the above representation can be used for this proposal ("only tuning" task) setting $MAX = k$ and ignoring the control set in obtaining initial population, evaluation function and genetic operators, i.e. all MAX rules are interpreted as actives. Next, we will see the implications of this representation on obtaining an initial population, the identification of nondominated individuals, the selection and variation operators and the generational replacement to achieve the "generate and tune" task.

2.2 INITIAL POPULATION

To create an initial population we take into account two criteria:

1. Use available information (examples, rule bases, etc.).
2. Explore (randomly) solution space to obtain an appropiate diversity.

According to these criteria, we propose three different alternatives (F1, F2 and F3) to initialize the solutions, which will be applied with probabilities p_1, p_2 and p_3 respectively. F1 and F2 assume the existence of an initial rule base with k rules $(1 \leq k \leq MAX)$.

(F1) All k rules of the initial rule base are introduced in random positions of the solution, and corresponding values in the control set are set to 1. The rest of the rules $(MAX - k)$ are obtained randomly according to the domains of the input and output variables, and the corresponding values in the control set are set to 0. Example:

$$\begin{array}{ll} \textit{Initial rule base } (k = 4): & R_1\ R_2\ R_3\ R_4 \\ \textit{Initial solution } (MAX = 7): & Rnd\ R_3\ R_2\ Rnd\ R_1\ R_4\ Rnd \quad 0\ 1\ 1\ 0\ 1\ 1\ 0 \end{array}$$

where Rnd indicates a rule that has been obtained randomly. Although other approaches can be used to randomly obtain fuzzy rules, we propose the following:

For each trapezoidal fuzzy number $\tilde{u} = (u_1, u_2, u_3, u_4)$ do

$u_1 \leftarrow$ random real number $\in [a, b]$;

$u_2 \leftarrow$ random real number $\in [u_1, b]$;

$u_3 \leftarrow$ random real number $\in [u_2, b]$;

$u_4 \leftarrow$ random real number $\in [u_3, b]$;

where $[a, b]$ is the domain of the variable.

(F2) Each rule of the solution, either is selected from the initial rule base or is obtained randomly with equal probability. All values in the control set are generated randomly. Example:

$$\begin{array}{ll} \textit{Initial rule base } (k = 4): & R_1\ R_2\ R_3\ R_4 \\ \textit{Initial solution } (MAX = 7): & R_2\ Rnd\ Rnd\ R_4\ Rnd\ R_1\ Rnd \quad 1\ 0\ 1\ 0\ 1\ 1\ 1 \end{array}$$

(F3) All MAX rules of the solution and all values in the control set are obtained randomly. Example:

$$\textit{Initial solution } (MAX = 7): \quad Rnd\ Rnd\ Rnd\ Rnd\ Rnd\ Rnd\ Rnd \quad 0\ 1\ 0\ 1\ 1\ 0\ 1$$

2.3 IDENTIFICATION OF NONDOMINATED INDIVIDUALS

Let f_k $(k = 1, \ldots, M)$ be the objective functions which represent the criteria considered in the fuzzy model. We assume, without loss of generality, that all objectives have to be minimized. An individual H_i $(i = 1, \ldots, popsize)$ is said to be a *nondominated individual* in the population if there is no other individual H_j in the population such that:

$$f_k(H_j) \leq f_k(H_i), \text{ for all } k = 1, \ldots, M$$

and

$$f_k(H_j) < f_k(H_i), \text{ for at least one } k.$$

Obviously, only those rules R_h with $d_h = 1$ $(1 \leq h \leq MAX)$ are taken into account in the evaluation process.

Note that nondominated individuals are not necessary Pareto-optimal solutions.

2.4 SELECTION MECHANISM AND GENERATIONAL REPLACEMENT

We propose to use the *Pareto domination tournament* (Horn and Nafpliotis, 1993). In this method, first a comparison set comprising a specific number (t_{dom}) of individuals is picked at random from the population at the beginning of each selection process. Next, once we have the comparison set, two random individuals are picked from the population and compared with the members of the comparison set for domination with respect to the objective function. If one is nondominated and the other is dominated, then the nondominated is selected. If both are either nondominated or dominated, a niche count is found for each individual by counting in the entire population the number of points within a certain distance (σ_{share}) from it. The individual with the least niche count within the two, is selected.

Variation operators are applied to the selected individuals and the offspring are copied to the next population. This process is repeated until the whole new population is generated (*complete generational replacement*). Moreover we use *elitism*, selecting in each generation a random nondominated individual from the entire population.

2.5 VARIATION OPERATORS

Since solutions contain two diferent types of information (denoted as rule set and control set) it is necessary to use different variation operators to achieve the recombination process. We used the following variation operators:

Rule set variation operators

We consider five variation operators in this group (three crossovers and two mutations) in a similar way as proposed in (Herrera, Lozano and Verdegay, 1998):

Simple crossover: We assume that two individuals H_i and H_j are selected as parents for the crossover operation. Then we produce two offsprings H'_i and H'_j as follow:

Step 1. $H'_i \leftarrow H_i$;
$H'_j \leftarrow H_j$;

Step 2. Select randomly an integer number $r \in [1..MAX \cdot (p+1)]$ (first cross point);

Step 3. Select randomly an integer number $s \in [r..MAX \cdot (p+1)]$ (second cross point);

Step 4. Interchange elements from H'_i and H'_j between positions r and s to obtain new individuals.

Arithmetical crossover: Given two individuals H_i and H_j selected as parents we obtain two offsprings H'_i and H'_j as follows:

Step 1. $H'_i \leftarrow H_i$;
$H'_j \leftarrow H_j$;

Step 2. Select randomly an integer number $r \in [1..MAX \cdot (p+1)]$ (first cross point);

Step 3. Select randomly an integer number $s \in [r..MAX \cdot (p+1)]$ (second cross point);

Step 4. Choice randomly a real number $\alpha \in [0,1]$;

Step 5. For each fuzzy number $\tilde{u}_k = (u_{k1}, u_{k2}, u_{k3}, u_{k4})$ from H'_i with $r \le k \le s$ and for each fuzzy number $\tilde{v}_k = (v_{k1}, v_{k2}, v_{k3}, v_{k4})$ from H'_j with $r \le k \le s$ do:

$u_{kl} \leftarrow \alpha \cdot u_{kl} + (1-\alpha) \cdot v_{kl},\ l = 1, \ldots, 4;$

$v_{kl} \leftarrow \alpha \cdot v_{kl} + (1-\alpha) \cdot u_{kl},\ l = 1, \ldots, 4;$

Max-Min crossover: This operator is identical to Arithmetical crossover except that Step 5 is replaced by the following:

Step 5. For each fuzzy number $\tilde{u}_k = (u_{k1}, u_{k2}, u_{k3}, u_{k4})$ from H'_i with $r \le k \le s$ and for each fuzzy number $\tilde{v}_k = (v_{k1}, v_{k2}, v_{k3}, v_{k4})$ from H'_j with $r \le k \le s$ do:

$u_{kl} \leftarrow Max\{u_{kl}, v_{kl}\},\ l = 1, \ldots, 4;$

$v_{kl} \leftarrow Min\{u_{kl}, v_{kl}\},\ l = 1, \ldots, 4;$

Simple mutation: We assume that fuzzy number $\tilde{u} = (u_1, u_2, u_3, u_4)$ has been selected for mutation. Then, we obtain a new fuzzy number $\tilde{u}' = (u'_1, u'_2, u'_3, u'_4)$ as follows:

$$u'_1 \leftarrow \text{random real number} \in \left[max\{a, u_1 - \tfrac{u_2-u_1}{2}\}, u_1 + \tfrac{u_2-u_1}{2}\right];$$

$$u'_2 \leftarrow \text{random real number} \in \left[u_2 - \tfrac{u_2-u_1}{2}, u_2 + \tfrac{u_3-u_2}{2}\right];$$

$$u'_3 \leftarrow \text{random real number} \in \left[u_3 - \tfrac{u_3-u_2}{2}, u_3 + \tfrac{u_4-u_3}{2}\right];$$

$$u'_4 \leftarrow \text{random real number} \in \left[u_4 - \tfrac{u_4-u_3}{2}, min\{u_4 + \tfrac{u_4-u_3}{2}, b\}\right];$$

where $[a, b]$ is the domain of the variable.

Non-uniform mutation: The action of this operator depends on the age of the population, and his effect is a fine local tuning in the last generations of the EA. This operator was proposed by Michalewicz (1992) and it has also been used by Herrera, Lozano and Verdegay (1998) in fuzzy modeling. Given a fuzzy number $\tilde{u} = (u_1, u_2, u_3, u_4)$ selected for this mutation, the result is a fuzzy number $\tilde{u}' = (u'_1, u'_2, u'_3, u'_4)$ as follows:

$$u'_l = \begin{cases} u_l + (b - u_l) \cdot (1 - r^{(1-\frac{t}{T})^c}) & \text{if a random digit is 0} \\ u_l - (u_l - a) \cdot (1 - r^{(1-\frac{t}{T})^c}) & \text{if a random digit is 1} \end{cases}$$

with $l = 1, \ldots, 4$, where $[a, b]$ is the domain of the variable, r is a random number from $[0, 1]$, T is the maximal generation number, t is the actual generation, and c is a system parameter determining the degree of non-uniformity.

Control set variation operators

As control set is a bit string, simple binary crossover and mutation (Goldberg, 1989) are used in this case.

2.6 PARAMETERS

The parameters to be considered by the EA are maximal generation number ($maxgen$), population size ($popsize$), maximal number of rules in the fuzzy model (MAX), probabilities F1, F2 and F3 used to obtain an initial population (p_1, p_2 and p_3, respectively), probability of crossover (p_c), probability of mutation (p_m), number of members in comparison sets (t_{dom}) and maximum phenotypic distance between individuals (σ_{share}). Note that to achieve the generating and tuning task when there is no initial rule base, it is necessary to set $p_1 = 0$, $p_2 = 0$ and $p_3 = 1$ to obtain an initial population. Note also that probabilities p_c and p_m correspond to the sums of the probabilities of all crossover and mutation operators respectively described above.

3 NUMERICAL EXAMPLES

Box and Jenkins's gas Furnace is a famous example of system identification. The well-known Box-Jenkins data set consists of 296 input-output observations, where the input $u(t)$ is the rate of gas flow into a furnace and the output $y(t)$ is the CO_2 concentration in the outlet gases. Since the process is dinamical, we can consider differents values in the time of the variables.

In table 1 we indicate different results presented in the literature, showing the number of variables chosen in each case, the number of rules and the performance of the models. We have chosen this example as a classical one, although our main objective is not to compare our performance with others results in the literature (see Gómez-Skarmeta and Jiménez (1998) for this proposal), but to show how the pareto-optimality behave, we will see that the results are comparable in general with the ones of the literature.

Model	**N. Inputs**	**N. Rules**	**PI/Error**
Tong(77)	2	19	0.469
Pedrycz(84)	2	81	0.320
Xu(87)	2	25	0.328
Peng(88)	2	49	0.245
Yoshinari(93)	2	6	0.299
Sugeno(93)	3	6	0.191
Wang(96)	2	5	0.172

Table 1: Errors in Literature.

After experiments, the following values of the parameters have produced good results:

Parameter	**Value**
$maxgen$	5000
$popsize$	20
MAX	10
$p_1 = p_2$	0.25
p_3	0.50
p_c	0.6
p_m	0.6
t_{dom}	10
σ_{share}	0.5

Next, in table 2 we present the results obtained with the proposed generating and tuning pareto-optimality EA, showing the nondominated solutions evolution with respect to the two objectives, the quadratic mean error and the number of rules in the different generations. In this algorithm we start with a random initialization.

We can see that the results obtained are near the optimal ones find in the literature. Once we have these results, is up to the decisor-maker to choose the better option to consider, either the one with the less rules, or the one with the less error.

4 CONCLUSIONS AND FUTURE TRENDS

In this paper we present an evolutionary algorithm to find multiple pareto-optimality solutions in the context of fuzzy modeling. This algorithm lets solve in a single step, the generation and/or tuning problems of fuzzy models. The results obtained up to now with the application of this algorithm although not definitive, show that its use can be of great interest for the decision makers as it offers a collection of solutions that can be used in a supervised learning environment.

REFERENCES

Delgado, M., Gómez-Skarmeta, A.F., Martín, F., Generating fuzzy rules using clustering based approach, Proc. Third European Congress on Fuzzy and Intelligent Technologies and Soft Computing, Aachen, Germany, pp.

Generation	Error	N. of rules
1.000	0.321	4
	0.313	5
2.000	0.2631	6
	0.2635	5
	0.310	3
	0.276	4
3.000	0.246	5
	0.2889	3
	0.259	4
4.000	0.243	5
	0.2885	3
	0.258	4
5.000	0.240	6
	0.257	4
	0.242	5

Table 2: Simulation results.

810-814, 1995.

Dubois, D., Prade, H., *Fuzzy Sets and Systems. Theory and Applications.* Academic Press, New York, 1980.

Fonseca, C.M, Fleming, P.J., Genetic algorithms for multi-objective optimization: formulation, discussion and generalization, in S. Forrest (ed.), Genetic Algorithms: Proc. of the Fifth Intern. Conf, Morgan Kaufmann, San Mateo, pp. 416-423, 1993.

Fonseca, C.M., Fleming, P.J., An overview of evolutionary algorithms in multiobjective optimization, Evolutionary Computation, vol. 3, no. 1, MIT Press, Spring, 1995.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

Gómez-Skarmeta, A.F., Jiménez, F., Fuzzy modeling with hibrid systems, Fuzzy Sets and Systems (to appear 1998).

Herrera, F, Lozano, M., Verdegay, J.L., A learning process for fuzzy control rules using genetic algorithms, Fuzzy Sets and Systems (to appear 1998).

Horn, J., Nafpliotis, N., Multiobjective optimization using the niched pareto genetic algorithm, IlliEAL Report No. 93005, July 1993.

Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, 1992.

Srinivas, N., Deb, K., Multiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary Computation, vol. 2, no. 3, pp. 221-248, 1995.

Sugeno, M., Yasukawa, T., A fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems, vol. 1, pp. 7-31, 1993.