10

Evolutionary Learning Processes for Data Analysis in Electrical Engineering Applications

OSCAR CORDÓN[1], **FRANCISCO HERRERA**[1], **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

10.1 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 de España) 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 have asked to redistribute 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.

High and medium voltage lines can be easily measured. But low voltage line is contained in cities and villages, and it would be very expensive to measure it. This kind of line uses to be very convoluted and, in some cases, one company may serve more than 10000 small nuclei. An indirect method for determining the length of line is needed.

Data Analysis (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 the population and size of a certain area and the length of line in it, making use of some known data, that may be employed to predict the real length of line in any other village, it is clear that it may be solved by means of DA techniques.

In this work we will analyze two different approaches that make use of the Evolutionary Algorithms (EAs) in the field of DA, the use of Genetic Algorithm Program (GA-P) [HD95] techniques for symbolic regression and the use of Genetic Algorithms (GAs) [Gol89] and Evolution Strategies (ESs) [Sch95] to design Mamdani and TSK-type Fuzzy Rule-Based Systems (FRBSs) [BD95, DHR93]. We will consider these two approaches to solve the introduced problem.

The paper is set up as follows. In Section 2, we introduce the use of the EAs in the field of DA and present the GA-P and Genetic Fuzzy Rule-Based Systems (GFRBSs) [CH95]. 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 and ESs to design FRBSs. In Section 5, the introduced Electrical Engineering problem is tackled by means of the proposed techniques and their performance is compared to the one obtained by some classical methods. Finally, some concluding remarks are pointed out.

10.2 EVOLUTIONARY ALGORITHMS FOR DATA ANALYSIS

10.2.1 Framework

In DA, objects described by some attributes are considered and the specific values of the attributes are the data to be analyzed. Objects can be, for example, things, time series, process states, and so on. The overall goal is to find structure (information) about these data. This leads to a complexity reduction in the considered application which allows us to obtain improved decisions based on the gained information.

The application of DA has a wide range and occurs in diverse areas where different problem formulations exist.

Different algorithmic methods for DA have been suggested in the literature, as clustering algorithms, regression techniques, neural netwoks, FRBSs, EAs, etc.

As regards the 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. For example, in [Gre94] different approaches for learning in the framework of GAs are to be found.

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, for example, evolutionay-neural processes [WS92], evolutionary regression models [Koz92] and evolutionary fuzzy sytems [HV96].

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

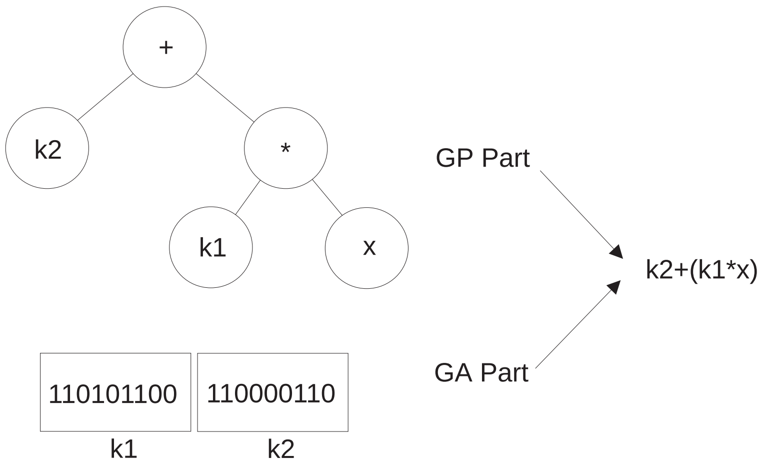

Figure 10.1 Member of population, GA-P algorithms

presented in Sections 4 and 5.

10.2.2 GA-P for symbolic regression

Genetic Programming (GP) [Koz92] has emerged as an effective mean of automatically generating computer programs to solve a variety of problems in many different problem domains, including the discovery of empirical formulae from numerical data.

GP methods generate symbolic expressions and can perform symbolic regressions. However, the way in which GP perform symbolic regressions is quite restrictive; the structure of an expression can be changed by crossover and mutation operations, but the value of the constants embedded in it —generated by the implementation program when the GP starts— can only be altered by mutations.

The GA-P [HD95] performs symbolic regression by combining the traditional GAs with the GP paradigm to evolve complex mathematical expressions capable of handling numeric and symbolic data. The GA-P combines GAs and GP, with each population member consisting of both a string and an expression as it is shown in Figure 10.1. The GP part of the GA-P evolves the expression. The GA part concurrently evolves the coefficients used in the expressions. Most of the GA-P's elements are the same as in either of the traditional genetic techniques.

The GA-P and GP make selection and child generation similarly, except that the GA-P's structure requires separate crossover and mutation operators for the expression and coefficient string components. In the GA-P, crossover and mutation take place independently for the coefficient string and the expressional component. Mutation and crossover rates for the coefficient string (using traditional GA methods) are independent from the rates for the expressional part (using standard GP methods).

By fusing the GA's capability of value optimization and the GP's capability of creating mathematical equations, it is improved the ability to describe the data. Therefore, the GA-P is a powerful DA tool.

A complete description of GA-P can be found in [HD95].

10.2.3 Genetic Fuzzy Rule-Based Systems

Nowadays, one of the most important applications of the Fuzzy Set Theory suggested by Zadeh in 1965 [Zad65] are the FRBSs. These kind of systems constitute an extension of the classical Rule-Based Systems because they deal with fuzzy rules instead of classical logic rules. Thanks to this, they have been succesfully applied to a wide range of problems presenting uncertainty and vagueness in different ways [BD95, Ped96, Wan94, YZ92].

An FRBS presents two main components: 1) the Inference System, which puts into effect the fuzzy inference process needed to obtain an output from the FRBS when an input is specified, and 2) the Knowledge Base (KB) representing the known knowledge about the problem being solved, composed of the Rule Base (RB) constituted by the collection of fuzzy rules, and of the Data Base (DB) containing the membership functions defining their semantics.

There exist two different kinds of fuzzy rules in the literature according to the expresion of the consequent:

1. Mamdani-type fuzzy rules consider a linguistic variable in the consequent [DHR93]:

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

   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.
2. TSK fuzzy rules are based on representing the consequent as a polynomial function of the inputs [TS85]:

$$\text{IF } X_1 \text{ is } A_1 \text{ and ... and } X_n \text{ is } A_n \text{ THEN } Y = p_1 \cdot X_1 + \ldots + p_n \cdot X_n + p_0$$

   with $X_1, \ldots, X_n$ and Y being the input and output linguistic variables, respectively, and $p_0, p_1, \ldots, p_n$ being real-valued weigths.

Knowledge-based methods are sutiable for fuzzy DA. In this approach, fuzzy If-Then rules are formulated and a process of fuzzification, inference and defuzzification leads to the final decision. Different efforts have been made to obtain an improvement on system performance by incorporating learning mechanisms to modify the rules and/or membership functions in the knowledge base (KB).

With the aim of solving this problem, in the last few years, many different approaches have been presented taking EAs, usually GAs, as a base, to automatically derive the KB, constituting the so called GFRBSs [CH95]. GFRBSs are considered nowadays as an important branch of the Soft Computing area [Bon97]. The promising results obtained by the EAs in the learning or tuning of the KB have extended the use of these algorithms in the last few years (see [CHL97a, CHL97b]).

It is possible to distinguish among three different groups of GFRBSs depending on the KB components included in the learning process: DB, RB, or both, i.e., KB [CH95]. The third group may be divided in two different subgroups depending on wheter the KB learning is performed in a single process or in different stages. For a wider description of each GFRBS group see [CH95, CH97b], and for an extensive bibliography see [CHL97a], Section 3.13. Different approaches may be found in [CH95, HV96].

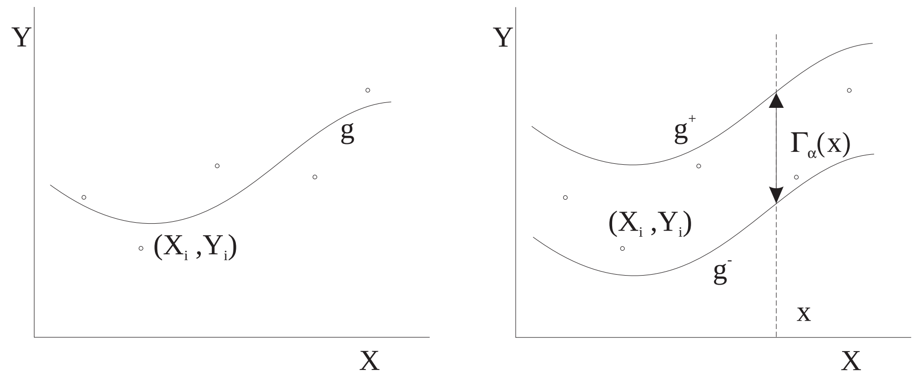

Figure 10.2 Linear and interval estimation

10.3 INTERVAL-VALUED GA-P FOR SYMBOLIC REGRESSION

In this Section we will introduce a modified version of the GA-P method, which we will call *Interval GA-P*. This approach —initially developed to solve an specific symbolic regression problem, [San97b]— is characterized by using interval values, instead of punctual ones, and by combining GA-P with local optimization techniques as well.

Regression analysis is concerned with the approximation of observed data by a function, when some variables (outputs) depend on other (inputs). Let the output Y be a random variable that will be estimated on the basis of the input variable $X = (X_1, \ldots, X_n)$. Usually, we understand that the regression analysis involves finding a function g, such that $g(X)$ is an admissible estimation of $E(Y|X)$. If the structure of g is unknown, the problem is named *symbolic regression*.

Symbolic regression produces a punctual estimation; anyway, sometimes it is necessary to obtain the margins in which we expect the output Y is, when the input variables X_i are known. Now, we should not look for a function g, but a multi-valued mapping $\Gamma_\alpha : \mathrm{Im}(X) \longrightarrow I(\mathbb{R})$, where $I(\mathbb{R})$ is the set formed by all closed intervals in $\mathbb{R}$, such that the random set $\Gamma_\alpha \circ X : \Omega \longrightarrow I(\mathbb{R})$ verifies

$$P\{\omega \in \Omega \mid Y(\omega) \in \Gamma_\alpha \circ X(\omega)\} \geq 1 - \alpha$$

for a given value of α.

We can assess this interval prediction in some different ways. We think that it is reasonable to admit that, given a value for α, the shorter Γ_α is, the better it is. So, if we define

$$\Gamma_\alpha \circ X = [g^- \circ X, g^+ \circ X]$$

for two continuous functions g^+ and g^- (see Figure 10.2) the margin of validity will be better when

$$E(g^+ \circ X - g^- \circ X)$$

is as low as possible, constrained by

$$P\{\omega \in \Omega \mid g^- \circ X(\omega) < Y(\omega) < g^+ \circ X(\omega))\} \geq 1 - \alpha.$$

In other words, given a region

$$R_{(g^+,g^-)} = \{(x,y) \in \mathbb{R}^{d+1} \mid g^-(x) < y < g^+(x)\}$$

it must be true that

$$P\{\omega \in \Omega \mid (X,Y)(\omega) \in R_{(g^+,g^-)}\} \geq 1 - \alpha.$$

Let us suppose now that g^+ y g^- also depend on a function $h_\theta : \mathbb{R}^d \longrightarrow \mathbb{R}$ in the following way:

$$[g^-(x), g^+(x)] = \{t \in \mathbb{R} \mid t = h_\theta(x), \theta \in [\theta_1^-, \theta_1^+] \times \ldots \times [\theta_m^-, \theta_m^+]\}$$

where the expression of h_θ is known except for the value of m parameters θ_i, and h_θ is continuous with respect to θ and x (so g^+ and g^- will also be continuous functions, as we had proposed). Then, for a random sample of size N, obtained from the random vector (X,Y),

$$((X_1,Y_1), \ldots, (X_N,Y_N))$$

we define θ_i^- and θ_i^+ to be the values that minimize

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

constrained by

$$1 - \epsilon \leq \frac{1}{N}\sum_{i=1}^{N} I_{R_{(g^+,g^-)}}(X_i,Y_i)$$

for a given value of ϵ. Notice that $\alpha \neq \epsilon$; once chosen a value for ϵ, we can only estimate α by means of a second sample

$$((X_1',Y_1'), \ldots, (X_M',Y_M')),$$

independent from the first one, by means of

$$\hat{\alpha}_M = 1 - \frac{1}{M}\sum_{i=1}^{M} I_{R_{(g^+,g^-)}}(X_i',Y_i').$$

The random variable $\hat{\alpha}_M$ follows a binomial distribution with parameters M and α and, by the strong law of the large numbers, it converges almost surely to the value α when $M \to \infty$.

In any case, to minimize $E(g^+ \circ X - g^- \circ X)$ with respect to the imposed constraints we should apply non linear constrained optimization techniques (say, for instance, non linear programming). And we cannot forget that the calculus is based in the knowledge of h_θ. Both problems (the search of the analytic expression of h and the values for θ_i^+ and θ_i^-) can be simultaneously solved by applying (with some modifications) the GA-P technique.

The adequacy of function h to a set of points is defined by the separation between g^+ and g^-, and both were defined in terms of h:

$$[g^-(x), g^+(x)] = \{t \in \mathbb{R} \mid t = h_\theta(x), \theta \in [\theta_1^-, \theta_1^+] \times \ldots \times [\theta_m^-, \theta_m^+]\}$$

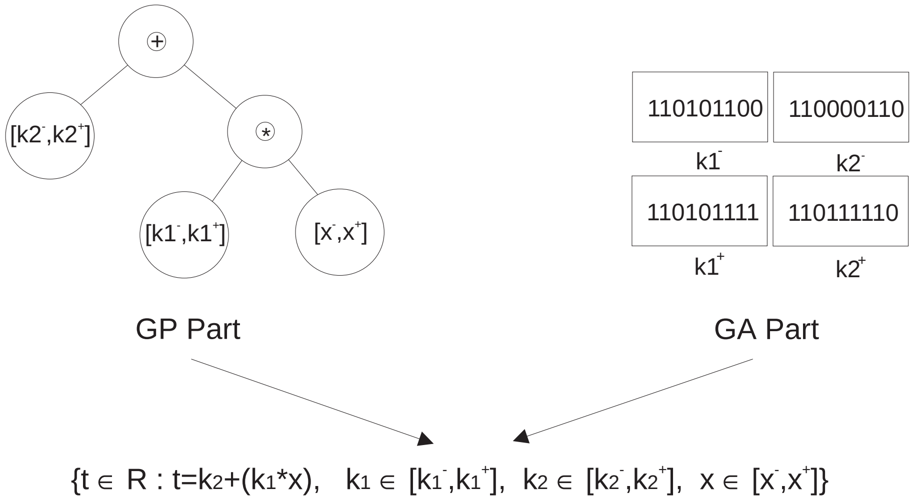

Figure 10.3 Interval arithmetic GA-P algorithms

that is, to find the value of $g^+(x)$ we should find the maximum of h inside the allowed range for its parameters,

$$g^+(x) = \max_{\mathbb{R}} \{h_\theta(x), \theta \in [\theta_1^-, \theta_1^+] \times \ldots \times [\theta_m^-, \theta_m^+]\}.$$

The same thing could be said of g^-. Fortunately, numerical calculus of this minimum and this maximum can be avoided if we choose an adequate representation for the expressional part of the GA-P algorithm.

The proposed representation is based in the use of interval arithmetic to perform all operations involved in the expressional part (see Figure 10.3). That is, we codify the function in a tree, whose terminal nodes represent intervals $[\theta_i^-, \theta_i^+]$ (that will contain the unknown values of the parameters). The internal nodes represent unary interval operations

$$O_u(A) = \{x \in \mathbb{R} \mid x = o_u(t) \wedge t \in A\}$$

or binary operations

$$O_b(A, B) = \{x \in \mathbb{R} \mid x = o_b(t, u) \wedge t \in A, u \in B\}$$

where $A, B \in I(\mathbb{R})$, $o_a : \mathbb{R} \longrightarrow \mathbb{R}$ and $o_b : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$. Then, the evaluation of the expressional part in an input value (point or interval) will be an interval. Moreover, by choosing operators O_u and O_b such that their evaluation depends only on the extremes of their arguments, the number of operations needed to evaluate the length of $\Gamma_{(g^+, g^-)}$ will be proportional to the number of operations needed to evaluate $h(x)$, so time of convergence will be proportional to conventional algorithms'.

A description on the unary and binary operators and the remaining characteristics of the algorithm are to be found in [San97b].

10.4 GENETIC ALGORITHMS AND EVOLUTION STRATEGIES TO DESIGN FUZZY RULE-BASED SYSTEMS

In this section, we analyze two different approaches for designing FRBSs by means of EAs that may be employed as DA techniques. In the first one, a GA will be used to refine a preliminary definition of a Mamdani-type KB, while in the second one, a $(\mu, \lambda) - ESs$ is considered to automatically derive a whole definition of a TSK-type KB.

10.4.1 Using Genetic Algorithms to Improve a Preliminary Definition of a Mamdani-Type Knowledge Base

In this first approach, EAs are considered to improve a previous definition of a KB. Thus, we are working with a GFRBS belonging to the first group mentioned in Section 2. These evolutionary methods are commonly known as *evolutionary tuning processes*, and many of them are to be found in the specialized literature (see [CHL97a], Section 3.13, and [CHL97b], Section 13). They all deal with the problem of refining a preliminary KB obtained from the linguistic information given by human experts, from an automatic learning process based on the numerical information available, or from a method combining both types of information [Wan94].

These kinds of processes may work over different DB components and adjust its previous definition by adapting it. The components that may be involved in the evolutionary tuning process are the following:

- The definitions of the fuzzy rule membership functions collected in the DB.
- The scaling factors.
- The gain of the different fuzzy partitions considered.

In this subsection we first present a very known inductive algorithm to derive Mamdani-type KBs, the Wang and Mendel's (WM) one [WM92], that is considered to generate a preliminary definition of the KB to solve any problem. Then, we will briefly introduce a genetic tuning process for adjusting the fuzzy membership functions of the different fuzzy partitions considered in the obtained KB [CHL96, CH97b]. The combination of both single methods in a two-stage process will allow us to automatically design high-performance Mamdani-type FRBSs. Anyway, it must be noted again that the proposed genetic tuning process may be used in combination with any other generation process able to obtain a preliminary definition of the KB.

The Wang and Mendel's Rule Base Generation Process

The inductive KB generation process presented in [WM92] has been widely known because of its simplicity and good performance. It is based on working with an input-output data set representing the behaviour of the problem being solved and with a previous definition of the DB composed of the input and output primary fuzzy partitions used. The fuzzy rule structure considered is the usual Mamdani-type rule with n input variables and one output variable presented in Section 2.

The generation of the fuzzy rules of this kind is performed by putting into effect the three following steps:

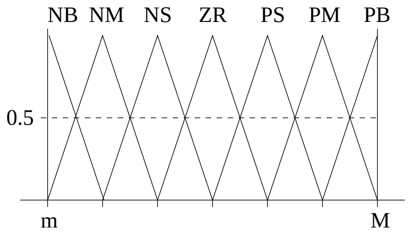

Figure 10.4 Graphical representation of a possible fuzzy partition

1. *To generate a preliminary linguistic rule set*: This set will be composed of the fuzzy rule best covering each example (input-ouput data pair) existing in the input-output data set. The composition of these rules is obtained by taking a specific example, i.e., a $n + 1$-dimensional real array (n values for the input variables and one for the output one), and setting each one of the rule variables to the linguistic label associated to the fuzzy set best covering every array component.
2. *To give an importance degree to each rule*: Let $R =$ *If* X_1 *is* A *and* X_2 *is* B *then* Y *is* C be the linguistic rule generated from the example (x_1, x_2, y) in a problem in which three variables, two input and one output ones, are involved. The importance degree associated to it will be obtained as follows:

$$G(R) = \mu_A(x_1) \cdot \mu_B(x_2) \cdot \mu_C(y)$$

3. *To obtain a final RB by using the preliminary rule set*: In the case in which all the rules in the preliminary set presenting the same antecedent values have associated the same consequent one, this linguistic rule is automatically put (only once) into the final RB. On the other hand, if there are conflictive rules, i.e., rules with the same antecedent and different consequent values, the rule considered for the final RB will be the one with higher importance degree.

A Genetic Tuning Process for Adjusting the Fuzzy Membership Functions in a Data Base

As all the GFRBSs in the same family, the genetic tuning process presented in [CHL96, CH97b] is based on the existence of a previous definition of the whole KB, i.e., an initial DB and an RB composed of T Mamdani-type fuzzy rules, called $\mathbf{R}$.

Each chromosome forming the genetic population will encode a different DB definition that will be combined with the existing RB to evaluate the individual adaption.

The GA designed for the tuning process presents real coding issue, uses the stochastic universal sampling [Bak87] as a selection procedure and Michaelewicz's non-uniform mutation operator [Mic96]. As regards the crossover operator, the max-min-arithmetical one [HLV95, HLV97], which makes use of fuzzy tools in order to improve the GA behaviour, is employed.

The primary fuzzy sets considered in the initial linguistic variables fuzzy partitions are triangular-shaped (see Figure 10.4). Thus, each one of the membership functions

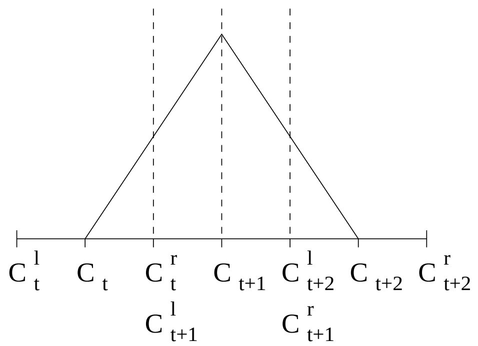

Figure 10.5 Intervals of performance

has an associated parametric representation based on a 3-tuple of real values and a primary fuzzy partition can be represented by an array composed of $3 \cdot N$ 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_r built by joining the partial representations of each one of the variable fuzzy partitions as it is shown in the following:

$$C_{ri} = (a_{i1}, b_{i1}, c_{i1}, \ldots, a_{iN_i}, b_{iN_i}, c_{iN_i}) \ ,$$
$$C_r = C_{r1}\ C_{r2}\ \ldots\ C_{rm}$$

The initial gene pool is created making use of the initial DB definition. This one is encoded directly into a chromosome, denoted as C_1. The remaining individuals are generated by associating an interval of performance, $[c_h^l, c_h^r]$ to every gene c_h in C_1, $h = 1 \ldots \sum_{i=1}^{m} N_i \cdot 3$. Each interval of performance will be the interval of adjustment for the corresponding gene, $c_h \in [c_h^l, c_h^r]$.

If $(t \bmod 3) = 1$ then c_t is the left value of the support of a fuzzy number. The fuzzy number is defined by the three parameters $(c_t,\ c_{t+1},\ c_{t+2})$, and the intervals of performance are the following:

$$c_t \in [c_t^l, c_t^r] = [c_t - \frac{c_{t+1}-c_t}{2}, c_t + \frac{c_{t+1}-c_t}{2}] \ ,$$
$$c_{t+1} \in [c_{t+1}^l, c_{t+1}^r] = [c_{t+1} - \frac{c_{t+1}-c_t}{2}, c_{t+1} + \frac{c_{t+2}-c_{t+1}}{2}] \ ,$$
$$c_{t+2} \in [c_{t+2}^l, c_{t+2}^r] = [c_{t+2} - \frac{c_{t+2}-c_{t+1}}{2}, c_{t+2} + \frac{c_{t+2}-c_{t+1}}{2}]$$

Figure 10.5 shows these intervals.

Therefore we create a population of chromosomes containing C_1 as its first individual and the remaining ones initiated randomly, with each gene being in its respective interval of performance.

As regards the fitness funtion, $E(\cdot)$, two different definitions for it may be considered [CH97b]. Both of them are based on an application specific measure usually employed in the design of GFRBSs, the mean square error (SE) over a training data set, E_{TDS}, composed of a number of input-output data pairs, $(ex_1^i, \ldots, ex_n^i, ey^i)$. The first definition is constituted directly by this criterion. Therefore, it is represented by the following expression:

$$E(C_j) = \frac{1}{2|E_{TDS}|} \sum_{e_l \in E_{TDS}} (ey^l - S(ex^l))^2$$

with $S(ex^l)$ being the output value obtained from the FRBS using the KB $\mathbf{R}(C_j)$, comprising the initial RB definition, $\mathbf{R}$, and the DB encoded in the chromosome C_j, when the input variable values are $ex^l = (ex^l_1, \ldots, ex^l_n)$, and ey^l is the known desired value.

The second fitness function definition is based on considering the *completeness property*, an important property of KBs [DHR93]. This condition is ensured by forcing every example contained in the training set to be covered by the considered KB to a degree greater than or equal to τ,

$$C_{\mathbf{R}(C_j)}(e_l) = \bigcup_{j=1..T} R_j(e_l) \geq \tau, \quad \forall e_l \in E_{TDS} \text{ and } R_j \in \mathbf{R}(C_j)$$

where $\tau \in [0, 1]$ is the minimal training set completeness degree, a value provided by the system designer.

Therefore, we define a *training set completeness degree* of $\mathbf{R}(C_j)$ over the set of examples E_{TDS} as

$$TSCD(\mathbf{R}(C_j), E_{TDS}) = \bigcap_{e_l \in E_{TDS}} C_{\mathbf{R}(C_j)}(e_l)$$

and the final fitness function penalizing the lack of the completeness property is:

$$F(C_j) = \begin{cases} E(C_j) & \text{if } TSCD(\mathbf{R}(C_j), E_{TDS}) \geq \tau \\ \frac{1}{2}\sum_{e_l \in E_{TDS}} (ey^l)^2 & \text{otherwise.} \end{cases}$$

10.4.2 Using Evolution Strategies to Derive a TSK Knowledge Base

In this second approach, we will consider a GFRBS that belongs to the third said group, the ones learning the complete KB. We are going to work with an evolutionary learning process presented in [CH97a], which is able to generate a whole definition of a TSK KB from examples.

The GFRBS is based on an iterative algorithm that equally divides the input space into a number of fuzzy subspaces and studies the existence of data in them. Each time data are located in a specific fuzzy input subspace, the process applies a *TSK rule consequent learning method* to determine the existing partial linear input-output relation, taking the data located in this input subspace as a base. The latter method is based on a (μ, λ)-ES using a new TSK rule consequent coding scheme, *the angular coding*, that was proposed in [CH97a], and a local measure of error, and takes into account the knowledge contained in this training data subset to improve the search process.

Next subsections will introduce the different process components. First of all, the TSK rule consequent learning method is introduced. Then we propose the use of the knowledge contained in the training data set to improve the search process. Finally we present the algorithm of the whole generation process, which makes use of the two previous aspects.

The TSK rule consequent learning method

In this method, the (μ, λ)-ES is considered to define TSK rule consequent parameters. The dimension n of the object variable vector $\vec{x}$ is determined by the number of input variables in the problem being solved. When there are iv input variables, there are $n = iv+1$ parameters to learn in the TSK rule consequent. The $\vec{x}$ part of the individuals forming the (μ, λ)-ES population is built by encoding the possible values using the *angular coding* proposed in [CH97a]. This coding scheme is based on encoding the angle value associated to the TSK rule consequent parameters instead of the tangent one by means of the function

$$C : \mathrm{R} \rightarrow (-\frac{\pi}{2}, \frac{\pi}{2}) \qquad ; \qquad C(y) = \arctan(y)$$

EA evolution is guided by a fitness function composed of a local measure of error. The expression of the measure used is the following:

$$\sum_{e_l \in E} h_l \cdot (ey^l - S(ex^l))^2$$

where E is the set of input-output data pairs $e_l = (ex^l_1, \ldots, ex^l_{iv}, ey^l)$ located in the fuzzy input subspace defined by the rule antecedent, $h_l = T(A_1(ex^l_1), \ldots, A_{iv}(ex^l_{iv}))$ is the matching between the antecedent part of the rule and the input part of the current data pair, ex^l, and $S(ex^l)$ is the ouput provided by the TSK fuzzy rule when it receives ex^l as input.

The object variables of the individuals in the first population are generated in the way shown in the next subsection, taking into account the knowledge contained in the input-output data set. As regards the composition of the remaining vectors, the components of $\vec{\sigma}$ are initiated to 0.001, and the ones in $\vec{\alpha}$, when considered, are set to *arctan (1)*.

Using available knowledge in the design process

To develop the knowledge-based generation of the initial population, we compute the following indices and obtain the following set from the input-output data set E:

$$y_{med} = \frac{\sum_{e_l \in E} ey^l}{|E|} \quad ; \quad y_{min} = \min_{e_l \in E} \{ey^l\} \quad ; \quad y_{max} = \max_{e_l \in E} \{ey^l\}$$
$$h_{max} = \max_{e_l \in E} \{h_l\}, \quad E_\theta = \{e_l \in E / h_l \geq \theta \cdot h_{max}\}$$

Therefore, we generate the initial ES population in three steps as follows:

1. Generate the $\vec{x}$ part of the first individual, $\vec{x}_1$, initiating parameters x_i, $i = 1, \ldots, iv$, to zero, and parameter x_0 to the angular coding of y_{med}.
2. Generate the $\vec{x}$ part of the following γ individuals, $\vec{x}_2, \ldots, \vec{x}_{\gamma+1}$, with $\gamma \in \{0, \ldots, \mu - 1\}$ defined by the GFRBS designer, initiating parameters x_i, $i = 1, \ldots, iv$, to zero, and x_0 to the angular coding of a value computed at random in the interval $[y_{min}, y_{max}]$.

3. Generate the $\vec{x}$ part of the remaining $\mu - (\gamma + 1)$ individuals, $\vec{x}_{\gamma+2}, \ldots, \vec{x}_{\mu}$, initiating parameters x_i, $i = 1, \ldots, iv$, to the angular coding of values computed at random in the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$, and x_0 to the angular coding of a value computed from a randomly selected element e in E_θ ($\theta \in [0.5, 1]$ is provided by the GFRBS designer as well) in such a way that e belongs to the hyperplane defined by the TSK rule consequent generated. Thus, we shall ensure that this hyperplane intersects with the swarm of points contained in E_θ, the most significative ones from E.
   Since with small angular values, large search space zones are covered, it seems interesting to generate small values for the parameters x_i in this third step. To do this, we make use of a modifier function that assigns greater probability of appearance to the smaller angles according to a parameter q, also provided by the GFRBS designer. We use the following function:

$$f : [0,1] \times \{-1,1\} \to (-\frac{\pi}{2}, \frac{\pi}{2}) \qquad ; \qquad f(x,z) = z \cdot \frac{\pi}{2} \cdot x^q$$

   Hence, the individual generation is performed as follows in this third step:
   For $j = \gamma + 2, \ldots, \mu$ do
   a) For $i = 1, \ldots, iv$ do
      a.1) Generate y at random in $[0,1]$.
      a.2) Generate z at random in $\{-1,1\}$.
      a.3) Set x_i to $f(y,z)$.
   b) Generate the value of x_0:
      b.1) Select e at random from E_θ.
      b.2) Set x_0 to $ey - \sum_{k=1}^{iv} C^{-1}(x_k) \cdot ex_k$, where $C^{-1}(\beta) = \tan(\beta)$ is the inverse of C.

Algorithm of the Evolutionary Generation Process

The generation process proposed is developed by means of the following steps:

1. Consider a fuzzy partition of the input variable spaces obtained from the expert information (if it is availaible) or by a normalization process. If the latter is the case, perform a fuzzy partition of the input variable spaces dividing each universe of discourse into a number of equal or unequal partitions, select a kind of membership function and assign one fuzzy set to each subspace. In this paper, we will work with symmetrical fuzzy partitions of triangular membership functions (see Figure 10.4).
2. For each multidimensional fuzzy subspace obtained by combining the individual input variable subspaces using the *and* conjunction do:
   (a) Build the set $E^{'}$ composed of the input-ouput data pairs $e \in E$ that are located in this subspace.
   (b) If $|E^{'}| \neq 0$, i. e., if there is any data in this space zone, apply the TSK rule consequent learning method over the data set $E^{'}$ to determine the partial linear input-output relation existing in this subspace. Therefore, no rules are considered in the fuzzy subspaces in which no data are located.
   (c) Add the generated rule to the KB.

10.5 PRACTICAL APPLICATION

We have mentioned that the development of the Interval GA-P Method was driven by a practical symbolic regression problem. That work dealt with maintenance cost estimation in some different kinds of electrical lines, and one of its intermediate results was the definition of a model relating the lenght of line in a rural population with its characteristics. We think that this problem is well suited to numerically compare the models we have defined, so we reproduce it here partially.

We were provided with the measured line length, the number of inhabitants and the mean distance from the center of the town to the three furthest clients in a sample of 495 rural nuclei. Our objective was to relate the first variable (line length) with the other two ones (population, radius of village), first by classical methods, later by applying the DA techniques presented in this paper. Numerical results will be compared in the next section.

Our variables are named as shown in Table 10.1.

Table 10.1 Notation considered for the problem variables

Symbol	Meaning
A_i	Number of clients in population
R_i	Radius of i population in the sample
n	Number of populations in the sample
l_i	Line length, population i
$\tilde{l}_i$	Estimation of l_i

10.5.1 Application of classical methods

In order to apply classical methods, we needed to make some hypothesis [San97a]. In the populations that are being studied, electrical networks are star-shaped and arranged in sectors. A main line passes near all clients inside them, and clients are connected to these main lines by small segments (see Figure 10.6).

To build a theoretical simplified model we have admitted that:

- A population comprises s_i sectors. Each sector covers an angle $2\theta_i$. All sectors in the same population cover the same angle. Each sector is served by one output of the only transformation center in the village.
- All sectors in a population have the same radius, R_i.
- Clients are uniformly distributed inside every sector.
- Inside a sector, the electrical line comprises a main nerve of length R_i and so many branches as consumers.

If we admit that customers are uniformly distributed, we can approximate the total length by multiplying the mean distance between one of them and the nerve by the number of inhabitants. Let us name this mean distance d_i for population i, and let the sector be $2\theta_i$ wide. Then

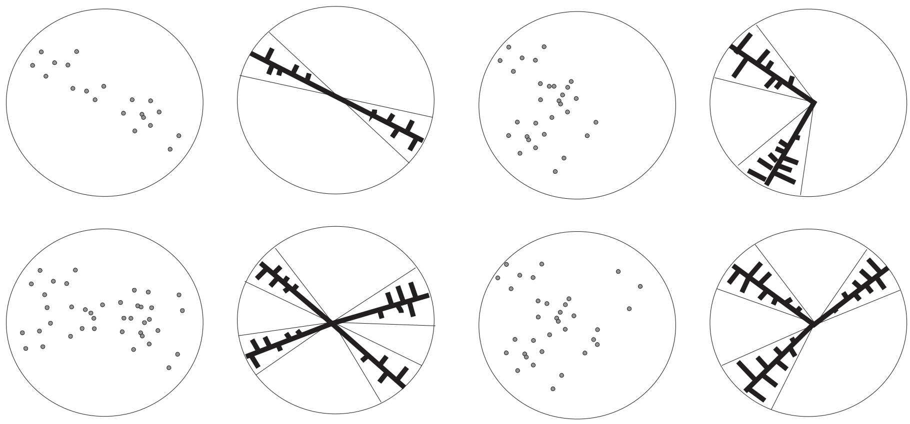

Figure 10.6 Models of some kind of nuclei

$$d_i = \frac{2(1 - cos\theta_i)}{3\theta_i} R_i$$

so cable length will be

$$\tilde{l_i} = s_i(R_i + \frac{A_i}{s_i} d_i) = s_i R_i + A_i \frac{2(1 - cos\theta_i)}{3\theta_i} R_i$$

10.5.2 Classical regression adjust

If the angles θ_i and the numbers s_i were similar enough between them, we could regard them as constants and estimate them by the parameters $\overline{\theta}_i = \theta$ y $\overline{s}_i = s$ of a least squares linear regression

$$\tilde{l_i}/R_i = s + k(\theta)A_i.$$

to a set of pairs $(x, y) = (A_i, l_i/R_i)$.

We can get a better adjust by allowing a certain dependence between the number of sectors, their angles and the number of inhabitants. This can be done by dividing the sample into classes or by mean of a change of variables. Both cases were studied, and the best adjust was obtained with the model

$$\frac{\tilde{l_i}}{R_i} = k_1 A_i^{k_2}$$

10.5.3 GA-P and Interval GA-P adjust

Let us apply GA-P algorithms to check whether we can obtain a formula that is comparable in complexity with the last one, while getting better adjust to the real

Table 10.2 Parameter values considered for the GA-P process

Parameter	Decision
Population size	100
Maximum number of generations	1000 (steady state)
Parent selection	(See text)
GA Part encoding	Real
GA Crossover operator	Two points
GP Crossover operation	Subexpression exchange, not context-dependant
GA Cross. probability	0.9
GP Cross. prob. internal nodes	0.9
GP Cross. prob. leaves	0.1
GA Mutation probability	0.01
GP Mutation probability	0.01
Expressional part limited to	20 nodes
Complexity individuals initial pop.	20 nodes
Maximum number of parameters	10
Enrichment initial population	1000 individuals
Edition probability	0
Encapsulation probability	0
Permutation probability	0
Decimation	No
ADFs maximum	0
Local GA optimization	Nelder and Mead's simplex

data. We will define "simple expression" as a formula that can be codified in a tree with no more than 20 nodes and depending on no more than 10 parameters. Binary operations will be sum, subtraction, product, quotient and power. The unary operation will be the square root. Other decisions (whose meaning is well known, see for instance [HD95, KR94, Mic96]) are shown in the Table 10.2.

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.

10.5.4 GFRBS fuzzy modeling

To solve the problem by means of the GFRBSs proposed, we have considered the parameter values shown in Tables 10.3 and 10.4. In both cases, the initial DB considered is 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 10.4), and the adequate scaling factors to translate the generic universe of discourse into the one associated with each problem variable.

Table 10.3 Parameter values considered for the genetic tuning process

Parameter	Decision
Population size	61
Maximum number of generations	1000
Non-uniform mutation parameter b	5
Max-min-arithmetical parameter a	0.35
Crossover probability	0.6
Mutation probability	0.1
Fitness function	E

Table 10.4 Parameter values considered for the TSK GFRBS

Parameter	Decision
Number of parents μ	15
Number of descendents λ	100
Maximum number of generations	500
Parameter γ	$0.2 \cdot \mu = 3$
Parameter θ	0.7
Parameter q	5
Recombination operators considered $\vec{r}$	$(3, 2, 0)$
Number of parents considered for recombination $\vec{\zeta}$	$(\mu, \mu, 1)$

Table 10.5 Results obtained in the problem being solved

Method	Training	Test	Complexity
Linear	287775	209656	7 nodes, 2 par.
Exponential	232743	197004	7 nodes, 2 par.
2th order polynomial	235948	203232	25 nodes, 6 par.
3rd order polynomial	235934	202991	49 nodes, 10 par.
3 layer perceptron 2-25-1	169399	167092	102 par.
GA-P	183693	159837	20 nodes, 3 par.
Interval GA-P	192908	158737	16 nodes, 3 par.
WM fuzzy model	175337	180102	13 rules
TSK fuzzy model	162609	148514	20 rules

10.5.5 Comparison between methods

To compare classical method, GA-P technique and GFRBS fuzzy modeling we have divided the sample into two sets comprising 396 and 99 samples. SE values over these two sets are labeled *training* and *test*. In this case, we define SE as

$$\frac{1}{2 \cdot N} \sum_{i=1}^{N} (\tilde{l}_i - l_i)^2$$

and the 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 every generated fuzzy model.

The parameters of the polynomial models were fitted by Levenberg-Marquardt; exponential and linear models were fitted by linear least squares and the multilayer perceptron was trained with the QuickPropagation algorithm. The number of neurons in the hidden layer was chosen to minimize the test error.

We can observe that fuzzy models and GA-P techniques clearly outperform classical non linear regression methods, being equal or superior to neural networks. This result has great significance, because it means that neural network performance can be achieved with a model with a high descriptive power. WM 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 genetic based method has found a very simple structure, comprising only 13 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. We observed that usually Interval GA-P finds a simpler expression that punctual GA-P, besides its convergence is somewhat slower. Observe that Interval GA-P is *not* intended to provide an estimation but a range of values in which the output is, with a probability higher than a preselected value. The number collected in the table is the scoring achieved by a punctual model formed when every interval of parameters is replaced by its mean point in the final model.

By last, observe that the best precision can only be obtained if we choose the less descriptive of fuzzy models, TSK. This model has a high complexity (20 rules) and definitely it is the selection that should be made when the precision is more important than the easiness of explanation. Anyway, this fuzzy model has associated a higher level of description than neural network models, because of the possibility of interpreting the antecedent part of the fuzzy rules.

10.6 CONCLUDING REMARKS

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

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 giving a linguistic local description of the problem.

ACKNOWLEDGEMENTS

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.

REFERENCES

[Bak87] Baker J. E. (1987) Reducing bias and inefficiency in the selection algorithm. In *Proc. Second International Conference on Genetic Algorithms (ICGA'87)*, pages 14–21. Hillsdale.

[BD95] Bardossy A. and Duckstein L. (1995) *Fuzzy Rule-Based Modeling With Application to Geophysical, Biological and Engineering Systems.* CRC Press.

[Bon97] Bonissone P. P. (1997) Soft computing: the convergence of emerging reasoning technologies. *Soft Computing* 1(1): 6–18.

[CH95] Cordón O. and Herrera F. (1995) A general study on genetic fuzzy systems. In Periaux J., Winter G., Galán M., and Cuesta P. (eds) *Genetic Algorithms in Engineering and Computer Science*, pages 33–57. John Wiley and Sons.

[CH97a] Cordón O. and Herrera F. (1997) Evolutionary design of TSK fuzzy rule-based systems using (μ, λ)-evolution strategies. In *Proc. Sixth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97)*, volume 1, pages 509–514. Barcelona.

[CH97b] Cordón O. and Herrera F. (December 1997) A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples. To appear in International Journal of Approximate Reasoning. Vol. 17, num. 4.

[CHL96] Cordón O., Herrera F., and Lozano M. (1996) A three-stage method for designing genetic fuzzy systems by learning from examples. In Voight H., Ebeling W., Rechemberg E., and Schwefel H. (eds) *Proc. Fourth International Conference on Parallel Problem Solving from Nature - PPSN IV. L.N.C.S. 1141*, pages 720–729. Springer-Verlag, Berlin.

[CHL97a] Cordón O., Herrera F., and Lozano M. (1997) A classified review on the combination fuzzy logic-genetic algorithms bibliography: 1989-1995. In Sanchez E., Shibata T., and Zadeh L. (eds) *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*, pages 209–241. World Scientific.

[CHL97b] Cordón O., Herrera F., and Lozano M. (1997) On the combination of fuzzy logic and evolutionary computation: a short review and bibliography. In Pedrycz W. (ed) *Fuzzy Evolutionary Computation*, pages 57–77. Kluwer Academic Press.

[DHR93] Driankov D., Hellendoorn H., and Reinfrank M. (1993) *An Introduction to Fuzzy Control.* Springer-Verlag.

[Gol89] Goldberg D. E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[Gre94] Grefensette J. J. (ed) (1994) *Genetic Algorithms for Machine Learning.* Kluwer Academic Press.

[HD95] Howard L. and D'Angelo D. (June 1995) The GA-P: a genetic algorithm and genetic programming hybrid. *IEEE Expert* pages 11–15.

[HLV95] Herrera F., Lozano M., and Verdegay J. L. (1995) Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning* 12: 299–315.

[HLV97] Herrera F., Lozano M., and Verdegay J. L. (November 1997) Fuzzy connectives based crossover operators to model genetic algorihtms population diversity. *Fuzzy Sets and Systems* 92(1): 21–30.

[HV96] Herrera F. and Verdegay J. L. (eds) (1996) *Genetic Algorithms and Soft Computing.* Physica-Verlag.

[Koz92] Koza J. (1992) *Genetic Programming. On the Programming of Computers by Means of Natural Selection.* The MIT Press.

[KR94] Koza J. and Rice J. (1994) *Genetic Programming.* The MIT Press.

[Mic96] Michalewicz Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag.

[Ped96] Pedrycz W. (ed) (1996) *Fuzzy Modelling: Paradigms and Practice.* Kluwer Academic Press.

[San97a] Sanchez L. (1997) Estudio de la red asturiana de baja tensión rural y urbana. Technical report, Hidroeléctrica del Cantábrico Research and Development Department.

[San97b] Sanchez L. (1997) Interval-valued GA-P algorithms. Technical report, University of Oviedo, Department of Computer Sciences. Oviedo. Spain.

[Sch95] Schwefel H. P. (1995) *Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series.* John Wiley and Sons.

[TS85] Takagi T. and Sugeno M. (February 1985) Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1): 116–32.

[Wan94] Wang L. X. (1994) *Adaptive Fuzzy Systems and Control.* Prentice-Hall.

[WM92] Wang L. X. and Mendel J. M. (November 1992) Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics* 22(6): 1414–1427.

[WS92] Whitley L. D. and Schaffer J. D. (eds) (1992) *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks.* IEEE Computer Society Press.

[YZ92] Yager R. R. and Zadeh L. A. (eds) (1992) *An Introduction to Fuzzy Logic Applications in Intelligent Systems.* Kluwer Academic Press.

[Zad65] Zadeh L. A. (February 1965) Fuzzy sets. *Information and Control* 8: 338–353.