3rd International Workshop on Genetic and Evolving Fuzzy Systems
Witten-Bommerholz, Germany, March 2008

A Novel Genetic Cooperative-Competitive Fuzzy Rule Based Learning Method using Genetic Programming for High Dimensional Problems

Francisco José Berlanga, María José del Jesus and Francisco Herrera

***Abstract*— In this contribution we present GP-COACH, a novel GFS based on the cooperative-competitive learning approach, that uses Genetic Programming to code fuzzy rules with a different number of variables, for getting compact and accurate rule bases for high dimensional problems. GP-COACH learns disjunctive normal form rules (generated by means of a context-free grammar) and uses a token competition mechanism to maintain the diversity of the population. It makes the rules compete and cooperate among themselves, giving out a compact set of fuzzy rules that presents a good performance. The good results obtained in an experimental study involving several high dimensional classification problems support our proposal.**

I. INTRODUCTION

In the design of any Fuzzy Rule-Based Systems (FRBSs) learning method, there are two main, and opposite, goals to be maximized: the accuracy and the interpretability of the extracted knowledge. In the 1990s, it was paid more attention on the accuracy maximization, and different approaches were developed to improve the accuracy of FRBSs, although such improve was usually done at the cost of their interpretability. However, more recent studies ([1], [2]) has pointed out the necessity of the existence of an interpretability-accuracy trade-off in the design of FRBSs.

Such trade-off is more difficult to reach when the problem to be solved presents a high dimensionality, that is, a high number of input features and/or a high number of instances or examples, because of the exponential growth of the fuzzy rule search space that makes the learning process more difficult and, in most cases, leads to an FRBS with a complex rule base (with respect to the number of rules and the structure of each rule).

There exist two main ways to tackle this problem:

1) *Carrying out a feature selection process*, that determines the most relevant variables before or during the inductive learning process of the FRBS. This process reduces the fuzzy rule search space and increases the efficiency and accuracy of the learning stage ([3], [4]).
2) *Compacting and reducing a previously learned rule set in a postprocessing stage*. This kind of methods acts combining rules and/or selecting a subset of them from a given rule set to achieve the goal of minimizing the number of rules used while maintaining (or even improving) the FRBS performance ([5], [6]).

F.J. Berlanga and M.J. del Jesus are with Department of Computer Science, University of Jaén, E-23071 Jaén, Spain {berlanga, mjjesus}@ujaen.es
F. Herrera is with the Department of Computer Science and Artificial Intelligence, University of Granada, E-18071 Granada, Spain herrera@decsai.ugr.es

In this contribution, we tackle the problem of the high dimensionality in Fuzzy Rule-Based Classification Systems (FRBCSs), presenting a method that carries out an embedded feature selection in the rule learning, getting a compact rule base by means of Genetic Programming (GP). GP [7] is a kind of evolutionary algorithm that uses variable-length trees to represent the different individuals in the population, which makes it very suitable for the learning of rules.

Our proposal, GP-COACH (GP-based evolutionary algorithm for the learning of COmpact and ACcurate FRBCS for High dimensional problems), learns disjunctive normal form (DNF) fuzzy rules (generated by means a context-free grammar) and it uses a competition mechanism between rules (the token competition) which simultaneously maintains the diversity into the population and deletes irrelevant rules during the learning process, allowing us to obtain compact FRBCSs (with few rules and conditions per rule) with a high-generalization capability.

The contribution is arranged as follows: In Section II some preliminaries are briefly described: an overview of GFSs for rule induction and a description of some proposals using GP for FRBSs learning. GP-COACH algorithm is completely explained in Section III. In section IV, the results obtained in the experimental study are analysed. Finally, in Section V the conclusions are pointed out.

II. PRELIMINARIES

In this section we will briefly describe some considerations about the use of GFSs in rule induction processes and a description of some proposals using GP for the learning of FRBSs.

A. Genetic Fuzzy Systems for Rule Induction Processes

Fuzzy systems have shown their usefulness in solving a wide range of problems in different application domains. The use of Evolutionary Algorithms (EAs), and particularly Genetic Algorithms (GAs), in the design of fuzzy systems allows us to equip them with the learning and adaptation capabilities. The result of this hybridization between Fuzzy Logic and GAs leads to Genetic Fuzzy Systems (GFSs) [8].

The most determining aspect of any GFS is the genetic representation of the solutions. In this sense, the proposals in the specialized literature follow two approaches in order to encode rules within a population of individuals [8]:

- The "Chromosome = Set of rules", also called the *Pittsburgh* approach, in which each individual represents a whole rule set. Thrift proposes in [9] a method that follows this approach.

- The "Chromosome = Rule" approach, in which each individual codifies a single rule.

In turn, within the "Chromosome = Rule" approach, there are three generic proposals:

- The *Michigan* approach, in which each individual codifies a single rule. This kind of system, usually called a learning classifier system, is rule-based, message-passing system that employs reinforcement learning and the GA to learn rules that guide its performance in a given environment. Casillas et al. propose in [10] a method using this approach, that extends the classical XCS algorithm [11], for fuzzy rules learning.
- The *IRL (Iterative Rule Learning)* approach, in which each chromosome represents a rule, but the solution is the best individual obtained and the global solution is formed by the best individuals obtained when the algorithm is run multiple times. SLAVE [12] is a proposal that follows this approach.
- The *GCCL (Genetic Cooperative-Competitive Learning)* approach, in which the complete population or a subset of it, codifies the rule base. This approach makes necessary to introduce a mechanism to maintain the diversity of the population in order to avoid that all individuals in the population converge to the same area of search space. Juang et al. propose in [13] a method using this approach for fuzzy rules learning.

In this contribution we consider the GCCL approach for learning FRBCSs.

B. GP-based learning of FRBSs

In the specialized bibliography, different methods have been developed to obtain set of fuzzy rules by using GP. An initial work in this topic is Fuzzy GP, developed by Geyer-Schulz [14], which combines a simple GA that operates on a context-free language with a context-free fuzzy rule language. In [15] and [16], Sánchez et al. propose different FRBCSs learning processes by combining GP operators with GA and simulated annealing respectively to establish the membership functions. Bastian propose in [17] a tree representation of fuzzy models which uses GP to simultaneously identify the rule structure and the parameters of the underlying membership functions. In [18], Hoffmann and Nelles propose a GP-based algorithm to identify an optimal partition of the input space into gaussian, axis-orthogonal fuzzy sets. CEFR-MINER [19], proposed by Mendes et al., is a co-evolutionary algorithm which includes a GP-based algorithm to obtain a fuzzy rule set and an evolutionary algorithm to evolve membership functions. Chien et al. learn discriminant functions with fuzzy attributes by means GP in [20]. In [21] Tsakonas investigates the effectiveness of GP-generated intelligent structures in classification tasks by means a context-free grammar that allows to encode several rules per individual in the population (Pittsburgh approach). Finally, the authors propose in [22] and [23] a GP-based algorithm for the learning of interpretable FRBCSs. The differences between this proposal and the one presented in this contribution will be later described in section IV.

III. GP-COACH ALGORITHM

The main features of GP-COACH are the following:

- It uses a context-free grammar that allows the learning of DNF fuzzy rules and the absence of some input features, thus obtaining interpretable rules.
- It follows the *cooperative-competitive* approach, that is, it encodes a single rule per individual and the rule base is formed by the whole population. That makes necessary to use two different fitness functions in GP-COACH:
  - On the one hand, a local fitness function that evaluates the goodness of each one of the different rules in the population of individuals. From now on, we will refer it simply as *fitness function*.
  - On the other hand, a global fitness function that evaluates the goodness of a whole population of individuals (a rule set). From now on, we will refer it as *global fitness score*.

    This last fitness function has been introduced in GP-COACH in order to obtain the best rule set generated during the evolutionary process.
- It includes a mechanism to promote the diversity into the population, in order to avoid that all individuals converge to the same area of search space. Specifically, it uses the *Token Competition* [24] diversity mechanism which makes rules compete among themselves during the evolutionary process, deleting irrelevant rules, and thus giving out a smaller number of rules that present a high-generalization capability.
- Finally, it is important to point out that GP-COACH uses a two level hierarchical inference process because it learns rule sets containing two different types of rules: *primary rules*, which are strong and general rules generated by the genetic operators, and *secondary rules*, which are weaker and more specific rules, generated after the token competition procedure to increase the diversity in the population.

In the following subsections we will explain each one of these components and the algorithm's description.

A. Context-free grammar for learning DNF fuzzy rules

GP-COACH learns DNF fuzzy rules:

$$\begin{aligned} R^k : & \; If \; X_1 \; is \; \hat{A}_{k1} \; and \; \ldots \; and \; X_{n_v} \; is \; \hat{A}_{kn_v} \\ & \; then \; Class \; is \; C^k \; with \; CD^k \end{aligned} \tag{1}$$

where each input variable X_i takes as a value a set of linguistic terms or labels $\hat{A}_{ki} = \left\{ L_i^1 \; or \; \ldots \; or \; L_i^{l_i} \right\}$ joined by a disjunctive operator, whilst the output variable ($Class$) is one of the class labels.

This rule also includes a certainty degree ($CD^k \in [0, 1]$), which represents the confidence of the classification in the class represented by the consequent. In our proposal, this certainty degree is obtained as the quotient S_j/S, where S_j is the sum of the matching degrees for the training examples belonging to class represented by the consequent which are covered by the antecedent of the rule, and S the sum of the matching degrees for all the training examples which are covered by the antecedent of the rule, independently of the class they belong to.

In GP-COACH, these DNF fuzzy rules are generated according to the production rules of a context-free grammar. In Table I, an example of the grammar for a classification problem with two features (X_1, X_2), three linguistic labels per feature (*Low*, *Medium*, *High*) and three classes (C_1, C_2, C_3) is shown.

TABLE I

GRAMMAR EXAMPLE

$Start$	$\longrightarrow$	$[If]$, $antec$, $[then]$, $conseq$, $[.]$.
$antec$	$\longrightarrow$	$descriptor1$, $[and]$, $descriptor2$.
$descriptor1$	$\longrightarrow$	$[any]$.
$descriptor1$	$\longrightarrow$	$[X_1\ is]$ $label$.
$descriptor2$	$\longrightarrow$	$[any]$.
$descriptor2$	$\longrightarrow$	$[X_2\ is]$ $label$.
$label$	$\longrightarrow$	$\{member(?a,\ [L,\ M,\ H,\ L\ or\ M,\ L\ or\ H,\ M\ or\ H,\ L\ or\ M\ or\ H])\}$, $[?a]$.
$conseq$	$\longrightarrow$	$[Class\ is]$ $descriptorClass$.
$descriptorClass$	$\longrightarrow$	$\{member(?a,\ [C_1,\ C_2,\ C_3])\}$, $[?a]$.

B. Evaluating an individual: Fitness function

Each one of the individuals in the population is evaluated according to a fitness function based on the estimation of:

- *Confidence*, which measures the accuracy of an individual, that is, the confidence of the consequent to be true if the antecedent is verified

$$Conf(R^k) = \frac{\mu_{tp}}{(\mu_{tp} + \mu_{fp})} \tag{2}$$

- *Support*, which measures the coverage of the knowledge represented in the individual

$$Supp(R^k) = \frac{\mu_{tp}}{N_{C^k}} \tag{3}$$

where μ_{tp} and μ_{fp} are the sums of the matching degrees for true and false positives, and N_{C^k} is the number of examples belonging to the class indicated in the consequent of the individual (R^k).

Both measures are combined to make up the fitness function in the following way

$$raw\_fitness = (\alpha * Conf) + ((1 - \alpha) * Supp) \tag{4}$$

where α parameter allow us to give more importance to any of these both measures.

Another different fitness function's definition, also based in the concepts of accuracy (confidence) and coverage (support), can be found in [25].

Finally, it is important to point out, that each time that an individual is created or modified it is also necessary to calculate its certainty degree.

C. Evaluating a population: Global fitness score

Global fitness score measure is defined as follows:

$$\begin{aligned} Global\_fitness =& (w_1 * \%Tra) + (w_2 * \overline{\#V}) + \\ & (w_3 * \overline{\#C}) + (w_4 * \overline{\#R}) \end{aligned} \tag{5}$$

where $\overline{\#A} = 1.0 - \#A$ (with $\#A = \{\#V, \#C, \#R\}$). As it can be seen, the global fitness score measure is formed by other four measures: a) *%Tra*, the normalized value of the correct percentage on training examples, b) *#V*, the normalized value of the number of variables per individual (rule) in the population, c) *#C*, the normalized value of the number of labels (or conditions) per individual, and d) *#R*, the normalized value of the number of rules in the population.

Finally, it is important to point out that some weights (w_i) have been included to give more importance to any of these four measures. A configuration obtaining good results for these weights is proposed below in table III. In this configuration, we have given more importance to the first of these four measures, that it, the correct percentage on training examples.

D. Token competition: Maintaining the diversity of the population

Token Competition [24] has been used as mechanism to maintain the diversity in the population in GP-COACH. The token competition idea is the following: In the natural environment, if an individual finds a good place to live (a niche) it will try to exploit this niche and prevent other newcomers from sharing its resources, unless the newcomer is stronger than it is. The other individuals are hence forced to explore and find their own niches. In this way, the diversity of the population is increased.

Based on this idea, it is assumed that each example in the training set can provide a resource called a token, for which all chromosomes in the population will compete to capture. If an individual (i.e. a rule) can match the example, it sets a flag to indicate that the token is seized. Other weaker individuals then cannot get the token.

The priority of receiving tokens is determined by the strength of the individuals. The individuals with a high fitness score will exploit their niches by seizing as many tokens as they can. The other ones entering the same niches will have their strength decreased because they cannot compete with the stronger ones. This is done introducing a penalization in the fitness score of each individual. This penalization is based on the number of tokens that each individual has seized:

$$Penalized\_fitness = raw\_fitness * \frac{count}{ideal} \tag{6}$$

where *raw_fitness* is the fitness score obtained from the evaluation function, *count* is the number of tokens that the individual actually seized and *ideal* is the total number of tokens that it can seize, which is equal to the number of examples that the individual matches.

As a result of token competition, there exist individuals that cannot seize any token. These individuals are considered as irrelevant, and they can be eliminated from the population due to all of their examples are covered by other stronger individuals.

Token competition mechanism makes the number of individual in population can change during the evolutionary process and that is what allows us to obtain compact fuzzy rule sets.

E. Secondary rules: Improving population diversity

Once token competition mechanism has finished, it is possible that exist training examples which have not been covered by any of the rules in the population. The generation of new specific rules covering these examples improves the diversity in the population, and helps the evolutionary process to easily find stronger and more general rules covering these examples.

Therefore, GP-COACH learns rule sets having two different types of fuzzy rules: A core of strong and general rules (*primary rules*) that covers most of the examples, and a small set of weaker and more specific rules (*secondary rules*) that are only taken into account if there not exist any primary rule matching with some of the examples. This two level hierarchical inference process allows GP-COACH to obtain rule sets having a better interpretability-accuracy trade-off.

Finally, it is important to point out that it is also possible that GP-COACH learns rule sets having no secondary rules, because their primary rules are strong enough to cover all the given examples.

F. Genetic operators

GP-COACH makes use of four different genetic operators (each one generating only one child) to generate new individuals during the evolutionary process:

1) *Crossover*: A part in the first parent is randomly selected and exchanged by another part, randomly selected, in the second one. It is important to point out that it is not allowed to choose cut points in the middle of a disjunction of labels.
2) *Mutation*: It operates on label sets level. A variable in the rule is randomly chosen and then one of the next three different actions is done (randomly chosen):
   a) A new label is added to the label set.
   b) A label is removed from the label set.
   c) A label in the label set is exchanged by another one not included in it.
3) *Insertion*: It looks for all the variables in the rule and adds another different one that was not been included in it with a linguistic label set randomly chosen, although it must have at least one label and it must be different from the ”any” set (see Table I).
4) *Dropping Condition*: Dropping condition randomly selects one variable in the rule and then turns it into ”any”. The label set associated with this variable is also removed. The variable is no longer considered in the rule, hence, the rule can be generalized.

G. Reproduction stage

It is important to point out that GP-COACH generates a number of children equal to the size of the current population. This size is not a constant and it can vary throw the evolutionary process because of the action of token competition.

This new population of children does no replace the current population of parents. Instead of this, a new join population is formed by adding the parents and children populations. Individuals in this join population are arranged by their fitness score in order to be able to apply the token competition diversity mechanism.

Each parent is selected from the current population by the binary tournament selection scheme. The chosen individual is then used to generate a new child by applying one of the previous described four genetic operators. The genetic operator election is done in a probabilistic way. Specifically, the probabilities of the four genetic operator are added in a single measure $P = P_c + P_m + P_i + P_{dp}$ (where P_c, P_m, P_i and P_{dp} represent the crossover, mutation, insertion and dropping condition probabilities, respectively) and then a random value $u \in [0, P]$ is obtained to choose the genetic operator to be applied.

H. Algorithm's description

GP-COACH algorithm begins creating an initial population according to the rules in the context-free grammar. Each individual in this population is then evaluated. After that, the initial population is kept as the best evolved population and its global fitness score is calculated. Then, the initial population is copied to the current population and the evolutionary process begins:

1) An offspring population, with the same size than the current one, is created. Parents are selected by using the binary tournament selection mechanism and children are created by using one of the four genetic operators. Each new child must be then evaluated.
2) Once the offspring population is created, it is joined to the current population, thus creating a new population whose size is double the current population size. Individuals in this new population are then sorted by their fitness and Token Competition mechanism is applied. Secondaries rules are created if they are necessary.
3) Global fitness score measure is then calculated for this new population. In the case of this measure outperforms the one for the best population, best population must be replaced by this new population (and best global fitness score measure must also be updated). In any case, the new population is copied as the current population in order to be able to apply the evolutionary process again.

The evolutionary process ends when the stop condition is verified. Then, the population kept as the best one is returned as solution to the problem and GP-COACH finishes.

IV. EXPERIMENTAL STUDY

In order to analyse the GP-COACH behaviour, we have carry out an experimental study using several high dimensional data sets (see Table II) obtained from the UCI repository of machine learning databases [26].

TABLE II
DATA SETS CHARACTERISTICS

Name	N. Instances	N. Features	N. Classes
Ionosphere	351	34	2
Pen-based	10992	16	10
Spambase	4597	57	2
Thyroid	7200	21	3
Vehicle	846	18	4
Wdbc	569	30	2

Our method has been compared to, a) 2SLAVE [4], an extension consisting in including a feature selection process to the SLAVE algorithm [12], b) Tsakonas' method in [21], which uses a GP algorithm with a Pittsburgh codification scheme to learn FRBCSs, and c) FRBCS_GP [22], a GP-based method for learning FRBCSs early proposed by the authors. The main differences between FRBCS_GP and GP-COACH are the following:

- FRBCS_GP does not uses a two level hierarchical inference process, and therefore it leans rule sets containing only one type of fuzzy rule.
- In FRBCS_GP the size of the current population does not change during the evolutionary process. GP-COACH uses a non-constant population size.
- FRBCS_GP does not uses any kind of fitness score to keep the best evolved population, so it returns the last evolved population as solution to the problem.
- FRBCS_GP uses a crisp fitness function, which uses the number of positive and negative examples, and not the sum of matching degrees like GP-COACH does.
- Finally, FRBCS_GP uses a ranking selection scheme to select parents from the population, and different genetic operators to generate new children.

The parameters of each one of the algorithms used in the study are summarised in Table III. We have used 5 linguistic labels per variable (with triangular membership functions) in all the experiments. All the data sets have been partitioned using the *ten fold cross-validation* (*10-fcv*) procedure. Finally, each method has been executed by using three different seeds, so the results shown stand the average of a total of 30 runs.

The average results are shown in Table IV, where #R indicates the average rule number, #Var the average antecedent variables per rule, #Cond the average antecedent conditions number per rule, and %Tra and %Test the correct percentage with training and test examples, respectively. The subscripts in the different methods are related to the fuzzy reasoning method (FRM) [27] used, so 1 corresponds to the classical FRM (max-min) and 2 with the normalised sum respectively.

Analysing the results, it can be pointed out:

TABLE III
PARAMETERS USED IN THE ALGORITHMS

Algorithm	Parameters
2SLAVE	$itrmax = 1000, itr\_no\_improve = 50, Pop = 20$ $\lambda = 0.8, k_1 = 0, k_2 = 1, P_{AND} = 0.1, P_{OR} = 0.1$ $P_c = 0.6, P_m\ (per\ gene) = 0.05, P_{Rotation} = 0.05$
Tsakonas	$max\_individual\_size\ = 650\ nodes, Pop = 2000$ $itrmax = 10000, Tournament = 6, P_c = 0.35$ $P_m\ (per\ node) = 0.4, P_{Shrink} = 0.6$
FRBCS_GP	$Eval = 20000, Pop = 200, P_c = 0.5, P_m = 0.4$ $P_{dp} = 0.1, min\_support = 0.01$
GP-COACH	$Eval = 20000, Pop = 200, P_c = 0.5, P_m = 0.2$ $P_{dp} = 0.15, P_i = 0.15, \alpha = 0.7, Tournament = 2$ $w_1 = 0.8, w_2 = w_3 = 0.05\ and\ w_4 = 0.1$

Regarding the compactness, 2SLAVE algorithm usually obtains the rule sets with the lower number of rules (GP-COACH slightly outperforms it with Ionosphere and Wdbc data sets). However, these learned rules present a high number of variables and conditions per variable, that makes the interpretability of the rule sets decrease. GP-COACH learns rule sets with a quite small number of rules, having them also a low number of variables and condition per variable. Tsakonas algorithm obtains rule sets with a low number of variables and conditions per variable, but with a high number of fuzzy rules. Finally, GP-COACH is able to learn more compact rule sets than the ones obtained by FRBCS_GP.

Regarding the accuracy, GP-COACH is the algorithm that obtains the best test accuracy results with all the data sets, except with the Wdbc data set, were FRBCS_GP slightly outperforms its results. 2SLAVE obtains good test accuracy results, comparable to the ones obtained by FRBCS_GP, but they do not outperform the ones obtained by GP-COACH. Finally, Tsakonas algorithm obtains the worst results on test accuracy because of the simplicity of its learned rules.

So we consider that GP-COACH is the proposal that presents the best interpretability-accuracy trade-off.

V. CONCLUSIONS

In this work we have presented GP-COACH, a Genetic Programming based algorithm for the learning of COmpact and interpretable fuzzy rule bases, that also present a high ACcuracy on test data, for High dimensional problems. GP-COACH main features are:

- It uses a context-free grammar that allows the learning of DNF fuzzy rules and the absence of some input features.
- It follows the *GCCL* approach that encodes a single rule per individual and thus the rule base is formed by the whole population.
- It includes a mechanism to increase the diversity into the population, the *Token Competition*, that makes the rules compete among themselves during the evolutionary learning process, giving out a smaller number of rules.
- Regarding FRBCS_GP, GP-COACH uses a new fitness function, new genetic operators, a global fitness score and learns rule sets having two different types of rules,

TABLE IV

DATA SETS RESULTS

	IONOSPHERE					PEN-BASED					SPAMBASE				
	#R	#Var	#Cond	%Tra	%Test	#R	#Var	#Cond	%Tra	%Test	#R	#Var	#Cond	%Tra	%Test
2SLAVE$_1$	6	12.73	25.16	82.73	81.88	**39.97**	11.76	26.55	75.98	75.85	**7.9**	22.25	49.19	73.66	73.74
2SLAVE$_2$	6	12.73	25.16	85.46	84.14	**39.97**	11.76	26.55	81.32	81.16	**7.9**	22.25	49.19	69.87	70.14
TSAKONAS$_1$	35.4	2.44	**2.92**	40.37	39.69	65	**1.17**	**1.2**	15.56	15.58	27.43	**3.03**	**3.68**	39.63	39.93
TSAKONAS$_2$	35.4	2.44	**2.92**	51.05	52.42	65	**1.17**	**1.2**	15.19	15.38	27.43	**3.03**	**3.68**	49.63	49.49
FRBCS_GP$_1$	14.03	2.41	6.63	83.65	82.81	87.07	3.25	7.67	73.43	73.44	43.93	13.46	31.57	73.72	73.50
FRBCS_GP$_2$	14.03	2.41	6.63	82.86	81.48	87.07	3.25	7.67	75.74	75.55	43.93	13.46	31.57	75.03	74.55
GP-COACH$_1$	**5.9**	2.14	5.20	93.61	92.14	85.9	4.22	9.08	78.97	78.85	14	5.27	9.63	77.29	76.62
GP-COACH$_2$	5.97	**2.06**	5.07	**94.26**	**92.31**	85.13	4.26	9.13	**82.51**	**82.38**	10.77	3.93	7.96	**83.70**	**83.59**
	THYROID					VEHICLE					WDBC				
	#R	#Var	#Cond	%Tra	%Test	#R	#Var	#Cond	%Tra	%Test	#R	#Var	#Cond	%Tra	%Test
2SLAVE$_1$	**3.7**	9.23	20.52	92.81	92.87	**11.73**	11.07	24.96	49.11	47.30	5.2	10.31	20.22	92.59	92.74
2SLAVE$_2$	**3.7**	9.23	20.52	92.85	92.88	**11.73**	11.07	24.96	49.21	47.38	5.2	10.31	20.22	92.43	92.33
TSAKONAS$_1$	37.27	2.02	**2.56**	80.31	80.38	37.73	**1.89**	**2.48**	26.15	26.09	27.8	2.76	3.69	51.89	53.28
TSAKONAS$_2$	37.27	2.02	**2.56**	43.23	43.77	37.73	**1.89**	**2.48**	26.32	27.00	27.8	2.76	3.69	52.39	52.71
FRBCS_GP$_1$	29.33	12.01	34.16	92.90	92.89	55.23	6.10	15.84	53.13	52.60	14.9	3.59	8.85	94.48	93.49
FRBCS_GP$_2$	29.33	12.01	34.16	92.82	92.82	55.23	6.10	15.84	56.47	55.05	14.9	3.59	8.85	**95.27**	**94.37**
GP-COACH$_1$	6.13	**1.59**	2.73	**93.27**	**93.26**	34.03	4.08	9.17	54.81	53.55	4.97	1.11	**2.61**	94.19	92.32
GP-COACH$_2$	6.3	1.60	2.80	93.26	93.25	33.53	4.22	9.57	**58.33**	**55.09**	**4.8**	**1.09**	2.67	95.03	93.38

obtaining more compact rule sets, that also present a good accuracy performance.

An experimental study involving several high dimensional data sets has been carried out. This study has demonstrated that GP-COACH is able to learn rule sets with a small number of rules, variables and conditions per variable for high dimensional problems. On the other hand, GP-COACH has outperformed the rest of algorithms with regard to the accuracy results on training and test data, showing a high performance for high-dimensional problems.

Therefore, we consider that GP-COACH can be an interesting alternative for the interpretability-accuracy trade-off in FRBCSs when dealing with high dimensional problems.

VI. ACKNOWLEDGMENTS

Supported by the Spanish Ministry of Science and Technology under Projects TIN-2005-08386-C05-(01 and 03)

REFERENCES

[1] J. Casillas, O. Cordón, F. Herrera and L. Magdalena (Eds.), *Interpretability Issues in Fuzzy Modeling*, Springer-Verlag, Series Studies in Fuzziness and Soft Computing, Vol. 128 (2003)

[2] H. Ishibuchi, T. Nakashima and M. Nii, *Classification And Modeling With Linguistic Information Granules: Advanced Approaches To Linguistic Data Mining*, Springer Verlag (2004)

[3] D. Chakraborty and N.R. Pal, A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification, *IEEE Transactions on Neural Networks* 15:1 (2004) 110–123

[4] A. González and R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm, *IEEE Transactions on Systems, Man and Cybernetics Part B* 31:3 (2001) 417–425

[5] J. Abonyi, J.A. Roubos and F. Szeifert, Data-driven generation of compact, accurate, and linguistically sound fuzzy classifiers based on a decision-tree initialization, *International Journal of Approximate Reasoning* 32:1 (2003) 1–21

[6] H. Ishibuchi and T. Yamamoto, Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining, *Fuzzy Sets and Systems* 141:1 (2004) 59–88

[7] J.R. Koza, *Genetic programming on the programming of computers by means of natural selection*, Cambridge MA, USA: MIT Press (1992)

[8] O. Cordón, F. Herrera, F. Hoffmann and L. Magdalena, *Genetic Fuzzy Systems. Evolutionary tuning and learning of fuzzy knowledge bases*, World Scientific (2001)

[9] P. Thrift, "Fuzzy logic synthesis with genetic algorithms", in *Proc. of the 4th Int. Conference on Genetic Algorithms* (1991) 509-513

[10] J. Casillas, B. Carse and L. Bull, Fuzzy-XCS: a Michigan genetic fuzzy system, *IEEE Trans. on Fuzzy Systems* 15:4 (2007) 536–550

[11] S. Wilson, Classifier fitness based on accuracy, *Evolutionary Computation* 3:2 (1995) 149-175.

[12] A. González and R. Pérez, SLAVE: A genetic learning system based on an iterative approach, *IEEE Trans. on Fuzzy Systems* 27 (1999) 176–191

[13] C.F. Juang, J.Y. Lin and C.T. Lin, Genetic reinforcement learning through symbiotic evolution for fuzzy controller design, *IEEE Transactions on Systems, Man and Cybernetics Part B* 30:2 (2000) 290–302

[14] A. Geyer-Schulz, *Fuzzy rule-based expert systems and genetic machine learning*, Heidelberg: Physica-Verlag (1995)

[15] L. Sánchez and J.A. Corrales, Niching Scheme for Steady State GA-P and its Application to Fuzzy Rule Based Classifiers Induction, *Mathware & Soft Computing* 7:2-3 (2000) 337–350

[16] L. Sánchez, I. Couso and J.A. Corrales, Combining GP operators with SA search to evolve fuzzy rule based classifiers, *Information Sciences* 136:1–4 (2001) 175–191

[17] A. Bastian, Identifying fuzzy models utilizing genetic programming, *Fuzzy Sets and Systems* 113:3 (2000) 333–350

[18] F. Hoffmann and O. Nelles, Genetic programming for model selection of TSK-fuzzy systems, *Information Sciences* 136:1-4 (2001) 7–28

[19] R.R.F. Mendes, F. de B. Voznika, A.A. Freitas and J.C. Nievola, "Discovering Fuzzy Classification Rules with Genetic Programming and Co-evolution", in *5th European Conf. on Principles of Data Mining and Knowledge Discovery*, LNAI Vol. 2168 (2001) 314–325

[20] B-C. Chien, J.Y. Lin and T-P. Hong: Learning discriminant functions with fuzzy attributes for classification using genetic programming. *Expert Systems with Applications* 23:1 (2002) 31–37

[21] A. Tsakonas, A comparison of classification accuracy of four genetic programming-evolved intelligent structures, *Information Sciences* 176:6 (2006) 691–724

[22] F.J. Berlanga, M.J. del Jesus and F. Herrera, "Learning fuzzy rules using Genetic Programming: Context-free grammar definition for high-dimensionality problems", in *Proc. of the I Workshop on Genetic Fuzzy Systems* (2005) 136–141

[23] F.J. Berlanga, M.J. del Jesus, M.J. Gacto and F. Herrera, "A Genetic-Programming-Based Approach for the Learning of Compact Fuzzy Rule-Based Classification Systems", in *8th Int. Conference on Artificial Intelligence and Soft Computing*, LNCS Vol. 4029 (2006) 182–191

[24] M.L. Wong and K.S. Leung, *Data Mining using grammar based genetic programming and applications*, Kluwer Academics Pub. (2000)

[25] T-P. Hong and Y-C. Lee, Mining coverage-based fuzzy rules by evolutional computation, *The 2001 IEEE International Conference on Data Mining* (2001) 218–224

[26] A. Asuncion and D.J. Newman, *UCI Machine Learning Repository [http://www.ics.uci.edu/∼mlearn/MLRepository.html]*, Irvine, CA: Univ. of California, Dpt. of Information and Computer Science (2007)

[27] O. Cordón, M.J. del Jesus and F. Herrera, A Proposal on Reasoning Methods in Fuzzy Rule-Based Classification Systems, *International Journal of Approximate Reasoning* 20 (1999) 21–45