Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Fuzzy Sets and Systems 141 (2004) 5–31

FUZZY sets and systems

www.elsevier.com/locate/fss

Ten years of genetic fuzzy systems: current framework and new trends

O. Cordón[a,*], F. Gomide[b], F. Herrera[a], F. Hoffmann[c], L. Magdalena[d]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain*
[b] *Department of Computer Engineering and Industrial Automation, FEEC, State University of Campinas, SP 13083-970, Brazil*
[c] *Royal Institute of Technology, Center for Autonomous Systems, S-10044 Stockholm, Sweden*
[d] *Department of Applied Mathematics, E.T.S.I. Telecomunicación, Politechnic University of Madrid, 28040 Madrid, Spain*

Abstract

Fuzzy systems have demonstrated their ability to solve different kinds of problems in various application domains. Currently, there is an increasing interest to augment fuzzy systems with learning and adaptation capabilities. Two of the most successful approaches to hybridise fuzzy systems with learning and adaptation methods have been made in the realm of soft computing. Neural fuzzy systems and genetic fuzzy systems hybridise the approximate reasoning method of fuzzy systems with the learning capabilities of neural networks and evolutionary algorithms.

The objective of this paper is to provide an account of genetic fuzzy systems, with special attention to genetic fuzzy rule-based systems. After a brief introduction to models and applications of genetic fuzzy systems, the field is overviewed, new trends are identified, a critical evaluation of genetic fuzzy systems for fuzzy knowledge extraction is elaborated, and open questions that remain to be addressed in the future are raised. The paper also includes some of the key references required to quickly access implementation details of genetic fuzzy systems.

Keywords: Fuzzy rule based systems; Genetic algorithms; Genetic fuzzy systems; Tuning; Learning

1. Introduction

Fuzzy systems have been successfully applied to problems in classification [25], modelling [105] control [43], and in a considerable number of applications. In most cases, the key for success was the

* Corresponding author.
E-mail addresses: ocordon@decsai.ugr.es (O. Cordón), gomide@dca.fee.unicamp.br (F. Gomide), herrera@decsai.ugr.es (F. Herrera), hoffmann@nada.kth.se (F. Hoffmann), llayos@mat.upm.es (L. Magdalena).

0165-0114/$ - see front matter
doi:10.1016/S0165-0114(03)00111-8

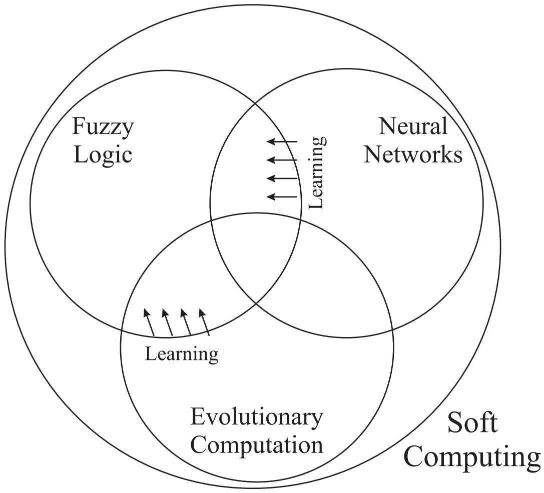

Fig. 1. Soft computing and learning in fuzzy systems.

ability of fuzzy systems to incorporate human expert knowledge. In the 1990s, despite the previous successful history, the lack of learning capabilities characterising most of the works in the field generated a certain interest for the study of fuzzy systems with added learning capabilities. Two of the most successful approaches have been the hybridisation attempts made in the framework of soft computing, were different techniques, such as neural and evolutionary, provide fuzzy systems with learning capabilities, as shown in Fig. 1. Neuro-fuzzy systems are one of the most successful and visible directions of that effort [44,78,99]. A different approach to hybridisation lead to genetic fuzzy systems (GFSs) [8,33,64,106,118].

A GFS is basically a fuzzy system augmented by a learning process based on a genetic algorithm (GA). GAs are search algorithms, based on natural genetics, that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes [50,52,67].

Genetic learning processes cover different levels of complexity according to the structural changes produced by the algorithm [41], from the simplest case of parameter optimisation to the highest level of complexity of learning the rule set of a rule based system. Parameter optimisation has been the approach utilised to adapt a wide range of different fuzzy systems, as in genetic fuzzy clustering or genetic neuro-fuzzy systems, that are briefly considered in Section 6.

Analysis of the literature shows that the most prominent types of GFSs are *genetic fuzzy rule-based systems* (GFRBSs) [33], whose genetic process learns or tunes different components of a fuzzy rule-based system (FRBS). Fig. 2 shows this conception of a system where genetic design and fuzzy processing are the two fundamental constituents. Inside GFRBSs it is possible to distinguish between either parameter optimisation or rule generation processes, that is, adaptation and learning.

The paper starts by briefly reviewing GAs (Section 2), next offering a general approach to GFRBSs (Section 3). Section 4 describes examples of GFS applications. Later, Section 5 describes new (or less common) lines of research in the field of GFRBSs. Section 6 presents a general view of GFSs beyond

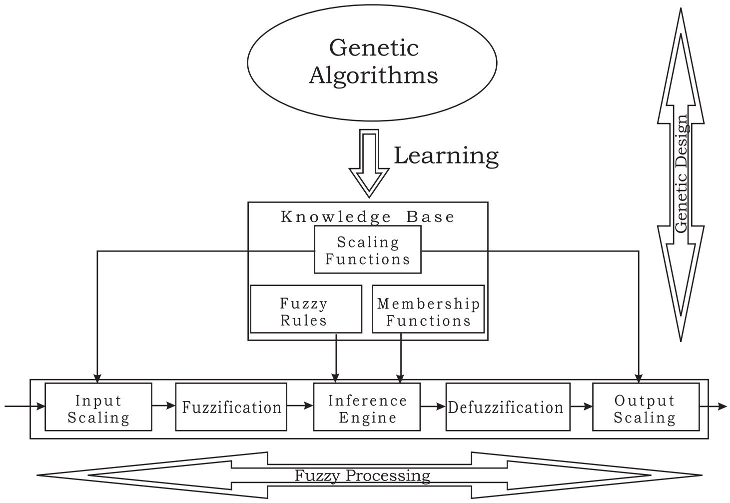

Fig. 2. Genetic design and fuzzy processing.

fuzzy rule-based approaches. Section 7 is concerned with a critical evaluation of the contribution of GAs to fuzzy knowledge extraction. Section 8 summarises examples of open questions and problems that remain to be solved in the future. Section 9 concludes the paper.

2. Genetic algorithms

GAs are general purpose search algorithms which use principles inspired by natural genetics to evolve solutions to problems [50,52,67]. The basic idea is to maintain a population of chromosomes (representing candidate solutions to the concrete problem being solved) that evolves over time through a process of competition and controlled variation.

A GA starts with a population of randomly generated *chromosomes*, and advances towards better chromosomes by applying genetic operators modelled on the genetic processes occurring in nature. The population undergoes evolution in a form of natural selection. During successive iterations, called *generations*, chromosomes in the population are rated for their adaptation as solutions, and on the basis of these evaluations, a new population of chromosomes is formed using a selection mechanism and specific genetic operators such as *crossover* and *mutation*. An *evaluation* or *fitness function* must be devised for each problem to be solved. Given a particular chromosome, a possible solution, the fitness function returns a single numerical value, which is supposed to be proportional to the utility or adaptation of the solution represented by that chromosome.

Although there are many possible variants of the basic GA, the fundamental underlying mechanism consists of three operations: evaluation of individual fitness, formation of a gene pool (intermediate population) through selection mechanism, and recombination through crossover and mutation operators. Fig. 3 illustrates this operation mode. The specific characteristics of the evaluation method are quite dependent on the application. Some comments concerning evaluation in GFSs will be introduced in Section 5.5 when describing the accuracy/interpretability trade-off.

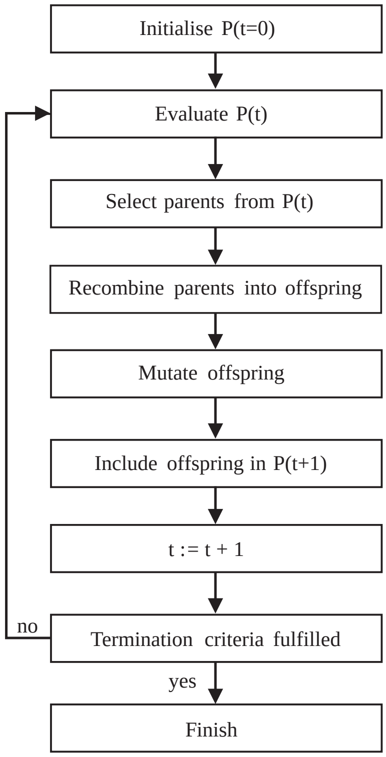

Fig. 3. Principal structure of a genetic algorithm.

As previously stated, genetic learning processes cover different levels of complexity, from parameter optimisation to learning the rule set of a rule based system. Genetic learning processes designed for parameter optimisation usually fit to the description given in previous paragraphs, but when considering the task of learning rules in a rule based system, a wider range of possibilities is open.

When considering a rule based system and focusing on learning rules, there are three main approaches that have been applied in the literature: Pittsburgh [124] (Fig. 4), Michigan [68] (Fig. 5) and iterative rule learning [131]. Pittsburgh and Michigan approaches are the most extended methods for rule learning developed in the field of GAs. The first one is characterised by representing an entire rule set as a genetic code (chromosome), maintaining a population of candidate rule sets and using selection and genetic operators to produce new generations of rule sets. The Michigan approach considers a different model where the members of the population are individual rules and a rule set is represented by the entire population. In the third approach, the iterative one, chromosomes code individual rules, and a new rule is adapted and added to the rule set, in an iterative fashion, in every run of the GA.

3. Genetic fuzzy rule-based systems

A number of papers have been devoted to the automatic generation of the knowledge base of an FRBS using GAs. The key point is to employ an evolutionary learning process to

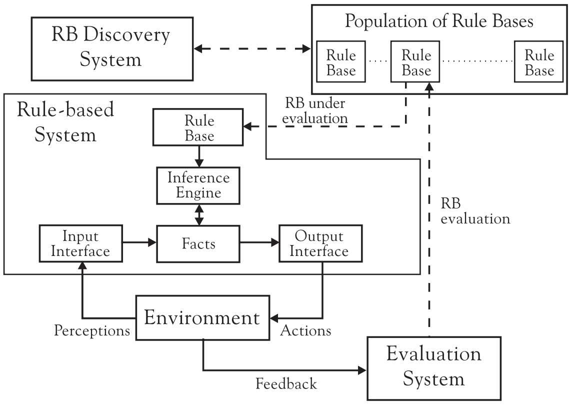

Fig. 4. Learning with the Pittsburgh approach.

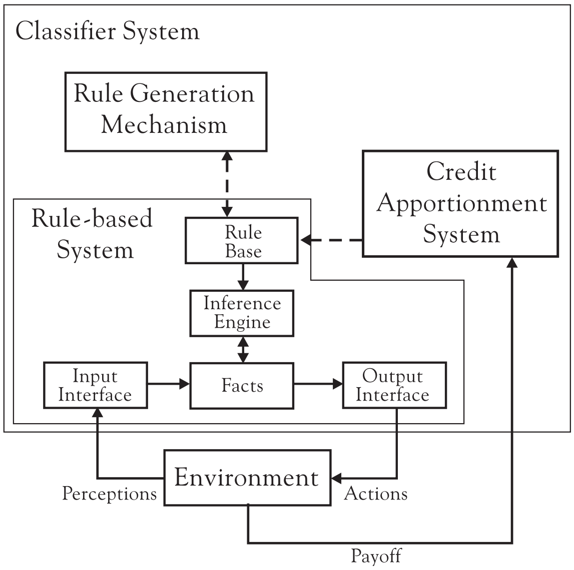

Fig. 5. Learning with the Michigan approach.

automate the design of the knowledge base, which can be considered as an optimisation or search problem.

From the viewpoint of optimisation, the task of finding an appropriate knowledge base (KB) for a particular problem, is equivalent to parameterise the fuzzy KB (rules and membership functions), and to find those parameter values that are optimal with respect to the design criteria. The KB parameters constitute the optimisation space, which is transformed into a suitable genetic representation on which the search process operates.

The first step in designing a GFRBS is to decide which parts of the KB are subject to optimisation by the GA. The KB of an FRBS does not constitute a homogeneous structure but is rather the union of qualitatively different components. As an example, the KB of a descriptive Mamdani-type FRBS (the one considered in Fig. 2) is comprised of two components:

- a data base (DB), containing the definitions of the scaling functions of the variables and the membership functions of the fuzzy sets associated with the linguistic labels, and
- a rule base (RB), constituted by the collection of fuzzy rules.

The decision on which part of the KB to adapt depends on two conflicting objectives: dimensionality and efficiency of the search. A search space of a smaller dimension results in a faster and simpler learning process, but the obtainable solutions might be suboptimal. A larger, complete search space that comprises the entire KB and has a finer dimensionality is therefore more likely to contain optimal solutions, but the search process itself might become prohibitively inefficient and slow.

With these considerations there is an obvious trade-off between the completeness and dimensionality of the search space and the efficiency of the search. This trade-off offers different possibilities for GFS design that are considered in the following subsections.

First of all, it is important to distinguish between tuning (alternatively, adaptation) and learning problems:

- Tuning is concerned with optimisation of an existing FRBS, whereas learning constitutes an automated design method for fuzzy rule sets that starts from scratch. Tuning processes assume a predefined RB and have the objective to find a set of optimal parameters for the membership and/or the scaling functions, DB parameters.
- Learning processes perform a more elaborated search in the space of possible RBs or whole KBs and do not depend on a predefined set of rules.

3.1. Genetic tuning

Tuning of the scaling functions and fuzzy membership functions is an important task in FRBS design. Parameterised scaling functions and membership functions are adapted by the GA according to a fitness function that specifies the design criteria in a quantitative manner.

As previously said, tuning processes assume a predefined RB and have the objective of finding a set of optimal parameters for the membership and/or the scaling functions (Fig. 6). It is also possible, as will be seen in Section 5.4, to perform the tuning process a priori, i.e., considering that a subsequent process will derive the RB once the DB has been obtained, that is a priori *genetic DB learning*.

3.1.1. Tuning scaling functions

Scaling functions applied to the input and output variables of FRBSs normalise the universes of discourse in which the fuzzy membership functions are defined. Usually, the scaling functions are parameterised by a single scaling factor [101] or a lower and upper bound [96] in case of linear scaling, and one or several contraction/dilation parameters in case of non-linear scaling [57,95]. These parameters are adapted such that the scaled universe of discourse better matches the underlying variable range.

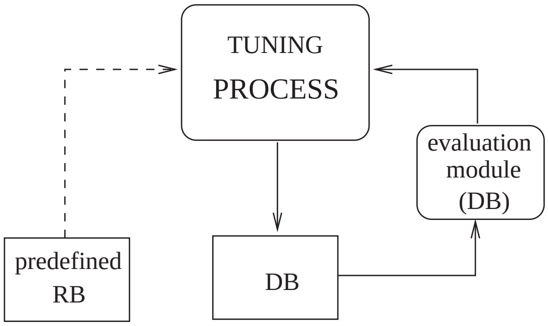

Fig. 6. Tuning the data base.

The usual approach of these kinds of processes is the adaptation of one to four parameters (defining the scaling function) per variable: one when using a scaling factor, two for linear scaling, and three or four in non-linear scaling. Most of the cited works consider a real coding scheme to represent the parameters of the function, but it is also possible to use binary codes, as in [101] where a three bits binary representation of each scaling factor is used.

Since the number of variables is predefined, as well as the number of parameters required to code each scaling function, this approach leads to a fixed length code.

3.1.2. Tuning membership functions

When tuning membership functions, an individual represents the entire DB as its chromosome encodes the parameterised membership functions associated to the linguistic terms in every fuzzy partition considered by the FRBS. The most common shapes for the membership functions (in GFRBSs) are triangular (either isosceles [80,103] or asymmetric [29,83]), trapezoidal [63,81] or Gaussian functions [58,60]. The number of parameters per membership function usually ranges from one to four, each parameter being either binary [119] or real coded [94].

The structure of the chromosome is different for FRBSs of the descriptive (using linguistic variables) or the approximate (using fuzzy variables) type. [1] When tuning the membership functions in a linguistic model [29], the entire fuzzy partitions are encoded into the chromosome and it is globally adapted to maintain the global semantic in the RB. These approaches usually consider a predefined number of linguistic terms for each variable (no need to be the same for each of them), leading to a code of fixed length in what concerns membership functions. But even having a fixed length for the code, it is possible to evolve the number of linguistic terms associated to a variable by simply defining a maximum number (that defines the length of the code) and letting some of the membership functions to be located out of the range of the linguistic variable (reducing the actual number of linguistic terms). This is the conception of [91] when designing a TSK system with linguistic input variables.

A particular case where the number of parameters to be coded is reduced, is that of descriptive fuzzy systems working with strong fuzzy partitions. In this case, the number of parameters to code

[1] Grid-based versus scatter fuzzy partitions.

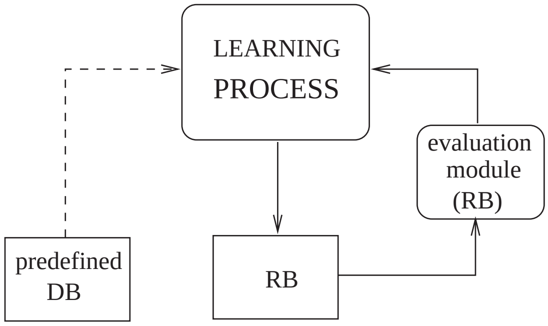

Fig. 7. Learning the rule base.

is reduced to those defining the core regions of the fuzzzy sets: the modal point for triangles [49], the extreme points of the core for trapezoidal shapes [14].

On the other hand, tuning the membership functions of a model working with fuzzy variables (scatter partitions) [5,63,120] is a particular instantiation of KB learning since the rules are completely defined by their own membership functions instead of referring to linguistic terms in the DB.

3.2. Genetic learning of rule bases

Genetic learning of RBs assumes a predefined set of fuzzy membership functions in the DB to which the rules refer to by means of linguistic labels (Fig. 7). It only applies to descriptive FRBSs, as in the approximate approach adapting rules is equivalent to modify the membership functions (see next section).

The three learning approaches described in previous section can be considered to learn RBs: Michigan approach [12,75,128], Pittsburgh approach [66,108,127], and iterative rule learning approach [29,38,54]. The RB can be represented by a relational matrix [127], a decision table [108], or a list of rules [54,66,128].

Representations through relational matrix and decision table are only useful when the system has a reduced number of variables, since they lead to an unaffordable length of the code when having more than two or three input variables. The result is a monolithic code that can be only managed by the Pittsburgh approach.

The list of rules is the most used representation, adopts quite different codes for the individual rules, and can be adapted to the three learning approaches. Often the number of rules in the list is variable (having in some cases an upper limit). A common approach to code individual rules is the use of the disjunctive normal form (DNF) represented in the form of a fixed length binary string [54,96,128]. DNF rules are also considered when working with variable length codes [66] based on messy GAs [50]. With a structure of list of rules, the chromosome can be generated by concatenating the code of individual rules (Pittsburgh, where each chromosome codes a RB) or will contain the code of a single rule.

To code a rule, either as an element of an RB that generates a chromosome or as a single rule generating a chromosome by itself, there are many different approaches. Rules are composed of

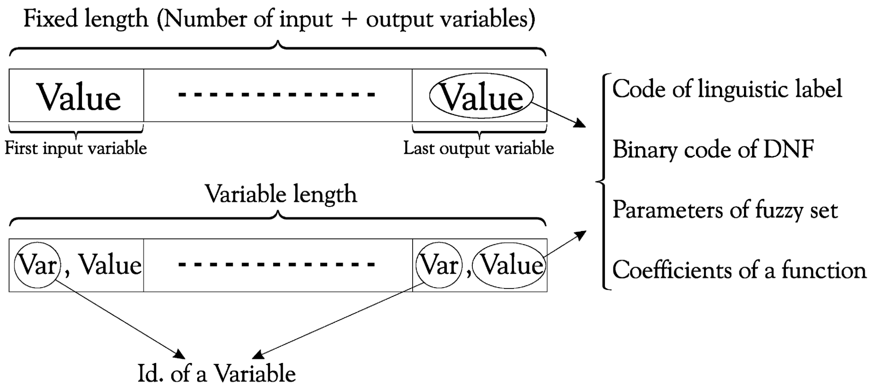

Fig. 8. Representing a rule with fixed or variable length codes.

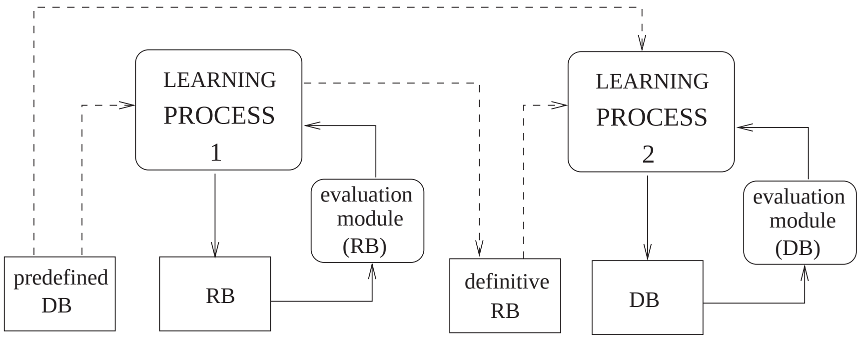

Fig. 9. Learning the rule base and, a posteriori, the data base.

propositions of the form *variable* is *value*, where the variable could be identified by position or by label, and the value could have quite different forms (Fig. 8). When using a code with positional structure (Fig. 8, top) there is a fixed location for each variable in which the information (*value*) related to that variable is placed. When using non-positional codes (Fig. 8, bottom), the code of a rule is composed of pairs (*var*,*value*), where *var* is a label identifying the variable. In both cases, positional and non-positional codes, the content of the *value* part can be: the label of a linguistic term (linguistic variables), the binary code of a DNF structure (linguistic variables), the parameters defining a fuzzy set (fuzzy variables) or the real values (coefficients) of the linear output (output variables of TSK rules).

In addition to the RB learning, other approaches try to improve the preliminary DB definition once the RB has been obtained [38]. That process is comprised by a learning process to obtain the RB considering a predefined DB, followed by a learning process similar to those described in the previous section. In this case, the tuning process that involves the DB learning is called a posteriori *DB learning*. Fig. 9 shows this approach.

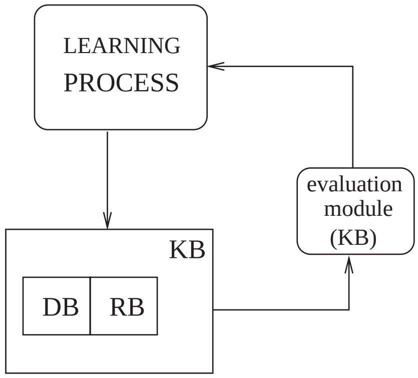

Fig. 10. Learning the knowledge base.

3.3. Genetic learning of knowledge bases

Since genetic learning of the KB deals with heterogeneous search spaces (Fig. 10), it encompasses different genetic representations such as variable length chromosomes, multi-chromosome genomes and chromosomes encoding single rules instead of a whole KB. The computational cost of the genetic search grows with the increasing complexity of the search space. A GFRBS that encodes individual rules rather than entire KBs is an option to maintain a flexible, complex rule space in which the search for a solution remains feasible and efficient. Again, the three learning approaches can considered: Michigan [104,130], Pittsburgh [10,20,91,96,103], and iterative rule learning approach [29,32].

Proposals to learn KBs include systems obtaining approximate Mamdani-type FRBSs with scatter partitions [20,29,32,130], linguistic Mamdani-type FRBSs (scaling functions and rules [96] or membership functions and rules [103]), and TSK fuzzy systems [30,91,102].

The way to code the KB in systems with linguistic variables involves the coding of rules and scaling factors/membership functions as independent parts of the chromosome, or in an iterative way using different chromosomes [61]:

- To code the RB (any of the methods for linguistic variables described in Section 3.2 can be considered).
- The DB will be coded similarly as described in Section 3.1.

Closely related to the coding scheme, the process of crossover of the genetic codes of two parents involves chromosomes containing substructures (rule and data bases) that can be managed in different ways:

- As a single one, by merging the substructures [91].
- As two unrelated substructures, applying a parallel process [103].
- As two related substructures, applying a sequential process where the result of crossing one substructure affects the crossover of the other [96].

3.4. Summary of classical GRBFS approaches

Summing up, the classical genetic learning procedures to evolve FRBSs are:

- genetic tuning of the DB,
- genetic learning of the RB,
- genetic learning of the KB.

Although the review is by no means exhaustive, this section reviewed the most important approaches found in the literature. In Section 5, we shortly report new (or less common) lines of development of GFRBSs.

4. Applications of genetic fuzzy systems

Soft Computing provides a computational framework to address design, analysis and modeling problems in the context of uncertain and imprecise information. Its constituents fuzzy logic, neural networks, probabilistic computing and evolutionary algorithms are considered as complementary and synergistic partners rather than competing methodologies.

Since the late 1980s fuzzy control has been successfully applied in consumer products and industrial plants in Japan. In the early 1990s Zadeh proposed the concept of soft computing, which quite rapidly lead to industrial applications in aerospace systems, communication systems, electric power systems, manufacturing automation, robotics, power electronics and transportation [42]. Soft computing techniques contributed in particular to improved solutions in application domains distinguished by imprecise data and incomplete knowledge such as diagnostics, system identification, estimation and control [13]. Neuro-fuzzy systems [78] are by far the most prominent and visible representative of hybrid systems in terms of number of applications. Compared to neuro-fuzzy systems, GFS applications until today remained less visible, in particular in an industrial setting.

In the following we review some successful applications of GFSs to real-world problems in control, manufacturing, consumer products, transportation, modeling and decision making. The selected references give a flavor of potential applications of GFSs but by no means provide a complete overview. The different applications will be reported in chronological order of publication.

In [14], Bonissone et al. describe a genetic tuning scheme for optimisation of a fuzzy controller that regulates the velocity of a freight train. The design goal is a controller that accurately tracks a desired velocity profile while at the same time maintains a smooth ride in order to minimise the stress load on the couplers connecting the rail-cars. Tuning proceeds in several stages, first the GA adapts the scaling factors of input and output variables as they globally affect the control behavior. In a second phase, the GA tunes the membership functions causing a local adaptation. The fuzzy controller is evaluated in a train simulation on different track profiles. In general, the tuned fuzzy controllers demonstrate a substantial improvement in terms of tracking accuracy and smoothness. The authors conclude that tuning the scaling factors accounts for most of the performance improvement. Adaptation of membership functions for a controller with properly tuned scaling factors only results in a marginal improvement. The scalability of the approach reduces the computational cost of tuning, such that it becomes feasible to individually customise the fuzzy controller for different tracks off-line.

In [71], Hwang addresses a similar transportation problem, namely to optimise trip time and energy consumption of a high-speed railway. Fuzzy *c*-means clustering and GAs identify the structure and parameters of a linguistic model that describes the relationship between the current velocity commands as input and the resulting trip time and energy consumption. The fuzzy model is then used by the railway operators to build efficient and economic control strategies. The parameters of the model are identified by a hybrid GA, that combines global evolutionary search with a local hill-climbing technique. The method is applied to derive a control strategy for a planned high-speed train line in Korea. An economical train run with a trip time margin of less than 7% and an energy saving of 5% is reported.

In [132], Voget et al. present a multi-objective optimization scheme, in which a fuzzy controller regulates the selection procedure and fitness function of the GA. This approach falls into the realm of so-called fuzzy evolutionary algorithms, in which a fuzzy system manages the resources and parameters of a GA such as mutation rate, population size and selective pressure to improve the performance [62]. In the particular approach, the fuzzy rules constitute a heuristic that allows the GA to identify the set of Pareto-optimal solutions. Based on the deviation between the current population and the multi-objective goal function, the fuzzy controller decides which selection scheme and fitness function the GA applies in order to achieve the optimisation goals and to cover the optimal Pareto front. The approach has been applied to optimise the timetable of railway networks, with the objective to reduce passenger waiting time when switching trains while at the same time minimise the cost of new investments to improve the necessary infrastructure. The result of the genetic optimisation is a cost–benefit curve that shows the effect of investments on the accumulated passenger waiting time and the trade-off between both criteria. Another application example of fuzzy evolutionary algorithms to agile manufacturing is given in [125].

In [70], Huang et al. design a fuzzy sliding mode controller by means of a real-coded GA. The authors report on an application to position control of an industrial XY-table. By using laser as a sensor, the table can be positioned with sub-micron level accuracy. The fuzzy controllers are evaluated on the physical hardware and the accumulated positioning error serves as a fitness. Special protective circuitry is used to prevent damages to the motors and the table, that otherwise might result from improper fuzzy controllers that lead to an unstable system. Evolutionary optimisation of fuzzy controllers is feasible for hardware-in-the-loop systems and can help to reduce the costs for controller design and tuning.

In [121], the authors report on applications of evolutionary computation in combination with neural networks and fuzzy systems for intelligent consumer products. The role of the evolutionary algorithm is to adapt the number of rules and to fine tune the membership functions to improve the performance of fuzzy systems for estimation and control. Genetic tuning is applied to fuzzy rules that predict the amount of dishes to be cleaned by a dish washer, estimate the amount of rice filled into a rice cooker and control a microwave oven. The paper also mentions evolutionary computation for fuzzy rule generation applied to process control.

In [97], Mizutani et al. propose a hybrid neuro-genetic–fuzzy system for computerised colour prediction, a challenging problem in paint production. Their architecture for colour paint manufacturing intelligence cannot be characterised as a conventional GFSs in which the evolutionary algorithm optimises the fuzzy knowledge base. Instead, colour expert knowledge is expressed by fuzzy rules.

The first generation of colour chromosomes in the GA is initialised based on the problem specific knowledge. The same rules are also used to evaluate the colour recipe quality. The system mimics the decision-making process of a professional colourist. A fuzzy knowledge base for predicting the pigment concentration of ten different colours for a given surface spectral reflectance is obtained by means of a neuro-fuzzy system. The fuzzy population generator uses this knowledge to seed the first generation of colour chromosomes. For example, if the target colour looks greenish yellow, the initial population is dominated by randomised copies of green and yellow template chromosomes. Expert knowledge of a colourist about the correct proportions of colourants, the number of necessary colourants and conflicts between complementary and similar colourants is summarised in the fuzzy rules. The GA calculates one component of colour chromosome fitness according to the compliance of the chromosome's colourant with the fuzzy expert rules.

In [39], Damousis et al. present a fuzzy expert system that forecasts the wind speed for power generation in wind farms. The TSK fuzzy model is optimised by a GA that adapts the input fuzzy membership functions and the gain factors in the rule conclusion. The training procedure minimises the error between forecast and actual wind speeds in the training set. The accuracy of the model was evaluated with real wind data obtained from groups of wind stations located in two different regions. The input to the model are the wind speed of one local and two remote up-wind stations, from which the expert system predicts the future wind speed at the local down-wind station. The results show that the fuzzy model improves the short- and long-term forecast of wind speeds and thereby is able to better predict the amount of power generated by the wind farm over the next few hours.

Bonissone et al. apply evolutionary techniques to tune a fuzzy decision system [15]. The fuzzy system automatically classifies the risk of an insurance application, which in turn determines the premium to be paid by the applicant. The evolutionary algorithm tunes decision thresholds and internal parameters of the fuzzy decision system in order to optimises the coverage and relative and absolute accuracy of the decision process. Maintenance of automated decision systems is critical, as the decision guidelines as well as the distribution of applicants and their profiles changes over time.

Finally, in [4], a GA is considered to develop a smart tuning strategy for fuzzy logic controllers dedicated to the control of heating, ventilating and air conditioning systems concerning energy performance and indoor comfort requirements. The problem was so complex as, on the one hand, the fuzzy controller provided by the experts was based on a hierarchical structure in order to be able to deal with 17 different variables. On the other hand, the tuning process needed to develop a multicriteria optimization by jointly reducing the energy consumption, augmenting the controller stability and satisfying three different comfort indices. Besides, it had to deal with large time restrictions due to the long computation time models required to assess the accuracy of each individual. To solve the problem, a very specific steady-state GA with a quick convergence was proposed based on an aggregated fitness function considering trustable weights provided by the experts. Several fuzzy logic controllers were produced for different seasons and tested in two real test cells, obtaining very promising results.

The list of applications above indicates that GFSs can contribute to solve industrial and commercial problems. The major driving force behind this development is the need for low-cost solutions that utilise intelligent tools for information processing, design and optimization. GFSs can reduce the cost and time required to design, autonomously operate and maintain systems with a high degree of machine intelligence for control, prediction, modelling and decision making.

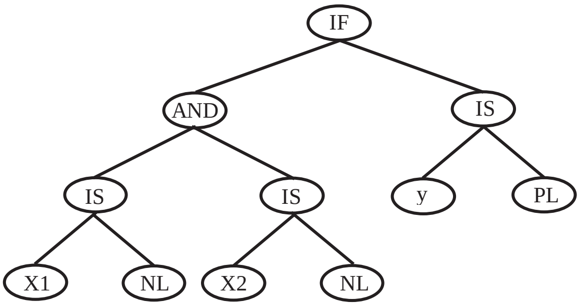

Fig. 11. A syntactic tree and the rule it represents: *IF* $X1$ is *NL* and $X2$ is *NL* THEN Y is *PL*.

5. New trends in genetic fuzzy rule-based systems

In addition to the classical systems addressed in Section 3, here new directions to apply genetic (evolutionary) techniques to FRBSs are explored. The next subsections offer a summary of them.

5.1. Designing fuzzy rule-based systems with genetic programming

Genetic programming (GP) is concerned with the automatic generation of computer programs [84]. Different proposals can be found when using GP to evolve fuzzy rule sets, internally represented as type-constrained syntactic trees [2,26,47,65]. In these kind of systems, fuzzy rules are represented by binary trees as the one depicted in Fig. 11. Fuzzy GP, suggested in [47], combines a simple GA that operates on a context-free language with a context-free fuzzy rule language. Nowadays, it is possible to distinguish among GPs that utilise a grammar to learn linguistic rules [2,47], and approaches that use domain-specific knowledge to define the function and terminal set which constitute the building blocks for the fuzzy rules to be learned [65]. Approaches where GAs or simulated annealing and GP are hybridised have also been proposed [116,117].

5.2. Genetic selection of fuzzy rule sets

In high-dimensional problems, the number of rules in the RB grows exponentially as more inputs are added. Rule reduction methods have been formulated using neural networks, clustering techniques, orthogonal transformation methods, and algorithms based on similarity measures among others. In recent years, genetic techniques have been considered to address the problem of high-dimensional spaces in FRBS design [29,53,73,77,114] with a great success.

A genetic multi-selection process, that at the same time eliminates unnecessary rules from the set of candidate rules and refines KBs for classification problems by means of a linguistic hedge learning process (with a double coding scheme, the first chromosome part for rule reduction and the second chromosome part for linguistic hedge learning evolution) is presented in [37]. Ref. [133] proposes a genetic integration process of multiple knowledge bases that can be also considered as a particular case of genetic selection.

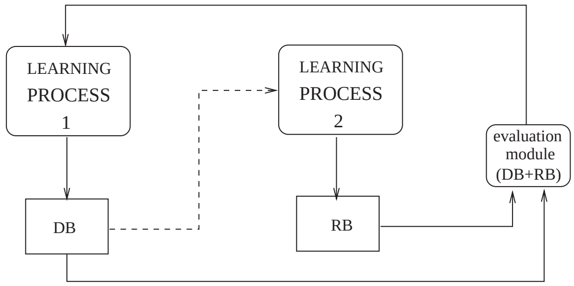

Fig. 12. Learning the data base a priori.

5.3. Genetic feature selection

Related to the question of reducing the number of rules in high-dimensional spaces, there is another option by working directly on the dimension of the search space. With this aim in mind, some papers have focused on feature selection approaches through GAs, allowing a reduction in the number of variables involved in the rules, and consequently, reducing the number of rules [21,69,90,123]. In [79], an interesting scheme is suggested for linguistic FRBSs, including the variable selection within a more complex derivation process (rule generation, DB tuning, and rule selection).

Another approach involves the selection of a subset of input variables for each rule [55]. In this case, the length of the antecedent is variable, avoiding the need of using all available features in the rule. However, this does not mean that a specific input variable ignored in one rule might not be used in another one.

5.4. Learning knowledge bases via genetic derivation of data bases

An innovative approach to learn both KB components is the a priori *genetic DB learning* that evolves some DB components.

Genetic tuning of the DB usually assumes that a predefined RB is used to evaluate the quality of the overall FRBS. A priori *genetic DB learning* refers to a KB learning process in which a GA adapts the DB components such as scaling functions, membership functions and granularity parameters, whilst an additional fuzzy rule generation method derives the RB from the DB definition encoded in the chromosome [34,35,48,72] (see Fig. 12). The RB generated by the second learning process is used to get the fitness function associated to the DB coded in the chromosome.

5.5. Maintaining interpretability via multi-objective genetic processes

Research on GFSs has for a long time considered the objective of the learning process in terms of accuracy. Consequently, the fitness (or evaluation) function of the GA was stated in terms of errors or distances from a target output. Recently, concepts of *linguistic fuzzy modelling*, *interpretability*, and other similar ideas that were considered as almost opposite to *accuracy*, have been reconsidered and today they are viewed as an interesting part of the design process of GFSs.

Recent works have suggested both, accuracy and interpretability, as objectives of learning systems [73,76,79]. This sets a new situation for the learning process where several, usually colliding, objectives have to be simultaneously considered. Some of the measures used to determine the level of interpretability of a fuzzy system are:

- Compactness [73], considered through the number of rules in the RB.
- Rule simplicity [22], evaluated through the number of input variables involved in each rule.

5.6. Genetic-based learning approaches considering different model structures

Improvements in linguistic fuzzy modeling can be accomplished to make learning and/or model structure more flexible. Three possibilities to relax the model structure using a GFS are as follows:

- *Use of double-consequent fuzzy rules*, that allows the model to present rules such that each combination of antecedents may have two consequents associated when it improves the model accuracy. A proposal that use GAs for getting a compact set of rules of this kind can be found in [31]. The GA acts as a genetic selection method to get a cooperative and compact set of fuzzy rules.
- *Consideration of weighted fuzzy rules* in which an importance factor (weight) is considered for each rule. By means of an evolutionary technique, the way in which these rules interact with their neighbor ones could be indicated [74].
- *Genetic selection with hierarchical knowledge bases*. In [36], the structure of the KB of FRBSs is extended in a hierarchical way. Linguistic rules defined over linguistic partitions of different granularity levels provide additional flexibility, and thus improve the model accuracy in those regions in which the usual non-hierarchical models demonstrate poor performance. This type of improvement is the starting point for the development of different hierarchical system of linguistic rules learning methodologies, which are considered as a refinement of the basic linguistic fuzzy models. These methodologies have been thought as a refinement of simple linguistic models which, preserving their descriptive power, introduces small changes to increase their accuracy. A GA is used to get a compact set of hierarchical rules.

5.7. Genetic-based learning approaches with sophisticated genetic algorithms

We should also mention some approaches that use some kind of sophisticated GAs, such as parallel GAs, cooperative coevolutionary algorithms, and Lamarckian co-adaption.

For instance, [24] proposes a new genetic learning approach based on a parallel GA with three populations to optimise FRBSs with RBs generated from fuzzy partitions of three different granularities: 3×3, 5×5 and 7×7. The process also employs a novel method to create migrants between the three populations of the parallel GA to increase the chances of optimisation.

Additionally, [113] suggests a parallel GA to learn FRBSs, separately evolving multiple populations and occasionally exchanging individuals. It simultaneously optimises the structure of the system (number of inputs, membership functions and rules) and tunes the parameters that define the fuzzy system. In the multideme GA system, various fuzzy systems with different number of input variables and with different structures are jointly optimised. Communication between the different demes is established by the migration of individuals presenting a difference in the dimensionality of the input space of a particular variable.

On the other hand, coevolutionary algorithms are advanced evolutionary techniques to solve decomposable complex problems. They involve two or more species (populations) that permanently interact through coupled fitness. Therefore, each species has its own coding scheme and reproduction operators. When an individual must be evaluated, its fitness is found considering some individuals of others species. The coevolution makes easier to find good solutions to complex problems. Cooperative coevolutionary algorithms [109] are those where all the species cooperate to build the problem solution. In this case, the fitness of an individual depends on its ability to cooperate with individuals from other species.

In [107], the author suggests a coevolutionary approach to learn RBs and tune membership functions. An approach to coevolve two species, the subset of best cooperating rules (rule selection) and the weights associated to them, is introduced in [6].

The approach introduced in [112] follows the ideas expressed in [109], "explicit notions of modularity must be introduced to provide reasonable opportunities for solutions to evolve in the form of interacting coadapted subcomponents". It presents a hierarchical evolutionary method to design FRBSs, where a GA works on different populations encoding information items at different levels, to finally evolve a population of complete FRBSs.

In [82], the authors introduce a new design method of neuro-fuzzy logic controllers using a Lamarckian co-adaption scheme that incorporates backpropagation learning into the GA evolution. The design parameters are determined by evolution and learning in a way that evolution performs the global search and makes inter-fuzzy logic controllers parameters adjustments to obtain both the optimal RB with a high covering value, a small number of fuzzy rules, and optimal membership functions.

5.8. Genetic-based machine learning approaches

GFSs with specific combination of evolution and bioinspired models have been developed. For instance, genetic schemes inspired on the virus theory of evolution have been derived to learn TSK fuzzy rule sets [122], including genetic recombination in bacterial genetics [46,100], and DNA coding schemes [45].

6. Other kinds of genetic fuzzy systems

The predominant type of GFS is the GFRBS. However other kinds of GFSs have been developed, with successful results. They include genetic fuzzy neural networks and genetic fuzzy clustering algorithms.

6.1. Genetic fuzzy neural networks

Genetic fuzzy neural networks are the result of adding genetic or evolutionary learning capabilities to systems integrating fuzzy and neural concepts. The result is a genetic-neuro-fuzzy system (or a genetic fuzzy neural network).

The usual approach of most genetic fuzzy neural networks found in the literature, is that of adding evolutionary learning capabilities to a fuzzy neural network that usually is a feed-forward multilayered network to which, previously, some fuzzy concepts were incorporated. The result is a feed-forward multilayered network having fuzzy and genetic characteristics [7,27,87,93,115].

Genetic fuzzy neural networks incorporate fuzzy numbers as weights, perform fuzzy operations in the nodes of the network, and/or consider fuzzy nodes to represent membership functions. In addition, the learning process uses GAs to obtain the weights of the neural network, to adapt the transfer functions of the nodes, and/or to adapt the topology of the net.

A different approach can be found in [110], where an adaptive mechanism is proposed based on the concept of *perpetual evolution*. Here, an adaptive control architecture uses evolutionary learning for initial learning and real-time tuning of a fuzzy logic controller. The initial learning phase involves identification of an artificial neural network model of the process and subsequent development of a fuzzy controller with parameters obtained via a genetic search.

6.2. Genetic fuzzy clustering algorithms

Several references found in the literature propose the use of GAs in fuzzy clustering, most of them devoted to improve the performance of fuzzy C-means (FCM)-type algorithms [9] using the GA to optimise some parameters of these algorithms.

The use of GAs to optimise the parameters of an FCM-type algorithm generates two different kinds of GFSs. *Prototype-based algorithms* encode the fuzzy cluster prototypes and evolve them by means of a GA guided by any centroid-type objective function [59,98], while *fuzzy partition-based algorithms* encode, and evolve, the fuzzy membership matrix [129].

A second possibility is to use the GA to define the distance norm of an FCM-type algorithm. The system considers an adaptive distance function and employs a GA to learn its parameters to obtain an optimal behaviour of the FCM-type algorithm [134].

Finally, a third group of genetic approaches are based on directly solving the fuzzy clustering problem without interaction with any FCM-type algorithm [16].

7. A critical evaluation of the contribution of GAs and evolutionary computation for fuzzy knowledge extraction

Until recently, there was no systematic procedure to design and develop fuzzy systems. A common approach was, and in some application domains still is, defining fuzzy systems based on expert knowledge and testing them to verify if the design is satisfactory. However, when expert knowledge is lacking or when considerable amount of data must be processed and analysed, purely knowledge-based design approaches become limited. Machine learning approaches have shown to be useful in these cases. For instance, neural networks can learn from data, but the linguistic representation of fuzzy rules and their transparency may be lost [126]. Fuzzy neural networks contribute to overcome lacking of linguistic representation and transparency, but the designer must decide on major design parameters such as universes granulation, rule antecedent aggregation operators, rule semantics, rule base aggregation operators and defuzzification methods [11,18,23,78,92]. Automatic methods based-on fuzzy clustering and rule induction from large collections of learning data are attractive alternatives [78], but they have, from the design point of view, the same limitations that the fuzzy neural network approaches.

As emphasised in Section 3, GA-based approaches have been developed to learn:

(i) membership functions with fixed fuzzy rules [80],
(ii) fuzzy rules with fixed membership functions [127],
(iii) fuzzy rules and membership functions using (i) and (ii) in alternate steps [83],
(iv) membership functions and RB simultaneously [20],
(v) rules and RB structure and parameters (granularity, rule antecedent aggregation operator, rule semantics, rule base aggregation operator, defuzzification, membership function shape and parameters) simultaneously [111,112].

Contrary to neural network, clustering, rule induction and many other machine learning approaches, GAs provide a means to encode and evolve rule antecedent aggregation operators, rule semantics, RB aggregation operators and defuzzification methods. Therefore, except for search techniques, GAs remain today as one of the fewest knowledge acquisition schemes available to design and, in some sense, optimise FRBSs with respect to the design decisions above. Potential exceptions are heuristic search techniques, but currently no systematic experiments have been performed to critically compare GAs with heuristic search. A first study can be found in [56].

Hybrid approaches for supervised learning and increasing design efficiency is another alternative. In particular, genetic-neuro learning algorithms combined with least squares and singular value decomposition have been proposed [115].

GFSs also effectively integrate multiple sources of fuzzy knowledge into a single KB. Fuzzy knowledge integration [133] is a key issue that still challenges machine learning methods.

Clearly, increasing chromosome complexity to encode operators, rule semantics and defuzzification methods enlarges the search space considerably and challenges the computational efficiency of GA-based methods. In this case, design effort can be attenuated only via judicious exploration of design requirements. In other words, a decision must be made to find which of the items (i)–(v) above is the case.

Another critical point is how to evaluate GFS models in particular, and fuzzy system learning models in general. Contrary to other machine learning approaches, currently there is no systematic evaluation methodology for GFSs. Statistical analysis tools may again be of help as they do in classical machine learning. To develop an acceptable evaluation method it is necessary:

- to manage adequate and unified sets of benchmark problems for learning from data, e.g. classification, data mining, regression and control problems.
- to design experimental analysis models under an equal and significant number of runs, iterations, parameters, and execution time.

For example, the use of the factorial analysis of variance (ANOVA) and multiple comparison tests (see [135]) may introduce an important advantage to evaluate the performance of new proposals, and to compare them with previous approaches.

8. Questions and problems

The hybridisation of fuzzy logic and evolutionary computation in GFSs became an important research area during the last decade. To date, several hundred papers, special issues of different journals and books have been devoted [33]. Nowadays, as the field of GFSs matures and grows in visibility, there is an increasing concern on the integration of these two topics from a novel and more sophisticated perspective.

In what follows, we enumerate some open questions and scientific problems that suggest future research:

- *Hybrid intelligent systems. Balance between interpretability and accuracy. Sophisticated evolutionary algorithms*. As David Goldberg stated [51], the integration of single methods into hybrid intelligent systems goes beyond simple combinations. For him, the future of computational intelligence "*lies in the careful integration of the best constituent technologies*" and subtle integration of the abstraction power of fuzzy systems and the innovating power of genetic systems requires a design sophistication that goes further than putting everything together.

  Along this line of sophistication, in this contribution we have mentioned several recent approaches integrating different components, RBs, DBs, granularity, feature selection,..., or the integration of multiple sources of fuzzy knowledge. Usually, these approaches have had the objective of getting FRBSs with transparency and accuracy properties, that is, trying to get a trade-off between interpretability and accuracy in FRBSs.

  However, to pursue the balance between

  - necessary accuracy for modelling complex systems, and
  - interpretability degrees to provide expert knowledge,

  remains an open issue in the development of future GFSs.

  On the other hand, this line of sophistication in the integration and evolution of fuzzy system components and the introduction of new objectives, may lead us to the use of new and more sophisticated GAs, such as:

  - more sophisticated components (genetic operators, adaptation, ...), and
  - new and advanced evolutionary models, multiobjetive genetic algorithms [28,40], coevolutionary algorithms [109], parallel genetic algorithms [3,19], estimation distribution algorithms [89], etc.

  The integration of all of these previously mentioned requirements may be an important point to build hybrid intelligent systems beyond simple combinations.

- *New coding approaches*. As stated in [1], "the most influential factors in the quality of the solutions found by an evolutionary algorithm are a correct coding of the search space and an appropriate evaluation function of the potential solutions".

  Despite the chance GAs bring to design more complex fuzzy systems, efficient chromosome representation of rules, DB and RB structures still needs to be found. The most appropriate coding scheme is not known yet. There is a need to define new approaches to encode FRBS components to get a good balance between exploration and exploitation in the search space. Coding of different integration proposals of fuzzy system component needs to be revisited to find the most appropriate "natural coding".

- *Other evolution learning models*. As seen, the three classical genetic learning approaches are Michigan, Pittsburgh and Iterative Rule Learning. Improving mechanisms for these approaches or the development of other new ones may be necessary. Especial attention must be paid to Michigan learning approach. It is a classical genetic learning scheme introduced by Holland [68] as a technique for evolutionary computation and temporal difference learning. However, only a few number of Michigan approaches for learning fuzzy systems have been proposed in the literature. They are called fuzzy classifier systems.

  Recently, many results have been presented and successful applications reported which demonstrate that learning classifier systems may represent an interesting alternative to more traditional machine learning techniques in many application domains ranging from autonomous robotics to data mining [17,85,86,88].

  These recent developments have brought a resurgence and a rapid growth of this area, and this may be a starting point for the development of new fuzzy classifier learning systems.

- *New application areas*. Chap. 11 in [33] describes applications in classification, system modelling, control systems and robotics. Internet search and distributed computing brings considerable challenges. To match Internet knowledgeable resources contents to the meaning intended by a user search query is a task that remains to be done. To find cooperation protocols to coordinate multiagent systems to achieve a desired behavior still is a promise. The recent developments of GFSs to improve fuzzy linguistic modelling accuracy, to get better trade-off between interpretability and accuracy, to integrate different fuzzy system components, allow us to manage a large set of GFS tools and can lead to a growth of applications in different areas such as robotics, control, scheduling, data mining, internet, medicine, drug discovery, DNA sequencing, molecular biology, etc.

9. Concluding remarks

This paper provided an account of the current status of GFSs after 10 years of considerable research and development effort. In addition to a brief overview of the field to address the classical models and applications, new trends have been identified. A critical evaluation of the contribution that GFSs bring to knowledge acquisition and fuzzy rule base design was conducted, and challenges for further developments in the field were outlined. In this context we emphasize the need to build hybrid intelligent systems that go beyond simple combinations. Development of GFSs that offer acceptable trade-off between interpretability and accuracy is also a major requirement for efficient and transparent knowledge extraction. Discovery of more sophisticated and new evolutionary learning models of GFSs and its application to new areas and problems still remain as key questions for the next 10 years of GFSs. To further motivate such an endeavour, we have also offered valuable information for both, beginners and researchers working with GFSs and fuzzy knowledge extraction. The references have been elaborated to serve as a directory to guide the reader to the valuable results that GFSs contribute to the research and application oriented soft computing community.

References

[1] J.S. Aguilar-Ruiz, J.C. Riquelme, C. Del Valle, Improving the evolutionary coding for machine learning tasks, in Proc. of the 15th European Conf. on Artificial Intelligence (ECAI'02), Lyon, France, 2002, pp. 173–177.
[2] E. Alba, C. Cotta, J.M. Troya, Evolutionary design of fuzzy logic controllers using strongly-typed GP, Mathware Soft Comput. 6 (1) (1999) 109–124.
[3] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evolutionary Computation 6 (5) (2002) 443–462.
[4] R. Alcalá, J.M. Benítez, J. Casillas, O. Cordón, R. Pérez, Fuzzy control of HVAC systems optimised by genetic algorithms, Appl. Intell. 18 (2003) 155–177.
[5] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, Fuzzy graphs: features and taxonomy of learning methods for non-grid oriented fuzzy rule-based systems, J. Intell. Fuzzy Systems 11 (2001) 99–119.
[6] R. Alcalá, J. Casillas, O. Cordón, F. Herrera, Linguistic modeling with weighted double-consequent fuzzy rules based on cooperative coevolutionary learning, Integrated Comput. Aided Eng., 2003, to appear.
[7] G. Alpaydin, G. Dandar, S. Balkir, Evolution-based design of neural fuzzy networks using self-adapting genetic parameters, IEEE Trans. Fuzzy Systems 10 (2) (2002) 211–221.
[8] P.P. Angelov, Evolving Rule-Based Models. A Tool for Design of Flexible Adaptive Systems, Physica-Verlag, Wurzburg, 2002.
[9] R. Babuska, Fuzzy Modeling for Control, Kluwer Academic Press, Dordrecht, 1998.
[10] L. Baron, S. Achiche, M. Balazinski, Fuzzy decision support system knowledge base generation using a genetic algorithm, Int. J. Approximate Reasoning 28 (1) (2001) 125–148.
[11] J.M. Benítez, J.L. Castro, I. Requena, Are artificial neural networks black boxes? IEEE Trans. Neural Networks 8 (5) (1997) 1156–1164.
[12] A. Bonarini, Evolutionary learning of fuzzy rules: competition and cooperation, in: W. Pedrycz (Ed.), Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Press, Norwell, MA, 1996, pp. 265–284.
[13] P. Bonissone, Y.-T. Chen, K. Goebel, P.S. Khedkar, Hybrid soft computing systems: industrial and commercial applications, Proc. IEEE 87 (9) (1999) 1641–1667.
[14] P.P. Bonissone, P.S. Khedkar, Y. Chen, Genetic algorithms for automated tuning of fuzzy controllers: a transportation application, in Proc. Fifth IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'96), New Orleans, USA, 1996, pp. 674–680.
[15] P.P. Bonissone, R. Subbu, K.S. Aggour, Evolutionary optimization of fuzzy decision systems for automated insurance underwriting, in Proc. of IEEE Int. Conf. on Fuzzy Systems FUZZ-IEEE'02, 2002, pp. 1003–1008.
[16] B.P. Buckles, F.E. Petry, D. Prabhu, R. George, R. Srikanth, Fuzzy clustering with genetic search, in Proc. 1st IEEE Conf. on Evolutionary Computation (ICEC'94), Orlando, FL, USA, 1994, pp. 46–50.
[17] M.V. Butz, Anticipatory Learning Classifier Systems, Kluwer Academic Publishers, Dordrecth, 2002.
[18] W. Caminhas, H. Tavares, F. Gomide, W. Pedrycz, Fuzzy set based neural networks: structure, learning and application, J. Adv. Comput. Intell. 3 (3) (1999) 151–157.
[19] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer Academic Publishers, Dordrecht, 2000.
[20] B. Carse, T.C. Fogarty, A. Munro, Evolving fuzzy rule based controllers using genetic algorithms, Fuzzy Sets and Systems 80 (1996) 273–294.
[21] J. Casillas, O. Cordón, M.J. del Jesus, F. Herrera, Genetic feature selection in a fuzzy rule-based classification system learning process for high dimensional problems, Inform. Sci. 136 (2001) 169–191.
[22] L. Castillo, A. González, R. Pérez, Including a simplicity criterion in the selection of the best rule in a fuzzy genetic learning algorithm, Fuzzy Sets and Systems 120 (2) (2001) 309–321.

[23] J.L. Castro, J.M. Benítez, C. Mantas, Interpretation of artificial neural networks by means of fuzzy rules, IEEE Trans. Neural Networks 13 (1) (2002) 101–116.
[24] F. Cheong, R. Lai, Constraining the optimization of a fuzzy logic controller using an enhanced genetic algorithm, IEEE Trans. Systems Man Cybernet. B 30 (2000) 31–46.
[25] Z. Chi, H. Yan, T. Pham, Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition, World Scientific, Singapore, 1996.
[26] B.-C. Chien, J.Y. Lin, T.-P. Hong, Learning discriminant functions with fuzzy attributes for classification using genetic programming, Expert Systems Appl. 23 (1) (2002) 31–37.
[27] I-F. Chung, C.J. Lin, C.T. Lin, A GA-based fuzzy adaptive learning control network, Fuzzy Sets and Systems 112 (1) (2000) 65–84.
[28] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, Dordrecht, 2002.
[29] O. Cordón, F. Herrera, A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples, Int. J. Approximate Reasoning 17 (4) (1997) 369–407.
[30] O. Cordón, F. Herrera, A two-stage evolutionary process for designing TSK fuzzy rule-based systems, IEEE Trans. Systems Man Cybernet. 29 (6) (1999) 703–715.
[31] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, IEEE Trans. Fuzzy Systems 8 (3) (2000) 335–344.
[32] O. Cordón, F. Herrera, Hybridizing genetic algorithms with sharing scheme and evolution strategies for designing approximate fuzzy rule-based systems, Fuzzy Sets and Systems 118 (2) (2001) 235–255.
[33] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems—Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, Singapore, 2001.
[34] O. Cordón, F. Herrera, L. Magdalena, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base, Inform. Sci. 136 (1–4) (2001) 85–107.
[35] O. Cordón, F. Herrera, P. Villar, Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base, IEEE Trans. Fuzzy Systems 9 (4) (2001) 667–674.
[36] O. Cordón, F. Herrera, I. Zwir, Linguistic modeling by hierarchical systems of linguistic rules, IEEE Trans. Fuzzy Systems 10 (1) (2002) 2–20.
[37] O. Cordón, M.J. del Jesús, F. Herrera, Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods, Int. J. Intell. Systems 13 (10–11) (1998) 1025–1053.
[38] O. Cordón, M.J. del Jesús, F. Herrera, M. Lozano, MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach, Int. J. Intelligent Systems 14 (11) (1999) 1123–1153.
[39] I.G. Damousis, P. Dokopoulos, A fuzzy expert system for the forecasting of wind speed and power generation in wind farms, in Proc. of Int. Conf. on Power Industry Computer Applications PICA 2001, Sydney, Australia, 2001, pp. 63–69.
[40] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, New York, 2001.
[41] K. DeJong, Learning with genetic algorithms: an overview, Mach. Learning 3 (3) (1988) 121–138.
[42] Y. Dote, S.J. Ovaska, Industrial applications of soft computing: a review, Proc. IEEE 89 (9) (2001) 1243–1265.
[43] D. Driankov, H. Hellendoorn, M. Reinfrank, An Introduction to Fuzzy Control, Springer, Berlin, 1993.
[44] R. Fuller, Introduction to Neuro-Fuzzy Systems, Physica-Verlag, Wurzburg, 1999.
[45] T. Furuhashi, Development of if-then rules with the use of DNA coding, in: W. Pedrycz (Ed.), Fuzzy Evolutionary Computation, Kluwer Academic Publishers, Dordrecht, 1997, pp. 107–125.
[46] T. Furuhashi, K. Nakaoka, Y. Uchikawa, A new approach to genetic based machine learning and an efficient finding of fuzzy rules, in Proc. 1994 IEEE/Nagoya University WWW on Fuzzy Logic and Neural Networks/Genetic Algorithms, Nagoya, Japan, 1994, pp. 114–121.
[47] A. Geyer-Schulz, Fuzzy Rule-Based Expert Systems and Genetic Machine Learning, Physica-Verlag, Heidelberg, 1995.
[48] P.Y. Glorennec, Constrained optimization of FIS using an evolutionary method, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996, pp. 349–368.
[49] P.Y. Glorennec, Coordination between autonomous robots, Int. J. Approximate Reasoning 17 (4) (1997) 433–446.

[50] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[51] D.E. Goldberg, A meditation on the computational intelligence and its future, Technical Report IlliGAL Report #2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 2000. Foreword of the Proc. 2000 Int. Symp. on Computational Intelligence.
[52] D.E. Goldberg, The Design of Competent Genetic Algorithms: Steps Toward a Computational Theory of Innovation, Kluwer Academic Publishers, Dordrecht, 2002.
[53] A.F. Gómez-Skarmeta, F. Jiménez, Fuzzy modeling with hybrid systems, Fuzzy Sets and Systems 104 (1999) 199–208.
[54] A. González, R. Pérez, SLAVE: a genetic learning system based on an iterative approach, IEEE Trans. Fuzzy Systems 7 (2) (1999) 176–191.
[55] A. González, R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm, IEEE Trans. System Man Cybernet. B 31 (3) (2001) 417–425.
[56] A. González, R. Pérez, An experimental study about the search mechanism in SLAVE learning algorithm: hill-climbing methods versus genetic algorithm, Inform. Sci. 136 (1–4) (2001) 159–174.
[57] R. Gudwin, F. Gomide, W. Pedrycz, Context adaptation in fuzzy processing and genetic algorithms, Int. J. Intell. Systems 13 (10/11) (1998) 929–948.
[58] H.B. Gurocak, A genetic-algorithm-based method for tuning fuzzy logic controllers, Fuzzy Sets and Systems 108 (1) (1999) 39–47.
[59] L.O. Hall, J.C. Bezdek, S. Boggavarapu, A. Bensaid, Genetic fuzzy clustering, In Proc. NAFIPS'94, San Antonio, Texas, USA, 1994, pp. 411–415.
[60] U.D. Hanebeck, G.K. Schmidt, Genetic optimization of fuzzy networks, Fuzzy Sets and Systems 79 (1) (1996) 59–68.
[61] H. Heider, T. Drabe, A cascade genetic algorithm for improving fuzzy-system design, Int. J. Approximate Reasoning 17 (4) (1997) 351–368.
[62] F. Herrera, M. Lozano, Adaptation of genetic algorithm parameters based on fuzzy logic controllers, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996, pp. 95–125.
[63] F. Herrera, M. Lozano, J.L. Verdegay, Tuning fuzzy controllers by genetic algorithms, Int. J. Approx. Reasoning 12 (1995) 299–315.
[64] F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996.
[65] F. Hoffmann, O. Nelles, Genetic programming for model selection of TSK-fuzzy systems, Inform. Sci. 136 (1–4) (2001) 7–28.
[66] F. Hoffmann, G. Pfister, Evolutionary design of a fuzzy knowledge base for a mobile robot, Int. J. Approximate Reasoning 17 (4) (1997) 447–469.
[67] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[68] J.H. Holland, J.S. Reitman, Cognitive systems based on adaptive algorithms, in: D.A. Waterman, F. Hayes-Roth (Eds.), Pattern-Directed Inference Systems, Academic Press, New York, 1978, pp. 313–329.
[69] T.-P. Hong, J.-B. Chen, Finding relevant attributes and membership functions, Fuzzy Sets and Systems 103 (3) (1999) 389–404.
[70] P.-Y. Huang, S.-C. Lin, Y.-Y. Chen, Real-coded genetic algorithm based fuzzy sliding-mode control design for precision positioning, in Proc. IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 1998, pp. 1247–1252.
[71] H.-S. Hwang, Control strategy for optimal compromise between trip time and energy consumption in a high-speed railway, IEEE Trans. Systems Man Cybernet. 28 (6) (1998) 791–802.
[72] H. Ishibuchi, T. Murata, A genetic-algorithm-based fuzzy partition method for pattern classification problems, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996, pp. 555–578.
[73] H. Ishibuchi, T. Murata, I.B. Türksen, Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems, Fuzzy Sets and Systems 89 (1997) 135–150.
[74] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Trans. Fuzzy Systems 9 (4) (2001) 506–515.

[75] H. Ishibuchi, T. Nakashima, T. Murata, Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems, IEEE Trans. System Man Cybernet. 29 (1999) 601–618.

[76] H. Ishibuchi, T. Nakashima, T. Murata, Three-objective genetics-based machine learning for linguistic rule extraction, Inform. Sci. 136 (1–4) (2001) 109–133.

[77] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms, IEEE Trans. Fuzzy Systems 3 (3) (1995) 260–270.

[78] J.-S.R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[79] Y. Jin, Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement, IEEE Trans. Fuzzy Systems 8 (2) (2000) 212–220.

[80] C. Karr, Genetic algorithms for fuzzy controllers, AI Expert 6 (2) (1991) 26–33.

[81] C. Karr, E.J. Gentry, Fuzzy control of pH using genetic algorithms, IEEE Trans. Fuzzy Systems 1 (1) (1993) 46–53.

[82] D. Kim, H.-P. Lee, An optimal design of neuro-FLC by lamarckian co-adaption of learning and evolution, Fuzzy Sets and Systems 118 (2001) 319–337.

[83] J. Kinzel, F. Klawoon, R. Kruse, Modifications of genetic algorithms for designing and optimizing fuzzy controllers, in Proc. First IEEE Conf. on Evolutionary Computation (ICEC'94), Orlando, FL, USA, 1994, pp. 28–33.

[84] J.R. Koza, Genetic Programming III; Darwinian Invention and Problem Solving, Morgan Kaufman, San Francisco, CA, 1999.

[85] T. Kovacs, Learning classifier systems resources, Soft Comput. 6 (3–4) (2002) 240–243.

[86] T. Kovacs, What should a classifier system learn and how should we measure it? Soft Comput. 6 (3–4) (2002) 171–182.

[87] P.V. Krishnamraju, J.J. Buckley, K.D. Reilly, Y. Hayashi, Genetic learning algorithms for fuzzy neural nets, In Proc. Third IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'94), Orlando, FL, USA, 1994, pp. 1969–1974.

[88] P.L. Lanzi, W. Stolzmann, S.W. Wilson, Learning Classifier Systems: From Foundations to Applications, Vol. 1813, Lecture Notes in Computer Science, Springer, Berlin, 2000.

[89] P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, Dordrecht, 2001.

[90] H.-M. Lee, C.-M. Chen, J.-M. Chen, Y.-L. Jou, An efficient fuzzy classifier with feature selection based on fuzzy entropy, IEEE Trans. Systems Man Cybernet. B 31 (3) (2001) 426–432.

[91] M.A. Lee, H. Takagi, Integrating design stages of fuzzy systems using genetic algorithms, In Proc. 2nd IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'93), San Francisco, USA, 1993, pp. 613–617.

[92] C. Lin, C. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Trans. Comput. 40 (1991) 1320–1336.

[93] D.A. Linkens, H.O. Nyongesa, Evolutionary learning in fuzzy neural control systems, in Proc. Third European Congress on Fuzzy and Intelligent Technologies (EUFIT'95), Aachen, Germany, 1995, pp. 990–995.

[94] J. Liska, S.S. Melsheimer, Complete design of fuzzy logic system using genetic algorithms, in Proc. 3rd IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'94), Vol. 2, Orlando, FL, USA, 1994, pp. 1377–1382.

[95] L. Magdalena, Adapting the gain of an FLC with genetic algorithms, Int. J. Approximate Reasoning 17 (4) (1997) 327–349.

[96] L. Magdalena, F. Monasterio, A fuzzy logic controller with learning through the evolution of its knowledge base, Int. J. Approx. Reasoning 16 (3–4) (1997) 335–358.

[97] E. Mizutani, H. Takagi, D.M. Auslander, J.-S.R. Jang, Evolving colour recipes, IEEE Trans. Systems Man Cybernet. 30 (4) (2000) 537–550.

[98] S. Nascimiento, F. Moura-Pires, A genetic approach for fuzzy clustering with a validity measure function, in: X. Liu, P. Cohen, M.R. Berthold (Eds.), Proc. 2nd Int. Conf. on Intelligent Data Analysis (IDA'97). (LNCS 1280), Berlin, 1997. Springer, pp. 325–335.

[99] D. Nauck, F. Klawoon, R. Kruse, Foundations of Neuro-Fuzzy Systems, Wiley, New York, 1997.

[100] N.E. Nawa, T. Furuhashi, Fuzzy system parameters discovery by bacterial evolutionary algorithm, IEEE Trans. Fuzzy Systems 7 (5) (1999) 608–616.

[101] K.C. Ng, Y. Li, Design of sophisticated fuzzy logic controllers using genetic algorithms, in Proc. 3rd IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'94), Vol. 3, Orlando, FL, USA, 1994, pp. 1708–1712.

[102] S.E. Papadakis, J.B. Theocharis, A GA-based fuzzy modeling approach for generating TSK models, Fuzzy Sets and Systems 131 (1) (2002) 121–152.
[103] D. Park, A. Kandel, G. Langholz, Genetic-based new fuzzy reasoning models with application to fuzzy control, IEEE Trans. Systems Man Cybernet. 24 (1) (1994) 39–47.
[104] A. Parodi, P. Bonelli, A new approach to fuzzy classifier systems, in Proceedings of Fifth International Conference on Genetic Algorithms (ICGA'93), Morgan Kaufmann, Los Altos, CA, 1993, pp. 223–230.
[105] W. Pedrycz (Ed.), Fuzzy Modelling: Paradigms and Practice, Kluwer Academic Press, Dordrecht, 1996.
[106] W. Pedrycz (Ed.), Fuzzy Evolutionary Computation, Kluwer Academic Publishers, Dordrecht, 1997.
[107] C.A. Peña-Reyes, M. Sipper, Fuzzy CoCo: a cooperative coevolutionary approach to fuzzy modeling, IEEE Trans. Fuzzy Systems 9 (5) (2001) 727–737.
[108] D.T. Pham, D. Karaboga, Optimum design of fuzzy logic controllers using genetic algorithms, J. System Eng. 1 (1991) 114–118.
[109] M.A. Potter, K.A. De Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, Evolutionary Comput. 8 (1) (2000) 1–29.
[110] A. Rajapakse, K. Furuta, S. Kondo, Evolutionary learning of fuzzy logic controllers and their adaptation through perpetual evolution, IEEE Trans. Fuzzy Systems 10 (3) (2002) 309–321.
[111] M. Regatieri-Delgado, Automatic design of fuzzy systems: a co-evolutionary approach, Ph.D. Thesis, State University of Campinas, Sao Paulo, Brazil, 2002 (in Portuguese).
[112] M. Regattieri-Delgado, F. Von-Zuben, F. Gomide, Hierarchical genetic fuzzy systems, Inform. Sci. 136 (1–4) (2001) 29–52.
[113] I. Rojas, J. Gonzalez, H. Pomares, F.J. Rojas, F.J. Fernández, A. Prieto, Multidimensional and multideme genetic algorithms for the construction of fuzzy systems, Int. J. Approx. Reasoning 26 (3) (2001) 179–210.
[114] H. Roubos, M. Setnes, Compact and transparent fuzzy models through iterative complexity reduction, IEEE Trans. Fuzzy Systems 9 (4) (2001) 515–524.
[115] M. Russo, FuGeNeSys: a fuzzy genetic neural system for fuzzy modeling, IEEE Trans. Fuzzy Systems 6 (2000) 373–388.
[116] L. Sánchez, J.A. Corrales, Niching scheme for steady state GA-P and its application to fuzzy rule base classifiers induction, Mathware Soft Comput. 7 (2–3) (2000) 337–350.
[117] L. Sánchez, I. Couso, J.A. Corrales, Combining GP operators with SA search to evolve fuzzy rule-based classifiers, Inform. Sci. 136 (1–4) (2001) 175–191.
[118] E. Sanchez, T. Shibata, L. Zadeh (Eds.), Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives, World Scientific, Singapore, 1997.
[119] A. Satyadas, K. KrishnaKumar, EFM-based controllers for space attitude control: applications and analysis, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996, pp. 152–171.
[120] M. Setnes, H. Roubos, GA-fuzzy modeling and classification: complexity and performance, IEEE Trans. Fuzzy Systems 8 (5) (2000) 509–522.
[121] M. Shim, S. Seong, B. Ko, M. So, Application of evolutionary computation at LG electronics, in Proc. of IEEE Int. Conf. on Fuzzy Systems FUZZ-IEEE'99, Seoul, South Korea, 1999, pp. 1802–1806.
[122] K. Shimojima, N. Kubota, T. Fukuda, Virus-evolutionary genetic algorithm for fuzzy controller optimization, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzburg, 1996, pp. 369–388.
[123] R. Silipo, M. Berthold, Input features' impact on fuzzy decision processes, IEEE Trans. System Man Cybernet. B 30 (6) (2000) 821–834.
[124] S.F. Smith, A learning system based on genetic adaptive algorithms, Doctoral Dissertation, Department of Computer Science, University of Pittsburgh, 1980.
[125] R. Subbu, A. Anderson, P.P. Bonissone, Fuzzy logic controlled genetic algorithms versus tuned genetic algorithms: an agile manufacturing application, in Proc. IEEE Int. Symp. on Intelligent Control (NIST), Gaithersburg, MD, USA, 1998.
[126] H. Takagi, I. Hayashi, NN-driven fuzzy reasoning, Int. J. Approx. Reasoning 5 (3) (1991) 191–212.
[127] P. Thrift, Fuzzy logic synthesis with genetic algorithms, in Proc. Fourth Int. Conf. on Genetic Algorithms (ICGA'91), San Diego, USA, Morgan Kaufmann, Los Altos, CA, 1991, pp. 509–513.

[128] M. Valenzuela-Rendón, The fuzzy classifier system: motivations and first results, in: H.P. Schwefel, R. Männer (Eds.), Proc. First Int. Conf. on Parallel Problem Solving from Nature-PPSN I, Springer, Berlin, 1991, pp. 330–334.
[129] T. Van Le, Evolutionary fuzzy clustering, in Proc. 2nd IEEE Conf. on Evolutionary Computation (ICEC'95), Vol. 2, Perth, Australia, 1995, pp. 753–758.
[130] J.R. Velasco, Genetic-based on-line learning for fuzzy process control, Int. J. Intelligent System 13 (10–11) (1998) 891–903.
[131] G. Venturini, SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts, In Proc. European Conf. on Machine Learning, Vienna, 1993, pp. 280–296.
[132] S. Voget, M. Kolonko, Multidimensional optimization with a fuzzy genetic algorithm, J. Heuristics 4 (3) (1998) 221–244.
[133] C.H. Wang, T.P. Hong, S.S. Tseng, Integrating fuzzy knowledge by genetic algorithms, IEEE Trans. Evol. Comput. 2 (4) (1998) 138–149.
[134] B. Yuan, G.J. Klir, J.F. Swan-Stone, Evolutionary fuzzy c-means clustering algorithm, in Proc. 4th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'95), Yokohama, Japan, 1995, pp. 2221–2226.
[135] J.H. Zar, Biostatistical Analysis, 4th Edition, Prentice Hall, Englewood Cliffs, NJ, 1999.