Linguistic Approximation of TSK Fuzzy Models with Multi-objective Neuro-Evolutionary Algorithms

Sánchez, G.
University of Murcia
gracia@dif.um.es

Sánchez, J.F.
University of Murcia
jsanchez@ditec.um.es

Alcaraz, J.M.
University of Murcia
josem.alcaraz@carm.es

Jiménez, F.
University of Murcia
fernan@dif.um.es

Abstract

In this paper, a multi-objective constrained optimization model is proposed to improve interpretability of TSK fuzzy models. This approach allows a linguistic approximation of the fuzzy models. Three different multi-objective evolutionary algorithms (MONEA, ENORA and NSGA-II) are used together with neural network techniques. These algorithms are checked out in the approximation of a dynamic non-linear system studied in literature. The results clearly show a real ability and effectiveness of the proposed approach to find accurate and interpretable TSK fuzzy models.

Keywords: TSK Fuzzy Models, Multi-objective Evolutionary Algorithms, RBF Neural Networks.

1 Introduction.

Evolutionary Algorithms (EA) [4] have been successfully applied to learn fuzzy models [8]. EAs have been also combined with other techniques like fuzzy clustering [5] and neural networks [12]. This has resulted in many complex algorithms and, as recognized in [17] and [15], often interpretability of the resulting rule base is not considered to be of importance. In such cases, the fuzzy model becomes a black-box, and one can question the rationale for applying fuzzy modeling instead of other techniques.

In the other hand, EAs have been recognized as appropriate techniques for multi-objective optimization because they perform a search for multiple solutions in parallel [2, 3]. Current evolutionary approaches for multi-objective optimization consist of multi-objective EAs based on the Pareto optimality notion, in which all objective are optimized simultaneously to find multiple non-dominated solutions in a single run of the EA. The decision maker can then choose the most appropriate solution according to the current decision environment at the end of the EA run. Moreover, if the decision environment changes, it is not always necessary to run the EA again. Another solution may be chosen out of the set of non-dominated solutions that has already been obtained.

The multi-objective evolutionary approach can be considered from the fuzzy modeling perspective [7]. Current research lines in fuzzy modeling mostly tackle improving accuracy in descriptive models, and improving interpretability in approximative models [1]. This paper deals with the second issue approaching the problem by means of multi-objective optimization in which accuracy and interpretability criteria are simultaneously considered.

In this paper, we propose a neuro-evolutionary multi-objective optimization approach to generate TSK fuzzy models considering accuracy and interpretability criteria. Section 2 describes the TSK type rule-based fuzzy model, and criteria taken into account for fuzzy modeling. A multi-

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 1129–1136
Torremolinos (Málaga), June 22–27, 2008

objective constraind optimization model is proposed. Section 3 shows the main components of the three multi-objective neuro-evolutionary algorithms used in this paper. Section 4 shows the experiments performed and the results obtained for a standard test problem. Finally, section 5 concludes the paper.

2 Improving Interpretability in TSK Fuzzy Models

2.1 Fuzzy Models Identification

We consider Takagi-Sugeno-Kang (TSK) type rule-based models [16] where rule consequents are taken to be linear functions of the inputs. The rules have, therefore, the following expression:

$$R_i: \ \textbf{If } x_1 \text{ is } A_{i1} \textbf{ and } \ldots \textbf{ and } x_n \text{ is } A_{in} \\ \textbf{then } y_i = \theta_{i1}x_1 + \ldots + \theta_{in}x_n + \theta_{i(n+1)}$$

where $i = 1, \ldots, M$, M is the number of rules, $\boldsymbol{x} = (x_1, \ldots, x_n)$, $x_i \in [l_i, u_i] \subset \Re$, is the input vector, $\theta_{ij} \in [l, u] \subset \Re$ are the consequent parameters, y_i is the output of the ith rule, A_{ij} are fuzzy sets defined in the antecedent space by membership functions $\mu_{A_{ij}} : \mathcal{X}_j \rightarrow [0, 1]$, being $\mathcal{X}_j$ the domain of the input variable x_j.

The total output of the model is computed by aggregating the individual contributions of each rule:

$$y = \frac{\sum_{i=1}^{M} \mu_i(\boldsymbol{x})\, y_i}{\sum_{i=1}^{M} \mu_i(\boldsymbol{x})}$$

where $\mu_i(\boldsymbol{x})$ is the normalized firing strength of the ith rule:

$$\mu_i(\boldsymbol{x}) = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j)$$

Each fuzzy set A_{ij} is described by a gaussian membership function:

$$\mu_{A_{ij}}(x_j) = \exp\left[-\frac{1}{2}\left(\frac{x_j - c_{ij}}{\sigma_{ij}}\right)^2\right]$$

where $c_{ij} \in [l_j, u_j]$ is the center, and $\sigma_{ij} > 0$ is the variance.

This fuzzy model can be defined by a radial basis function neural network. The number of neurons in the hidden layer of an RBF neural network is equal to the number of rules in the fuzzy model. The firing strength of the ith neuron in the hidden layer matches the firing strength of the ith rule in the fuzzy model. We apply a gaussian membership function defined by two parameters, (c, σ): the center c and variance σ. Therefore, each neuron in the hidden layer has these two parameters that define its firing strength value.

The neurons in the output layer perform the computations for the first order linear function described in the consequents of the fuzzy model, therefore, the ith neuron of the output layer has the parameters $\boldsymbol{\theta}_i = \left(\theta_{i1}, \ldots, \theta_{i(n+1)}\right)$ that correspond to the linear function defined in the ith rule of the fuzzy model.

2.2 Criteria for Fuzzy Modeling

We consider three main criteria: accuracy, transparency, and compactness. It is necessary to define quantitative measures for these criteria by means of appropriate objective functions which define the complete fuzzy model identification.

Accuracy. The accuracy of a model can be measured with the *mean squared error*:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y_k - t_k)^2 \tag{1}$$

where y_k is the model output for the kth input vector, t_k is the desired output for the kth input vector, and N is the number of data samples.

Transparency. For transparency, there are many possible measures, however we consider one of the most used, the *similarity* [14]. The similarity S among distinct fuzzy sets in each variable can be expressed as follows:

$$S = \max_{\substack{i=1,\ldots,M \\ j=1,\ldots,n \\ k=1,\ldots,M \\ A_{ij} \neq A_{kj}}} S(A_{ij}, A_{kj}) \quad (2)$$

Similarity between two different fuzzy sets A and B can be measured using different criteria. In our case we use the following measure:

$$S(A,B) = \max\left\{\frac{|A \cap B|}{|A|}, \frac{|A \cap B|}{|B|}\right\} \quad (3)$$

Common characteristics
Pittsburgh approach, real-coded representation.
Trainning of the RBF network consequents.
Constraint-handling technique.
Variation operators.
Rule-set simplification technique.
Elitist generational replacement strategy.
Specific characteristics
MONEA: Preselection over 10 children, steady-state replacement ($n = 2$).
ENORA: Non-dominated radial slots sorting.
NSGA-II: Non-dominated crowded sorting.

Table 1: Common and specific characteristics of MONEA, ENORA and NSGA-II.

Compactness. Measures for compactness are the number of rules (M), and the number of different fuzzy sets (L) of the fuzzy model. It is assumed that models with a small number of rules and fuzzy sets are compact.

2.3 An Optimization Model for Fuzzy Modeling

According to the previous remarks, we propose the following multi-objective constrained optimization model:

$$\begin{array}{ll} Minimize & f_1 = MSE \\ Minimize & f_2 = M \\ Subject\ to & S < g_s \end{array} \quad (4)$$

where $g_s \in [0, 1]$ is a threshold for similarity defined by the decision maker (we use $g_s = 0.25$).

An "a posteriori" articulation of preferences applied to the non-dominated solutions of the problem is used to obtain the final compromise solution.

3 Multi-objective Neuro-Evolutionary Algorithms

We propose an hybrid learning system to find multiple Pareto-optimal solutions simultaneously, considering accuracy, transparency and compactness criteria. We study different multi-objective evolutionary algorithms to evolve the structure and parameters of TSK-type rule sets, together with gradient-based learning to train rule consequents. Additionally, a rule set simplification operator is used to encourage rule base transparency and compactness. This method may be applied to a wide variety of classification and control problems.

Considering the multi-objective constrained optimization model (4), we use three Pareto-based multi-objective evolutionary algorithms: MONEA, ENORA and NSGA-II. MONEA and ENORA are algorithms proposed by authors in [6], and [13] respectively, while NSGA-II is the well-known MOEA proposed by Deb in [3].

Table 1 summarizes common and specific characteristics of the algorithms MONEA, NSGA-II and ENORA.

3.1 Representation of Solutions

The EAs have a variable-length, real-coded representation using a Pittsburgh approach. Each individual of a population contains a variable number of rules between 1 and max, where max is defined by a decision maker. Fuzzy numbers in the antecedents and parameters in the consequent are coded by floating-point numbers c_{ij}, σ_{ij} and θ_{ij}.

3.2 Initial Population

The population is initialized by generating individuals with different numbers of rules. Each individual is generated randomly with a uniform distribution within the boundaries of the search space, defined by the learning data and trained with the gradient technique described in 3.3.

3.3 Training of the RBF Neural Networks

In RBF neural networks, each neuron in the hidden layer can be associated with a fuzzy rule, therefore RBF neural networks are suitable to describe fuzzy models. The RBF neural networks associated with the fuzzy models can be trained with a gradient method to obtain more accuracy. However, in order to maintain the transparency and compactness of the fuzzy sets, only the consequent parameters are trained. The training algorithm incrementally updates the parameters based on the currently presented training pattern. The network parameters are updated by applying the gradient descent method to the MSE error function. The error function for the ith training pattern is given by the MSE function error defined in equation (1).

3.4 Constraint-Handling

The EAs use the following constraint-handling rule proposed in [9]. This rule considers that an individual I is better than an individual J if any of the following conditions is true:

- I is feasible and J is not.
- I and J are both unfeasible, but $S_I < S_J$. (S_I and S_J are similarity of I and J)
- I and J are feasible and I dominates J.

3.5 Variation Operators

In order to achieve an appropriate exploitation and exploration of the potential solutions in the search space, variation operators working in the different levels of the individuals are necessary. In this way, we consider three levels of variation operators: rule set level, rule level and parameter level. Table 2 summarizes the variation operators.

3.6 Rule Set Simplification Technique

Automated approaches to fuzzy modeling often introduce redundancy in terms of several similar fuzzy sets and fuzzy rules that describe almost the same region in the domain of some variable. According to some similarity measure, two similar fuzzy sets can be merged or separated. The merging-separation process is repeated until fuzzy sets for each model variable are not similar. This simplification may results in several identical rules, which must be removed from the rule set. The proposed algorithm is the following:

1. While there be i, j, k such that $S(A_{ij}, A_{kj}) > \eta_2$

    If $S(A_{ij}, A_{kj}) > \eta_1$ then merge A_{ij} and A_{kj}

    in other case, split A_{ij} and A_{kj}

2. While there be i, k such that the antecedentes of rules R_i and R_k are the same

    Substitute the consequent of R_i by the average of the consequents of R_i and R_k and eliminate R_k

Similarity between two fuzzy sets, $S(A, B)$ is measure using the expression in equation (3). The values η_1 and η_2 are the threshold to perform the merging or the separation and must be $0 < \eta_2 < \eta_1 < 1$. We use the values $\eta_1 = 0.9$ and $\eta_2 = 0.6$.

4 Experiments and Results.

We consider the second order non-linear plant studied by Wang and Yen in [18, 19]:

$$
\begin{array}{l}
y(k) = g(y(k-1), y(k-2)) + u(k) \\
with \\
g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1+y^2(k-1)+y^2(k-2)}
\end{array}
$$

The objective is the approximation of the non-linear component of the plant $g(y(k-1), y(k-2))$ using a fuzzy model. 200 trainning values and 200 evaluation values are obtained starting at the initial state $(0, 0)$ with a random input signal $u(k)$ uniformly distributed in the interval $[-1.5, 1.5]$.

MONEA, ENORA and NSGA-II are executed 100 times for 10000 evaluation, with a population of 100 individuals, cross and mutation probabilities of 0.8 and 0.4. The different variation operators are applied with

Rule Set Level Variation Operators	
Rule Set Crossover.	Exchanges a random number of rules.
Rule Set Increase Crossover.	Adds a random number of rules of the parents to children.
Rule Set Mutation.	Adds or deletes, with the same probability, a rule.
Rule Level Variation Operators	
Rule Arithmetic Crossover.	Performs an arithmetic crossover of two random rules.
Rule Uniform Crossover.	Performs an uniform crossover of two random rules.
Parameter Level Variation Operators.	
Arithmetic and Uniform Crossover Uniform and Non-Uniform Mutation Small Mutation	These operators have been studied and described by other authors [4]. The small mutation produces a small change in the individual and it is suitable for fine tuning of the real parameters.

Table 2: Variation Operators.

Ref.	M	L	*Train* MSE	*Eval* MSE
[18]	40 (initial)	40	$3.3 \cdot 10^{-4}$	$6.9 \cdot 10^{-4}$
	28 (optimized)	28	$3.3 \cdot 10^{-4}$	$6.0 \cdot 10^{-4}$
[19]	36 (initial)	12	$1.9 \cdot 10^{-6}$	$2.9 \cdot 10^{-3}$
	24 (optimized)	12	$2.0 \cdot 10^{-6}$	$6.4 \cdot 10^{-4}$
[11]	7 (initial)	14	$1.8 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
	5 (optimized)	5	$5.0 \cdot 10^{-4}$	$4.2 \cdot 10^{-4}$

Table 3: Fuzzy models for the second order non-linear plant reported in literature.

M	L	Train MSE	Eval MSE	S
MONEA				
1	2	0.041882	0.043821	0.000000
2	3	0.004779	0.005533	0.249887
3	4	0.002262	0.002749	0.232016
4	4	0.000216	0.000248	0.249021
ENORA				
1	2	0.041882	0.043821	0.000000
2	3	0.004951	0.005722	0.242090
3	4	0.001906	0.002411	0.249391
4	4	0.000161	0.000194	0.249746
NSGA-II				
1	2	0.041882	0.043821	0.000000
2	3	0.004870	0.005639	0.249998
3	4	0.001885	0.002343	0.249999
4	4	0.000249	0.000314	0.250000

Table 4: Non-dominated solutions (best results over 100 runs) obtained in this paper for the second order non-linear plant.

equal probability. We can compare our results with the results obtained by other approaches proposed in [18], [19] and [11] which are shown in Table 3. Table 4 shows the best non-dominated solutions in the last population over 100 runs. Solutions with 4 rules are chosen which are shown in Figure 1.

To compare the algorithms, we use the hypervolume indicator (ν) which calculates the fraction of the objective space which is non-dominated by any of the solutions obtained by the algorithm [3, 10, 20]. Algorithms were executed 100 times, so we have obtained a 100 sample for each algorithm.

The statistics showed in Table 6 indicate that MONEA and ENORA obtain lower localization values than NSGA-II while NSGA-II obtains the greatest dispersion values. Finally, the 90% confidence intervals for the mean obtained with t-test show that ENORA obtains lower values than MONEA and this obtains lower than NSGA-II. That is, the approximation sets obtained by ENORA are preferable to those of MONEA and those of NSGA-II under hypervolume indicator ν. t-test is robust with no normal samples which are greater than 30 individuals, so the results are significant and we can conclude that there is statistical difference between the hypervolume values obtained by the algorithms. The Boxplots showed in Figure 2 confirm the above conclusions.

Taking all the above, we can conclude that the hypervolume values obtained with ENORA are significantly better than the values obtained with MONEA and NSGA-II. The statistical analysis shows, therefore, that for the kind of multi-objective problems we are considering, Pareto search based on the space search partition in linear slots is most efficient than general search strategies exclusively based on diversity functions, as in NSGA-II.

R_1	**If** y_{k-1} is LOW **and** y_{k-2} is LOW	**then** $g = 0.4327y_{k-1} + 0.0007y_{k-2} - 0.2008$
R_2	**If** y_{k-1} is LOW **and** y_{k-2} is HIGH	**then** $g = -0.4545y_{k-1} - 0.0131y_{k-2} + 0.2368$
R_3	**If** y_{k-1} is HIGH **and** y_{k-2} is LOW	**then** $g = -0.3968y_{k-1} - 0.0044y_{k-2} + 0.1859$
R_4	**If** y_{k-1} is HIGH **and** y_{k-2} is HIGH	**then** $g = 0.43645y_{k-1} - 0.0052y_{k-2} - 0.2110$

y_{k-1}	$LOW = (-1.5966, 2.0662)$	$HIGH = (1.7679, 2.6992)$
y_{k-2}	$LOW = (-1.7940, 3.1816)$	$HIGH = (1.5271, 2.1492)$

Table 5: Fuzzy model with 4 rules for the non-linear dynamic plant obtained by ENORA.

	MONEA	**ENORA**	**NSGA-II**
Minimum	0.3444	0.3337	0.3318
Maximum	0.4944	0.4591	0.9590
Mean	0.3919	0.3799	0.5333
S.D.	0.0378	0.0334	0.1430
C.I. Low	0.3856	0.3743	0.5096
C.I. High	0.3982	0.3854	0.5571

S.D = Standard Deviation of Mean
C.I. = Confidence Interval for the Mean (90%)

Table 6: Statistics for the hypervolume obtained with 100 runs of MONEA, ENORA and NSGA-II for the second order non-linear plant.

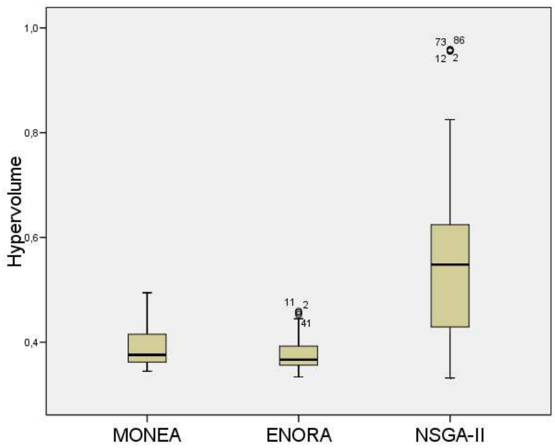

Figure 2: Boxplots for the hypervolume obtained with 100 runs of MONEA, ENORA and NSGA-II for the second order non-linear plant.

5 Conclusions.

This paper remarks on some results in the combination of Pareto-based multi-objective evolutionary algorithms, neural networks and fuzzy modeling. A multi-objective constrained optimization model is proposed in which criteria such as accuracy, transparency and compactness have been taken into account. Three multi-objective evolutionary algorithms (MONEA, ENORA and NSGA-II) have been implemented in combination with neural network based and rule simplification techniques. The results obtained improve on other more complex techniques reported in literature, with the advantage that the proposed technique identifies a set of alternative solutions. Statistical tests have been performed over the hypervolume quality indicator values to compare the algorithms and it has shown that, for the non linear plant problem, ENORA obtains better results than MONEA and NSGA-II algorithms.

Acknowledgements

Research supported in part by MEC and FEDER under project DPI2007-65501 and PET2006_0406.

References

[1] Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (2003) Interpretability improvements to find the balance interpretability-accuracy in fuzzy modeling: an overview, in J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Studies in Fuzziness and Soft Computing, Springer, pp. 3-22.

[2] Coello, C.A., Veldhuizen, D.V., Lamont, G.B. (2002). Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic/Plenum publishers, New York.

[3] Deb, K. (2001). Multi-Objective Optimization using Evolutionary Algorithms. John Wiley and Sons, LTD.

[4] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

[5] Gómez-Skarmeta, A.F., Jiménez, F. (1999). Fuzzy modeling with hibrid systems. Fuzzy Sets and Systems, 104:199-208.

[6] Gómez-Skarmeta, A.F., Jiménez, F., Sánchez, G. (2007). Improving Interpretability in Approximative Fuzzy Models via Multiobjective Evolutionary Algorithms. International Journal of Intelligent Systems, 22:943-969, 2.007

[7] Ishibuchi, H., Murata, T., Trksen, I. (1997). *Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems.* Fuzzy Sets and Systems, 89:135-150.

[8] Ishibuchi, H., Nakashima, T., Murata, T. (1999). Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. IEEE Transactions on Systems, Man, and Cubernetics - Part B: Cybernetics, 29(5):601-618.

[9] Jiménez, F., Gómez-Skarmeta, A.F., Sánchez, G., Deb, K. (2002). An evolutionary algorithm for constrained multiobjective optimization, in: Proceedings IEEE World Congress on Evolutionary Computation.

[10] Laumanns, M., Zitzler, E., and Thiele, L. (2001). On the Effects of Archiving, Elitism, and Density Based Selection in Evolutionary Multi-objective Optimization. Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001) E. Zitzler et al. (Eds.), 181-196.

[11] Roubos, J.A., Setnes, M. (2000). Compact fuzzy models through complexity reduction and evolutionary optimization. In Proceedings of FUZZ-IEEE-2000, 762-767, San Antonio, Texas, USA.

[12] Russo., M. (1998). FuGeNeSys - a fuzzy genetic neural system for fuzzy modeling. IEEE Transactions on Fuzzy Systems, 6(3):373-388.

[13] Sánchez, G., Jiménez, J., Vasant, P. (2007). Fuzzy Optimization with Multi-Objective Evolutionary Algorithms: a Case Study. IEEE Symposium of Computational Intelligence in Multicriteria Decision Making (MCDM). Honolulu, Hawaii

[14] Setnes, M. (1995). Fuzzy Rule Base Simplification Using Similarity Measures. M.Sc. thesis, Delft University of Technology, Delft, the Netherlands.

[15] Setnes, M., Babuska, R., Verbruggen, H. B. (1998). Rule-based modeling: Precision and transparency. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications & Reviews, 28:165-169.

[16] Takagi, T., Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, 15:116-132.

[17] Valente de Oliveira, J. (1999). Semantic constraints for membership function optimization. IEEE Transactions on Fuzzy Systems, 19(1):128-138.

[18] Wang, L., Yen, J. (1999). Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter. Fuzzy Sets and Systems, 101:353-362.

[19] Yen, J., Wang, L. (1998). Application of statistical information criteria for optimal fuzzy model construction. IEEE Transactions on Fuzzy Systems, 6(3):362-371.

[20] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V. (2003) Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, 7(2):117-132.

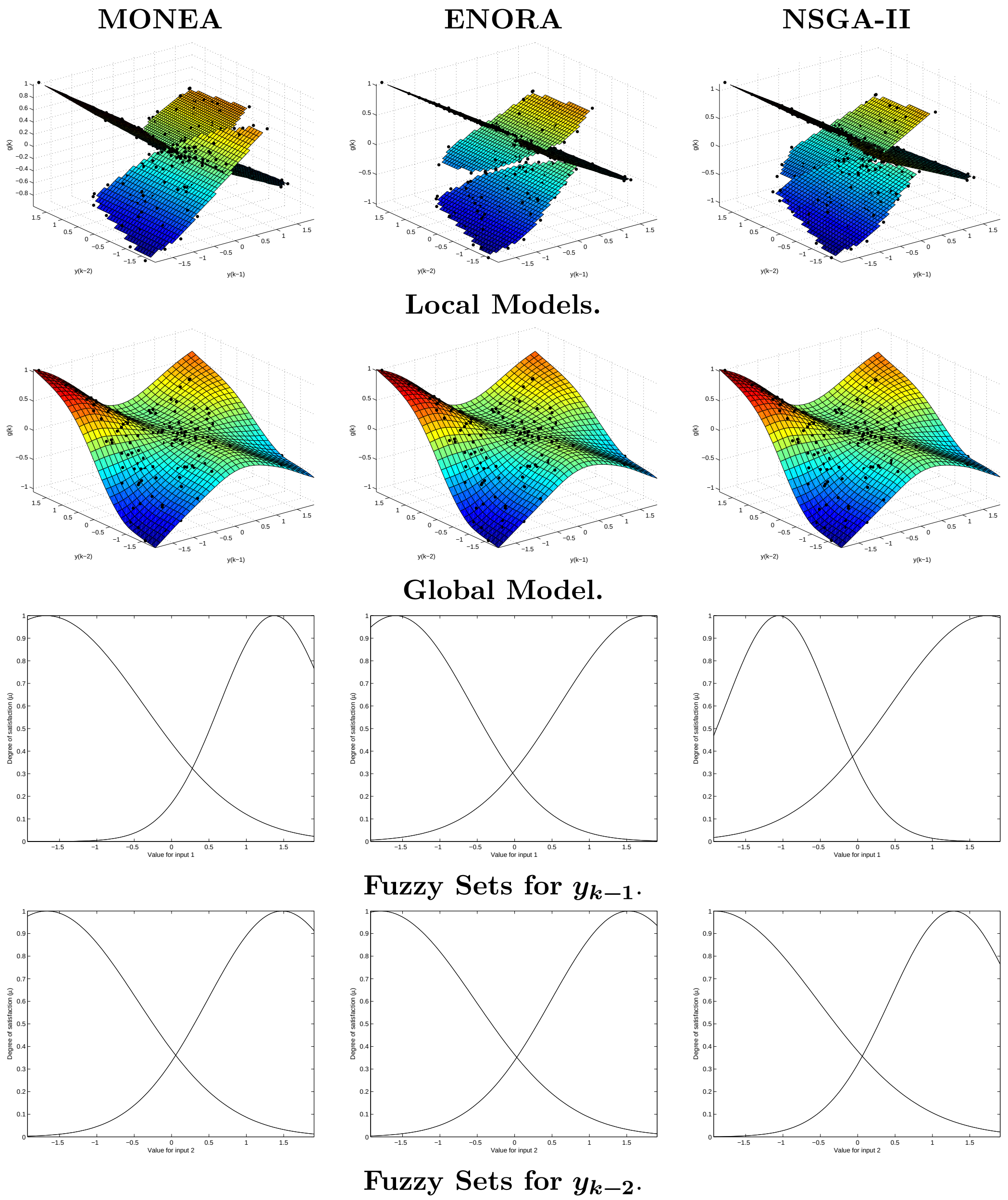

Figure 1: Solutions with 4 rules obtained in this paper for the second order non-linear plant.