14

Improving Interpretability of Fuzzy Models Using Multi-Objective Neuro-Evolutionary Algorithms

Gracia Sánchez Carpena, José Francisco Sánchez Ruiz,
José Manuel Alcaraz Muñoz and Fernando Jiménez
University of Murcia
Spain

1. Introduction

Evolutionary Algorithms (EA) (Goldberg, 1989) have been successfully applied to learn fuzzy models (Ishibuchi et al., 1999). EAs have been also combined with other techniques like fuzzy clustering (Gómez-Skarmeta & Jiménez 1999) and neural networks (Russo, 1998). This has resulted in many complex algorithms and, as recognized in (Valente de Oliveira, 1999) and in (Setnes et al., 1998), 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.

On the other hand, EAs have been recognized as appropriate techniques for multi-objective optimization because they perform a search for multiple solutions in parallel (Coello et al., 2002) (Deb, 2001). Current evolutionary approaches for multi-objective optimization consist of multi-objective EAs based on the Pareto optimality notion, in which all objective are simultaneously optimized 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 also be considered from the fuzzy modeling perspective (Ishibuchi et al., 1997). Current research lines in fuzzy modeling mostly tackle improving accuracy in descriptive models, and improving interpretability in approximative models (Casillas et al., 2003). This chapter 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 chapter, we propose a multi-objective neuro-evolutionary optimization approach to generate TSK fuzzy models considering accuracy and interpretability criteria. This approach allows a linguistic approximation of the fuzzy models. The rule-based fuzzy model and criteria taken into account for fuzzy modeling are explained in the text, where a multi-objective constrained optimization model is proposed.

Two different multi-objective evolutionary algorithms (MONEA, ENORA-II) are proposed and compared with the well-known algotithm NSGA-II (Deb et al., 2000) for the

Source: Advances in Evolutionary Algorithms, Book edited by: Witold Kosiński, ISBN 978-953-7619-11-4, pp. 468, November 2008, I-Tech Education and Publishing, Vienna, Austria

approximation of a non linear system (studied by Wang & Yen, 1998, 1999). The results of the experiments performed for this standard test problem show a real ability and effectiveness of the proposed approach to find accurate and interpretable TSK fuzzy models.

2. Improving interpretability in TSK fuzzy models

2.1 Fuzzy models identification

We consider Takagi-Sugeno-Kang (TSK) type rule-based models (Takagi & Sugeno, 1985) where rule consequents are taken to be linear functions of the inputs. The rules have, therefore, the following expression:

$$\begin{aligned} R_i:\ & \mathit{If}\ x_1\ \mathit{is}\ A_{i1}\ \mathit{and}\ \ldots\ \mathit{and}\ x_n\ \mathit{is}\ A_{in} \\ & \mathit{then}\ y_i = \theta_{i1}x_1 + \ldots + \theta_{in}x_n + \theta_{i(n+1)} \end{aligned}$$

where:

$i = 1,\ldots,M$, M is the number of rules,

$\mathbf{x} = (x_1,\ldots,x_n)$, $x_j \in [l_j\ u_j] \subset \Re$ is the input vector ($j = 1,\ldots,n$),

$\theta_{ij} \in [l, u] \subset \Re$ are the consequent parameters ($j = 1,\ldots,n+1$),

y_i is the output of the ith rule, and

A_{ij} are fuzzy sets defined in the antecedent space by membership functions $\mu_{A_{ij}} : \mathsf{X}_j \rightarrow [0,1]$,

being X_j the domain of the input variable x_j ($j = 1,\ldots,n$).

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

$$y = \frac{\sum_{i=1}^{M} \mu_i(\mathbf{x}) f_i(\mathbf{x})}{\sum_{i=1}^{M} \mu_i(\mathbf{x})} \tag{1}$$

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

$$\mu_i(\mathbf{x}) = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j) \tag{2}$$

and $f_i(\mathbf{x})$ is the function defined in the consequent of the ith rule:

$$f_i(\mathbf{x}) = \theta_{i1}x_1 + \ldots + \theta_{in}x_n + \theta_{i(n+1)} \tag{3}$$

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

$$\mu_{A_{ij}}(x_j) = \exp\left[-\frac{1}{2}\left(\frac{x_j - c_{ij}}{\sigma_{ij}} \right)^2 \right] \tag{4}$$

where:

$c_{ij} \in \lfloor l_j, u_j \rfloor$ is the center,

$\sigma_{ij} > 0$ is the variance,

$i = 1, \ldots, M$ and

$j = 1, \ldots, n$.

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 *i*th neuron in the hidden layer matches the firing strength of the *i*th rule in the fuzzy model. We apply a symmetric gaussian membership function defined by two parameters, the center *c* and the 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 *i*th 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 *i*th rule of the fuzzy model.

2.2 Criteria for fuzzy modeling

We consider three main criteria: (i) accuracy, (ii) transparency, and (iii) 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*:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}\left(y_k - t_k\right)^2 \tag{5}$$

where:

y_k is the model output for the *k*th input vector,

t_k is the desired output for the *k*th input vector, and

N is the number of data samples.

Transparency.

For the second criterion, transparency, there are many possible measures, however we consider one of the most used, the *similarity* (Setnes, 1995). The similarity S among distinct fuzzy sets in each variable can be expressed as follows:

$$S = \max_{\substack{i=1,\ldots,M \\ j=1,\ldots,n \\ k=1,\ldots,M \\ A_{ij} \neq A_{kj}}} S\left(A_{ij}, A_{kj}\right) \tag{6}$$

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

$$S(A,B) = \max\left\{\frac{|A \cap B|}{|A|}, \frac{|A \cap B|}{|B|}\right\} \tag{7}$$

The value of S is, therefore, an aggregated similarity measure for the fuzzy rule-based model with the objective to minimize the maximum similarity between the fuzzy sets in each input domain.

Compactness.

Finally, measures for the third criterion, the 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.

Table 1 summarizes the three criteria considered for the fuzzy models and the measures defined for each criterion.

Criteria	Measures
Accuracy	MSE
Transparency	S
Compactness	M, L

Table 1. Criteria for the fuzzy models and their measures

2.3 An optimization model for fuzzy modeling

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

$$\begin{aligned} &Minimize \quad f_1 = MSE \\ &Minimize \quad f_2 = M \\ &Subject\ to \quad g_1 : S - g_s \leq 0 \end{aligned} \tag{8}$$

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 a 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 (8), we use three Pareto-based multi-objective evolutionary algorithms: MONEA, ENORA-II and NSGA-II. MONEA and ENORA-II are algorithms proposed by authors in (Gómez-Skarmeta et al., 2007), and (Sánchez et al., 2007) respectively, while NSGA-II is the well-known multi-objective EA proposed by Deb in (Deb, 2001).

The main common characteristics are the following:

- The algorithms are Pareto-based multi-objective EAs for fuzzy modeling; that is, they have been designed to find, in a single run, multiple non-dominated solutions according to the Pareto decision strategy. There is no dependence between the objective functions and the design of the EAs; thus, any objective function can easily be incorporated.

- Constraints with respect to the fuzzy model structure are satisfied by incorporating specific knowledge about the problem. The initialization procedure and variation operators always generate individuals that satisfy these constraints.
- The EAs have a variable-length, real-coded representation. 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 the parameters in the consequent are coded by floating-point numbers.
- The initial population is generated randomly with a uniform distribution within the boundaries of the search space, defined by the learning data and model constraints.
- The EAs search among rule sets treated with the technique described in Section 3.6 and trained as defined in Section 3.3, which is an added ad hoc technique for transparency, compactness, and accuracy.

Table 2 summarizes common and specific characteristics of the algorithms MONEA, NSGA-II and ENORA-II.

Common characteristics	
Pittsburgh approach, real-coded representation.	
Training 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-II:	Non-dominated radial slots sorting.
NSGA-II:	Non-dominated crowded sorting.

Table 2. Common and specific characteristics of MONEA, ENORA-II and NSGA-II.

3.1 Representation of solutions

The EAs have a variable-length, real-coded representation using a Pittsburgh approach. An individual I for this problem is a rule set of M (between 1 and *max*, where *max* is defined by a decision maker) rules defined by the weights of the RBF neural network. With n input variables, we have for each individual the following parameters:

- Parameters of the fuzzy sets A_{ij} :

$$\text{centers } c_{ij} \text{ and variances } \sigma_{ij},\ i=1,\ldots,M,\ j=1,\ldots,n$$

- Coefficients for the linear function of the consequents:

$$\theta_{ij},\ i=1,\ldots,M,\ j=1,\ldots,n+1$$

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 subsection 3.3.

An individual with M rules is generated with the following procedure:

1. For each fuzzy set A_{ij} ($i=1,\ldots,M$, $j=1,\ldots,n$), generate two real values: c_{ij} in the interval $\lfloor l_j, u_j \rfloor$ and the parameter of the gaussian fuzzy set , σ_{ij} .
2. Parameters θ_{ij} ($i=1,\ldots,M$, $j=1,\ldots,n+1$) are random real values in the interval $[l,u]$.
3. The individual is treated with the technique to improve transparency and compactness describe in subsection 3.6.
4. The individual is trained using the gradient technique described in subsection 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 *i*th training pattern is given by the *MSE* function error defined in equation (5). The updating rule is the following:

$$\theta_{ij} \leftarrow \theta_{ij} + \eta \Delta\theta_{ij} = \theta_{ij} - \eta \frac{\partial MSE}{\partial \theta_{ij}}$$

where:
$i=1,\ldots,M$,
$j=1,\ldots,n+1$, and
η is the learning rate.

This rule is applied during a number of iterations (epochs). We use a value $\eta = 0.01$ and a number of 10 epochs. The negative gradients of MSE with respect to each parameter are calculated in the following way:

$$\Delta\theta_{ij} = -\frac{\partial MSE}{\partial \theta_{ij}} = \left(t_k - y_k\right)\frac{1}{z}\mu_i\left(\mathrm{x}\right)x_j,\ j=1,\ldots,n$$

$$\Delta\theta_{i(n+1)} = -\frac{\partial MSE}{\partial \theta_{i(n+1)}} = \left(t_k - y_k\right)\frac{1}{z}\mu_i(\mathbf{x})$$

where:
$i=1,\ldots,M$,

$\mu_i(\mathbf{x})$ is the firing strength for the *i*th rule defined in equation (2), and

$z = \sum_{i=1}^{M} \mu_i(\mathbf{x})$.

3.4 Constraint-handling

The EAs use the following constraint handling rule proposed in (Jiménez et al., 2002). 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

As already said, an individual is a set of M rules. A rule is a collection of n fuzzy numbers (antecedent) plus $n+1$ real parameters (consequent), and a fuzzy number is composed of two real numbers. 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.

Rule Set Level Variation Operators

Rule Set Crossover

This operator exchanges a random number of rules. Given two parents $I_1 = \left(R_1^1 \ldots R_{M_1}^1\right)$and $I_2 = \left(R_1^2 \ldots R_{M_2}^2\right)$generate two children:

$$I_3 = \left(R_1^1 \ldots R_a^1 R_1^2 \ldots R_b^2\right)$$

$$I_4 = \left(R_{a+1}^1 \ldots R_{M_1}^1 R_{b+1}^2 \ldots R_{M_2}^2\right)$$

with:

$$a = round(\alpha M_1)$$

$$b = round((1-\alpha)M_2)$$

where α is a random real number in $[0,1]$. The number of rules of the children is therefore in $[M_1, M_2]$.

Rule Set Increase Crossover

This operator increases the number of each child rules adding a random number of rules of the other parent. Given two parents $I_1 = \left(R_1^1 \ldots R_{M_1}^1\right)$ and $I_2 = \left(R_1^2 \ldots R_{M_2}^2\right)$ generate two children:

$$I_3 = \left(R_1^1 \ldots R_{M_1}^1 R_1^2 \ldots R_a^2\right)$$

$$I_4 = \left(R_1^2 \ldots R_{M_2}^2 R_1^1 \ldots R_b^1\right)$$

with:

$$a = \min\{\max - M_1, M_2\}$$

$$b = \min\{\max - M_2, M_1\}$$

Rule Set Mutation

This operator adds or deletes, with the same probability, a rule. Given an individual $I = (R_1 \ldots R_M)$ generates other individual I' :

$$\begin{array}{ll} I' = (R_1, \ldots R_{a-1} R_{a+1} \ldots R_M), & \textit{if } \alpha \le 0.5 \\ I' = (R_1, \ldots R_M R_{M+1}), & \textit{in other case} \end{array}$$

where:

α is a random real number in $[0,1]$,

a a random index in $[1, M]$, and

R_{M+1} a new random rule generated with the initialization procedure.

Rule Level Variation Operators

Rule Arithmetic Crossover

It performs an arithmetic crossover of two random rules. Given two parents $I_1 = \left(R_1^1 \ldots R_{M_1}^1\right)$ and $I_2 = \left(R_1^2 \ldots R_{M_2}^2\right)$ generates two children:

$$I_3 = \left(R_1^1 \ldots R_i^3 \ldots R_{M_1}^1\right)$$

$$I_4 = \left(R_1^2 \ldots R_j^4 \ldots R_{M_2}^2\right)$$

with R_i^3 and R_j^4 obtained by arithmetic crossover:

$$R_i^3 = \alpha R_i^1 + (1 - \alpha) R_j^2$$

$$R_j^4 = \alpha R_j^2 + (1 - \alpha) R_i^1$$

where:

α is a random real number in $[0,1]$,

i, j are random index in $[1, M_1]$ and $[1, M_2]$, respectively.

The product αR_i is defined as follows:

$$\alpha R_i : \quad \alpha A_{i1} \ldots \alpha A_{in} \quad \alpha\theta_{i1} \ldots \alpha\theta_{in} \; \alpha\theta_{i(n+1)}$$

The fuzzy set αA_{ij} is defined as follows:

$$\alpha A_{ij} = \{\alpha a_{ij}, \alpha b_{ij}, \alpha c_{ij}, \alpha d_{ij}\}$$

Rule Uniform Crossover

It performs a uniform crossover of two random rules. Given two parents $I_1 = \left(R_1^1 \dots R_{M_1}^1\right)$ and $I_2 = \left(R_1^2 \dots R_{M_2}^2\right)$ generates two children:

$$I_3 = \left(R_1^1 \dots R_i^3 \dots R_{M_1}^1\right)$$

$$I_4 = \left(R_1^2 \dots R_j^4 \dots R_{M_2}^2\right)$$

where:

R_i^3 and R_j^4 are obtained by uniform crossover,

i, j are random index in $[1, M_1]$ and $[1, M_2]$.

Parameter Level Variation Operators

The operators considered at this level are arithmetic crossover, uniform crossover, non-uniform mutation, uniform mutation and small mutation. These operators excluding the last one have been studied and described by other authors (Goldberg, 1989). The small mutation produces a small change in the individual and it is suitable for fine tuning of the real parameters.

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
                Calculate $C$ as the merging of $A_{ij}$ and $A_{kj}$
                Substitute $A_{ij}$ and $A_{kj}$ by $C$
        in other case
                Split $A_{ij}$ and $A_{kj}$
2 While there be $i, k$ such that the antecedents of rules $R_i$ and $R_k$ are the same
        Calculate a new consequent with the average of the parameters of the consequents
        of $R_i$ and $R_k$
        Substitute the consequent of $R_i$ by the new consequent
        Eliminate $R_k$
```

Similarity between two fuzzy sets, $S(A, B)$, is measured using the expression in equation (7). 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 $\eta_1 = 0.9$ and $\eta_2 = 0.6$)

If $S(A,B) > \eta_1$, fuzzy sets *A* and *B* are merged in a new fuzzy set *C* as follows:

$$c_C = \alpha c_A + (1-\alpha)c_B$$

$$\sigma = \max\{c_C - \min\{c_A - \sigma_A, c_B - \sigma_B\}, \max\{c_A + \sigma_A, c_B + \sigma_B\} - c_C\}$$

where $\alpha \in [0,1]$ determines the influence of A and B in the new fuzzy set C :

$$\alpha = \frac{c_A^r - c_A^l}{c_A^r - c_A^l + c_B^r - c_B^l}$$

If $\eta_2 < S(A,B) < \eta_1$, fuzzy sets A and B are splitted as follows:

$$\begin{array}{ll} \textit{If } \sigma_A < \sigma_B \textit{ then} & \sigma_A \leftarrow \sigma_A(1-\beta) \\ \textit{in other case} & \sigma_B \leftarrow \sigma_B(1-\beta) \end{array}$$

where $\beta \in [0,1]$ indicates the amount of separation between A and B (we use $\beta = 0.1$).

3.7 Algorithm descriptions

In order to describe the algorithms, we consider the following formulation as a general form of the multi-objective constrained optimization model (8):

$$\begin{array}{ll} \textit{Minimize} \quad f_k & k = 1,..,n \\ \textit{Subject to } g_i \le 0 & i = 1,..,m \end{array} \tag{9}$$

Where f_k, g_i are arbitrary functions.

Multi-objective neuro-evolutionary algorithm (MONEA)

The main characteristic of MONEA is that Chromosome selection and replacement are achieved by means of a variant of the Preselection scheme. This technique is, implicitly, a niche formation technique and an elitist strategy. Moreover, an explicit niche formation technique has been added to maintain diversity with respect to the number of rules of the individuals.

Algorithm MONEA

```
1.  t ← 0
2.  Initialize P (t)
4.  while t < T do
5.      parent$_1$,parent$_2$ ← Random selection from P(t)
6.      Generate a new individual best$_1$ ← parent$_1$
7.      Generate a new individual best$_2$ ← parent$_2$
8.      Repeat nChildren times
9.          child$_1$,child$_2$ ← Crossing and Mutation of parent$_1$ and parent$_2$
10.         Improve transparency and compactness in child$_1$ and child$_2$
11.         Train child$_1$and child$_2$ by the gradient technique
12.         For i=1 to 2
```

```
                If $child_i$ is better than $best_i$ and
                (the number of rules of $child_i$ is equal to the number of rules of $parent_i$) or
                (the niche count of $parent_i$ is greater than minNS and the niche count of the
                $child_i$ is smaller than maxNS) then
                    $best_i \leftarrow child_i$
        P (t + 1) $\leftarrow$ P(t) – {$parent_1$, $parent_2$} $\cup$ {$best_1$, $best_2$}
        t $\leftarrow$ t + 1
    end while
```

The preselection scheme is an implicit niche formation technique to maintain diversity in the population because an offspring replaces an individual similar to itself (one of its parents). Implicit niche formation techniques are more appropriate for fuzzy modeling than explicit techniques, such as the sharing function, which can provoke excessive computational time. However, we need an additional mechanism for diversity with respect to the number of rules of the individuals in the population. The added explicit niche formation technique ensures that the number of individuals with *M* rules, for all *M* ∈ *[1, max]*, is greater or equal to *minNS* and smaller or equal to *maxNS*. Moreover, the preselection scheme is also an elitist strategy because the best individual in the population is replaced only by a better one.

The better function

Given two individuals *k* and *l*, *k* is better than *l* if:

- *k* is feasible and *l* is unfeasible, or
- *k* and *l* are unfeasible and $\max_{j=1...m}\{g_j^k\} \leq \max_{j=1...m}\{g_j^l\}$, or (10)
- *k* and *l* are feasible and *k* dominates *l*, or

ENORA-II: An Elitist Pareto-Based Multi-Objective Evolutionary Algorithm

ENORA-II uses a real-coded representation, uniform and arithmetical cross, and uniform and non-uniform mutation. Diversity among individuals is maintained by using an ad-hoc elitist generational replacement technique.

ENORA-II has a population *P* of *N* individuals. The following algorithm shows the pseudocode of ENORA-II.

Algorithm ENORA-II

```
t $\leftarrow$ 0
Initialize P (t)
Evaluate P (t)
while t < T do
    Q (t) $\leftarrow$ Random Selection, Crossing and Mutation of N individuals from P (t)
    Improve transparency and compactness in Q(t)
    Train all individuals in Q(t) by the gradient technique
    Evaluate Q(t);
    P (t + 1) $\leftarrow$ Best individuals from P (t) $\cup$ Q(t);
    t $\leftarrow$ t + 1;
    end while;
return the non dominated individuals from P(t);
```

Given a population *P* of *N* individuals, *N* children are generated by random selection, crossing and mutation. The new population is obtained selecting the *N* best individuals from the union of parents and children.

Better individuals

The better individuals are obtained by using the ranking established by the operator *best*. It assumes that every individual *i* has two attributes:

- a ranking in its slot (r_i), and
- a crowding distance (d_i).

Based on these attributes, an individual *i* is better than an individual *j* if:

- $r_i < r_j$ or
- $r_i = r_j$ and $d_i > d_j$.

Crowding distance

Quantity d_i is a measure of the search space around individual *i* which is not occupied by any other individual in the population. This quantity d_i serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices.

$$d_i = \begin{cases} \infty & , \ \ if\ f_j^i = f_j^{\max}\ or\ f_j^i = f_j^{\min}\ for\,any\ j \\ \sum_{j=1}^{n} \dfrac{f_j^{\sup_j^i} - f_j^{\inf_j^i}}{f_j^{\max} - f_j^{\min}}, & in\ other\ case \end{cases} \tag{11}$$

Where $f_j^{\max} = \max\limits_{i=1\ldots N}\left\{f_j^i\right\}$, $f_j^{\min} = \min\limits_{i=1\ldots N}\left\{f_j^i\right\}$, $f_j^{\sup_j^i}$ is the value of the *j*th objective for the individual higher adjacent in the *j*th objective to individual *i*, and $f_j^{\inf_j^i}$ is the value of the *j*th objective for the individual lower adjacent in the *j*th objective to individual i.

Ranking of individuals in its slot

Individuals are ordered in $\left(\left\lfloor \sqrt[n-1]{N} \right\rfloor + 1\right)^{n-1}$ slots. An individual *i* belongs to slot s_i such that:

$$s_i = \sum_{j=2}^{n} \left\lfloor \frac{f_j^i - f_j^{r\min}}{f_j^{r\max} - f_j^{r\min}} \left\lfloor \sqrt[n-1]{N} \right\rfloor \right\rfloor \left(\left\lfloor \sqrt[n-1]{N} \right\rfloor\right)^{n-j} \tag{12}$$

where $f_j^{r\max}$ and $f_j^{r\min}$ are the maximum and minimum values for the *j*th objective if the objective space is bounded; if it is not, then these are bounding reference points so that $f_j^{r\max} \geq f_j^i$ and $f_j^{r\min} \leq f_j^i$ for any individual *i*.

The ranking inside slots is established as an adjustment of the better function (10): given two individuals *k* and *l* belonging to same slot, ranking of individual *k* is lower than ranking of individual *l* in the slot if:

- *k* is feasible and *l* is unfeasible, or
- *k* and *l* are unfeasible and $\max\limits_{j=1\ldots m}\left\{g_j^k\right\} \leq \max\limits_{j=1\ldots m}\left\{g_j^l\right\}$, or
- *k* and *l* are feasible and *k* dominates *l*, or
- *k* and *l* are feasible and does not dominated each other and $d_k > d_l$.

4. Experiments and results

We consider the second order non-linear plant studied in (Wang & Yen, 1999) and (Yen & Wang, 1998):

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

The objective is the approximation of the non-linear component of the plant $g\big(y(k-1), y(k-2)\big)$ using a fuzzy model. 200 training 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-II and NSGA-II are executed 100 times for 10000 evaluations, with a population of 100 individuals, cross and mutation probabilities of 0.8 and 0.4 respectively. The different variation operators are applied with equal probability. We can compare our results with the results obtained by other approaches proposed in (Wang & Yen, 1999), (Yen & Wang, 1998) and (Roubos & Setnes, 2000) 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 and Table 5.

Reference	M	L	Train *MSE*	Eval *MSE*
Wang & Yen, 1999	40 (initial)	40	3.3 E-4	6.9 E-4
	28 (optimized)	28	3.3 E-4	6.0 E-4
Yen & Wang, 1998	36 (initial)	12	1.9 E-6	2.9 E-3
	24 (optimized)	12	2.0 E-6	6.4 E-4
Roubos & Setnes, 2000	7 (initial)	14	1.8 E-3	1.0 E-3
	5 (optimized)	5	5.0 E-4	4.2 E-4

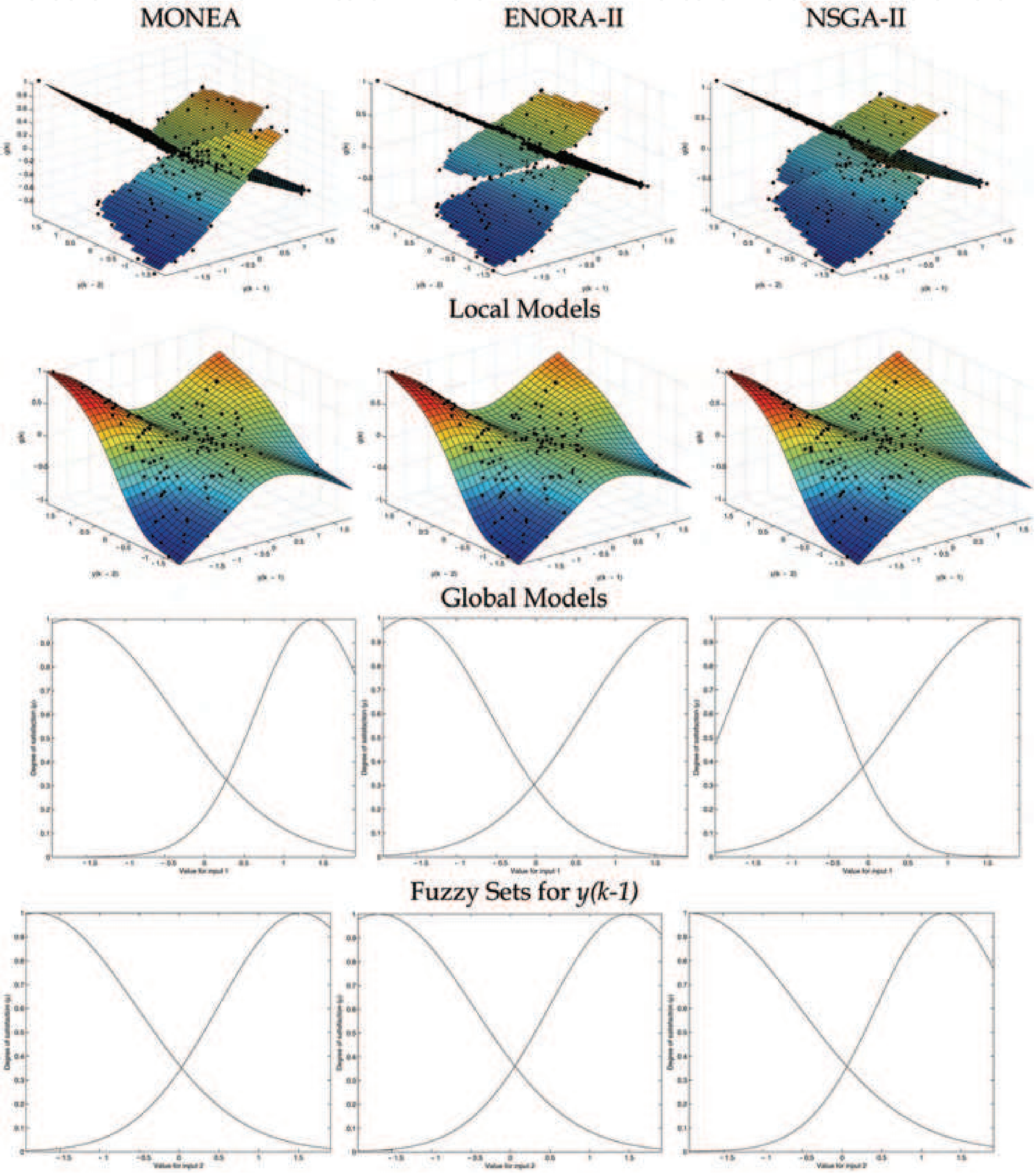
Table 3. Fuzzy models for the second order non-linear plant reported in literature.

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-II		
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.

*R*1	If *y(k-1)* is LOW and *y(k-2)* is LOW	then *g* = 0.4327*y(k-1)* + 0.0007*(k-2)* – 0.2008
*R*2	If *y(k-1)* is LOW and *y(k-2)* is HIGH	then *g* = -0.4545*y(k-1)* – 0.0131*(k-2)* + 0.2368
*R*3	If *y(k-1)* is HIGH and *y(k-2)* is LOW	then *g* = -0.3968*y(k-1)* – 0.0044*(k-2)*+ 0.1859
*R*4	If *y(k-1)* is HIGH and *y(k-2)* is HIGH	then *g* = 0.43645*y(k-1)* – 0.0052*(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-II.


MONEA ENORA-II NSGA-II

Local Models

Global Models

Fuzzy Sets for *y(k-1)*

Fuzzy Sets for *y(k-2)*

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

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 in (Deb, 2001), (Laumans et al., 2001) and (Zitzler et al., 2003). The aim is to minimize the value of ν. This indicator estimates both the distance of solutions to the real Pareto front and the spread. Whenever a set of solutions is preferable to other with respect to weak Pareto dominance, the indicator value for the first set of solution will be at least as good as the indicator value for the second; it is, therefore, a Pareto compliant quality indicator. Value ν can be calculated for a population P_0 which is composed by the N_0 non-dominated solutions of P.

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-II 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-II obtains lower values than MONEA and this obtains lower than NSGA-II. That is, the approximation sets obtained by ENORA-II 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.

	MONEA	ENORA-II	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-II and NSGA-II for the second order non-linear plant.

Taking all the above, we can conclude that the hypervolume values obtained with ENORA-II 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.

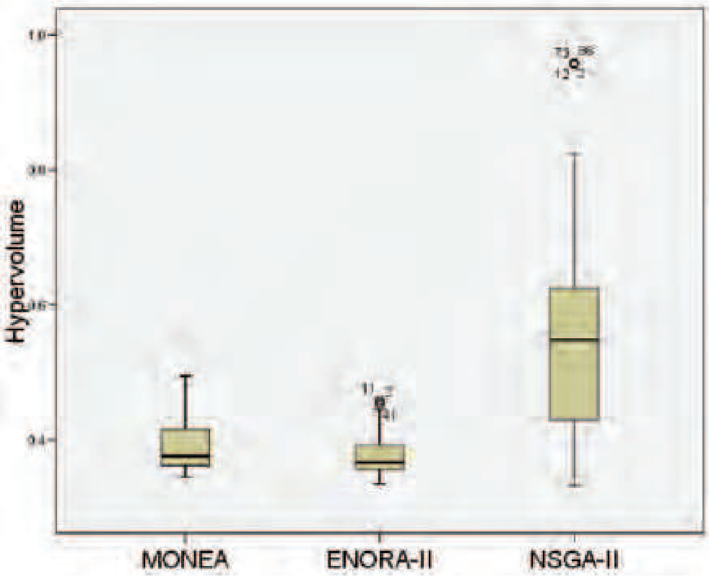


Figure 2. Boxplots for the hypervolume obtained with 100 runs of MONEA, ENORA-II and NSGA-II for the second order non-linear plant.

5. Conclusions

This chapter 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-II 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 to compare the algorithms and it has shown that, for the non linear plant problem, ENORA-II obtains better results than MONEA and NSGA-II algorithms.

Future improvements of the algorithms will be the automatic parameter tuning, and a next application of these techniques will be on medicine data.

6. Acknowledgements

Research supported in part by MEC and FEDER under project PET2006 0406 and TIN 2006-15460-C04-01.

7. References

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.

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

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*. In Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI), pp. 849-858.

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

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

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

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

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

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.

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

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.

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.

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

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

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

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.

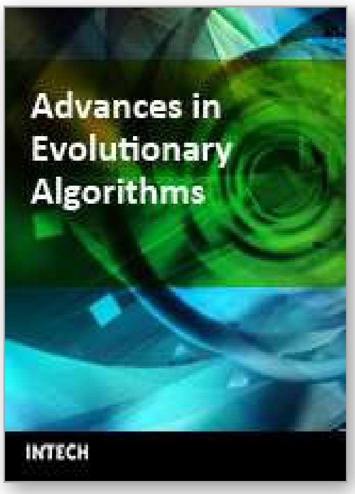
Takagi, T., Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15:116-132.

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

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.

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

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.



Advances in Evolutionary Algorithms
Edited by Xiong Zhihui

ISBN 978-953-7619-11-4
Hard cover, 284 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

With the recent trends towards massive data sets and significant computational power, combined with evolutionary algorithmic advances evolutionary computation is becoming much more relevant to practice. Aim of the book is to present recent improvements, innovative ideas and concepts in a part of a huge EA field.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821