Computing the Spanish Medium Electrical Line Maintenance Costs by means of Evolution-Based Learning Processes*

Oscar Cordón[1], Francisco Herrera[1] and Luciano Sánchez[2]

[1] Department of Computer Science and Artificial Intelligence.
University of Granada, 18071 - Granada, Spain.
E-mail: ocordon, herrera@decsai.ugr.es
[2] Department of Computer Science.
University of Oviedo. Oviedo, Spain.
E-mail: luciano@lsi.uniovi.es

Abstract. In this paper, we deal with the problem of computing the maintenance costs of electrical medium line in spanish towns. To do so, we present two Data Analysis tools taking as a base Evolutionary Algorithms, the Interval Genetic Algorithm-Programming method to perform symbolic regression and Genetic Fuzzy Rule-Based Systems to design fuzzy models, and use them to solve the said problem. Results obtained are compared with other kind of techniques: classical regression and neural modeling.

1 Introduction

In Spain, electrical industries do not charge the energy bill directly to the final user, but they share the ownership of an enterprise (called R.E.E., Red Eléctrica Española) which gets all payments and then distributes them according to some complex criteria (amount of power generation of every company, number of customers, etc.)

Recently, some of these companies asked to revise the rules. One of the proposed modifications involved a redistribution of the maintenance costs of the network. Since maintenance costs depend on the total length of electrical line each company owns, and on their kind (high, medium, urban low and rural low voltage) it was necessary to know the exact length of every kind of line each company was maintaining.

To compute the maintenance costs of town medium voltage lines, there is a need to know which would be the total line length if the instalation made would have been the optimal one. Clearly, it is impossible to obtain this value by directly measuring it, since the medium voltage lines existing in a town have been installed incrementally, according to its own electrical needs in each moment.

* This research has been supported by Hidroeléctrica del Cantábrico, under Contract CN-96-055-B1 with the University of Oviedo, and by CYCIT, under Project TIC96-0778 with the University of Granada.

Therefore, we need to solve the problem using other kind of techniques, which are able to relate some characteristics of a certain town with its maintenance cost [16], and the solution obtained has to verify another requeriment. It has not only to be accurate in the problem solving, but able to explain how a specific value is computed for a certain town. That is, *this solution has to be interpretable by human experts*, i.e, to be understood by lawyers and published by the Industry Ministry.

In this contribution, we propose two possible solutions to the said problem in the field of Data Analysis (DA). DA can be considered as a process in which starting from some given data sets, information about the respective application is generated. In this sense, DA can be defined as a search for structure in data. Since in our problem there is a need to find a relationship between some variables (the town characteristics and its associated maintenance cost), it is clear that it may be solved by means of DA techniques.

The two said approaches will make use of the Evolutionary Algorithms (EAs) [1]. We will consider the use of Genetic Algorithm Program (GA-P) [10] techniques for symbolic regression and the use of Genetic Algorithms (GAs) [8] to design Mamdani-type Fuzzy Rule-Based Systems (FRBSs) [7] to solve the said problem.

The paper is set up as follows. In Section 2, we briefly introduce the use of EAs in the field of DA and present the GA-P and Genetic Fuzzy Rule-Based Systems (GFRBSs) [3]. Sections 3 and 4 are devoted to present the two different approaches commented, the use of GA-P algorithms for symbolic regression problems and the use of GAs to design FRBSs. In Section 5, the introduced Electrical Engineering problem is tackled by means of the proposed techniques and their performance is compared with other kind of techniques, clasical regression and neural methods. Finally, some concluding remarks are pointed out.

2 Preliminaries: Evolutionary Algorithms for Data Analysis

Different algorithmic methods for DA have been suggested in the literature, as Clustering algorithms, regression techniques, Neural Netwoks (NNs), FRBSs, EAs, etc. As regards DA in the light of EAs, a representation of the information structure is considered and evolved until having an abstraction and generalization of the problem, reflected in the fitness function.

Recently, a lot of research efforts have been directed towards the combination of different methods for DA. In this way, EAs have been combined with different techniques either to optimize their parameters acting as evolutionary tuning processes or to obtain hybrid DA methods. Next, we briefly introduce two specific hybrid approaches, the GA-P to perform symbolic regressions and GFRBSs. Two particular developments in each field will be presented in Sections 3 and 4.

2.1 GA-P for Symbolic Regression

One of the main applications of the Genetic Programming (GP) [12] is the field of the *symbolic regression*. However, the way in which GP perform symbolic regressions is quite restrictive. To solve these GP limitations, a new technique, able to perform symbolic regression by combining the traditional GAs with the GP paradigm to evolve complex mathematical expressions capable of handling numeric and symbolic data, was proposed, the GA-P [10].

The GA-P combines GAs and GP, with each population member consisting of both a string and an expression. The GP part evolves the expression and the GA part concurrently evolves the coefficients used in it. Most of the GA-P's elements are the same as in either of the traditional genetic techniques. A complete description of GA-P can be found in [10].

2.2 Genetic Fuzzy Rule-Based Systems

Nowadays, one of the most important applications of the Fuzzy Set Theory are the FRBSs [7]. An FRBS presents two main components: 1) the Inference System, which puts into effect the fuzzy inference process, and 2) the Knowledge Base (KB) representing the known knowledge about the problem being solved, composed of a collection of fuzzy rules. Mamdani-type FRBSs work with the following rule structure:

$$\text{IF } X_1 \text{ is } A_1 \text{ and ... and } X_n \text{ is } A_n \text{ THEN } Y \text{ is } B_i$$

with $X_1, \ldots, X_n$ and Y being the input and output linguistic variables, respectively, and $A_1, \ldots, A_n$ and B being linguistic labels, each one of them having associated a fuzzy set defining its meaning. The fuzzy rules themselves are collected in a component named Rule Base, while linguistic labels and membership functions are stored in the Data Base (DB).

With the aim of automatically generating the KB, many different approaches have been presented taking EAs, usually GAs, as a base, constituting the so called GFRBSs [3]. These systems have obtained promising results that have extended their use in the last few years (see [5]). For a wider description of GFRBSs and different specific approaches see [3, 4].

3 Interval Valued GA-P for Symbolic Regression

Regression techniques are intended to find an adequate expression for a function g so that given a variable Y that depend on the value of a variable X, $g(X)$ is a good aproximation to Y. In practice, this means that we know N pairs (X_i, Y_i) and we search for a function g such that the mean square error $\frac{1}{N}\sum_{i=1}^{N}[Y_i - g(X_i)]^2$ is minimum [14].

Where the expression of g is not known, the techniques used to solve the problem are known as *symbolic regression*. Note that symbolic regression methods find *punctual* estimates for $\mathbf{Y}$ given $\mathbf{X}$. If we need a confidence interval for $\mathbf{Y}$ (that

is, an interval that contains the unknown value of $\mathbf{Y}$ with a certain probability) we must resort to fuzzy regression procedures [13] or neural network based ones [11].

We have extended GP methods to the intervalar case [15]. We will search for a set-valued function Γ that depends on m interval parameters and uses interval arithmetic [2] (for example: $\Gamma(x) = k \otimes x$ with $k = [1, 2]$ would be $\Gamma(x) = [x, 2x]$. "$\otimes$" means "product" in interval arithmetic) such that the probability that $\mathbf{Y} \in \Gamma(\mathbf{X})$ is higher or equal than a confidence degree β and the mean distance between the extremes of $\Gamma(x)$ is as low as possible.

That set-valued function is defined with a pair of functions g^+ and g^- such that $\Gamma(x) = [g^-(x), g^+(x)]$. The numerical problem that we need to solve is as follows: Given a value ϵ near to zero and *two* independent samples that contain N pairs $(\mathbf{X_i}, \mathbf{Y_i})$ and M pairs $(\mathbf{X'_j}, \mathbf{Y'_j})$ respectively, find two functions $g^-(x)$ and $g^+(x)$ such that

$$\frac{1}{N}\sum_{i=1}^{N}(g^+(\mathbf{X_i}) - \mathbf{g}^-(\mathbf{X_i}))$$

is minimum and the fraction of elements of the *first* sample for which $g^-(\mathbf{X_i}) \leq \mathbf{Y_i} \leq \mathbf{g}^+(\mathbf{X_i})$ is higher than $1 - \epsilon$.

The second sample is used to estimate the confidence degree. β is estimated with the fraction of the M elements of this last sample for which $g^-(\mathbf{X'_i}) \leq \mathbf{Y'_i} \leq \mathbf{g}^+(\mathbf{X'_i})$.

This method can also be used with imprecisely measured data, if the imprecision is characterized by interval values for $\mathbf{X}$ and $\mathbf{Y}$ (for example, data with tolerance: 5±10%).

4 Genetic Algorithms for Learning Mamdani-type Fuzzy Rule Bases

In this section, we analyze an specific GFRBS that may be employed as a DA technique. The genetic learning process was first proposed in [4] and it is composed of the following three different stages:

1. An *inductive generation process* for generating Mamdani-type fuzzy rules from examples, with two components: a fuzzy rule generating method based on a non-evolutionary inductive algorithm, and an iterative covering method of the example set.
2. A *genetic multisimplification process* for selecting rules, based on a binary coded GA with a genotypic sharing function and a measure of the FRBS performance. It will remove the redundant rules generated by the previous component with the aim of obtaining different simplified KBs presenting the best possible cooperation among the fuzzy rules composing them.
3. A *genetic tuning process*, based on a real coded GA and a measure of the FRBS performance. It will give the final KB as output by tuning the membership functions in each possible KB derived from the genetic multisimpli-

fication process. The more accurate KB obtained in this stage will constitute the final output of the whole genertic learning process.

Next subsections will briefly describe each one of the learning stages.

4.1 The inductive generation process

The generation process is based on a previously defined DB, composed of different fuzzy partitions of the variable spaces, as the one shown in Fig. 4.1.

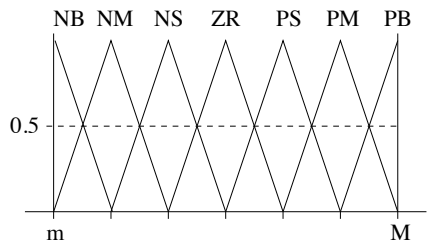

Fig. 1. Graphical representation of a possible fuzzy partition

The covering method is developed as an iterative process that allows us to obtain a set of fuzzy rules covering the example set. In each iteration, it runs the generating method, obtaining the best fuzzy rule according to the current state of the training set, considers the relative covering value this rule provokes over it, and removes from it the examples with a covering value greater than ϵ, provided by the system designer. It ends up when the training set becomes empty.

Each time the generating method is run, it produces a set of candidate fuzzy rules by generating the fuzzy rule best covering every example from the training set. The accuracy of the candidates is measured by using a multicriteria fitness function, composed of three different criteria measuring the covering that each rule provokes over the training set. Their expressions can be found in [4]. Finally, the best fuzzy rule is selected from the set of candidates and given as method output.

4.2 The genetic multisimplification process

Since the generation process works in an iterative way, it may obtain a KB containing redundant rules that do not cooperate adequatelly between them. The aim of this second stage is to simplify the previous KB, removing from it the rules not cooperating well.

The main idea of the genetic multisimplification process, proposed in [6], is that it does not only generate one simplified definition of the previous fuzzy rule set, but several different ones. To do so, it runs the genetic simplification process proposed in [9]. This process is based on a binary-coded GA which encodes the

set of rules obtained from the generation process into a chromosome, representing a value 1 the belonging of this rule to the final KB, and a 0 its absence. Two-point crossover and uniform mutation operators are used to alter the individuals and the stochastic universal sampling procedure, along with an elitist selection scheme, to perform selection. The fitness function combines an error measure, the medium square error (SE), and a term penalizing the lack of the encoded KB completeness property.

Each time the genetic simplification process obtains a simplified KB definition, the multisimplification one penalizes the search space zone where it is located to not generate it in future runs. A genotypic sharing scheme is used to penalize individuals according to its space proximity to the previous solutions found. The process ends up when the desired number of simplified KBs is generated.

For a wider description of this process, see [6].

4.3 The genetic tuning process

The genetic tuning process was proposed in [4]. It is based on a real coded GA that encodes a different DB definition in each chromosome. A primary fuzzy partition (as the one shown in Fig. 4.1) is represented as an array composed by $3 \cdot N$ real values, with N being the number of terms forming the linguistic variable term set. The complete DB for a problem, in which m linguistic variables are involved, is encoded into a fixed length real coded chromosome C_j built by joining the partial representations of each one of the variable fuzzy partitions as is shown in the following:

$$\begin{array}{l} C_{ji} = (a_{i1}, b_{i1}, c_{i1}, \ldots, a_{iN_i}, b_{iN_i}, c_{iN_i}) \ , \\ C_j = C_{j1}\ C_{j2}\ \ldots\ C_{jm} \ . \end{array} \tag{1}$$

The initial fuzzy partitions are used to define the interval of adjustment $[c_h^l, c_h^r]$ associated to every gene c_h in C_j, $h = 1 \ldots \sum_{i=1}^{m} N_i \cdot 3$. Max-min-arithmetical crossover and Michalewicz's non uniform mutation are considered to alter the individuals, and the selection is performed in the same way that in the genetic simplification process. The fitness function is the same used in that process.

On the other hand, the initial DB definition is used to generate the initial gene pool as well. It is encoded directly into a chromosome, denoted as C_1. The remaining individuals are generated in the interval of performance associated to each membership function.

5 Solving the Electrical Engineering Application by means of the proposed Evolution-Based Learning Processes

To solve the mentioned electrical problem [16], we were provided with data related to four different characteristics of the towns:

x_1: The sum of the lenghts of all streets in the town
x_2: The total area of the town

x_3: The area that is occupied by buildings
x_4: The energy supply to the town

and to the maintenance costs of line (y) in each one of them in a sample of 1059 simulated towns. Our objective was to relate the last variable (maintenance costs) with the other four ones, first by classical methods and NNs, and later by applying the DA techniques presented in this paper. Numerical results will be compared in the following.

As regards classical methods, we have considered linear, polynomial and and neural network models. The parameters of the polynomial models were fitted by Levenberg-Marquardt method and the neural model (a three layer perceptron) was trained with the QuickPropagation algorithm. The number of neurons in the hidden layer was chosen to minimize the test error; note that the training error could be made much lower than the shown, but not without making the test error higher. We used 4 input nodes, 5 hidden nodes, 1 output node.

GA-P algorithms have been applied to check whether we can obtain a formula that is comparable in complexity with the ones considered for the classical solution, while getting better adjust to the real data. We will define "simple expression" as a formula that can be codified in a tree with no more than 50 nodes and depending on no more than 10 parameters. Binary operations will be sum, difference, product, ratio and power. The unary operation will be the square root. We randomly select three individuals every generation. The worst one of them is replaced with the best descendent of the crossover of the remaining ones. Observe that this strategy is elitist and steady state.

Finally, to solve the problem by means of the GFRBS proposed, we have considered an initial DB constituted by some primary equally partitioned fuzzy partitions formed by *seven linguistic terms* with triangular-shaped fuzzy sets giving meaning to them (as shown in Figure 4.1), and the adequate scaling factors to translate the generic universe of discourse into the one associated with each problem variable.

To compare the mentioned techniques, we have divided the sample into two sets comprising 847 and 212 samples. SE values over these two sets are labeled *training* and *test*. Results obtained in the different experiments developed are shown in Table 1, where column *complexity* contains the number of parameters and the number of nodes in the parse tree of the expression, as well as the number of rules in the KB of the generated fuzzy model.

In view of them, we can conclude that fuzzy models and GA-P techniques clearly outperform classical non linear regression methods, being equal or superior to NNs. This result has great significance, because it means that NN performance can be achieved with a model with a high descriptive power. Mamdani-type fuzzy models provide the most comprehensive explanation of its functioning, and should be used when a human-readable, rule based, description of the problem is needed. In this case, the GFRBS has found a structure comprising 63 rules. When a mathematical formula is preferred to the rule bank, GA-P methods provide a suitable expression where the user can select the balance between complexity and precision. Punctual GA-P found a mathematical expression that can be co-

Table 1. Results obtained in the problem being solved

Method	Training	Test	Complexity
Linear	164662	36819	17 nodes, 5 par.
2th order polynomial	103032	45332	77 nodes, 15 par.
3 layer perceptron 4-5-1	86469	33105	35 par.
GA-P	18168	21884	50 nodes, 5 par.
Interval GA-P	16263	18325	15 nodes, 4 par.
fuzzy model	19679	22591	63 rules

dified in 50 nodes (note that a second order polynomial needs 77 nodes and that a linear model uses 17) and that explains the data almost identically than the fuzzy model. Interval GA-P results reflect the ability of the method to eliminate outliers. The results shown in Interval GA-P row were calculated over the subset of examples that were in the confidence interval (97,9% of total) and therefore it found the simplest expression (15 nodes and it does not use the input x_2, the total area of the town). Interval GA-P can only be used when we are allowed to discard some elements of the sample. If we cannot do that, we should apply punctual GA-P or fuzzy models. For example, if we apply the obtained Interval GA-P model to the whole dataset, without discarding outliers, we obtain roughly the same test error, but a much higher training error (202385) that may not be admissible.

6 Concluding Remarks

In this contribution we have solved a real-world spanish Electrical Engineering problem by means of two hybrid EA-based DA methods, the Interval GA-P for symbolic regression and GFRBSs.

Both techniques have demonstrated to be powerful DA tools capable of making abstraction on the data with good generalization properties in view of the results obtained in the application tackled. The first one allows us to obtain expressions with algebraic operators while the second one is able to generate KBs readable by human experts.

References

1. Bäck, T.: Evolutionary algorithms in theory and practice. Oxford University Press (1996).
2. Bojadziev, G. *Fuzzy Sets, Fuzzy Logic, Applications.* World Scientific. 1995.
3. Cordón, O., Herrera, F.: A general study on genetic fuzzy systems. In: J. Periaux, G. Winter, M. Galán, P. Cuesta, Eds., Genetic algorithms in engineering and computer science. John Wiley and Sons (1995) 33-57.

4. Cordón, O., Herrera, F.: A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples. International Journal of Approximate Reasoning **17(4)** (1997) 369-407.
5. Cordón, O., Herrera, F., Lozano, M.: On the combination of fuzzy logic and evolutionary computation: a short review and bibliography. In: W. Pedrycz, Ed., Fuzzy evolutionary computation. Kluwer Academic Press (1997) 57-77.
6. Cordón, O., Herrera, F.: Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems. Technical Report DECSAI-96126, Dept. of Computer Science and A.I. University of Granada. Spain (1997).
7. Driankov, D., Hellendoorn, H., Reinfrank, M.: An introduction to fuzzy control. Springer-Verlag (1993).
8. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley (1989).
9. Herrera, F., Lozano, M., Verdegay, J.L.: A learning process for fuzzy control rules using genetic algorithms. Fuzzy Sets and Systems (1998) to appear.
10. Howard, L., D'Angelo, D.: The GA-P: a genetic algorithm and genetic programming hybrid. IEEE Expert (1995) 11-15.
11. Ishibuchi, H., Tanaka, H., Okada, H. "An architecture of neural networks with interval weights and its application to fuzzy regression analysis". Fuzzy Sets and Systems 57. 27-39. 1993.
12. Koza, J.: Genetic programming. On the programming of computers by means of natural Selection. The MIT Press (1992).
13. Kacprzyk, J. *Fuzzy Regression Analysis.* Omnitech Press, Warsaw. 1992.
14. Ljung, L. *System Identification: Theory for the User* Prentice Hall. 1987.
15. Sánchez, L.: Interval-valued GA-P algorithms. Technical Report. Dept. of Computer Sciences. University of Oviedo. Oviedo. Spain (1997).
16. Sánchez, L.: Estudio de la red asturiana de baja tensión rural y urbana. Technical Report. Hidroeléctrica del Cantábrico Research and Development Department (in spanish). Asturias. Spain (1997).

This article was processed using the LaTeX macro package with LLNCS style