EVOLUTIONARY APPROACHES TO THE LEARNING OF FUZZY RULE-BASED CLASSIFICATION SYSTEMS

O. Cordón[1], **M.J. del Jesus**[2] and **F. Herrera**[1]

[1]Dept. of Computer Science and Artificial Intelligence. E.T.S.
Ingeniería Informática.
University of Granada, 18071 – Granada
Spain
[2] Dept. of Computer Science. Escuela Politécnica Superior
University of Jaén, 23071 – Jaén
Spain
(ocordon, herrera@decsai.ugr.es, mjjesus@ujaen.es)

The learning of a Fuzzy Rule-Based Classification System (FRBCS) by means of a supervised inductive process fundamentally implies four tasks that are complementary among them: the selection of the most informative variables to the classification problem to solve, the generation of a set of rules, the selection of the subset of rules with the best co-operation and the least redundancy, and the establishment and tuning of the fuzzy partitions for the domains of the problem variables.

The automatic definition of any of these tasks may be faced as an optimisation or search process. Evolutionary Algorithms, and particulary Genetic Algorithms, are search techniques which use operations based on the natural genetics, proving theoretic and empirically their robust search capacity in complex spaces. This is the reason why they offer a valid approach for problem solving, as the Classification System design, that requires an efficient and effective search.

In this chapter we analyse the evolutionary approaches to the learning of FRBCSs. We present the different proposals based on evolutionary algorithms for the four aforementioned tasks.

1 Introduction

The construction of a Classification System has been tackled many times using fuzzy rules as knowledge representation tool ([4], [7] and [12]). The resulting systems are called Fuzzy Rule-Based Classification Systems (FRBCSs) and their success is fundamentally due to two reasons. On the one hand, the use of fuzzy logic makes possible the treatment of imprecise, uncertain or incomplete information, very common in real classification problems. On the other hand, rules represent the knowledge in a comprehensible form for those who will use the Classification System, making possible the use of this kind of systems as a tool in decision making processes. All in all, the fuzzy rules allow us to work in a transparent way in a feasible computer environment with the opaque classification schemes often used by human beings for these kinds of tasks [81].

The design of an FRBCS by means of a supervised learning process, which describes with higher possible precision the classes to represent, fundamentally implies four tasks that are complementary among them: the selection of the most informative variables to the classification problem to solve, the generation of a set of rules, the selection of the subset of rules with the best co-operation and the least redundancy, and the establishment and/or tuning of the fuzzy partitions of the problem variable domains.

The automatic definition of any of these tasks may be faced as an optimisation or search process. Evolutionary Algorithms [2], and particularly Genetic Algorithms (GAs) ([24], [37] and [62]) are considered as a general adaptable concept for problem solving, especially well suited to solving difficult optimisation problems. It has been theoretically and practically proven that they provide a valid solution to those problems that require an efficient and effective search in complex spaces. Due to this reason, and although GAs can not be considered learning algorithms, they constitute a powerful optimisation tool, independent of the domain, and applicable to the different tasks composing the learning of the FRBCS.

In this chapter, we will analyse different proposals on the use of GAs in the learning of a FRBCS, introducing some examples about its application in a classic classification problem, the IRIS problem.

To achieve this objective, in Section 2 we describe the structure of an FRBCS and how it works classifying new patterns, and we explain the different aspects from the FRBCS design that can be treated with processes based on GAs. We study several evolutionary approaches to solve them in the rest of the contribution: Section 3 describes the genetic feature selection and extraction process; Section 4 discusses the general characteristics of the genetic fuzzy rule learning approaches; Section 5 shows the use of GAs in the fuzzy rule selection process; and Section 6 studies the genetic tuning of the fuzzy partitions for the variables involved in the fuzzy rules. Finally, in Section 7, we point out the conclusions of the study we have carried out.

2 Fuzzy Classification Framework

The pattern classification problem involves assigning a class C_j from a predefined class set $C = \{C_1, \ldots, C_M\}$ to an object, described as a point in a certain feature space $x \in S^N$.

The key problem of designing a Classification System is to find a mapping

$$D : S^N \longrightarrow C$$

optimal in the sense of a certain criterion $\delta(D)$ that determines the classifier performance. Usually, the final goal is to design a Classification System that assigns class labels with the smallest possible error across the whole feature space.

2.1 Fuzzy Rule-Based Classification Systems

As mentioned, in this contribution we will focus on the Classification Systems that utilise the fuzzy rules as a knowledge representation tool, FRBCSs.

The FRBCS design involves to obtain a Fuzzy Reasoning Method (FRM) and to learn a Knowledge Base (KB).

The FRM is an inference procedure, which derives conclusions from the KB and a pattern. Traditionally, in the specialised bibliography, the FRM which is used considers only one rule, the winner one ([30], [40], and [57]). Several researchers have shown the increase of the FRBCS generalisation capacity when considering an FRM that integrates, in the inference process, the information from all the rules fired or from a subset of them ([4], [12], [16], and [17]).

The KB is composed of the Data Base (DB) and the Rule Base (RB):

- The DB contains the definition of the fuzzy sets associated with the linguistic terms used in the RB.

- The RB contains the set of fuzzy classification rules of one of the following rule types traditionally used in the specialised bibliography for FRBCSs:

    1. Fuzzy rules with a class in the consequent [30]. This kind of rule has the following structure:

       $$R_k : \text{ If } x_1 \text{ is } A_1^k \text{ and} \ldots \text{and } x_N \text{ is } A_N^k \text{ then } Y \text{ is } C_j$$

       where $x_1, \ldots, x_N$ are the outstanding selected features for the classification problem, $A_1^k, \ldots, A_N^k$ are linguistic labels used to discretise the continuous domain of the variables, and Y is the class C_j to which the pattern belongs.

    2. Fuzzy rules with a class and a certainty degree of the classification in that class in the consequent [40]:

       $$R_k : \text{ If } x_1 \text{ is } A_1^k \text{ and} \ldots \text{and } x_N \text{ is } A_N^k \text{ then } Y \text{ is } C_j \text{ with } r^k$$

       where r^k is the certainty degree of the classification in the class C_j for a pattern belonging to the fuzzy subspace delimited by the antecedent.

3. Fuzzy rules with certainty degree for all classes in the consequent [57]:

$$R_k: \quad \text{If } x_1 \text{ is } A_1^k \text{ and} \ldots \text{and } x_N \text{ is } A_N^k \text{ then } (r_1^k, \ldots, r_M^k)$$

where r_j^k is the soundness degree for the rule k to predict the class C_j for a pattern belonging to the fuzzy region represented by the antecedent of the rule.

The learning of a KB is carried out through an inductive supervised learning process, which starts with a set of correctly classified examples (training examples) and its ultimate objective is to design a Classification System, which will assign class labels to new examples with minimum error. Finally, the system performance on the test data is computed, to gain an estimate of the FRBCS real error. This process is described in Figure 1.

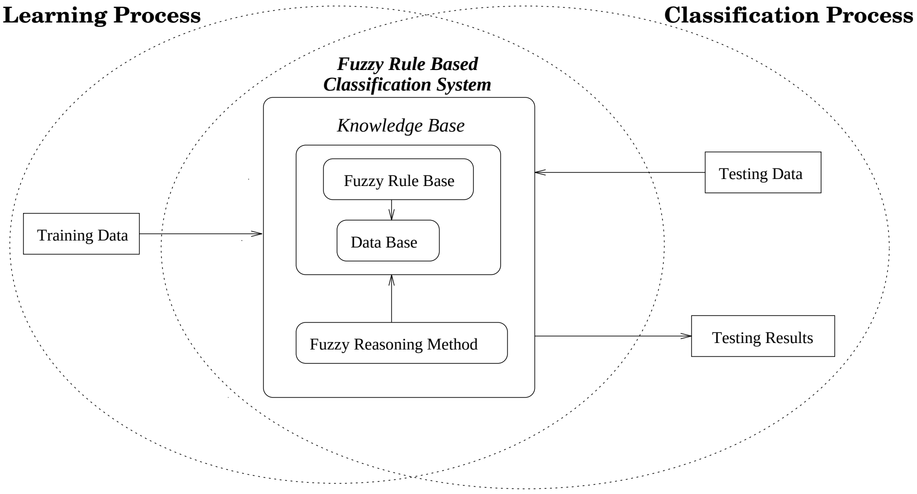


Figure 1. Design of an FRBCS (learning/classification).

There are many real applications of FRBCSs, among them we can point out the following: The recognition of license plates [1]; the detection of roadlike structures [58]; the segmentation of salient facial landmarks in a face recognition system [3]; the recognition of handwritten numerals [11] and [12], vowel sounds [57], and printed upper-case English letters [12]; the segmentation of map images [12]; the classification of biological patterns [68]; the medical diagnostic of diseases as the determination of thyroid dysfunctions [70], and myocardial infartions [26].

2.2 Fuzzy Reasoning Methods

An FRM is an inference procedure, that uses the information contained in the RB to predict a class for an unknown example. Cordón et al. in [17] introduced a general reasoning model that, particularised to an RB composed of rules with a class and its certainty degree in the consequent, is described in the following.

In the classification of an example $E^t = \left(e_1^t, \ldots, e_N^t\right)$, the RB $R = \left\{R_1, \ldots, R_L\right\}$ is divided into M subsets according to the class indicated by its consequent,

$$R = R_{C_1} \cup R_{C_2} \cup \ldots \cup R_{C_M}$$

and the next scheme is followed:

1. **Compatibility degree**. The compatibility degree of the antecedent with the example is computed for all the rules in the RB, applying a t-norm over the membership degree of the values of the example $\left(e_i^t\right)$ to the corresponding fuzzy subsets.

$$R^k\left(E^t\right) = T\left(\mu_{A_1^k}\left(e_1^t\right), \ldots, \mu_{A_N^k}\left(e_N^t\right)\right), \qquad k = 1, \ldots, L$$

2. **Association degree**. The association degree of the example E^t with the M classes is computed according to each rule in the RB.

$$b_i^k = h\left(R^k\left(E^t\right), r^k\right), \qquad k = 1, \ldots, \left|R_{C_i}\right| \\ i = 1, \ldots, M$$

3. **Weighting function**. The values obtained are weighted by means of a function *g*. An expression which promotes the highest values and penalises the smallest ones seems to be the most adequate choice for this function.

$$B_i^k = g\left(b_i^k\right), \qquad k = 1,\dots,\left|R_{C_i}\right| \\ i = 1,\dots,M$$

4. **Pattern classification soundness degree for all classes**. To compute this value, an aggregation operator is used which combines, for each class, the positive association degrees computed in the previous step

$$Y_i = f\left(B_i^k, \quad k = 1,\dots,\left|R_{C_i}\right| \quad \text{and} \quad B_i^k > 0\right), \\ i = 1,\dots,M$$

   with *f* being an aggregation operator that returns a value between the minimum and the maximum.

5. **Classification**. A decision function *F* is applied to the classification degrees of the example. This function will return the class label corresponding to the maximum value.

$$C_l = F\left(Y_1,\dots,Y_M\right) \quad \text{such that} \quad Y_l = \max_{j=1,\dots,M} Y_j$$

It is clear that if we select the aggregation operator $f(\cdot)$ as the maximum operator, we have the classical FRM, i.e., the FRM based on the winner rule that considers, in the classification process, the rule with the highest association degree. By using this reasoning method, we do not consider the information provided by the other rules that also are compatible (have also been fired) with this example. Cordón et al. in [16] and [17] proposed to use FRMs that combine the information given by the different rules fired by a pattern, and they distinguished two kinds of inference models whose difference is based on the use of the function $f(\cdot)$ in step 4:

- FRMs integrating all fuzzy rules, and

- FRMs selecting a subset of fuzzy rules (the classical FRM is a particular case of this group in which this subset is composed of a single rule, the one with the highest association degree).

Some proposals for the function $f(\cdot)$ belonging to both said types are described and analyzed in [16] and [17].

2.3 Learning the Knowledge Base for an FRBCS

The learning of an FRBCS by means of a supervised inductive process fundamentally implies four tasks that are complementary among them:

- feature selection and extraction, i.e., determining the set of relevant variables and their relevance for the classification problem,

- learning of fuzzy rules, that is, generation of an RB (if the DB is learnt together with the RB, we learn a complete KB),

- selection of fuzzy rules, simplifying the RB, and

- tuning the membership functions that describe the semantic associated to the linguistic labels used by the linguistic variables.

Before a detailed description of these four tasks is given, we should make some remarks:

- The feature selection process is usually considered as a previous stage, that is independent of the Classification System construction. In our case, we have included it into the design tasks that are needed to obtain the KB for the FRBCS, due to the fact that there is a need for determining the relevant variables in order to specify the structure and to learn the composition of the fuzzy classification rules included in it.

- As we mentioned in the first point, in many cases, the feature selection and extraction is carried out in a pre-processing data stage, previous to the Classification System design and regardless of the type of Classification System and of the inductive learning process used. Taking the models which consider both aspects in the selection of the most important variables into account, the main research has been focused in the development of methods for Classification Systems based on decision trees, neural networks, and k-nearest neighbour rule. This is the reason why in the following section we will analyse feature extraction and selection based on GAs and applied to Classification System design, not necessarily based on fuzzy rules. Anyway, due to the nature of the process to develop, the way in which it is carried out is not totally dependent on the type of Classification System to be built. For this reason, they may be considered feature selection and extraction systems analogous to those, applied to FRBCS design.

However, the generation and selection of fuzzy classification rules, and the tuning of membership functions are specific tasks in the FRBCS design, and they will be studied in Sections 4, 5 and 6.

3 Genetic Feature Selection

An important aspect in the developing process of a Classification System is selection of the most informative variable set for the specific problem. The algorithms that establish a solution to this optimisation problem with restrictions are called feature selection algorithms. Their objective is to find the subset from the whole feature or variable set making the inductive learning algorithm able to generate a Classification System with the least error.

The feature selection algorithms, removing the variables that introduce noise and those which are not representative, not only allow to increase the system performance, but also their simplicity, and to decrease the cost associated with the data acquisition and the time needed to learn the Classification System by means of an inductive learning process.

As Kohavi pointed out in [49], the feature selection problem is not interesting from the theoretical point of view, because the Bayes rule forecasts the most probable class for any instance, on the basis of the knowledge of the underlying probability distribution, with the highest possible accuracy. However, in most of the real problems the distribution function is unknown, and therefore, the inductive learning algorithms try to obtain approximate solutions to NP-hard optimisation problems and a selection of the most informative variables for a problem determines the performance of the system finally obtained.

Feature selection algorithms have three main components:

1. A *search algorithm*, which explores the space of variable subsets.

2. An *evaluation function* that provides a measure of the goodness of a specific feature subset in the searching process.

3. A *performance function*, which determines the validity of the subset obtained.

Depending on the type of search algorithm used, the feature selection methods have been traditionally grouped into three categories:

- Exponential algorithms, as the exhaustive search algorithms, which explore all the subset space to find the variable subset with the best behaviour, as well as the branch and bound algorithms. These kinds of methods have $O(2^n)$ complexity for a problem with n characteristics, so they cannot be used in most real problems.

- Random-search algorithms, as GA or Simulated Annealing-based ones.

- Sequential search algorithms, which use hill climbing strategies, removing or adding a variable to a given subset.

We will focus our attention on the second group, specifically on the feature selection methods based on GAs. Among them, and taking into account the evaluation function used, we may distinguish:

- *wrapper models*, which work with the inductive learning algorithm of the own classifier as evaluation function, and

- *filter models*, which assign evaluation results to the candidate solutions, independently on the inductive learning algorithm.

Graphically, we can see how these models work in Figures 2 and 3.

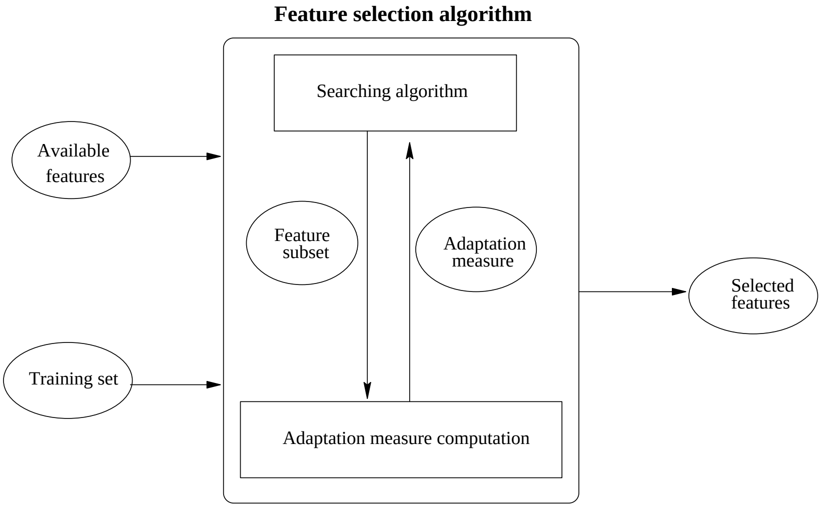


Figure 2. Filter model.

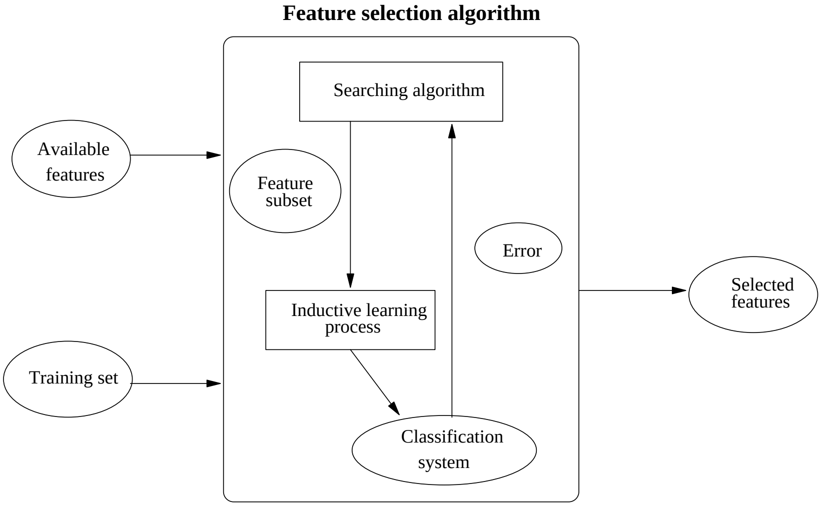


Figure 3. Wrapper model.

The main disadvantage of the wrapper model is that the time needed to carry out the selection process is increased, because of the inclusion of the Classification System construction in it, to evaluate each one of the potential solutions. The filter approach solve this problem including an evaluation function which does not consider the inductive process followed in the Classification System construction. Therefore, the behaviour of the selected features in the inductive learning algorithms is ignored, and this is the reason by which the given solution does not sometimes permit to design a Classification System with the highest performance.

Although most of the inductive learning processes consider the characteristic selection in a data pre-processing stage, some of them face the problem as a part of the learning process. These kinds of models are called *embedded models*, and we will analyse them in Section 4.

In the specialised bibliography, the terms feature *selection* and *extraction* have been confused. The feature selection objective is, as pointed out, the minimisation of the variable set considered in the learning problem which maximises the correct classification percentage. On the contrary, feature extraction processes look for a transformation, usually linear, which helps to remove redundant and irrelevant information, and that shows the relative importance of the variables selected in the classification problem.

In the following subsections, we will describe several proposals of genetic selection and/or extraction methods based on the filter or wrapper models, and on combinations of them both.

3.1 Genetic Feature Selection Methods Based on the Filter Model

The feature selection algorithms based on the filter model are characterised by the use of an evaluation function based on a measure of the selected characteristic suitability, which is independent from the inductive construction algorithm of the Classification System. In this philosophy, we should highlight the works carried out by Liu et al. [53], and Lanzi [51].

- **Liu et al.** [53] propose a GA for feature selection with fixed length integer coding, in which the i^{th} gen represents the index of the i^{th} selected variable. The evaluation function is based on the expected mutual information measure between two variables [5], and tries to increase the information content of the variable subset represented by the individual, and to reduce the redundancy. It uses two genetic operators, adapted to the problem and to the coding: The partial-complementary-crossover and the random-delete-mutation that introduces variation in a chromosome, maintaining the stability in the chromosome where it comes from.

- **Lanzi** [51] developes a binary coded GA in which each binary digit stands for the presence (1) or the absence (0) of a given feature. The algorithm uses standard genetic operators, crossover and mutation without modification, and a fitness function based on the inconsistency rate [54] introduced by the feature elimination.

Other authors ([50] and [55]) have proposed some approximate measures to evaluate feature subsets without applying an inductive learning algorithm that can be used in the fitness function in a filter GA for feature selection.

3.2 Genetic Feature Extraction/Selection Methods Based on the Wrapper Model

The feature selection and extraction algorithms based on the wrapper model are characterised by the construction of the Classification System by means of an inductive learning process, for the evaluation of the variable subsets. In this type of algorithms, we should highlight the works carried out by Siedlecki and Sklansky [71], Brill et al. [9], Punch et al. [68], Ray et al. [70] and Yang and Honavar [79].

- **Siedlecki and Sklansky** [71] introduce the use of GAs for the selection of features in the design of Classification Systems. For this aim, they propose a binary coded GA, in which a 0 value represents the absence of the feature and an 1 value its presence. The fitness function combines the error measure obtained from the inductive algorithm used for learning the Classification System and a penalty function to preserve the verification of a feasibility property in the feature subset being evaluated.

- **Brill et al.** [9] propose a GA with punctuated equilibria (GAPE) to select variables for a neural network classifier [33].

  GAPE is based on the following basic ideas:

  - It uses, like Siedlecki and Sklansky in [71], a binary coding with fixed length to indicate the present variables with an 1, and those not present with a 0.

  - Each individual receives a scoring, which is a linear combination of the error and the number of the features. One of the most sensible models to the presence of variables with noise and/or irrelevant, the nearest neighbour [19], is used to compute the classification error. The nearest neighbour rule provides a good measure of the variable set suitability, and is more efficient than the corresponding neural network construction.

  - The fitness function depends on the current population, because it is a normalised linear combination of the individual score with the average score of the individual from the population, and the deviation of them with respect to the average.

  - It uses the binomial crossover (2-point shuffle), which avoids the positional bias of the point crossover operators, and the distributional bias of the uniform crossover [22].

  - Several populations independently evolve in different processors during a number of generations, interchanging the best individuals among neighbour subpopulations at the end.

  This method obtains feature subsets for a neural network with the same or better prediction capacity than the neural network that uses all the variables. One of the limitations of the wrapper methods based on an evaluation function that uses a different learning algorithm from the one finally used for the Classification System construction, is that they could obtain a feature set which originates a Classification System with bad performance. The authors propose, as a solution to this problem, to carry out a little evolution with the best obtained solutions, using as fitness function the error made by the Neural Network Classification System, which is learnt starting from the selected feature set.

The main percentage of computing time of this algorithm is consumed by the population evaluation process, reason by which Brill et al. in [9] propose the *training set sampling* technique, in which only a proportion of the training example set is used in the individual evaluation, but this subset is resampled every generation. But in this GAPE the evaluation function is a measure that concerns to all the population, and this is why the application of the training set sampling technique forces to re-evaluate the whole population each time the crossover is carried out. Even though, the number of evaluations carried out is reduced.

- **Punch et al.** [68] develop some genetic methods for feature extraction and selection, based on the Siedlecki and Sklansky's initial works for large-scale feature selection problems [71].

  They utilise a GA with binary or real coding, depending on whether the process carried out is feature selection or extraction. In the first case, the coding scheme is that used in Siedlecki and Sklansky's work [71], and, in the second scheme, a real value $r \in [0.0, 10.0)$ is encoded in the i^{th} gen, which represents the weight (or discriminate value) assigned to the i^{th}-variable.

  The evaluation function is a linear combination of the error ratio committed by a K-Nearest Neighbour (KNN) [67] scalable classifier and the neighbour quotient that classifies incorrectly. This makes it more likely that the vote of the majority neighbours is the correct classification.

  Just like in the method proposed by Brill et al., an alternative which improves the method efficiency is suggested. Because most of the required computing time in this method is invested in the chromosome evaluation as well, a GA parallel processing (micro-gain parallelism [52]) in which the individual evaluation is distributed in several nodes is proposed.

This described approach makes a KNN space scaling, such that in the evaluation of each individual, as well as in the final solution interpretation, those variables with a value less than a specific threshold are not considered in the KNN classifier.

- **Ray et al.** [70] propose a characteristic explicit selection and extraction algorithm using GAs. The GA considered works with chromosomes composed of two parts:

  - The first one in which weights are coded as in [68], and

  - a second part representing the feature masking vector. To code this second part two schemes have been used: the traditional one, used in [9] and [71], with binary coding and fixed length equal to the total number of variables, indicating with a 0 the non-selected variables, and with an 1 the selected ones, and therefore, those to which the corresponding weight of the first part of the chromosome will be applied to. On the other hand, an alternative scheme is proposed in which a set of *m* bits is associated to each variable, and it is considered that the feature participates in the classification if the total number of 1s from the set is greater o equal to $\left| m/2 \right|$. This second scheme tries to reduce the great phenotypic variation associated to each genotypic change in a single bit in the first scheme.

  The fitness function is composed of several elements: the error ratio of the KNN classifier, the number of features used in the classification, the number of incorrect votes from the neighbourhood in the classification, and a term that prevents the bias introduced by the existence of different example proportions in the different classes. Therefore, the search is guided to obtaining a minimum characteristic number, which reaches an important percentage of classification, maximising the number of neighbours which correctly classifies and correcting the possible bias caused by different example proportions in the training set (very common in real problems).

Comparing this algorithm with other techniques that independently carry out the feature selection and extraction, and with the algorithm developed by the same authors in [68], which makes the feature selection as a consequence of the feature extraction process, it is realised that the integration of both processes in the search process allows the GA to find interrelationships between data, which are missed when those processes are independently carried out. Nevertheless, we have to point out that the search space is increased, as well as the time needed to find a good solution, and also the possibilities of falling into a local optimum if the GA does not receive enough information to correctly orientate the search.

- **Yang and Honavar** [79] propose a binary coded GA with the same coding scheme used in [9], [51] and [71], the classical crossover and mutation, and a fitness function that combines two objectives, the maximisation of the classification rate obtained by a fast constructive neural network learning algorithm [78] and the cost of performing the classification. The latter aspect depends on the real problem considered and can include the cost of measuring the value of a particular feature or the risk involved, among others.

3.3 Genetic Feature Selection Methods Based on the Hybridisation of Wrapper and Filter Models

Bala et al. [3] develop a feature selection genetic method with aspects from the wrapper and filter models.

The GA uses binary coding (as in [9], [51], [71] and [79]) and a fitness function that linearly combines three independent measures: a theoretical information measure which estimates the discriminatory power of each variable -using an entropy measure-, a measure of the selected feature extraction cost, and the classification ratio obtained by a decision tree, built by means of C4.5 [69] using the selected variables. In this way, it is got a hybrid model, in which the GA looks for inside the feature subset genotypic space, and the inductive C4.5 explores the decision tree phenotypic space.

4 Genetic Fuzzy Rule Learning

An inductive rule learning process, which starts from an example set with known classification, obtains an RB that properly describes the problem classification mechanisms. GAs have been used to evolve sets of rules, which describe concepts or classes from three different points of view:

- *The Michigan approach*, that considers a chromosome as a single rule, and represents the RB by the entire population.

- *The Pittsburgh approach*, in which each chromosome encodes a whole RB.

- *The Iterative Rule Learning approach*, that considers, as the Michigan approach does, each chromosome like a single rule, but contrary to the latter, the solution is obtained by only considering the best individual found in each GA run, i.e., the GA provides a partial solution to the learning problem each time it is run.

The following subsections will describe these learning models and their applications to FRBCS design.

4.1 The Michigan Approach

This learning model is based on a GA in which the chromosomes represent single rules and the whole population an RB. Classification Systems developed in this way are called, in the specialised bibliography, Classifier Systems (CSs), and are obtained as a consequence of the evolution of the rules over time, by means of credit assignment, rule discovering and genetic operations applied at the level of individual rules.

The learning algorithms which employ this approach contain three main components:

1. A *performance system* that interacts with the environment.

2. A *credit assignment system*, which learns the reward associated with the individual rules with respect to the global behaviour of the RB obtained by the population. Traditionally, two different schemes have been used to assign credits: the Bucket Brigade algorithm [39], and the Profit-Sharing plan [31].

3. A *classifier discovering process*, that generates new rules from a set of rules by means of GAs.

It was Holland [37] who introduced the first ideas about CSs, and developed, jointly with Reitman in [38], the first CS, named CS-1. Since that moment, different non-fuzzy CSs have been proposed in the specialised literature ([24], [59], [76] and [77]), and also some algorithms based on this approach for designing Fuzzy Rule-Based control Systems ([8], [64], [66], [73] and [74]).

As regards the Michigan approach to design FRBCSs, we may highlight the method proposed by Yuan and Zhuang in [80] to generate fuzzy rule bases that maximise the consistency [60], correctness and concision properties.

This algorithm learns fuzzy rules (in the modified normal disjunctive form proposed by Michalski [60]) in which the variables from the antecedent, as well as that which describes the class, are linguistic variables. Each rule is represented by a chromosome with binary coding, distinguishing a segment per variable in it (using as many bits as linguistic labels have the variable). This coding scheme allows the system to evolve in the learning of rules not necessarily composed of all the initial linguistic variables, determining the most relevant variables for each one of the rules.

The initial population comes from a random generation (50 %) and from rules generated considering the training examples (50 %). This combination provides a better starting point for reproduction than the random population, because combines accurate and specific rules generated from training examples, with rules randomly generated that may contain some information not covered by these examples.

Over this population, two classic operators are applied: the two-point crossover and the standard random mutation. Both operators are used in a segment level, to achieve that the genetic alterations are produced between units with a common meaning. Also, the cross is produced between two randomly selected individuals of the same species (i.e. two rules with the same consequent). This limitation is due to the belief that interspecies crossover tends to generate low-performance offsprings [37].

After the application of these operators, a viability check is performed to eliminate redundant offsprings (for example to eliminate an individual that represents a rule covered by another rule in the population) and therefore to improve the quality of new generations.

The genetic fuzzy rule learning process developed by Yuang et al. starts from a predefined fuzzy partition and selects the most representative variables for each rule. The fitness function joins three goals: to maximise the accuracy of the rule, the covering of the rule and its contribution in the correct classification of the examples with respect to the action of the whole RB. The last objective avoids the necessity of a credit assignment mechanism present in classical CSs, and orientates the search to a co-operative set of rules. This fact increases the learning process efficiency.

4.2 The Pittsburgh Approach

In this model each chromosome encodes a whole RB [72]. The algorithms based on this philosophy are described easily because the learning method is the description of the GA aspects:

- *Search space and representation of rules*. These genetic learning methods use two languages for representing the rules: the modified disjunctive normal form, proposed by Michalski [60], and the VL1 language, subsequently proposed by Michalski et al. [61]. For these languages, in most cases, chromosomes of variable length are used because the number of rules needed to describe the system is unknown.

- *Evaluation function*. As each chromosome encodes a whole classifier set, credit is assigned to the complete set of rules via interaction with the environment and usually depicts concepts from the classic learning theory like consistency and completeness [60].

- *Genetic operators*. These processes need to redefine the traditional genetic operators with the aim of adapting them to the problem of learning concepts and to the variable-length and position-independent genomes. The crossover operator serves to provide a new combination of rules and the mutation operator gives new rules. Furthermore, new operators are introduced to provide a faster evolution of the system towards suitable solutions as operators of generalisation and specialisation of rules [21], or adaptive operators, which work in a different way depending on the evolution level of the system ([32] and [46]).

The Pittsburgh approach was initially proposed by Smith [72]. Recent examples of it in a non fuzzy environment are the GABIL [21] and GIL [47] systems. This approach has not been applied to the generation of fuzzy rules for FRBCSs (see [10], [36] and [56] for some examples of its application in the design of Fuzzy Rule-Based Control Systems), although its basic philosophy has been used in RB selection and tuning processes, as we will describe in Sections 5 and 6.

If we compare both classical genetic approaches to rule learning, we see that the roles of GAs in the Pittsburgh and Michigan approaches are rather different, and the distinction arises from the difference in the level at which the GAs are applied [10].

The Michigan approach has proven to be most useful in an on-line, real-time environment in which radical changes in behaviour cannot be tolerated, whereas the Pittsburgh approach is more useful for off-line environments in which more leisurely exploration and more radical behavioural changes are acceptable [21].

González and Herrera [28] reported that the major problem with the Michigan approach is that of resolving the conflict between the individual and collective interests of rules within the system. The ultimate aim of a Classification System learning process is to obtain a set of co-adapted rules which act together in solving some problem. In a Michigan style system, with selection and replacement at the level of the individual rule, rules which co-operate to effect good actions and receive payoff also compete with each other under the action of the GA. Moreover, in FRBCSs with an FRM that considers the information provided by all rules or by a subset of rules of the RB, this problem is augmented.

With Pittsburgh-style Classification Systems such conflict between individual and collective interest of individual rules does not arise since reproductive competition occurs between complete RBs rather than individual rules. However, maintenance and evaluation of a population of complete rule sets in Pittsburgh-style systems can often lead to a much greater computational burden (in terms of both memory and processing time). Therefore, problems with the Pittsburgh approach have proven to be, at least, equally challenging. Although the approach avoids the problem of explicit competition between rules, large amounts of computing resources are required to evaluate a complete population of RBs.

4.3 The Iterative Rule Learning Approach

In recent bibliographies, a new learning model based on GAs and named the iterative learning approach ([14], [28], [30] and [75]) appears as an alternative to the Michigan and Pittsburgh approaches. This new model considers, as does the Michigan approach, that each chromosome in the population represents a single rule, but contrary to the Michigan approach, only the best individual is considered as the solution, discarding the remaining chromosomes in the population. Therefore, in the iterative model, the GA provides a partial solution to the problem of learning.

Its basic working scheme in order to obtain a set of rules that describes the classes represented in the examples is [30]:

0. Let *E* be a set of training examples which describes the behaviour in the past time of the system to be learnt. Let GLOBAL and PARTIAL be two sets of rules, initialised to the empty set.

1. Select a class *B* which has not been learnt.
2. While PARTIAL is not complete and is composed of consistent solutions,
    a) Find a consistent rule *d* by means of a GA.
    b) Add *d* to the set of rules PARTIAL.
    c) Penalise those rules belonging to PARTIAL, eluding the possibility of being chosen in subsequent steps.
3. Add the new rules to GLOBAL.
4. Delete the rules from PARTIAL.
5. If there are more classes to be learnt, go to step 1. Otherwise, return GLOBAL as the solution of the classification problem.

If we compare this approach with the Michigan and Pittsburgh models, we see that the iterative genetic model, as well as the Pittsburgh model, does not require any adaptation of the learning algorithm. Furthermore, it considerably reduces the size of the search space because of its decomposing scheme of the problem. This reduction of the search space, with respect to the two general models studied in the last sections, is shown in Table 1 [75]. This table shows, for a problem with *M* classes, *N* variables, *V* values per variable, and needing *k* rules to describe the system, the size of the search space for the GA in each one of the three approaches.

Table 1. Size of the search space for different genetic learning models.

Genetic model	Size of the search space
Pittsburgh	$k^{M(V+1)^M}$
Michigan	$M(V+1)^N$
Iterative	2^N

The first algorithms that have used this approach to develop a Classification System are SIA (Supervised Inductive Algorithm) [75] for learning crisp Classification Systems, and SLAVE (Structure Learning Algorithm in Vague Environment, [25] and [30]), developed for designing crisp or fuzzy rule-based systems, used both for control and classification problems. Herrera et al. [14], and [18] and González and Pérez [27] use this model for control problems. The latter authors describe in [28] a multi-stage genetic learning process for fuzzy systems.

As mentioned, SLAVE is a system for FRBCS design, developed by González and Pérez [30], based on the basic scheme of the iterative model for rule learning that demands the verification of the completeness and consistency properties to the KB. It is known that both properties are difficult to verify in a real-world classification problem. Furthermore, if a system which works with fuzzy rules, as SLAVE does, is considered, the condition of consistency turns to be a degree problem. Due to this, González and Pérez [30] define the condition of *soft consistency* -which allows a number of negative examples, which is a percentage of the number of positive examples-, and the *weak completeness* that attempts to determine when the number of examples covered by a set of rules on a fixed class is sufficient to represent this class.

The main components of the genetic learning algorithm considered in Step 2(a) in SLAVE are the following:

- It learns rules in a modified disjunctive normal form, as happens in the genetic learning process proposed by Yuan and Zhuang [80], but the difference is that the consequents are crisp. It utilises the same binary coding scheme but this algorithm does not only allow us to carry out a learning of the more significant variables for each rule, but also to make a change in the granularity of the domain, combining different elements [30].

- The initial population is obtained from the most specific rules generated from a random subset of the training set to orientate the search.

- It uses the two point crossover operator and the random mutation genetic operators applied at the bit level. Also, other specific operators for the classification problem are included: a modified release of the traditional inversion operator, the rotation operator [27], which produces an increase in the level of diversity of the population; a generalisation operator, which tries to obtain stable variables, eliminating their unstable zones [30]; and the And - Or operators, two special crossover operators, which try to represent concepts of generalisation and specialisation between two individuals from the population.

- The evaluation function provides a measure of the consistency, completeness, simplicity and description of the rules.

- The stopping condition is carried out in different ways: when a maximum number of iterations is reached, or when the population does not improve during a maximum number of generations.

As mentioned, we use in this contribution, as example base, Fisher's Iris database [23] composed of a set of 150 examples of iris flowers with four attributes (petal and sepal width and length) and three classes (setosa, versicolor and virginica). This example base is available by anonymous FTP to `ics.uci.edu` in the directory `pub/machine-learning-databases`.

González and Pérez [30] show the test results obtained by SLAVE on the IRIS example base, using as error estimation technique random resampling, comparing them with those obtained with other classical learning algorithms: C4.5 [69] and BP [33].

Table 2. Test results for IRIS.

Algorithm	Test
SLAVE	95.43
C4.5	91.13
BP	91.56

One of the more important characteristics of this learning algorithm is the high linguistic description level of the obtained RB. The coding scheme used, along with the genetic operators, make possible to obtain an RB with a low cardinality, in which the defined partition granularity is fitted to the information represented by the training example set.

In the following, an RB obtained by SLAVE for the Iris problem and composed of three rules is shown [30].

```
IF  Petal_l is Very_low
THEN Class is Setosa
IF Petal_w is High or Very_high
THEN Class is Virginica
IF Petal_l is Medium AND Petal_w is less or equal to
Medium
THEN Class is Versicolor
```

González and Pérez [29] propose a modification of the learning process to include a feature selection in it, thus obtaining an embedded selection method (see Section 3). In the proposed GA, each individual in the population is represented by two chromosomes: the *chromosome in the variable level*, and the *chromosome in the value level*. The second component follows the rule representation coding scheme previously described. The first can use one of the following schemes: binary coding (representing with 1 the present variables, and with 0 those that do not appear), or real coding (showing the variable activation probability and considering that a variable will be present in the rule when this probability is greater than a given threshold). This real coding scheme allows us to use the information provided by the training set. Also, it is possible to have small bit genotypic changes that do not cause a strong phenotypic change.

It can be seen that the selection process is carried out in the rule level, and due to this, it can not be used in a situation where the aim is to reduce the training data acquisition cost. Nevertheless, it selects variables at two levels, the variable and value, and this leads to a more accurate selection of the variables participating in a certain classification rule.

5 Genetic Fuzzy Rule Selection

The selection of a set of non-redundant rules with a high degree of co-operation amongst them is an important problem in the design of an FRBCS. Furthermore, in many of the real classification problems the number of variables involved is high. As a result, any generation method will give an RB with high cardinality.

In the specialised bibliography, two genetic methods for selecting fuzzy rules in FRBCSs have been proposed: the genetic rule selection process developed by Ishibuchi et al. [43], and the multiselection genetic process for FRBCSs, developed by Cordón et al. [15]. Both are described in the following subsections.

5.1 Genetic Rule Selection Process [43]

Ishibuchi et al. in [41], [42], [43], and [44] develop a genetic process for simplifying fuzzy rule bases used in classification. This process does not modify the semantics of the fuzzy rules, and starts from an RB obtained from a learning method which, from different fuzzy partitions of the attribute domains, generates in each zone of the space a rule whose consequent is delimited by the examples of such subspace.

The proposed rule selection GA has the following characteristics:

- It follows the philosophy of the Pittsburgh approach: each chromosome represents a complete RB. It uses integer coding, which notes with an 1 the rules that are present in the RB, with a -1, those which are not present, and with a 0, the rules not generated.

- It uses the random crossover in one point in [43] and the uniform crossover in [42], as crossover operator.

- The mutation operator used in their first works is the simple random mutation in [42], replaced by a biased mutation operator that enhances the elimination of rules in [43].

- The fitness function, which determines the goodness of the RB, is in [41] a weighted combination of two objectives: maximising the number of examples that are correctly classified and minimising the cardinality of the RB. In [42], the second objective is removed and the minimisation of the sum of the linguistic terms used by the variables is added. In this way, the fuzzy rules in a coarse fuzzy partition are enhanced, because from the point of view of knowledge acquisition, a fuzzy rule with linguistic labels from a coarse fuzzy partition is a general rule that can be valid in a large subspace of the pattern space. In [44], different values are given to the weights of the weighted combination in each generation, thus orientating the search of the GA in different directions simultaneously.

Ishibuchi et al. [44] consider that the basic approach to multiobjective optimisation problems is to try to find not a single solution but a set of non-dominated solutions. The final solution should be determined by the decision maker from the non-dominated solutions depending on his/her preference. Due to this, a modification of the proposed selection algorithm is suggested, transforming it into a multiobjective GA that obtains a better set of non-dominated solutions for the classification problem. Furthermore, the genetic method may include a reinforcement learning process of the certainty degree included in the consequent of the rules, thus obtaining a hybrid algorithm, which increases the classification power of the FRBCS.

The results obtained by the proposed genetic selection method in the Iris problem with consideration to the fitness function proposed by Ishibuchi et al. in [42] are shown in Table 3.

Table 3. Classification results for Iris.

Algorithm	Correct classification	NR
GA with uniform crossover and bias mutation	99.20	6.2
GA with one-point-crossover and bias mutation	99.47	5.8

In Table 4, some non-dominated solutions obtained by the two-objective GA developed by Ishibuchi et al. in [44] are shown.

Table 4: Classification results for Iris.

Correct classification	94.67	97.33	98	98.67
Number of rules	3	4	5	6

The final solution should be selected by human users from four non-dominated solutions in the last table depending on preference.

Note that the results shown in Tables 3 and 4 are obtained providing all the available examples (150) to the proposed selection GA as training examples.

5.2 Genetic Multiselection of FRBCSs [15]

Cordón et al. in [13] and [15] develop a GA-based process that not only selects the best subsets of fuzzy rules, but also learns the best set of linguistic hedges for the linguistic variables co-operating with the FRMs. Therefore, this rule selection process includes a local tuning of the membership functions used by the linguistic variables.

The genetic multiselection process obtains different simplified KBs, with the best co-operation between the rules. It includes:

- The Sequential Niche Technique [6] to induce niches [20], using as basic optimisation technique the genetic selection process proposed in [35], iterated in each run of the multiselection process.

- A search process that looks for the best set of modifiers or linguistic hedges associated with the linguistic labels of the variables.

- A local search posterior to each selection process, so that for the best individual, i.e., the best KB, it looks for the best modification, adding or eliminating a rule and/or modifying a linguistic hedge.

Different KB definitions are obtained by selecting the rules best co-operating from the initial fuzzy rule set and by selecting the best hedges for these rules, by means of the use of the above mentioned subprocesses.

In the following subsections the genetic method and the composition of the multiselection process are analysed.

5.2.1 The Basic Genetic Selection Method

The genetic selection process eliminates unnecessary rules from the RB and looks for the best set of hedges modifying these fuzzy rules. The learning of the hedges may be carried out from two different points of view:

- To obtain a hedge for each fuzzy set related to a linguistic label in the fuzzy partitions of the DB. In this case, this set of hedges is shared for all rules in the RB.

- To obtain the best set of hedges for each fuzzy rule in the RB.

In the first case, the semantic related to the linguistic variables is uniform for all rules and it is specified in the DB. In the second, the meaning is specific for each individual rule, but it keeps the descriptive nature of the FRBCS. In the following, the first kind of hedges will be referred to as Hedges I, and the latter as Hedges II.

The selection process is based on a GA and uses the philosophy of the Pittsburgh approach, because all the information relative to the KB is coded in a chromosome. The selection of individuals is developed using the stochastic universal sampling procedure together with an elitist selection scheme, and the generation of the offspring population is put into effect by using the classical binary multipoint crossover (performed at two points) and uniform mutation operators.

The coding scheme generates fixed-length chromosomes with two outstanding parts, one related to the selected rules and the other referring to the hedges associated with the linguistic labels. Considering an RB with *m* fuzzy rules, and depending on the hedge learning process to be carried out, there are two different coding schemes:

- *Hedges I*: The chromosome length is $h = m + \sum_{i=1}^{N} l_i$, with l_i being the number of linguistic labels for the variable *i*. A chromosome $C_j = (c_i, \ldots, c_h)$ is divided into two parts: The first one has as many binary genes as rules exist in the RB, i.e., *m* genes. $c_1, \ldots, c_m$ represents a subset of candidate rules to form the RB finally obtained as this stage output, *B*, such that,

  $$\text{If } c_i = 1 \text{ then } R_i \in B \text{ else } R_i \notin B$$

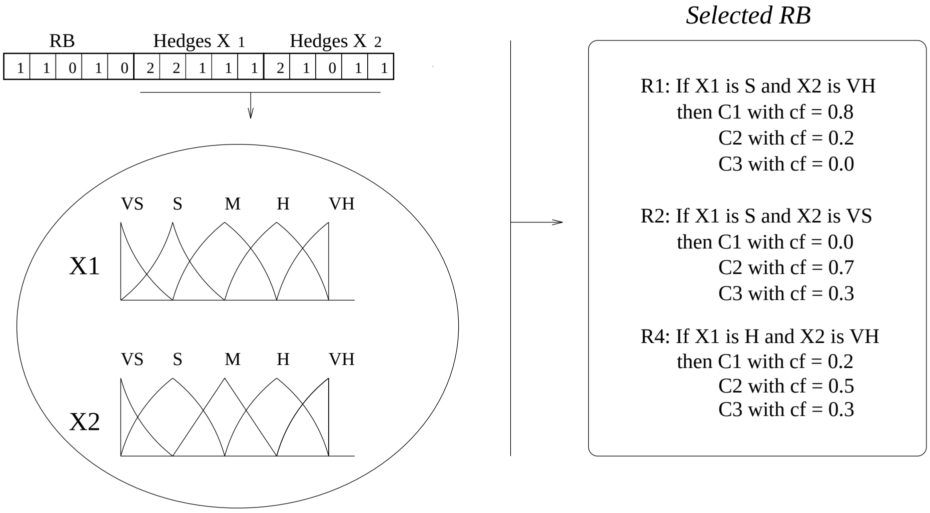
  In addition, the second part has as many genes as different linguistic terms are considered for each variable. For these genes, the number of digits considered as values will be equal to the number of different hedges taken into account. For instance, if we use the linguistic hedges "more or less" and "very", we could code the information of each gen with one of the following values: 0, if the linguistic term does not have any modifier, 1 if it has the modifier "more or less", and 2, if the hedge is "very".

  In Figure 4, this coding scheme and the resulting KB (with an RB composed of rules with a certainty degree for each one class in the consequent) are described, representing the values 1 and 2 the modifiers *more or less* and *very*, respectively, and the value 0 being associated with the original membership function without hedges.

- *Hedges II*: The chromosome length is $h' = m \cdot (N + 1)$, with *N* being the number of variables. The chromosome is again divided into two parts. In the first one we follow the coding scheme presented in the latter point. The $m \cdot N$ remaining genes represent the hedges for each of the rules.

  In Figure 5, this coding scheme as well as the type of resulting KB are described.

In both cases, the initial population is generated by introducing a chromosome representing the complete previously obtained rule set, that is, with all $c_i = 1, i \in \{1, \ldots, m\}$, without hedges. For each type of value of hedge considered, a chromosome representing the complete RB, and with all the genes that code the linguistic hedges with the value of the mentioned hedge, is included. The remaining chromosomes are selected at random.

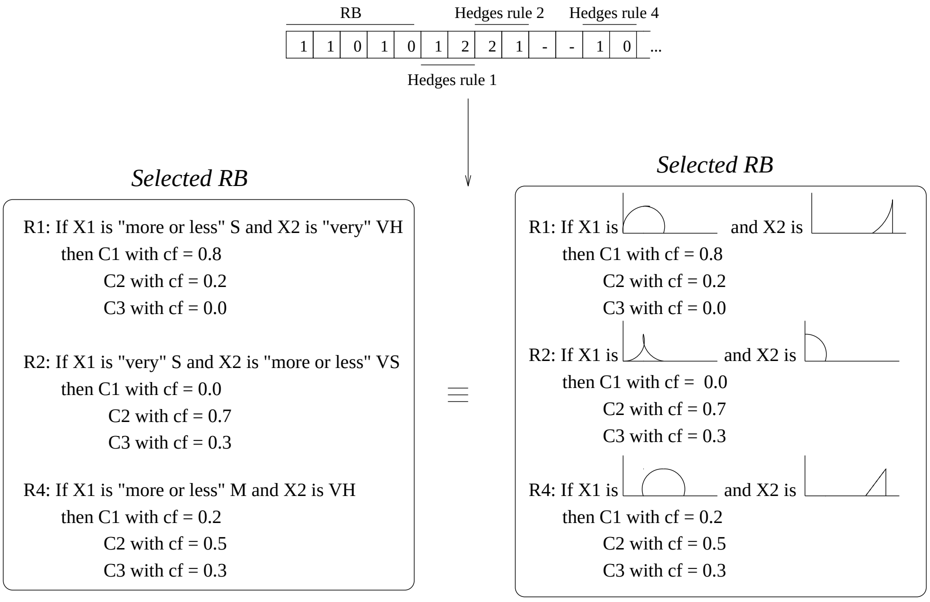


R1: If X1 is S and X2 is VH
then C1 with cf = 0.8
C2 with cf = 0.2
C3 with cf = 0.0

R2: If X1 is S and X2 is VS
then C1 with cf = 0.0
C2 with cf = 0.7
C3 with cf = 0.3

R4: If X1 is H and X2 is VH
then C1 with cf = 0.2
C2 with cf = 0.5
C3 with cf = 0.3

Figure 4. A chromosome with type I hedges.



R1: If X1 is "more or less" S and X2 is "very" VH
then C1 with cf = 0.8
C2 with cf = 0.2
C3 with cf = 0.0

R2: If X1 is "very" S and X2 is "more or less" VS
then C1 with cf = 0.0
C2 with cf = 0.7
C3 with cf = 0.3

R4: If X1 is "more or less" M and X2 is VH
then C1 with cf = 0.2
C2 with cf = 0.5
C3 with cf = 0.3

R1: If X1 is and X2 is
then C1 with cf = 0.8
C2 with cf = 0.2
C3 with cf = 0.0

R2: If X1 is and X2 is
then C1 with cf = 0.0
C2 with cf = 0.7
C3 with cf = 0.3

R4: If X1 is and X2 is
then C1 with cf = 0.2
C2 with cf = 0.5
C3 with cf = 0.3

Figure 5. A chromosome with type II hedges.

With regard to the fitness function, $F(\cdot)$, it is based on the classifier's error rate over a training data set using a certain FRM, demanding the verification of the k-consistency property [30].

5.2.2 The Multiselection Genetic Process

The multiselection genetic process takes as a base the Sequential Niche Technique [6] for inducing niches in the search space to obtain different KB definitions [15]. In each stage, the genetic selection process proposed in the last subsection is used.

Each time the genetic selection process obtains a new KB definition, the multiselection one penalises the search space zone where it is located in order that it will not be selected in future runs. A genotype sharing scheme [20] is used to penalise individuals according to their space proximity to the previous solutions found. To do so, there is a need to define a *distance metric* which, given two individuals, returns a value of how close they are. In this genetic multiselection process, chromosomes are not binary encoded because of their second part, which encodes the linguistic hedges. Therefore, the use of the following distance function is proposed: With $A = (a_i, \ldots, a_h)$ and $B = (b_1, \ldots, b_h)$ being two individuals, it is defined as follows:

$$D(A, B) = \sum_{i=1}^{h} d_i$$

$$d_i = \begin{cases} 1, & \text{if } a_i \neq b_i \\ 0, & \text{otherwise} \end{cases}$$

Making use of this distance function, the *modified fitness function* guiding the search on the multiselection process is based on modifying the value associated with an individual by the basic algorithm fitness function, multiplying it by a *derating function*, $G(C_j, S)$, penalising the closeness of this individual to the solutions S previously obtained. We consider the following function taking into account the fact that the problem to be dealt with is one of minimisation:

$$G(C_j,S)=\begin{cases}\infty, & \text{if } d=0\\ 2-\left(\dfrac{d}{r}\right)^{\beta}, & \text{if } d<r \text{ and } d\neq 0\\ 1 & \text{if } d\geq r\end{cases}$$

where d is the minimum value of the distance between C_j and the solutions s_i included in *S*, i.e., $d=\mathrm{Min}_i\{H(C_j,s_i)\}$, and the penalisation is considered for the most close solution, *r* is the *niche radius*, and β is the *power factor* determining how concave $(\beta>1)$ or convex $(\beta<1)$ the derating curve is. Therefore, the penalisation given by the derating function takes its maximum value when the individual C_j encodes one of the solutions already found. There is no penalisation when the C_j is far away from *S* in a value greater than or equal to the niche radius *r*.

Moreover, a local search algorithm is considered to individually optimising each of the KB definitions obtained, inserting or eliminating a rule and/or changing a hedge, changes that will lead to improvement in KB behaviour. It may be observed that this is a very simple and quick optimisation process.

The local search is carried out at the end of each iteration stage in the multiselection process. It is divided into two phases: First of all, the rule selection is optimised by means of a search in the RB space with distance one to the optimum, i.e., with one rule more or one less in the RB obtained as a result of one of the iterations of the multiselection process. To reduce the search space, when the RB part is optimised, the best set of hedges with distance one to the set of hedges which belongs to the KB represented by the optimum is looked for.

The algorithm of the genetic multiselection process is shown below:

1. Initialisation: Equate the multiselection modified fitness function to the basic selection fitness function: $F'(C_j)\leftarrow F(C_j)$.

2. Run the basic genetic selection process, using the modified fitness function, keeping a record of the best individual found in the run.

3. Run the local optimisation process to optimise the KB definition generated.

4. Update the modified fitness function to give a depression in the region near this individual, producing a new modified fitness function.

5. If all the adapted KBs desired have not been obtained, return to step 2.

Hence, the number of runs of the sequential algorithm performed (iterations of the multiselection process) is the number of solutions to be obtained, i.e., the number of selected KBs to generate, value determined by the FRBCS designer.

In Table 5, some test results for Iris achieved with this multiselection method and with five different FRMs (see also [16] and [17]) are shown. In this table, the type of hedge learning from which the results are obtained is noted with I, II or nothing if the result corresponds to a multiseleccion process without hedge learning. To describe the effects of the multiselection process, the results obtained with the original RB (in which the consequent is composed of a class and a certainty degree) are shown in columns 2 and 3. The error estimation technique used is random resampling.

Table 5. Test results for Iris.

	Initial RB		Multiselection		
FRM	Test	NR	Hedges	Test	NR
Normalized sum	96.22	70		96.71	48
Quasiarithmetic mean	95.21	70	II	96.18	48.2
QuasiOWA	95.21	79	I	95.70	42.6

6 Genetic Tuning

One of the more difficult aspects to specify in the design of any Fuzzy Rule-Based System (FRBS), and at the same time, one of the best determinants of its accuracy, is the fuzzy partition used. If the set of rules has been obtained by an expert or been generated with a supervised inductive learning process, it is necessary to carry out a tuning of the used membership functions to obtain a system with better performance. This tuning could be performed at two levels:

- In the DB level, modifying the parameters that define the membership functions of the linguistic labels in a common way for all the rules. This tuning process maintains the descriptive character of the resulting system.

- In the KB level, tuning for each rule the parameters that define the membership functions of the linguistic labels. Therefore, the semantic of the linguistic terms depends on the specific rule in which such terms appear. The RB obtained will have an approximate behaviour ([14] and [18]).

Both approaches have been used in the genetic tuning of FRBSs for modelling and control. Karr [48] uses a GA to learn the parameters of the fuzzy sets with triangular membership functions related to each one of the linguistic terms of the DB. Herrera et al. [34] propose a genetic process to tune trapezoidal and triangular membership functions, coding with real parameters the complete definition of the KB in a chromosome and obtaining a control FRBS of approximate type.

As was mentioned in Section 1, Classification Systems are frequently used as support systems in decision making processes, and this is why they are usually designed with a descriptive approach. As a result of, there are more processes developed for the descriptive than the approximate tuning. Among them, we may highlight two genetic tuning methods for FRBCSs: Ishibuchi et al.'s approach [45] and the one proposed by Cordon et al. in [14] and [15].

Apart from the descriptive Classification Systems context, there are proposals to optimise the parameters of fuzzy classification rules, which approximate regions of the searching space. Some of them, such as that described in [63], uses a GA with real coding as the base of the tuning procedure of the two parameters for the distance function utilised to determine the fuzzy region to which the example belongs in a greater degree, and subsequently, the fuzzy rule that is fired for its classification.

6.1 Genetic Tuning Method to Obtain the Fuzzy Partitions [45]

Ishibuchi and Murata [45] introduce a genetic process that determines the fuzzy partition of the pattern space for a classification problem. The resulting partition establishes, along with the process of obtaining the consequent described in [40], the set of fuzzy rules that composes the FRBCS. As a result of this, this tuning process is also a fuzzy rule learning process. All in all, the GA simultaneously determines the number of fuzzy rules and the membership function for each fuzzy set belonging to the antecedents.

To carry out this task, a genetic process is developed, using as the coding scheme an extension of that proposed by Nomura et al. [65] for tuning the membership functions of a control system with the following differences:

- It allows an unique label to be a possible solution for the partition of one variable (it will be equivalent to not consider that variable in the corresponding rule).

- It permits trapezoidal membership functions for the labels from the extremes of the variable domain, in contrast to the original proposal designed to tune triangular membership functions. This leads to a greater descriptive power in the rules, as well as a reduction in the number of them needed to obtain a certain level of performance.

The coding scheme generates binary chromosomes of fixed length, with a segment per variable. Each segment has a predefined length which determines the precision of tuning. In the chromosome, an 1 indicates the centre of a triangular membership function, and the extremes of the neighbour membership functions. If a bit extreme of a segment is equal to 0, it indicates that the membership function of this extreme linguistic value is trapezoidal. If a bit $l_i = 1$ then its position is determined by the expression $(i-1)(N_i-1)$, N_i being the pre-established length of the segment. An example of this type of coding for a classification problem with two variables is shown in Figure 6.

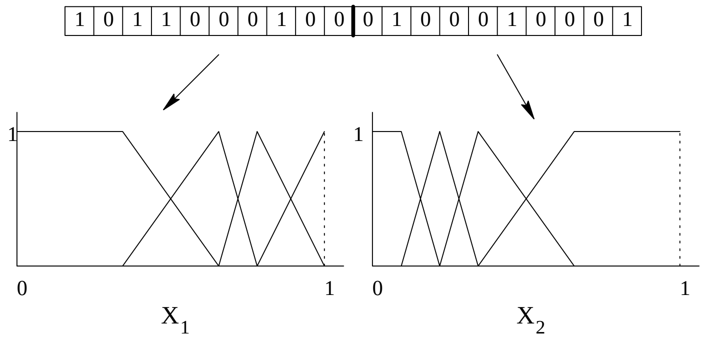


Figure 6. Coding scheme used by Ishibuchi and Murata in [45].

Over a randomly generated population, three recombination operators are used: an operator of multiple crossover between segments, which is equivalent to interchanging the partition of a certain variable between two individuals without losing the meaning; and two mutation operators. The first of them interchanges adjacent bits inside a segment thus making a fine tuning of the partition, that is, a soft modification of the width for the support of the corresponding fuzzy set. The second one changes the value of one bit with different probability, and this is equivalent to adding or removing a linguistic term from the partition of a certain variable. In Figures 7, 8, and 9, the effect of these operators is shown.

In this tuning algorithm, as happens in the one proposed by the same authors for rule selection (see Section 5.1), the fitness function has a double objective: to maximise the number of patterns correctly classified, and to minimise the number of rules.

The process depends, in a high degree, on the length of the segments N_i, which determines the tuning precision, and consequently, the classification percentage obtained by the resulting FRBCS. Although the increase on that length produces a deterioration in the GA efficiency, due to the expansion of the search space, highlighting the necessity for orientating the GA with the available information to produce good results.

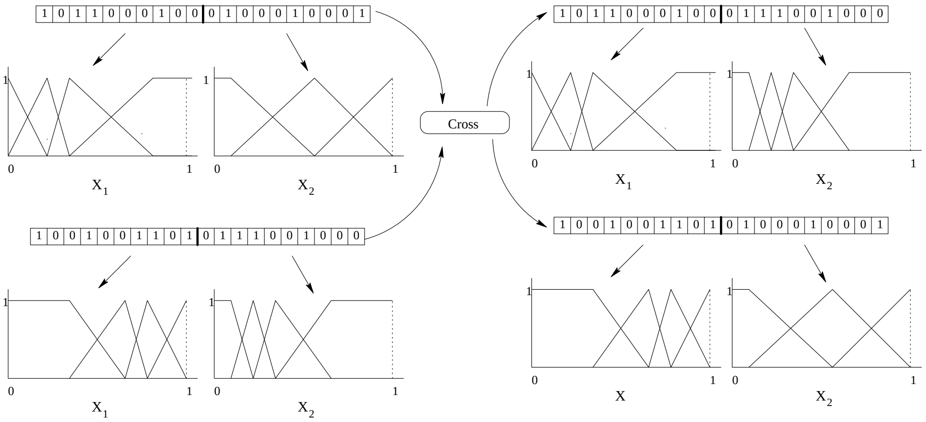


Figure 7. Crossover operator used by Ishibuchi and Murata in [45].

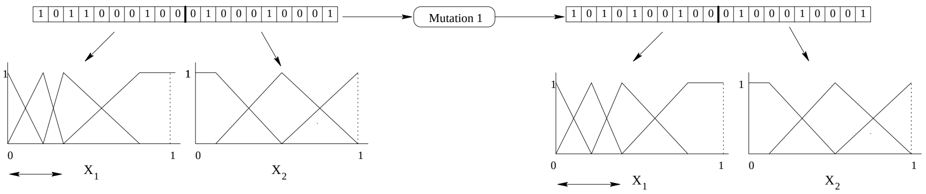


Figure 8. Type 1 mutation operator used by Ishibuchi and Murata in [45].

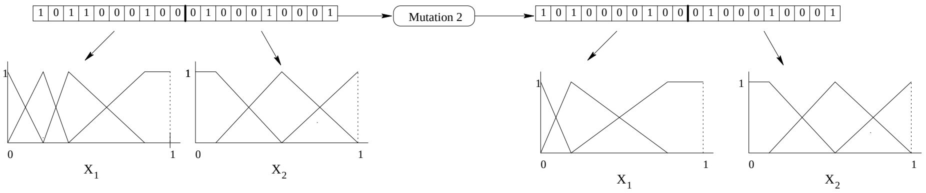


Figure 9. Type 2 mutation operator used by Ishibuchi and Murata in [45].

In Table 6, the results obtained by the described tuning process for IRIS database with different values of N_i are shown [45]. Note that the results correspond to a learning process in which all the available examples (150) have been used to build the FRBCS.

Table 6: Classification results for IRIS.

N_i	Classification rate (%)	NR
6	97.3	6
11	98.0	12
16	97.3	9
21	97.3	18
31	96.0	40

6.2 Genetic Data Base Tuning Process [15]

Cordón et al. in [15] propose a genetic tuning process that optimises the parameters which define the membership of a DB with a predefined granularity. This optimisation process is performed at a superior DB level to retain the linguistic approach of the resulting Classification System and constitutes a solution to the problem of finding the search space partition that best represents the knowledge about the problem, as happens in the process described in the last subsection.

The genetic tuning process starts from a set of predefined fuzzy partitions -uniform and with triangular membership functions, as the one shown in Figure 10- and finds a new set of fuzzy partitions, in which the fuzzy sets are modified in width and location.

Each chromosome forming the genetic population will encode a different DB definition that will be combined with the existing RB to evaluate the individual adaptation. The GA designed for the tuning process presents a real coding issue and uses the stochastic universal sampling as a selection procedure.

It has been mentioned that the primary fuzzy sets considered in the initial fuzzy partitions are triangular-shaped. Thus, each one of the membership functions 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 $3 \cdot L$ real values, with L being the number of terms forming the linguistic variable term set. The complete DB for a problem in which N input 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:

$$C_{ri} = (a_{i1}, b_{i1}, c_{i1}, \ldots, a_{iLi}, b_{iL_i}, c_{iL_i}),$$
$$C_r = C_{r1}\, C_{r2} \ldots C_{rN}$$

During the reproduction phase of the GA, the non-uniform mutation and the max-min-arithmetical crossover operators have been used. The non-uniform mutation operator [62] has a dynamic behavior: the proportion in which a real gene is mutated decreases as the GA's execution advances. The max-min-arithmetical crossover [34] makes use of fuzzy tools in order to improve the GA behaviour in this form: If C_v and C_w are two chromosomes to be crossed, four offspring are generated

$$\begin{gathered} C_1 = aC_w + (1-a)C_v \\ C_2 = aC_v + (1-a)C_w \\ C_3 \text{ with } c_{3k} = \min\{c_k, c_k'\} \\ C_4 \text{ with } c_{4k} = \max\{c_k, c_k'\} \end{gathered}$$

This operator can use a parameter a which is either a constant, or a variable whose value depends on the age of the population. The resulting offspring are the two best of the four aforementioned offspring.

The initial gene pool is created making use of the DB definition of the FRBCS to tune. This is encoded directly into a chromosome, denoted C_1. The remaining individuals are generated by associating a performance interval, $[c_h^l, c_h^r]$, to every gene c_h in C_1, $h = 1 \ldots \sum_{i=1}^{N} L_i \cdot 3$. 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-hand 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:

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

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

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

Figure 10 shows these intervals.

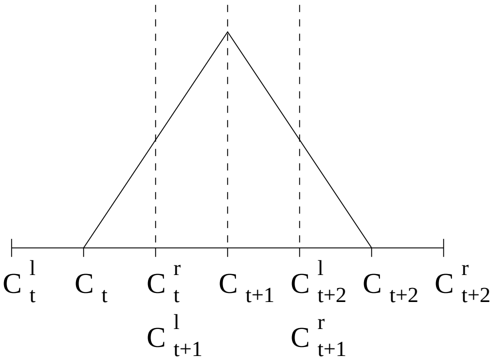


Figure 10. Intervals of performance.

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

Finally, with regard to the fitness function, it is the same one used in the multiselection process presented in Subsection 5.2: if the KB is complete to a τ degree, the function will be equal to classification error, value belonging to [0,1]. Otherwise, it will be equal to 1.

The coding scheme and the tuning algorithm that includes it do not consider variations in the level of granularity established for each variable, in contrast to the one proposed by Ishibuchi and Murata in [45], and shown in the previous section. This is as a result of its objective being the optimisation of a partition defined *a priori* to improve the performance of an FRBCS, whose RB has been previously obtained, either by means of a learning process or an expert, who could not usually define the membership function of the linguistic labels that he/she uses with precision.

In Table 7, the test results obtained for an RB with a class and a certainty degree, with different FRMs are shown ([16] and [17]). To describe the effects of the tuning process we also show the results obtained with the initial KB. Random resampling as error estimation technique is used.

Table 7: Test results for Iris.

	Initial KB		Tuning	
FRM	Test	NR	Test	NR
Classic	94.48	47.4	95.66	47.4
OWA	94.84	44.8	95.76	44.8
QuasiOWA	94.71	49.4	95.62	49.4

7 Summary

Designing a Classification System through a supervised inductive learning process, when the subjacent data probability distribution is unknown is an NP-hard problem to which inductive learning algorithms try to provide an approximate solution. GAs are search algorithms which use operations based on natural genetics, proving theoretically and empirically their robust searching capacity in complex spaces, and hence offering a valid approach for problem solving, as the Classification System design, requiring an efficient and effective search.

GAs have been applied to different FRBCS design aspects, and among them the following can be highlighted:

- Selecting the most informative variable set for the problem to solve, and/or determining the importance of each variable in the classification problem.

- Learning a fuzzy rule set, which expresses the extracted knowledge from the training examples about the problem to be solved, and allowing generalisations to make inferences regarding classes among unknown samples.

- Simplifying the RB, selecting a new RB with co-operation among the rules and with the least redundancy.

- Aspects traditionally grouped into FRBS design under the term "tuning", in which the following are included:

  - Determining the most suitable fuzzy partition for each one of the linguistic variables.

  - Tuning of the predefined membership functions, corresponding to the labels used by the linguistic variables.

  - Tuning of the fuzzy sets used by the fuzzy variables. In this way, the semantic of the linguistic terms depends on the specific rule in which such terms appear and therefore the system has an approximate nature [14].

The work carried out in the FRBCS design shows the utility of GAs in learning processes. The proposals are fundamentally centred in the suitable definition, for the different problem they solve, of three aspects:

- The *coding scheme* that allows us to adequately represent potential solutions. Note that in the different analysed genetic methods an important incidence in the use of the binary coding is observed that makes possible to use traditional recombination operators. Nevertheless, and given that it is known that the GA capabilites do not depend on the utilisation of a binary coding scheme, in feature extraction and tuning proposals a real coding allows representation of problem solutions in a more natural way.

- The *recombination operators* which allow us to adequately explore and exploit the search space, adapting it to the adopted representation. In this form, there are proposals to represent traditional operations in inductive learning process ([30] and [43]); operators that work among units with meaning [80], or even among individuals from the same species, in our case, from the same class [80]; or adaptive operators that work in a different way depending on the evolution level of the GA ([15], [32] and [46]), amongst others. In the latter case, the objective is to guide the GA evolution towards a proper relation between exploration and exploitation. The former cases try to determine the recombination operator actuation at the phenotypic level, to adequately combine the genotypic information, and orientate the GA towards solutions with good results.

- *Mechanisms which improve the GA efficiency*. In most proposals the most resource-consuming task is the individual evaluation. Some approaches have been introduced in two different senses for reducing this computing time:

  – Parallel genetic algorithms: Brill et al. [9] propose a GA with punctuated equilibria, in which different populations evolve in parallel, making a periodic interchange of the best individuals among neighbour subpopulations. Punch et al. [52] developed a GA based on parallel processing, in which the individual evaluation is distributed in several nodes.

  – Alternative techniques of training example utilisation, as the training set sampling technique proposed by Brill et al. [9], in which only a proportion of the training example set is used in each evaluation.

The feature selection is a very important aspect in the Classification System design, because it will determine the design performance, simplicity, linguistic description and complexity. The feature extraction is a process that increases the system accuracy, providing an approximate character to the Classification System that includes it, because it is difficult to linguistically justify the assignation of a real value as discriminative value to each variable considered in the Classification System.

The selection of the most important variables for each rule in the genetic rule learning algorithm, as Yuan et al. [80] and Gonzalez et al. [30] do in their FRBCS inductive genetic learning, is useful and descriptive. Note that this kind of learning does not exclude a previous feature selection, which facilitates the learning and decreases the training data obtaining cost.

Sometimes, the rule generation methods include in the final RB, redundant rules that do not co-operate in the correct classification of new examples, and in these cases the RB selection is important. Cordón et al. in [15] develop a genetic RB selection method which includes a learning of the best linguistic modifier set for the labels used by the linguistic variables. The multiselection genetic process takes as a base the Sequential Niche Technique [6] for inducing niches in the search space to obtain different KB definitions. This selection process is independent from the inductive rule generation method used to obtain the RB. Ishibuchi et al. [42] propose a rule selection method which allows a reduction of the cardinality of the RB, and obtains an RB with fuzzy partitions with different cardinality, which increases the FRBCS generalisation power. It is a rule selection process orientated to the simplification of RBs that are obtained with a rule generation process with multiple partitions.

If the FRBCS has been obtained through an automatic inductive learning process, as well as by an expert, it is usually necessary to carry out tuning of the fuzzy partitions used in the fuzzy rules. This aspect is treated by Ishibuchi et al. [45] and Cordón et al. [15] with good results. Ishibuchi and Murata [45] developed a fuzzy partition determination process, and subsequently, an RB generation. Cordón et al. [15] obtained a tuning process independent on from the way in which the RB is generated.

All these genetic proposals provide adequate results for the partial problem that they solve. Results, as those obtained by Cordón et al. in [13], [15] and [16], point out that a possible solution could be to develop a genetic learning method including everything. Obviously, to develop a GA that simultaneously searches for the most suitable variable set, the best partition for each variable, and the rule set with least redundancy and greatest co-operation is not possible because of the huge size of the search space to explore. This is the reason why a multistage genetic learning process that determines the variable set, their partition and the fuzzy rules in several stages, by means of independent genetic processes, may obtain an FRBCS with best behaviour. The genetic process of each stage can be adapted to the problem to be solved, orientating the search in the best possible way.

Acknowledgments

This research has been supported by CICYT TIC96-0778.

References

[1] Abe, S. and Thawonmas, R. (1997), “A fuzzy classifier with ellipsoidal regions,” *IEEE Transactions on fuzzy systems*, Vol. 5 (3), pp. 358-368.

[2] Bäck, T. (1996), *Evolutionary algorithms in theory and practice,* Oxford University Press.

[3] Bala, J, De Jong, K., Huang, J, Vafaie, and H. Wechsler, H. (1997), “Using learning to facilitate the evolution of features for recognizing visual concepts,” *Evolutionary Computation*, Vol. 4 (3) , pp. 297-311.

[4] Bárdossy, A. and Duckstein, L. (1995), *Fuzzy rule-based modelling with applications to geophysical, biological and engineering systems,* CRC Press.

[5] Battiti, R. (1994), “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on Neural Networks,* Vol. 5 (4), pp. 537-550.

[6] Beasly, D., Bull, D.R., Martin, R.R. (1993), “A sequential niche technique for multimodal function optimisation,” *Evolutionary Computation*, Vol. 1 (2), pp. 101-125.

[7] Bezdek, J.C. and Pal, S.K. Eds. (1992), *Fuzzy models for pattern recognition*, IEEE Press.

[8] Bonarini, A., and Basso, F. (1997), “Learning to compose fuzzy behaviours for autonomous agents,” *International Journal of Approximate Reasoning*, Vol. 17 (4), pp. 433-446.

[9] Brill, F.Z., Brown, D.E. and Martin, W.N. (1992), “Fast genetic selection of features for neural network classifiers,” *IEEE Transactions on Neural Networks*, Vol. 3 (2), pp. 324-328.

[10] Carse, B., Fogarty, T.C. and Munro, A. (1996), “Evolving fuzzy rule based controllers using genetic algorithms,” *Fuzzy Sets and Systems*, Vol. 80, pp. 273-293.

[11] Chi, Z., Wu, J. and Yan, H. (1995), “Handwritten numeral recognition using self-organizing maps and fuzzy rules,” *Pattern Recognition*, Vol. 28 (1), pp. 59-66.

[12] Chi, Z., Yan, H. and Pham, T. (1996), *Fuzzy algorithms with applications to image processing and pattern recognition*, World Scientific.

[13] Cordón, O., del Jesus, M.J., Herrera, F. and López, E. (1997), “Selecting fuzzy rule-based classification systems with specific reasoning methods using genetic algorithms,” *Proceedings of the Seventh International Fuzzy Systems Association World Congress* (IFSA’97), pp. 424-429.

[14] Cordón, O. and Herrera, F. (1997), “A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples,“ *International Journal of Approximate Reasoning*, Vol. 17 (4), pp. 369-407.

[15] Cordón, O., del Jesus, M.J and Herrera, F. (1998), “Genetic learning of fuzzy rule-based classification systems co-operating with fuzzy reasoning methods,” *International Journal of Intelligent Systems*, Vol. 13 (10/11), pp.1025-1053.

[16] Cordón, O., del Jesus, M.J. and Herrera, F. (1998), “Reasoning methods based on OWA operators under fuzzy majority in fuzzy rule-based classification systems,” Technical Report DECSAI-98121, Dept. Computer Science and A.I., University of Granada.

[17] Cordón, O., del Jesus, M.J. and Herrera, F. (1999), "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, Vol. 20 (1), pp. 21-45.

[18] Cordón, O. and Herrera, F. (1999), "Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing fuzzy rule-based systems," *Fuzzy Sets and Systems*. To appear.

[19] Cover, T.M and Hart, R.E. (1967), "Nearest neighbour pattern classification," *IEEE Transactions on Information Theory,* IT-13 (1), pp. 21-27.

[20] Deb, K. and Goldberg, D.E. (1989), "An investigation of niche and species formation in genetic function optimisation," *Proceedings of the Third International Conference on Genetic Algorithms* (ICGA'89), pp. 42-50.

[21] Jong, K.A., Spears, W.M. and Gordon, D.F. (1993), "Using genetic algorithms for concept learning," *Machine Learning*, Vol. 13, pp. 161-188.

[22] Eshelman, L., Caruana, R., and Schaffer, J. (1989), "Biases in the crossover landscape," *Proceedings of the Third International Conference on Genetic Algorithms* (ICGA'89), pp. 10-19.

[23] Fisher, F.A. (1936), "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, Vol. 7, pp. 179-188.

[24] Goldberg, D.E. (1989), *Genetic algorithms in search, optimisation, and machine learning*, Addison-Wesley.

[25] González, A., Pérez, R. and Verdegay, J.L (1994), "Learning the structure of a fuzzy rule: a genetic approach," *Fuzzy Systems and Artificial Intelligence*. Vol. 3, pp. 57-70.

[26] González, A., Pérez, R. and Valenzuela, A. (1995), "Diagnosis of myocardial infarction through fuzzy learning techniques," *Proceedings of the Sixth International Fuzzy Systems Association World Congress* (IFSA'95), pp. 273-276.

[27] González, A. and Pérez, R. (1996), “A learning system of fuzzy control rules,” in: Herrera, F. and Verdegay, J.L. (Eds.), *Genetic algorithms and soft computing*, Physica-Verlag, pp. 202-225.

[28] González, A., and Herrera, F. (1997), “Multi-stage genetic fuzzy systems based on the iterative rule learning approach,” *Mathware & Soft Computing*, Vol. 4 (3), pp. 233-249.

[29] González, A. and Pérez, R. (1997), “Using information measures for determining the relevance of the predictive variables in learning problems,“ *Proceedings of the Seventh IEEE International Conference on Fuzzy Systems*, pp.1423-1428.

[30] González, A. and Pérez, R. (1998), “Completeness and consistency conditions for learning fuzzy rules,” *Fuzzy Sets and Systems*, Vol. 96, pp. 37-51.

[31] Grefenstette, J.J. (1988), “Credit assignment in rule discovery systems based on genetic algorithms,” *Machine Learning*, Vol. 3, pp. 225-245.

[32] Grefenstette, J.J. (1991), “Lamarckian learning in multi-agent environments,” *Proceedings of the Fourth International Conference on Genetic Algorithms* (ICGA’91), pp. 303-310.

[33] Hecht-Nielsen, R. (1989), “Applications of counterpropagation networks,” *Neural Networks*, Vol. 1, pp. 131-139.

[34] Herrera, F., Lozano, M. and Verdegay, J.L. (1995), “Tuning fuzzy controllers by genetic algorithms,” *International Journal of Approximate Reasoning*, Vol. 12, pp. 299-315.

[35] Herrera, F., Lozano M. and Verdegay, J.L. (1998), “A learning process for fuzzy control rules using genetic algorithms,” *Fuzzy Sets and Systems*, Vol. 100, pp. 143-158.

[36] Hoffmann, F. and Pfister, G. (1997), “Evolutionary design of a fuzzy knowledge base for a mobile robot,” *International Journal of Approximate Reasoning*, Vol. 17 (4), pp. 447-469.

[37] Holland, J.H.(1975), *Adaptation in natural and artificial systems*, Ann Arbor: The University of Michigan Press.

[38] Holland, J.H. and Reitman, J.S. (1978), "Cognitive systems based on adaptive algorithms," in Waterman, D.A. and Hayes-Roth, F. (Eds.), *Pattern-Directed Inference Systems*, Academic Press, pp. 313-329.

[39] Holland, J.H. (1985), "Properties of the bucket brigade algorithm," *Proceedings of the First International Conference on Genetic Algorithms* (ICGA'85), pp. 1-7.

[40] Ishibuchi, H., Nozaki, K. and Tanaka, H. (1992), "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, Vol. 52, pp. 21-32.

[41] Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H. (1993), "Genetic operations for rule selection in fuzzy classification systems," *Proceedings of the Fifth International Fuzzy Systems Associations World Congress* (IFSA'93), pp. 15-18.

[42] Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H. (1994), "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets and Systems*, Vol. 65, pp. 237-253.

[43] Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H. (1995), "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, Vol. 3 (3), pp. 260-270.

[44] Ishibuchi, H., Murata, T. and Türksen, I.B. (1997), "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets and Systems*, Vol. 89, pp. 135-150.

[45] Ishibuchi, H. and Murata, T. (1997), "A genetic-algorithm-based fuzzy partition method for pattern classification problems," in Herrera, F. and Verdegay, J.L. (Eds.), *Genetic algorithms and soft computing*, Physica-Verlag, pp. 555-578.

[46] Janikow, C.Z. (1991), "Inductive learning of decision rules from attribute-based examples. A knowledge-intensive genetic algorithm approach," Technical Report 91-030, Dpt. of Computer Science, University of North Carolina.

[47] Janikow, C.Z. (1993), "A knowledge intensive genetic algorithm for supervised learning," *Machine Learning*, Vol. 13, pp. 198-228.

[48] Karr, C. (1991), "Genetic algorithms for fuzzy controllers," *AI Expert*, Vol. 6, pp. 26-33.

[49] Kohavi, R. and John, G.H. (1997), "Wrappers for feature subset selection," *Artificial Intelligence*, Vol. 97, pp. 273-324.

[50] Koller, D. and Sahami, M. (1996), "Toward optimal feature selection," *Proceedings of the 13th International Conference on Machine Learning* (ICML'96).

[51] Lanzi, P.L. (1997), "Fast feature selection with genetic algorithms: a filter approach," *Proceedings of the International IEEE on Evolutionary Computation* (ICEC'97), pp. 537-540.

[52] Lin, S.-C., Punch, W.F. and Goodman, E.D. (1994), "Coarse-grain parallel genetic algorithms: categorization and new approach," *Proceedings of the International IEEE on Evolutionary Computation*, (ICEC'94), pp. 28-37.

[53] Liu, W., Wang, M. and Zhong, Y. (1995), "Selecting features with genetic algorithm in handwritten digits recognition," *Proceedings of the International IEEE on Evolutionary Computation* (ICEC'95), pp. 396-399.

[54] Liu, H. and Setiono, R., (1996) "A probabilistic approach to feature selection: a filter solution," *Proceedings of the 13th International Conference on Machine Learning* (ICML'96).

[55] Lu, H., Sung, S.Y. and Lu, Y. (1996), "On preprocesing data for effective classification," *Workshop on Research Issue on Data Mining and Knowledge Discovery in Databases*.

[56] Magdalena, L. (1997), "A fuzzy logic controller with learning through the evolution of its knowledge base," *International Journal of Approximate Reasoning*, Vol. 16, pp. 335-358.

[57] Mandal, D.P., Murthy, C.A. and Pal, S.K. (1992), "Formulation of a multivalued recognition system," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 22 (4), pp. 607-620.

[58] Mandal, D.P., Murthy, C.A. and Pal, S.K. (1994), "A remote sensing application of a fuzzy classifier," *International Journal of Unicertainty, Fuzziness and Knowledge-Based Systems*, Vol. 2 (3), pp. 287-295.

[59] McCallum, R.A. and Spackman, K.A. (1990), "Using genetic algorithms to learn disjunctive rules from examples," *Proceedings of the Seventh International Conference on Machine Learning* (ICML'90), pp. 149-153.

[60] Michalski, R.S. (1983), *A theory and methodology of inductive learning*, Machine Learning: An artificial intelligence approach I, Morgan Kaufmann, pp. 83-134.

[61] Michalski, R.S. (1986), *Understanding the nature of learning*, Machine Learning: An artificial intelligence approach II, Morgan Kaufmann.

[62] Michalewicz, Z. (1996), *Genetic algorithms + data structures = evolution programs*, Springer-Verlag.

[63] Mitra, S., Pal, S.K. and Barnejee, S. (1995), "Tuning of class membership using genetic algorithms," *Proceedings of the European Conference on Fuzzy and Intelligent Technologies* (EUFIT'95), pp. 1420-1424.

[64] Nakaoka, K., Furuhashi, T., Uchikawa, Y. and Maeda, H., "A proposal on payoffs and apportionment of credits of fuzzy classifier system - finding the knowledge for large scale systems," *Japanese Journal of Fuzzy Theory and Systems*, Vol. 8 (1), pp. 25-34.

[65] Nomura, H., Hayashi, I. and Wakami, N. (1992), “A self-tuning method of fuzzy reasoning by genetic algorithm,” Proceedings of the 1992 International Fuzzy Systems and Intelligent Control Conference, pp. 236-245.

[66] Parodi, A. and Bonelli, P. (1993), “A new approach to fuzzy classifier system,” *Proceedings of the Fifth International Conference on Genetic Algorithms* (ICGA'93), pp. 223-230.

[67] Patrick, E.A. and Fischer III, F.P. (1970), “A generalized k-nearest neighbor rule,” *Information and Control*, Vol. 16, pp. 128-152.

[68] Punch, W.F., Goodman, E.D., Pei, M., Chia-Shun, L., Hovland, P. and Enbody, R. (1993), “Further research on feature selection and classification using genetic algorithms,” *Proceedings of the Fifth International Conference on Genetic Algorithms* (ICGA'93), pp. 557-564.

[69] Quinlan, J.R. (1993), *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers.

[70] Raymer, M.L., Punch, W.F., Goodman, E.D., Sanschagrin, P.C. and Kuhn, L.A. (1997), “Simultaneous feature scaling and selection using a genetic algorithm,” *Proceedings of the Seventh International Conference on Genetic Algorithms* (ICGA'97), pp. 561-567.

[71] Siedlecki, W. and Sklansky, J. (1989), “A note on genetic algorithms for large-scale feature selection,” *Pattern Recognition Letters*, Vol. 10, pp. 335-347.

[72] Smith, S.F. (1980), “A learning system based on genetic algorithms,” Ph. D. Thesis, University of Pittsburg.

[73] Valenzuela-Rendón, M. (1991), “The fuzzy classifier system: motivations and first results,” *Proceedings of Parallel Solving from Nature* (PPSN II), pp. 330-334.

[74] Velasco, J.R. (1998), “Genetic-based on-line learning for fuzzy process control,” *International Journal of Intelligent Systems*, Vol. 13, pp. 891-903.

[75] Venturini, G. (1993), “SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts,” *Lecture Notes in Artificial Intelligence: Machine Learning* (ECML-93), pp. 280-296.

[76] Wilson, S.W. (1994), “ZCS: A zeroth order classifier system,” *Evolutionary Computation*, Vol. 2 (1), pp. 1-18.

[77] Wilson, S.W. “Classifier fitness based on accuracy,” *Evolutionary Computation*, Vol. 3 (2), pp. 149-175.

[78] Yang, J., Parekh, R. and Honavar, V. (1997), “DistAI: An inter-pattern distance-based constructive learning algorithm,” Technical Report ISU-CS-TR 97-05, Iowa State University.

[79] Yang, J. and Honavar, V. (1998), “Feature subset selection using a genetic algorithm,” *IEEE Intelligent Systems*, Vol. 13, pp. 44-49.

[80] Yuan, Y. and Zhuang, H. (1996), “A genetic algorithm for generating fuzzy classification rules,” *Fuzzy Sets and Systems*, Vol. 84, pp. 1-19.

[81] Zadeh, L.A. (1977), “Fuzzy sets and their applications to classification and clustering,” in: J. van Ryzin, (Ed.), *Classification and clustering*, Academic Press, pp. 251-299.