Fuzzy Evolutionary Algorithms and Genetic Fuzzy Systems: A Positive Collaboration between Evolutionary Algorithms and Fuzzy Systems

F. Herrera and M. Lozano

Abstract. There are two possible ways for integrating fuzzy logic and evolutionary algorithms. The first one involves the application of evolutionary algorithms for solving optimization and search problems related with fuzzy systems, obtaining genetic fuzzy systems. The second one concerns the use of fuzzy tools and fuzzy logic-based techniques for modelling different evolutionary algorithm components and adapting evolutionary algorithm control parameters, with the goal of improving performance. The evolutionary algorithms resulting from this integration are called fuzzy evolutionary algorithms. In this chapter, we shortly introduce genetic fuzzy systems and fuzzy evolutionary algorithms, giving a short state of the art, and sketch our vision of some hot current trends and prospects. In essence, we paint a complete picture of these two lines of research with the aim of showing the benefits derived from the synergy between evolutionary algorithms and fuzzy logic.

1 Introduction

Computational intelligence techniques such as artificial neural networks [157], fuzzy logic [204], and genetic algorithms (GAs) [87, 63] are popular research subjects, since they can deal with complex engineering problems which are difficult to solve by classical methods [109].

Hybrid approaches have attracted considerable attention in the computational intelligence community. One of the most popular approaches is the hybridization between fuzzy logic and GAs leading to genetic fuzzy systems (GFSs) [38] and fuzzy evolutionary algorithms [79, 149, 183]. Both are well known examples of a positive collaboration between soft computing techniques.

A GFS is basically a fuzzy rule based system (FRBS) augmented by a learning process based on evolutionary computation, which includes GAs, genetic programming, and evolution strategies, among other evolutionary algorithms (EAs) [56].

F. Herrera and M. Lozano
Department of Computer Science and Artificial Intelligence University of Granada, 18071 - Granada, Spain
e-mail: herrera@decsai.ugr.es, lozano@decsai.ugr.es

C.L. Mumford and L.C. Jain (Eds.): Computational Intelligence, ISRL 1, pp. 83–130.
springerlink.com

The automatic definition of a FRBS can be seen as an optimization or search problem, and GAs are a well known and widely used global search technique with the ability to explore a large search space for suitable solutions only requiring a performance measure. In addition to their ability to find near optimal solutions in complex search spaces, the generic code structure and independent performance features of GAs make them suitable candidates to incorporate a priori knowledge. In the case of FRBSs, this a priori knowledge may be in the form of linguistic variables, fuzzy membership function parameters, fuzzy rules, etc. These capabilities extended the use of GAs in the development of a wide range of approaches for designing FRBSs over the last few years.

The behaviour of the EAs in general, and GAs in particular, is strongly determined by the balance between exploration (to investigate new and unknown areas in a search space) and exploitation (to make use of knowledge acquired by exploration to reach better positions on the search space). The GA control parameter settings, such as mutation probability, crossover probability, and population size, are key factors in the determination of the exploitation versus exploration tradeoff. It has long been acknowledged that they have a significant impact on GA performance. If poor settings are used, the exploration/exploitation balance may not be reached in a profitable way; the GA performance shall be severely affected due to the possibility of premature convergence. Finding robust control parameter settings is not a trivial task, since their interaction with GA performance is a complex relationship and the optimal ones are problem-dependent. Furthermore, different control parameter values may be necessary during the course of a run to induce an optimal exploration/exploitation balance. For these reasons, adaptive GAs have been built that dynamically adjust selected control parameters or genetic operators during the course of evolving a problem solution. Their objective is to offer the most appropriate exploration and exploitation behaviour. FRBSs provide a tool which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. They are particularly suited to model the relationship between variables in environments that are either ill-defined or very complex. The adaptation of GA parameters is one such complex problem that may benefit from the use of FRBS, producing the so-called fuzzy adaptive GAs. If we consider any kind of EA that can be improved by means of fuzzy logic based techniques, then we can use the name of fuzzy EAs.

In this chapter we shortly introduce GFSs and fuzzy EAs, giving a short state of the art, and sketch our vision of some hot current trends and prospects.

The remainder of this article is organized as follows. In Section 2, we provide an overview of FRBSs. In Section 3, we focus our attention to GFSs. In Section 4, we tackle fuzzy EAs. Finally, in Section 5, we provide some concluding remarks of this work.

2 Fuzzy Rule Based Systems

FRBSs constitute one of the main contributions of fuzzy logic. The basic concepts which underlie these fuzzy systems are those of linguistic variables and fuzzy

IF-THEN rules. A linguistic variable, as its name suggests, is a variable whose values are words rather than numbers, e.g., small, young, very hot and quite slow. Fuzzy IF-THEN rules are of the general form: if antecedent(s) then consequent(s), where antecedent and consequent are fuzzy propositions that contain linguistic variables. A fuzzy IF-THEN rule is exemplified by "if the temperature is high then the fan-speed should be high". With the objective of modelling complex and dynamic systems, FRBSs handle fuzzy rules by mimicking human reasoning (much of which is approximate rather than exact), reaching a high level of robustness with respect to variations in the system's parameters, disturbances, etc. The set of fuzzy rules of an FRBS can be derived from subject matter experts or extracted from data through a rule induction process.

In this section, we present a brief overview of the foundations of FRBSs, with the aim of illustrating the way they behave. In particular, in Section 2.1, we introduce the important concepts of fuzzy sets and linguistic variables. In Section 2.2, we deal with the basic elements of FRBSs. Finally, in Section 2.3, we describe a simple instance of FRBS, a fuzzy logic controller for the inverted pendulum.

2.1 Preliminaries: Fuzzy Set and Linguistic Variable

A *fuzzy set* is distinct from a crisp set in that it allows its elements to have a degree of membership. The core of a fuzzy set is its membership function: a surface or line that defines the relationship between a value in the set's domain and its degree of membership. In particular, according to the original ideal of Zadeh [208], membership of an element x to a fuzzy set A, denoted as $\mu_A(x)$ or simply $A(x)$, can vary from 0 (full non-membership) to 1 (full membership), i.e., it can assume all values in the interval $[0,1]$. Clearly, a fuzzy set is a generalization of the concept of a set whose membership function takes on only two values $\{0,1\}$.

The value of $A(x)$ describes a degree of membership of x in A. For example, consider the concept of *high temperature* in an environmental context with temperatures distributed in the interval [0, 50] defined in degree centigrade. Clearly 0^{o}C is not understood as a high temperature value, and we may assign a null value to express its degree of compatibility with the high temperature concept. In other words, the membership degree of 0^{o}C in the class of high temperatures is zero. Likewise, 30^{o}C and over are certainly high temperatures, and we may assign a value of 1 to express a full degree of compatibility with the concept. Therefore, temperature values in the range [30, 50] have a membership value of 1 in the class of high temperatures. From 0^{o}C to 30^{o}C, the degree of membership in the fuzzy set high temperature gradually increases, as exemplified in Figure 1, which actually is a membership function $A : T \to [0,1]$ characterizing the fuzzy set of high temperatures in the universe $T = [0,50]$. In this case, as temperature values increase they become more and more compatible with the idea of high temperature.

Linguistic variables are variables whose values are not numbers but words or sentences in a natural or artificial language. This concept has clearly been developed as a counterpart to the concept of a numerical variable. More precisely, a linguistic

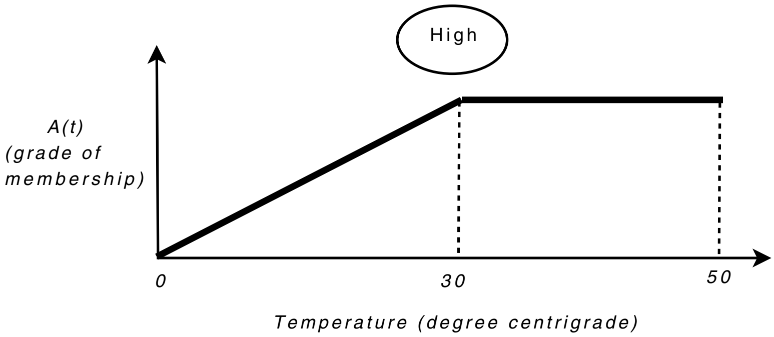


Fig. 1 Membership function

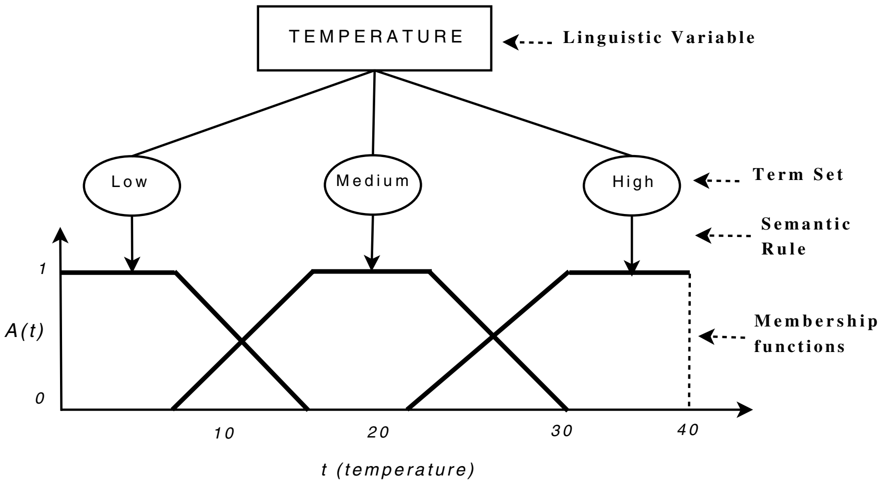


Fig. 2 Example of linguistic variable *Temperature* with three linguistic terms

variable L is defined as a quintuple [107]: $L = (x, A, X, g, m)$, where x is the base variable, $A = \{A_1, A_2, \ldots, A_N\}$ is the set of *linguistic terms* of L (called *term-set*), X is the domain (universe of discourse) of the base variable, g is a syntactic rule for generating linguistic terms and m is a semantic rule that assigns to each linguistic term its *meaning* (a fuzzy set in X). Figure 2 shows an example of a linguistic variable *Temperature* with three linguistic terms "Low, Medium, and High". The base variable is the temperature given in appropriate physical units.

Each underlying fuzzy set defines a portion of the variable's domain; but this portion is not uniquely defined. Fuzzy sets overlap as a natural consequence of their elastic boundaries. Such an overlap not only implements a realistic and functional semantic mechanism for defining the nature of a variable when it assumes various data values but provides a smooth and coherent transition from one state to another.

2.2 Basic Elements of FRBSs

The essential part of FRBSs is a set of IF-THEN linguistic rules, whose antecedents and consequents are composed of fuzzy statements, related by the dual concepts of fuzzy implication and the compositional rule of inference.

An FRBS is composed of a *knowledge base* (KB), that includes the information in the form of IF-THEN fuzzy rules;

IF *a set of conditions are satisfied*
THEN *a set of consequents can be inferred*

and an *inference engine module* that includes:

- A *fuzzification interface*, which has the effect of transforming crisp data into fuzzy sets.
- An *inference system*, that uses them together with the KB to make inference by means of a reasoning method.
- A *defuzzification interface*, that translates the fuzzy rule action thus obtained to a real action using a defuzzification method.

FRBSs can be broadly categorized into different families:

- The first includes linguistic models based on collections of IF-THEN rules, whose antecedents are linguistic values, and the system behaviour can be described in natural terms. The consequent is an output action or class to be applied. For example, we can denote them as:
  R_i : If X_{i1} is A_{i1} and $\cdots$ and X_{in} is A_{in} then Y is B_i
  or
  R_i : If X_{i1} is A_{i1} and $\cdots$ and X_{in} is A_{in} then C_k with w_{ik}
  with $i = 1$ to M, and with X_{i1} to X_{in} and Y being the input and output variables for regression respectively, and C_k the output class associated to the rule for classification, with A_{i1} to A_{in} and B_i being the involved antecedents and consequent labels, respectively, and w_{ik} the certain factor associated to the class. They are usually called *linguistic* FRBSs or *Mamdani* FRBSs [134].
- The second category is based on a rule structure that has fuzzy antecedent and functional consequent parts. This can be viewed as the expansion of piece-wise linear partition represented as
  R_i : If X_{i1} is A_{i1} and $\cdots$ and X_{in} is A_{in} then $Y = p(X_{i1}, \cdots, X_{in})$,
  with $p(\cdot)$ being a polynomial function, usually a linear expression, $Y = p_0 + p_1 \cdot X_{i1} + \cdots + p_n \cdot X_{in}$. The approach approximates a nonlinear system with a combination of several linear systems. They are called *Takagi and Sugeno's* type fuzzy systems [177] (TS-type fuzzy systems).
- Other kinds of fuzzy models are the approximate or scatter partition FRBSs, which differ from the linguistic ones in the direct use of fuzzy variables [4]. Each fuzzy rule thus presents its own semantic, i.e., the variables take different fuzzy sets as values (and not linguistic terms from a global term set). The fuzzy rule structure is then as follow:

R_i : If X_{i1} is $\hat{A}_{i1}$ and $\cdots$ and X_{in} is $\hat{A}_{in}$ then Y is $\hat{G}_i$
with $\hat{A}_{ij}$ to $\hat{A}_{in}$ and $\hat{G}_i$ being fuzzy sets. The major difference with respect to the rule structure considered in linguistic FRBSs is that rules of an approximate nature are semantics free, whereas descriptive rules operate in the context formulated by means of the linguistic semantics.

In linguistic FRBSs, the KB is composed of two components, a *data base* (DB) and a *rule base* (RB).

- A DB, containing the linguistic term sets considered in the linguistic rules and the membership functions defining the semantics of the linguistic labels.
  Each linguistic variable involved in the problem will have an associated fuzzy partition of its domain representing the fuzzy set associated with each of its linguistic terms. Figure 5 shows an example of a fuzzy partition with five labels. This can be considered as a discretization approach for continuous domains where we establish a membership degree to the items (labels), we have an overlapping between them, and the inference engine manages the matching between the patterns and the rules, providing an output according to the rule consequents with a positive matching. The determination of the fuzzy partitions is crucial in fuzzy modelling [11], and the granularity of the fuzzy partition plays an important role for the FRBS behaviour [39].
  If we manage approximate FRBSs, then we do not have a DB due to the fact that rules have associated the fuzzy values.
- An RB, comprises a collection of linguistic rules that are joined by a rule connective ("also" operator). In other words, multiple rules can fire simultaneously for the same input.

The inference engine of FRBSs acts in a different way depending on the kind of problem (classification or regression) and the kind of fuzzy rules (linguistic ones, TS-ones, etc). It always includes a fuzzification interface that serves as the input to the fuzzy reasoning process, an inference system that infers from the input to several resulting output (fuzzy set, class, etc) and the defuzzification interface or output interface that converts the fuzzy sets obtained from the inference process into a crisp action that constitutes the global output of the FRBS, in the case of regression problems, or provide the final class associated to the input pattern according to the inference model.

The generic structure of an FRBS is shown in Figure 3.

For more information about fuzzy systems the following books may be consulted [204, 113, 38, 94]. For different issues associated with the trade-off between the interpretability and accuracy of FRBSs, the two following edited books present a collection of contributions on the topic [25, 26].

Finally, we must point out that we can find a lot of applications of FRBSs in all areas of engineering, sciences, medicine, etc. At present it is very easy to search for these applications using the publisher web search tools focusing the search in journals of different application areas.

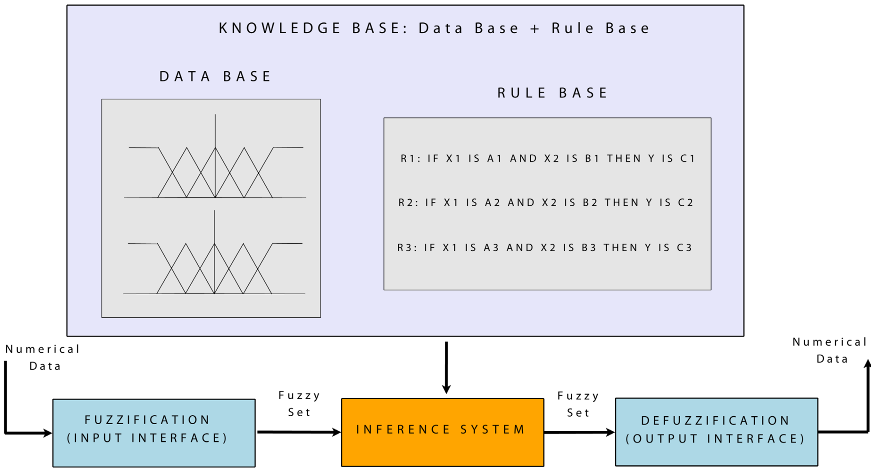


Fig. 3 Structure of an FRBS

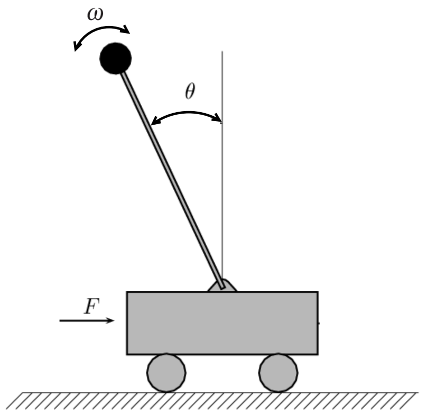
2.3 Example of FRBS: Fuzzy Logic Control of an Inverted Pendulum

Fuzzy logic controllers [53] are a particular model of FRBS that provide a tool which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. In these controllers, the domain knowledge is represented by a set of fuzzy IF-THEN rules that approximate a mapping from a state space **X** to an output space **Y**. They have been used in many practical applications, especially industrial ones in Japan and Europe. Industrial success stories of fuzzy control include portable video cameras, automatic transmission of automobiles, furnace temperature, robotics, urban underground railway, and banking.

The example of the inverted pendulum given in [205] is selected to illustrate elementary fuzzy control principles. Consider the problem of keeping an inverted pendulum (which is fixed) articulated at a fixed point on a mobile cart. The cart can move forward and backward, and the controller decides on the direction and acceleration of the cart (Figure 4).

To balance an upright pendulum, we know from naive physics that the control force F should be chosen according to the magnitudes of the input variables θ and ω that measure the angle from the upright position and the angular velocity, respectively. The relation between these variables is linguistic, a much weaker form than differential equations. That is exactly what happens in a human mind that processes information qualitatively. Humans choose F using common sense knowledge in the form of rules such as “if the pendulum is in a balanced position, then hold very still, that is, do not apply any force”. By taking all such rules into account, the inverted pendulum can be successfully controlled.

Fig. 4 Inverted pendulum



A sensor measures θ and ω (state variables) and a fuzzy logic controller may adjust F (output or control space) via a real time feedback loop with the objective of taking the pendulum to the vertical position. While the classical equations of motion of this system are extremely complicated and depend upon the specific characteristics of the pendulum (mass distribution, length), Yamakawa [205] found a set of linguistic fuzzy rules providing a stable fuzzy control of the pendulum independently of its characteristics. They are the following:

Rule 1. IF θ is PM AND ω is ZR THEN F is PM.
Rule 2. IF θ is PS AND ω is PS THEN F is PS.
Rule 3. IF θ is PS AND ω is NS THEN F is ZR.
Rule 4. IF θ is NM AND ω is ZR THEN F is NM.
Rule 5. IF θ is NS AND ω is NS THEN F is NS.
Rule 6. IF θ is NS AND ω is PS THEN F is ZR.
Rule 7. IF θ is ZR AND ω is ZR THEN F is ZR.

The linguistic term set for θ, ω, and F is {Negative Large (NL), Negative Medium (NM), Negative Small (NS), Zero (ZR), Positive Small (PS), Positive Medium (PM), Positive Large (PL)}, which has associated the fuzzy partition of their corresponding domains shown in Figure 5.

Given a sensor measured state (θ, ω), the inference obtained from the fuzzy controller is the result of interpolating among the response of these linguistic fuzzy rules. The inference's outcome is a membership function defined on the output space, which is then aggregated (defuzzified) to produce a crisp output.

The fuzzy logic controller described above is an example of linguistic FRBS. However, the problem of controlling the inverted pendulum may be tackled as well by means of a fuzzy logic controller based on the TS-type fuzzy system model. In this case, possible TS-type rules may include:

If θ is ZR and ω is ZR then $F = 0$.
If θ is PS and ω is ZR then $F = 0.5 \times \theta$.
If θ is PS and ω is NS then $F = 0.4 \times \theta + 0.6 \times \omega$.

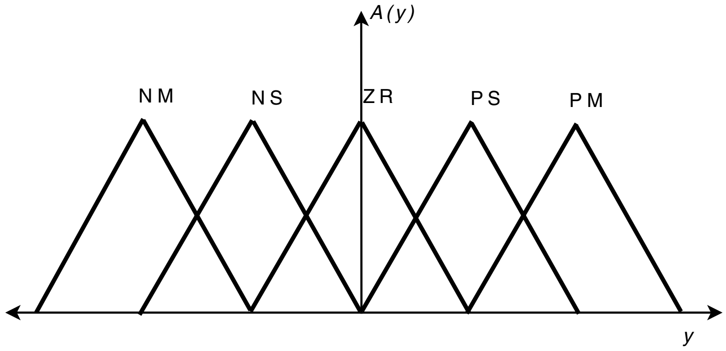


Fig. 5 Membership functions of the linguistic variables (where y stands for θ, ω, and F)

3 Genetic Fuzzy Systems

FRBSs constitute an extension to classical rule-based systems, because they deal with "IF-THEN" rules, whose antecedents and consequents are composed of fuzzy logic statements, instead of classical ones. They have demonstrated their ability for control problems [146], modelling [148], classification or data mining [113, 94] in a huge number of applications.

A GFS is basically a fuzzy system augmented by a learning process based on evolutionary computation, which includes GAs, genetic programming, and evolution strategies, among other EAs. Figure 6 illustrates this idea, where the genetic process learns or tunes different components of an FRBS.

The central aspect of the use of a GA for automatic learning of an FRBS is that the KB design process can be analyzed as an optimization problem.

From the optimization point of view, to find an appropriate KB is equivalent to coding it as a parameter structure and then finding the parameter values that give us the optimum for a fitness function. The KB parameters provide the search space that is transformed according to a genetic representation. Therefore, the first step in designing a GFS is to decide which parts of the KB are subject to optimization by the GA.

In the last few years we observe an increase of published papers in the topic due to the high potential of GFSs. In contrast to neural networks, clustering, rule induction and many other machine learning approaches, GAs provide a means to encode and evolve rule antecedent aggregation operators, different rule semantics, rule base aggregation operators and defuzzification methods. Therefore, GAs remain today as one of the few knowledge acquisition schemes available to design and, in some sense, optimize FRBSs with respect to the design decisions, allowing decision makers to decide what components are fixed and which ones evolve according to the performance measures.

The predominant type of GFS is that focused on FRBSs. However other kinds of GFSs have been developed, with successful results. They include genetic fuzzy

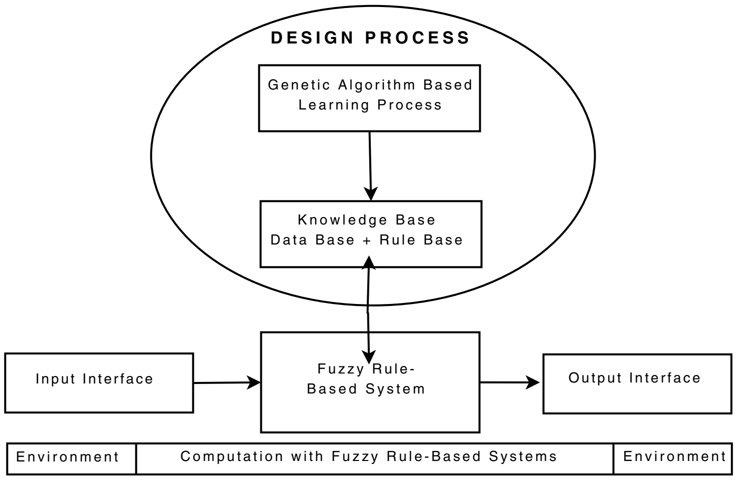


Fig. 6 Genetic fuzzy systems

neural networks and genetic fuzzy clustering algorithms. We will not analyze them in this papers. Readers can find an extended introduction to them in [38] (chapter 10).

In this section, we propose a taxonomy of GFSs focused on the FRBS components and sketch our vision of some hot current trends of GFSs [73].

3.1 Taxonomy of Genetic Fuzzy Systems

The central aspect on the use of GAs for automatic learning of FRBSs is that the design process can be analyzed as a search problem in the space of models, such as the space of rule sets, by means of the coding of the model in a chromosome.

From the optimization point of view, to find an appropriate fuzzy model is equivalent to code it as a parameter structure and then to find the parameter values that give us the optimum for a concrete fitness function. Therefore, the first step in designing a GFS is to decide which parts of the fuzzy system are subjected to optimization by the GA coding them into chromosomes.

We divide the GFS approaches into two processes, tuning and learning. It is difficult to make a clear distinction between tuning and learning processes, since establishing a precise borderline becomes as difficult as defining the concept of learning itself. The first fact that we have to take into consideration is the existence or not of a previous KB, including DB and RB. In the framework of GFSs we can briefly introduce the following distinction.

- Genetic tuning. If there exists a KB, we apply a genetic tuning process for improving the FRBS performance but without changing the existing RB. That is, to adjust FRBS parameters for improving its performance, maintaining the same RB.

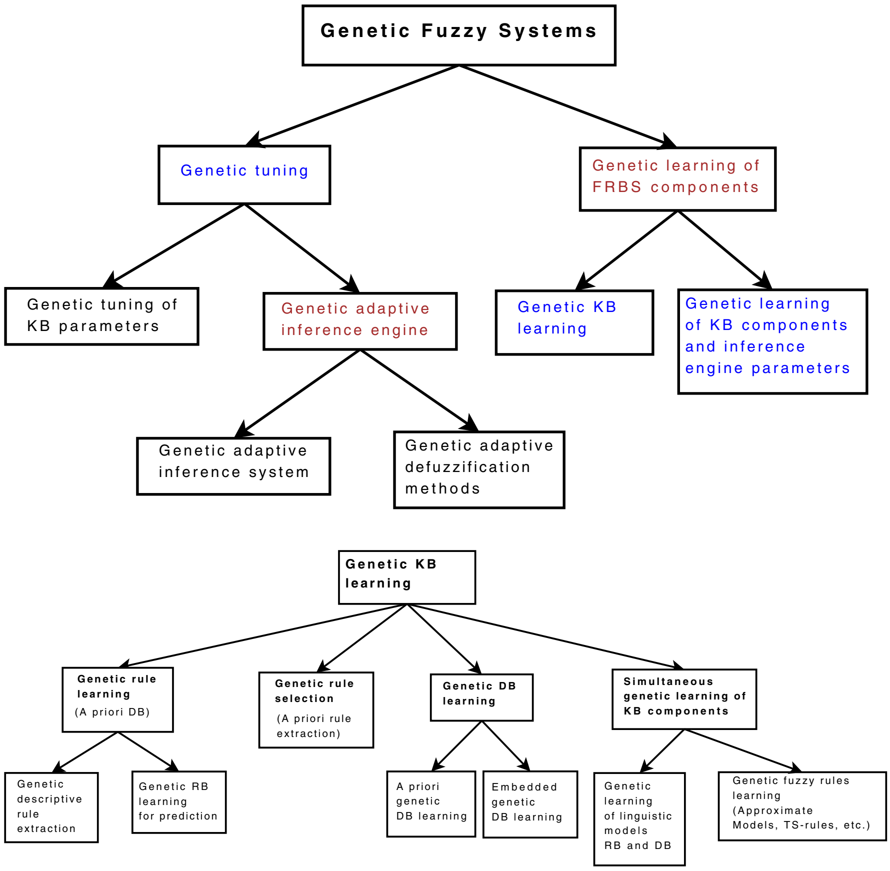


Fig. 7 GFSs Taxonomy

- Genetic learning. The second possibility is to learn KB components (where we can even include an adaptive inference engine). That is, to involve the learning of KB components among other FRBS components.

We classify the proposals according to these two processes and according to the FRBS components involved in the genetic learning process. In this way, we consider the taxonomy shown in Figure 7 [73].

There are three main areas in the taxonomy that we can observe in the first tree: genetic tuning, genetic KB learning, and genetic learning of KB components and inference engine parameters.

In the following, we briefly analyze the three areas. We will provide some references as examples for every approach, but we do not present an exhaustive list of papers, this is far from the chapter's objective.

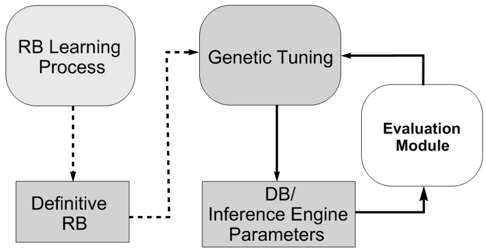


Fig. 8 Genetic tuning process

Genetic tuning

With the aim of making the FRBS perform better, some approaches try to improve the preliminary DB definition or the inference engine parameters once the RB has been derived. A graphical representation of this kind of tuning is shown in Figure 8.

The following three tuning possibilities can be considered (see the sub-tree under "genetic tuning").

1. *Genetic tuning of KB parameters.* In order to do so, a tuning process considering the whole KB obtained (the preliminary DB and the derived RB) is used a posteriori to adjust the membership function parameters. Nevertheless, the tuning process only adjusts the shapes of the membership functions and not the number of linguistic terms in each fuzzy partition, which remains fixed from the beginning of the design process. In [100], we can find a first and classic proposal on tuning. We can also find recent proposals that introduce linguistic modifiers for tuning the membership functions, see [24]. This latter approach is close to the inference engine adaptation.
2. *Genetic adaptive inference systems.* The main aim of this approach is the use of parameterized expressions in the Inference System, sometimes called Adaptive Inference Systems, for getting higher cooperation among the fuzzy rules and therefore more accurate fuzzy models without loosing the linguistic rule interpretability. In [8, 42, 43], we can find proposals in this area focused in regression and classification.
3. *Genetic adaptive defuzzification methods.* The most popular technique in practice, due to its good performance, efficiency and easier implementation, is to apply the defuzzification function to every inferred rule fuzzy set (getting a characteristic value) and to compute them by a weighted average operator. This method introduces the possibility of using parameter based average functions, and the use of GAs can allow us to adapt the defuzzification methods. In [105], we can find a proposal in this area.

Genetic KB learning

As a second big area we find the learning of KB components. We will now describe the four approaches that can be found within the genetic learning of a KB (see the second tree under "genetic KB learning").

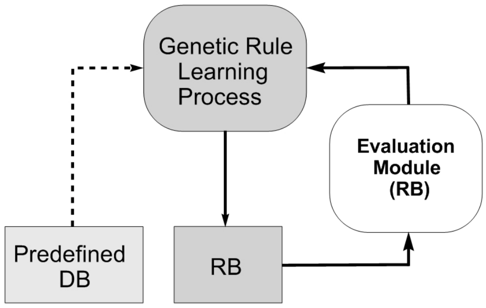


Fig. 9 Genetic rule learning process

1. *Genetic rule learning.* Most of the approaches proposed to automatically learn the KB from numerical information have focused on the RB learning, using a predefined DB. The usual way to define this DB involves choosing a number of linguistic terms for each linguistic variable (an odd number between 3 and 9, which is usually the same for all the variables) and setting the values of the system parameters by an uniform distribution of the linguistic terms into the variable universe of discourse. Figure 9 shows this type of RB learning graphically. The pioneer proposal for this approach can be found in [180].

   On the other hand, we also find approaches that are focused on the extraction of some descriptive rules for data mining problems (association rules, subgroup discovery, etc.) [102, 48].
2. *Genetic rule selection.* Sometimes we have a large number of rules extracted via a data mining method that subsequently provide us with a large number of rules associated with our problem. A big RB and an excessive number of rules makes it difficult to understand the FRBS behaviour. Thus we can find different kinds of rules in a fuzzy rule set: irrelevant rules, redundant rules, erroneous rules and conflictive rules, which perturb the FRBS performance when they coexist with others. To face this problem we can use a genetic rule selection process for obtaining an optimized subset of rules from a previous fuzzy rule set, by selecting some of them. Figure 10 illustrates this idea graphically. In [95] we can find the most classic and first contribution in this area and in [92] we can find the first journal paper on multiobjective genetic rule selection.

   We must point out that rule selection can be combined with tuning approaches, to try to get a good rule set together with a tuned set of parameters. In [24, 5], we can find two recent proposal that combines genetic tuning with rule selection.
3. *Genetic DB learning.* There is another way to generate the whole KB that considers two different processes to derive each component, DB and RB. A DB generation process allows us to learn the shape or the membership functions and other DB components such as the scaling functions, the granularity of the fuzzy partitions, etc. This DB generation process can use a measure for evaluating the quality of the DB, we can call it "a priori genetic DB learning". The second possibility is to consider an embedded genetic learning process where the DB generation process wraps an RB learning one working as follows: each time a

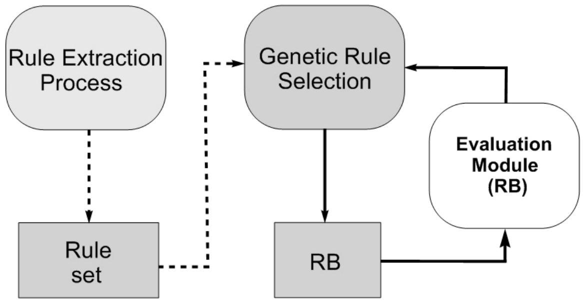


Fig. 10 Genetic rule selection process

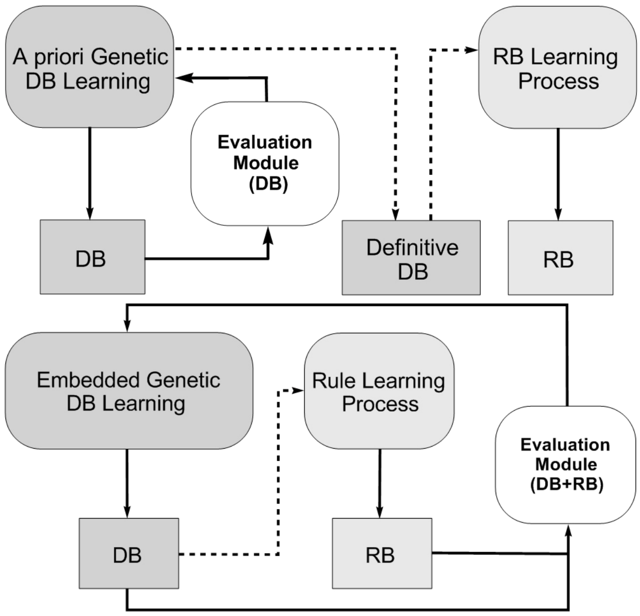


Fig. 11 Genetic DB learning (embedded and a priori)

DB has been obtained by the DB definition process, the RB generation method is used to derive the rules, and some type of error measure is used to validate the whole KB obtained. We should note this operation mode involves a partitioning of the KB learning problem. These two kinds of learning models are represented in Figure 11. In [41], we can find a proposal following the embedded genetic DB learning.

4. *Simultaneous genetic learning of KB components.* Other approaches try to learn the two components of the KB simultaneously. This kind of learning is depicted in Figure 12. Working in this way, they have the possibility of generating better definitions but there is a need to deal with a larger search space that makes the

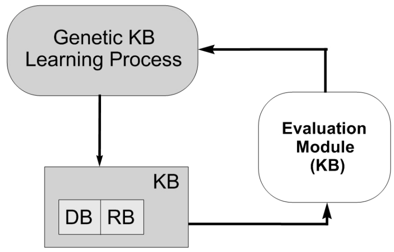


Fig. 12 Genetic KB learning process

learning process more difficult and slow. In [85], we can find a contribution that uses the simultaneous genetic KB learning process.

Genetic learning of KB components and inference engine parameters

This is the last area of GFSs taxonomy, belonging to a hybrid model between an adaptive inference engine and KB components learning. We can find novel approaches that try to find high cooperation between the inference engine via parameter adaptation and the learning of KB components, including both in a simultaneous learning process. In [135], we can find a recent proposal to learn a linguistic RB and the parametric aggregation connectors of the inference and defuzzification in a single step. Figure 13 presents the coding scheme of the model proposed in this paper.

3.2 Genetic Learning: Rule Coding and Cooperation/Competition Evolutionary Process

Although GAs were not specifically designed for learning, but rather as global search algorithms, they offer a set of advantages for machine learning. Many methodologies for machine learning are based on the search for a good model inside the space of possible models. In this sense, they are very flexible because the same GA can be used with different representations. Genetic learning processes cover

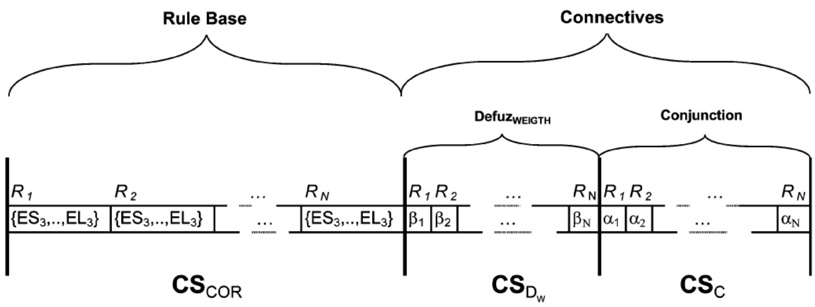


Fig. 13 Example of the coding scheme for learning an RB and the inference connective parameters

different levels of complexity according to the structural changes produced by the algorithm, from the simplest case of parameter optimization to the highest level of complexity for learning the rule set of a rule-based system, via the coding approach and the cooperation or competition between chromosomes.

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, the different genetic learning methods follow two approaches in order to encode rules within a population of individuals:

- The "Chromosome = Set of rules", also called the Pittsburgh approach, in which each individual represents a rule set (Smith 1980). In this case, a chromosome evolves a complete RB and they compete among them along the evolutionary process. GABIL is a proposal that follows this approach [47].
- The "Chromosome = Rule" approach, in which each individual codifies a single rule, and the whole rule set is provided by combining several individuals in a population (rule cooperation) or via different evolutionary runs (rule competition). In turn, within the "Chromosome = Rule" approach, there are three generic proposals:
  - The Michigan approach, in which each individual encodes a single rule. These kinds of systems are usually called learning classifier systems [88]. They are rule-based, message-passing systems that employ reinforcement learning and a GA to learn rules that guide their performance in a given environment. The GA is used for detecting new rules that replace the bad ones via the competition between the chromosomes in the evolutionary process. An interesting study on the topic can be found in [110].
  - The IRL (Iterative Rule Learning) approach, in which each chromosome represents a rule. Chromosomes compete in every GA run, choosing the best rule per run. The global solution is formed by the best rules obtained when the algorithm is run multiple times. SIA [188] is a proposal that follows this approach.
  - The GCCL (genetic cooperative-competitive learning) approach, in which the complete population or a subset of it encodes the RB. In this model the chromosomes compete and cooperate simultaneously. COGIN [67], REGAL [62] and LOGENPRO [200] are examples with this kind of representation.

These four genetic learning approaches (Pittsburgh, Michigan, IRL and GCCL) have been considered for learning KB components, and we can find different examples of them in the literature. Two of the pioneer GFS proposals were focused on the Pittsburgh [180] and Michigan [186] approaches. MOGUL [37, 83, 35] and SLAVE [64] are two proposals that follow the IRL approach in the framework of GFSs. In [93, 97], we find two proposals following the GCCL approach.

3.3 Some GFS Milestones: Books and Special Issues

For beginners, in the following we present the GFS milestones associated to the books and special issues published in the specialized literature.

We can find two authored books and three edited ones:

- A, Geyer-Schulz. Fuzzy Rule-Based Expert Systems and Genetic Machine Learning. Physica-Verlag, 1995. This is the first GFS book. It is a very specific book focused on fuzzy classifier systems (Michigan approach) and RB learning with genetic programming.
- O. Cordón, F. Herrera, F. Hoffmann and L. Magdalena. Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, 2001. This is the first general GFS book. It covers the overall state of the art of GFSs at that time.

These three following books compile an important number of contributions that gave maturity to the topic.

- F. Herrera and J.L. Verdegay (Eds.). Genetic Algorithms and Soft Computing. Physica-Verlag, 1996.
- E. Sanchez, Shibata and L. Zadeh (Eds.). Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives. World Scientific, 1997.
- W. Pedrycz (Ed.). Fuzzy Evolutionary Computation. Kluwer Academic Publishers, 1997.

In the following we provide a list of the journal special issues devoted to GFSs, including important contributions to all topics of GFSs.

- F. Herrera. Special Issue on Genetic Fuzzy Systems for Control and Robotics. International Journal of Approximate Reasoning, Volume 17, Number 4, November 1997.
- F. Herrera and L. Magdalena. Special Issue on Genetic Fuzzy Systems. International Journal of Intelligent Systems, Volume 13, Numbers 10-11, Oct.-Nov. 1998.
- O. Cordón, F. Herrera, F. Hoffmann and L. Magdalena. Special Issue on Recent Advances in Genetic Fuzzy System. Information Sciences, Volume 136, Numbers 1-4 , August 2001.
- O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena. Special Issue on Genetic Fuzzy Systems. Fuzzy Sets and Systems, Volume 141, Number 1, January 2004.
- J. Casillas, M.J. del Jesus, F. Herrera, R. Pérez, P. Villar. Special Issue on Genetic Fuzzy Systems and the Interpretability-Accuracy Trade-off. International Journal of Approximate Reasoning. Volume 44, Number 1, February 2007.
- O. Cordón, R. Alcalá, J. Alcalá-Fdez, I. Rojas. Genetic Fuzzy Systems. Special Section on Genetic Fuzzy Systems: What's Next?. IEEE Transactions on Fuzzy Systems. Volume 15, Number 4, August 2007.
- B. Carse, A.G. Pipe. Special Issue on Genetic Fuzzy Systems. International Journal of Intelligent Systems. Volume 22, Number 9, September 2007.
- J. Casillas, B. Carse. Special Issue on Genetic Fuzzy Systems: Recent Developments and Future Directions. Soft-Computing Volume 13, Number 5, March 2009.

The collection of papers that we could find on these special issues give us a historical tour on the different stages we can find in the evolution of GFSs research:

- The two first special issues (1997, 1998) contain contributions devoted to learning KB components using the different learning approaches (Michigan, IRL, Pittsburgh) together with some applications. We can find representative approaches of different areas of the taxonomy.
- In the next two special issues (2001, 2004) we can find contributions that exploit the mentioned genetic learning approaches together with contributions that stress new branches such as genetic rule selection, multiobjective genetic algorithms for rule selection, the use of genetic programming for learning fuzzy systems, hierarchical genetic fuzzy systems, coevolutionary genetic fuzzy systems, the combination of boosting and evolutionary fuzzy systems learning, embedded genetic DB learning, and first studies for dealing with high dimensional problems, among others. We would like to point out the review paper that was published in the last issue [36] that was the first review in the topic, briefly introducing GFS models and applications, trends and open questions. Another short review was presented in [72]. The present chapter can be considered as a continuation of those, with the novelty of the taxonomy, the GFSs outlook based on the pioneer papers, the ISI Web of Science based visibility and the milestones along the GFSs history and new trends and prospects.
- The next three special issues, published in 2007, emphasize three different directions. Carse and Pipe's special issue collect papers focused in the mentioned areas (multiobjective evolutionary learning, boosting and evolutionary learning, etc) and stress some new ones such as evolutionary adaptive inference systems. Casillas et al.'s special issue is focused on the trade-off between interpretability and accuracy, collecting four papers that proposed different GFSs for tackling this problem. Cordón et al.'s special issue focuses its attention on novel GFS proposals under the title "What's Next?", collecting highly innovative GFS proposals that can mark new research trends. The four collected papers are focused on: a new Michigan approach for learning RBs based on XCS [22], GFSs for imprecisely observed data (low quality data) [162], incremental evolutionary learning of TS-fuzzy systems [86], and evolutionary fuzzy rule induction for subgroup discovery [48].
- The last special issue, co-edited by J. Casillas and B. Carse, is devoted to new developments, paying attention to multiobjective genetic extraction of linguistic fuzzy rule based systems from imprecise data [163], multiobjetive genetic rule selection and tuning [60], parallel distributed genetic fuzzy rule selection [144], context adaptation of fuzzy systems [17], compact fuzzy systems [28], neuro-coevolutionary GFSs [153], evolutionary learning of TSK rules with variable structure [140] and genetic fuzzy association rules extraction [29].

3.4 Current Research Trends in GFSs

In this subsection, from the abundant GFSs literature published, we focus our attention into six current trends that are of high interest at the present and show considerable potential in the near future.

Evolutionary Multiobjective learning of FRBSs: interpretability-precision trade-off

Multiobjective evolutionary algorithms (MOEAs) are one of the most active research areas in the field of evolutionary computation, due to population-based algorithms being capable of capturing a set of non-dominated solutions in a single run of the algorithm. A large number of algorithms have been proposed in the literature [45, 34]. Among them, NSGA-II [46] and SPEA2 [209] are well known and frequently-used MOEAs.

Obtaining high degrees of interpretability and accuracy is a contradictory aim, and, in practice, one of the two properties prevails over the other. Nevertheless, a new tendency in the fuzzy modelling scientific community that looks for a good balance between interpretability and accuracy is increasing in importance. The improvement of the interpretability of rule based systems is a central issue in recent research, where not only the accuracy is receiving attention but also the compacting and the interpretability of the obtained rules [114, 138].

In multiobjective GFSs it is desirable to design genetic learning algorithms in which the learning mechanism itself finds an appropriate balance between interpretability and accuracy. We consider objectives based on accuracy and objectives that include different complexity/interpretability measures. Figure 14 from [91] illustrates this idea where each ellipsoid denotes a fuzzy system. There exists a large number of non-dominated fuzzy systems along the accuracy-complexity trade-off curve.

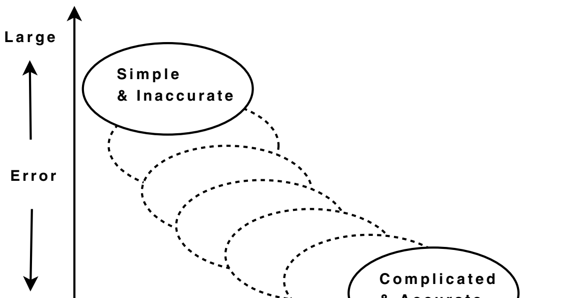


Fig. 14 Non-dominated fuzzy systems

There exists an important number of contributions focused on this topic, in fact, Chapter 5 of this book is devoted to this topic. Therefore, we will not extend our description on the topic.

GA-based techniques for mining fuzzy association rules and novel data mining approaches

Fayyad et al. defined knowledge discovery (KD) as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [57]. KD may not be viewed as synonymous with DM, but they are intimately related. KD is a wide ranging process which covers distinct stages: the comprehension of the problem, the comprehension of the data, pre-processing (or preparation) of the data, DM and post-processing (assessment and interpretation of the models). The DM stage is responsible for automatic KD at a high level, and from information obtained from real data. Some of the important problems that DM and KD deal with are: rule extraction, identification of associations, feature analysis, linguistic summarization, clustering, classifier design and novelty/anomaly detection.

The interpretability of knowledge is crucial in the field of DM/KD where knowledge should be extracted from data bases and represented in a comprehensible form, or for decision support systems where the reasoning process should be transparent to the user. In fact, the use of linguistic variables and linguistic terms in a discovered process has been explored by different authors.

Frequent pattern mining has been a focused theme in DM research for over a decade. Association analysis is a methodology that is useful for the discovery of interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of association rules or sets of frequent items. Abundant literature can be found presenting tremendous progress in the topic [179, 71].

As claimed in [54], the use of fuzzy sets to describe associations between data extends the types of relationships that may be represented, facilitates the interpretation of rules in linguistic terms, and avoids unnatural boundaries in the partitioning of the attribute domains.

Linguistic variables with linguistic terms can contribute in a substantial way to the advance in the design of association rules and the analysis of data to establish relationships and identify patterns, in general [90]. On the other hand, GAs in particular, and EAs in general, are widely used for evolving rule extraction and patterns association in DM/KD [59]. The conjunction in the GFS field provides novel and useful tools for pattern analysis and for extracting new kinds of useful information with a distinct advantage over other techniques: its interpretability in terms of fuzzy IF-THEN rules. We find interesting recent contributions focused on the genetic extraction of fuzzy association rules in [102, 89, 101, 184].

We would like to pay attention to a subdivision of descriptive induction algorithms which has recently received attention from researchers, called subgroup discovery. It is a form of supervised inductive learning of subgroup descriptions in which, given a set of data and having a property of interest to the user, attempts to locate subgroups which are statistically “most interesting” for the user. Subgroup discovery has the objective of discovering interesting properties of subgroups obtaining simple rules (i.e. with an understandable structure and with few variables), highly significant and with high support (i.e. covering many of the instances of the target class). The concept was initially formulated by Klösgen in his rule learning algorithm EXPLORA [108] and by Wrobel in the algorithm MIDOS [201]. Both use a rule-extraction model based on decision trees, in order to obtain the best subgroups among the population. In order to evaluate the subgroups, evaluation measurements are defined which determine the interest of an expression through a combination of unusualness and size. MIDOS tackles, within this same approach, the problem of discovery in multi-relational databases. A recent study on the topic can be found in [118]. In [48] we find a first approach to the use of GFSs for subgroup discovery.

The use of GFSs for association analysis is a topic that would provide interesting future contributions focusing attention on the different research problems that we can find in the frequent pattern mining area [71].

Learning genetic models based on low quality data (noise data and vague data)

There are many practical problems requiring learning models from uncertain data. The experimental designs of GFSs learning from data observed in an imprecise way are not being actively studied by researchers. However, according to the point of view of fuzzy statistics, the primary use of fuzzy sets in classification and modelling problems is for the treatment of vague data. Using vague data to train and test GFSs we could analyze the performance of these classifiers on the type of problems for which fuzzy systems are expected to be superior. Preliminary results in this area involve the proposals of different formalizations for the definition of fuzzy classifiers, based on the relationships between random sets and fuzzy sets [161] and the study of fitness functions (with fuzzy values) defined in the context of GFSs [162].

This is a novel area that is worth being explored in the near future, which may provide interesting results.

Genetic learning of fuzzy partitions and context adaptation

The DB learning comprises the specification of the universes of discourse, the number of labels for each linguistic variable, as well as the definition of the fuzzy membership functions associated with each label. In [39] the influence of fuzzy partition granularity in the FRBS performance was studied. Showing that using an appropriate number of terms for each linguistic variable, the FRBS accuracy can be significantly improved without the need of a complex RB learning method.

On the other hand, the idea of introducing the notion of context into fuzzy systems comes from the observation that, in real life, the same basic concept can be perceived differently in different situations. In some cases, this information is related to the physical properties or dimensions of the system or process, including restrictions imposed due to the measurement acquisition or actuators. In the literature, context adaptation in fuzzy systems has been mainly approached as the scaling of fuzzy sets from one universe of discourse to another by means of non-linear scaling functions whose parameters are identified from data.

Different approaches have been proposed to deal with the learning of membership functions, granularity, non-linear contexts using GAs, etc. [133, 69, 40, 41, 15, 16, 6].

Although there is a large number of contributions in the area of DB Learning, we think that this remains a promising research area, due to the importance of using adequate membership functions and an appropriate context. The use of GFSs has much potential due to its flexibility for encoding DB components together with other fuzzy system components.

Genetic adaptation of inference engine components

We know that it is possible to use parametric aggregation operators in the design of the inference system and the defuzzification method, in an attempt to get the most appropriate parameter configuration in any application. The tuning of these components can be considered to get more accurate fuzzy models. We have come across different GFS approaches for finding the most appropriate parameters [42, 8].

This is an interesting research area that can provide us with the opportunity to adapt the inference parameters to an FRBS and to design learning models that can coevolve the inference engine parameters together with the KB components.

Revisiting the Michigan-style GFSs

The first description of a Michigan-style GFS was given in [186]. All the initial approaches in this area were based on the concept of“rule strength” in the sense that a rule (classifier) gains “strength” during interactions with the environment (through rewards and /or penalties). This strength can then be used for two purposes: resolving conflicts between simultaneously matched rules during learning episodes; and as the basis of fitness for the GAs.

A completely different approach can be considered in which a rule’s fitness, from the point of view of the GA, is based on its “accuracy”, i.e., how well a rule predicts payoff whenever it fires. Notice that the concept of accuracy used here is different from that traditionally used in fuzzy modelling (i.e., the capability of the fuzzy model to faithfully represent the modelled system). This accuracy-based approach offers a number of advantages, such as avoiding overgeneral rules, obtaining optimally general rules, and learning a complete covering map. The first accuracy-based evolutionary algorithm, called XCS, was proposed in [199] and it is currently of major interest to the research community in this field.

Casillas et al. proposed in [22] a new approach to achieve accuracy-based Michigan-style GFSs. The proposal, Fuzzy-XCS, is based on XCS but properly adapted to fuzzy systems, with promising results for function approximation problems and for robot simulation online learning. In [145], an extension of the UCS algorithm is proposed, a recent Michigan-style genetic learning algorithm for classification [14].

These approaches build a bridge between the Michigan-style genetic learning studies and the fuzzy systems models. This is a promising research line that can provide interesting results in the near future.

4 Fuzzy Evolutionary Algorithms

Nowadays, there exists an increasing interest in the use of fuzzy tools and fuzzy logic-based techniques for modelling different EA components or adapting EA control parameters, with the aim of enhancing the performance of these search algorithms [79, 149, 183]. Generally, EAs resulting from this integration are called *fuzzy* EAs.

This section focuses on fuzzy EAs. We give an overview of the existing research on this topic, describing several instances grouped into three categories that were identified after revising specialized literature. The first one involves the adaptation of GA control parameters by means of FRBSs (in particular, fuzzy logic controllers) and, at present, it has a consolidated background of knowledge (Section 4.1). The second one includes those EA models whose components (genetic operators, representation, stop criterion, etc.) are designed using fuzzy tools (Section 4.2). The third one consists of different innovative EA models (particle swarm optimization algorithms, ant colony optimization algorithms, differential evolution, etc.) that make use of fuzzy logic as way to improve their performance (Section 4.3). In addition, we attempt to identify some open issues and summarize a few new promising research directions for fuzzy EAs (Section 4.4).

4.1 Fuzzy Adaptive GAs

Adaptive GAs dynamically adjust their parameters during the course of evolving a solution with the aim of inducing exploitation/exploration relationships that avoid the premature convergence problem and improve the final results [185, 55]. However, the design of this type of GA is very difficult, because the interaction of GA control parameter settings and GA performance is generally acknowledged as a complex relationship which is not completely understood. Although there are ways to understand this relationship (for instance, in terms of stochastic behavior), this kind of understanding does not necessarily result in a normative theory.

Fuzzy logic controllers (FLCs) [53] are a particular model of FRBS (Section 2) that provide a tool which can convert the linguistic control strategy based on expert knowledge into an automatic control strategy. They are particularly suited to model the relationship between variables in environments that are either ill-defined or very complex.

The adaptation of GA parameters is one such complex problem that may benefit from the use of FLCs, producing the so-called fuzzy adaptive GAs (FAGAs) [78, 123]. The rule-bases of FLCs facilitate the capture and representation of a broad range of adaptive strategies for GAs (for example, they may provide the support for the automatic learning of such strategies). The main idea of FAGAs is to use an FLC whose inputs are any combination of GA performance measures or current control parameters and whose outputs are GA control parameters. Current performance measures of the GA are sent to the FLC, which computes new control parameter values that shall be used by the GA. Figure 15 shows this process.

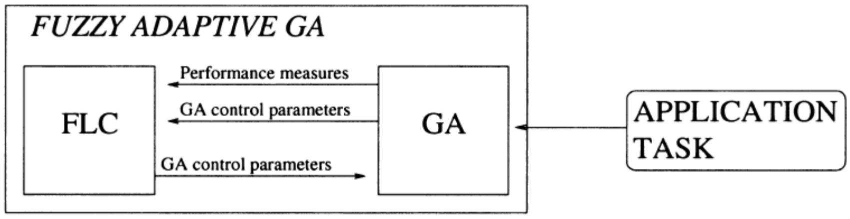


Fig. 15 FAGA model

4.1.1 Designing FAGAs

In this section, we briefly describe the issues that should be tackled in order to build the FLC used by an FAGA. They include the choice of inputs and outputs, the definition of the data base associated with them, and the specification of the rule-base:

Inputs, Outputs, and Data Base

- *Inputs.* They should be robust measures that describe GA behaviour and the effects of the genetic setting parameters and genetic operators. Some possible inputs may be: diversity measures, maximum, average, and minimum fitness, etc. The current control parameters may also be considered as inputs.
- *Outputs.* They indicate values of control parameters or changes in these parameters. In [182], the following outputs were reported: mutation probability, crossover probability, population size, selective pressure, the time the controller must spend in a target state in order to be considered successful, the degree to which a satisfactory solution has been obtained, etc.
- *Data Base.* Each input and output should have an associated set of linguistic labels. The meaning of these labels is specified through membership functions of fuzzy sets, the fuzzy partition, contained in the Data Base. Thus, it is necessary that every input and output have a bounded range of values in order to define these membership functions over it.

Rule-Base

After selecting the inputs and outputs and defining the Data Base, the fuzzy rules describing the relations between them should be defined. They facilitate the capture and representation of a broad range of adaptive strategies for GAs.

Although, the experience and the knowledge of GA experts may be used to derive rule-bases, many authors have found difficulties in doing this. In this sense, the following three reflections were quoted by different authors:

> *"Although much literature on the subject of GA control has appeared, our initial attempts at using this information to manually construct a fuzzy system for genetic control were unfruitful."* [120].

"Statistics and parameters are in part universal to any evolutionary algorithm and in part specific to a particular application. Therefore it is hard to state general fuzzy rules to control the evolutionary process." [182].

"The behaviour of GAs and the interrelations between the genetic operators are very complex. Although there are many possible inputs and outputs for the FLCs, frequently fuzzy rule-bases are not easily available: finding good fuzzy rule bases is not an easy task." [74].

Automatic learning mechanisms to obtain rule-bases have been introduced to avoid this problem. By using these mechanisms, relevant relations and membership functions may be automatically determined and may offer insight to understand the complex interaction between GA control parameters and GA performance [120]. Two types of rule-base learning techniques have been presented: the *offline* learning technique [120, 121] and the *online* learning technique [77]:

- The *offline* learning mechanism is an evolutionary algorithm that is executed once, before the operation of the FAGA, however it has associated with it a high computational cost. It works by considering a fixed set of test functions, following the same idea as the meta-GA of Grefenstette [68]. Unfortunately, the test functions may have nothing to do with the particular problem to be solved, which may limit the robustness of the rule-bases returned.
- In the *online learning* technique, the rule-bases used by the FLCs come from an evolutionary process that interacts concurrently with the GA to be adapted. The learning technique underlying this approach only takes into account the problem to be solved (in contrast to the previous one, which never considers it). In this way, the rule-bases obtained will specify adaptation strategies particularly appropriate for this problem.

4.1.2 A Taxonomy for FAGAs

In this section, we present a taxonomy for FAGAs, focussing on the combination of two aspects:

- The way in which the rule-bases are derived:
  - Through the *expertise, experience*, and *knowledge* of GAs, which have become available as a result of empirical studies conducted over a number of years.
  - Using an *offline learning mechanism*, which finds rule-bases that induce a suitable FAGA behaviour on a fixed set of test functions. It is executed before the application of the FAGA on any real problem.
  - By means of an *online learning mechanism*, which learns rule-bases during the run of the FAGA on a real problem.
- The level where the adaptation takes place in FAGAs:
  - *Population-level* adaptations adjust control parameters that apply for the entire population.

– *Individual-level* adaptations tune control parameters that have an effect on the individual members of the population.

Table 1 outlines the main features of several FAGA instances presented in the literature. It includes the inputs and outputs of the FLCs, the adaptation level, and the method considered to derive the rule-base. A visual inspection of Table 1 allows one to conclude that:

1. The study of FAGAs has been an active line of research in the evolutionary computation community that has produced a significant amount of work during the last fifteen years.
2. Most FAGAs presented in the literature involve population-level adaptation. However, adaptive mechanisms at the individual level based on FLCs may be interesting to adjust control parameters associated with genetic operators [210, 77]. In this case, the control parameters will be defined on individuals instead of on the whole population. Inputs to the FLCs may be central measures and/or measures associated with particular chromosomes or sets of them, and outputs may be control parameters associated with genetic operators that are applied to those chromosomes. A justification for this approach is that it allows for the application of different search strategies in different parts of the search space. This is based on the reasonable assumption that, in general, the search space will not be homogeneous, and that different strategies will be better suited to different kinds of sublandscapes.
3. Most instances use rule-bases derived from GA experts. The use of an online learning mechanism has been less explored, though nowadays it is becoming one of the most prospective alternatives (see Section 4.4.1). An example of is approach was proposed in [77], which was called *coevolution with fuzzy behaviours*. Its main ideas are:

   - It incorporates genetic operator adaptation at an individual-level based on FLCs. Control parameter values for a genetic operator are computed for each set of parents that undergo it, using an FLC that considers particular features associated with the parents as inputs.
   - The rule-bases of the FLCs applied are learnt implicitly throughout the run by means of a separate GA that *coevolves* with the one that applies the genetic operator to be controlled. The goal of this GA is to obtain the rule-bases that produce suitable control parameter values to allow the genetic operator to show an adequate performance on the particular problem to be solved.

   Since the learning technique underlying this approach only takes into account the problem to be solved (in contrast to the approaches based on offline learning mechanisms), the rule-bases obtained shall specify adaptation strategies particularly appropriate for this problem.

Table 1 Instances of FAGAs in the literature

FAGA Instances	Inputs	Outputs	Adaptation Level	Method to Derive Rule-Base
Xu and Vukovich (1993, 1994) [202, 203]	Generation and population size	p_c and p_m	Population-level	GA expert knowledge
Lee and Takagi (1993, 1994) [120, 121]	Two phenotypical diversity measures and change in the best fitness since the last control action	Changes to p_c and p_m, and population size	Population-level	Offline learning
Bergmann, Burgard, and Hemker (1994) [13]	Entropy evolution	Inversion rate, p_c, and p_m	Population-level	GA expert knowledge
Herrera and Lozano (1996) [74]	Genotypical diversity measure and phenotypical diversity measure	Frequency of application of two crossover operators and selection pressure	Population-level	GA expert knowledge
Wang et al (1996) [198]	Change in average fitness of the population at two consecutive generations	Changes to p_c and p_m	Population-level	GA expert knowledge
Zeng and Rabenasolo (1997) [210]	Variance of fitness values, distance between the fitness of the best parent and the best fitness, distance between parents, and normalized fitness values of the parents	p_c for every pair of parents	Individual-level	GA expert knowledge
Song et al. (1996, 1997) [172, 173]	Change in average fitness of the population at two consecutive generations	Changes to p_c and p_m	Population-level	GA expert knowledge
Clintock, Lunney, and Hashim (1997) [31, 32]	Population statistics and diversity statistics	p_c, p_m, and parameter that determines the application of different crossover operators	Population-level	GA expert knowledge
Subbu, Sanderson, and Bonissone (1998) [175]	Genotypic and phenotypic diversity measures of the population	Population size, p_c, and p_m	Population-level	Offline learning
Shi, Eberhart, and Chen (1999) [168]	Best fitness, number of generations for unchanged best fitness, and variance of fitness	p_c and p_m	Population-level	GA expert knowledge
Herrera and Lozano (2000) [76]	Current pm and convergence measure	p_m	Population-level	GA expert knowledge
Matousek, Osmera, and Roupec (2000) [136]	Variability of population, coefficient of partial convergente, and H-characteristics	p_m and selection pressure	Population-level	GA expert knowledge
Wang (2001) [196]	Genetic drift degree, phenotypical diversity measure and number of generations without improving the best individual	p_c and p_m	Population-level	GA expert knowledge
Herrera and Lozano (2001) [77]	Ranks associated with the parents with regards to their fitness values in the population	Control parameter associated with fuzzy recombination	Individual-level	Online learning
Zhu, Zhang, and Jing (2003) [211]	Population size, generation number, and two phenotypic measure for both diversity and convergence	p_c, p_m, and selection pressure	Population-level	Online learning
Yun and Gen (2003) [207]	Changes of average fitness in population of two continuous generations	Changes to p_c and p_m	Population-level	GA expert knowledge
Subbu and Bonissone (2003) [176]	Genotypic diversity and percentage completed trials	Changes to the population size and p_m	Population-level	GA expert knowledge
Ah King, Radha, and Rughooputh [1]	Change in average fitness of the population at two consecutive generations	Changes to p_c and p_m	Population-level	GA expert knowledge
King, Radha, and Rughooputh (2004) [106]	Changes in average fitness at two consecutive steps	Changes to p_c and p_m	Population-level	GA expert knowledge
Last and Eyal (2005, 2006). [115, 116]	Age and lifetime of the chromosomes to be crossed over (parents) and the population average lifetime	p_c	Individual-level	GA expert knowledge
Liu, Xu, and Abraham (2005) [126]	Changes of the best fitness and average fitness in the GA population of two continuous generations	Changes to p_c and p_m	Population-level	GA expert knowledge
Li et al (2006) [122]	Average fitness value of the individuals and standard deviation between two consecutive generations	p_c and p_m	Population-level	GA expert knowledge
Hamzeh, Rahmani, and Parsa (2006) [70]	Measures associated with an XCS learning classifier system	Exploration probability rate	Population-level	GA expert knowledge
Lau, Chan, and Tsui (2007) [117]	Average fitness values in the population and measure of population diversity	Changes to p_c and p_m	Population-level	GA expert knowledge
Sahoo et al (2006, 2007) [159, 160]	Standard deviation of fitness distribution of population and incremental change in average fitness of the population from generation to generation	p_m	Population-level	GA expert knowledge

4.2 EA Components Based on Fuzzy Tools

In this section, we review different EA components built using fuzzy tools that have appeared in the literature.

Fuzzy Genetic Operators

Fuzzy connectives and triangular probability distributions have been considered for designing powerful real-parameter crossover operators that establish adequate population diversity levels and thus help to avoid premature convergence:

- *FCB-crossovers* [82]. These are crossover operators for real-coded GAs based on the use of fuzzy connectives: t-norms, t-conorms and average functions. They were designed to offer different exploration and exploitation degrees.
- *Heuristic FCB-crossovers* [75]. These produce a child each whose components are closer to the corresponding component of its fitter parent.
- *Dynamic FCB-crossovers* [81]. These are crossover operators based on the use of parameterized fuzzy connectives. These operators keep a suitable sequence between the exploration and the exploitation along the GA run: *"to protect the exploration in the initial stages and the exploitation later"*.
- *Dynamic Heuristic FCB-crossovers* [81]. These operators put together the heuristic properties and the features of the Dynamic FCB-crossover operators. They showed very good results as compared with other crossover operators proposed for RCGAs, even better than the FCB-crossover operators and the dynamic ones.
- *Soft Genetic Operators.* In [192, 193, 195], crossover and mutation operators were presented, which are based on the use of triangular probability distributions. These operators, called *soft modal* crossover and mutation, are a generalization of the discrete crossover operator and the BGA mutation, respectively, proposed for the *Breeder* GA [141]. The term *soft* is gleaned from fuzzy set theory only to help grasp the main idea, since probability distributions are considered instead of membership functions.

Fuzzy Representations

Classical EAs, such as GAs and evolution strategies, do not take into account the *development* of an individual or organism from the gene level to the mature phenotype. There are no one-gene, one-trait relationships in natural evolved systems. The phenotype varies as a complex, non-linear function of the interaction between underlying genetic structures and current environmental conditions. Nature follows the universal effects of *pleiotropy* and *polygeny*. Pleiotropy is the fact that a single gene may simultaneously affect several phenotype traits. Polygeny is the effect when a single phenotypic characteristic may be determined by the simultaneous interaction of many genes [58]. An attempt to deal with more complex genotype/phenotype relations in EAs was presented in [191, 194]. A *fuzzy representation* is proposed for the case of tackling optimization problems of parameters with variables on continuous domains. Each problem parameter has associated a number (m) *fuzzy decision* *variables* belonging to the interval $[0,1]$. The chromosomes are formed by linking together the values of the decision variables for each parameter. For each parameter, the decoding process is carried out using a function $g : [0,1]^m \rightarrow [0,1]$, and a linear transformation from the interval [0,1] to the corresponding parameter domain. As an example of such a function the authors presented the following:

$$\forall d = (d_1, \ldots, d_m) \in [0,1]^m, \quad g(d) = \frac{1}{2^{m-1}-1} \sum_{j=1}^{m} d_j 2^{j-1}.$$

When $m > 1$, this coding type breaks the one-to-one correspondence between genotype and phenotype (assumed by classical EAs), since two different genotypes may induce the same phenotype. So, it is impossible to find inferences from phenotype to genotype, i.e., the mapping from genotype to phenotype is not *isomorphic*. Different experiments carried out in [194] with $m = 1$ and $m = 2$ showed that the use of a fuzzy representation allows robust behavior to be obtained. In some cases, a better performance than the Breeder GA was achieved. Furthermore, another important conclusion was stated: for a small population size the performance for $m = 2$ is slightly better than for $m = 1$, whereas the opposite is true for large population sizes.

Sharma and Irwin [167], addressed the use of appropriate fuzzy sets to represent a parameter depending upon its contribution within a problem domain. They proposed a chromosome encoding method, named *fuzzy coding*, for representing real number parameters in a GA. Fuzzy coding is an indirect method for representing a chromosome, where each parameter is represented by two sections. In the first section, the fuzzy sets associated with each parameter are encoded in binary bits with a "1" representing the corresponding set selected. In the second section, each parameter contains degrees of membership corresponding to each fuzzy set. These are encoded as real numbers and represent the degrees of firing. The actual parameter value of interest is obtained through the information contained in the chromosome by means of a defuzzification method. This coding method represents the knowledge associated with each parameter and is an indirect method of encoding compared with the alternatives in which the parameters are directly represented in the encoding. Two test examples, along with neural identification of a nonlinear *pH* (measure of acidity or alkalinity of water) process from experimental data, were studied. It was shown that fuzzy coding is better than the conventional methods (binary, gray, and floating-point coding) and is effective for parameter optimization in problems where the search space is complicated. In addition, the authors claim that this new technique also has the flexibility to embed prior knowledge from the problem domain which is not possible in the regular coding methods. We should point out that an additional investigation was carried out by Pedycz [149] into the exploitation of fuzzy sets as a basis for encoding an original search space.

Finally, in [174], an algorithm for adaptively controlling GA parameter coding using fuzzy rules is presented, which was called fuzzy GAP. This uses an intermediate mapping between the genetic strings and the search space parameters. In particular, each search parameter is specified by the following equation:

$$p_s = (\frac{p_g}{2^l - 1}) \cdot R + O,$$

where p_s is the search parameter, p_g is the genetic parameter, l is the number of bits in the genetic parameter, R is a specified parameter range, and O is a specified offset. By controlling the offset and range, more accurate solutions are obtained using the same number of binary bits.

Fuzzy GAP performs a standard genetic search until the population of strings has converged. Convergence was measured by evaluating the average number of bits which differ between all the genetic strings. Each string is compared to every other string and the number of different bits is counted. If the average number of differing bits per string pair is less than a threshold, the GA has converged. After the genetic strings have converged, a new range and offset for the search parameters are determined by means of an FLC with an input that measures the distance between the centre of the current range and the best solution found in the search. After applying the FLC, the GA is executed again with the new values for the range and offset. The performance of fuzzy GAP on a hydraulic brake emulator parameter identification problem was investigated. It was shown to be more reliable than other dynamic coding algorithms (such as the dynamic parameter encoding algorithm), providing more accurate solutions in fewer generations.

Fuzzy Stopping Criteria

Due to the possibility of premature convergence, GAs do not guarantee that the optimal solution shall be found. Therefore, if the optimal solution is not known, GA performance is difficult to measure accurately. In [137], a fuzzy stopping criterion mechanism (FSCM) is developed to provide a useful evaluation of the GA's real time performance. FSCM is based on achieving a user-defined level of performance for the given problem. In order to do so, it includes a predicting process based on statistics for estimating the value of the GA optimal solution, then it compares the current solution to this optimal one by checking if an acceptable percentage (specified by the user) of the latter is reached. If so, the GA stops and returns belief and uncertainty measures that provide reliability measure for the GA chosen solution. The acceptable percentage optimal solution defined by the user represents a fuzzy stopping criterion for halting GA if an appropriate solution is reached. The predicting process is invoked every 40 iterations and uses performance values such as the minimum solution value, average solution value and belief and plausibility values, all obtained during these iterations. The underlying idea for the FSCM is that the user does not need to find the global solution, but rather an approximate solution that is close to the optimal one, i.e., the GA is used for solving a *fuzzy goal* instead of a crisp one because of the vagueness of the term approximate. This term is quantitatively measured by the user through the acceptable percentage of the optimal solution that he requires in the final solution. Results obtained on a 25-city TSP problem indicate this approach is preferable to a simple GA, in term of cost/performance and in decreasing the amount of time the GA searches for acceptable solutions.

4.3 Other Fuzzy EA Models

Different fuzzy logic tools have been employed to improve the behavior of other EA models, such as EAs for multiobjective problems, parallel EAs, genetic programming, differential evolution, particle swarm optimization algorithms, ant colony optimization algorithms, and cultural algorithms. Next, we briefly explain the way these EA approaches benefited from fuzzy logic.

Fuzzy EAs for Multiobjective Optimization Problems

In [189], a FAGA is presented for multiobjective optimization problems. In each generation, an FLC decides what transformation of the cost components into a one-dimensional fitness function is taken. In this vein, [152], Rachmawati and Srinivasan present an algorithm that employs a fuzzy inference system to model and aggregate different objectives. They are represented as fuzzy variables, which act as inputs to a fuzzy inference system evaluating the fitness of the associated candidate solution. The fuzzy system captures preferences of the decision maker in the compromise between various objectives, thereby guiding the search to interesting regions in the objective space. In [190], a more complex method, called a *fuzzy reduction* GA, is proposed. It attempts to enable a uniform approximation of the Pareto optimal solutions (those that cannot be improved with respect to any cost function without making the value of some other worse). The authors started by explicitly formulating desirable goals for the evolution of the population towards the target Pareto optimal solutions (which could be expressed in vague terms only). Then, they defined deviation measures for a population from these goals, which were the inputs to an FLC. Later, they fixed a set of possible actions that could serve as countermeasures to decrease the deviations. These actions are different selection mechanisms based on classical ones proposed to tackle multiobjective optimisation problems. The FLC determines activation rates for the actions. The action that should actually be taken is decided according to the activation rates found. As an application, a timetable optimisation problem is presented where the method was used to derive cost-benefit curves for the investment into railway nets. The results showed that the fuzzy adaptive approach avoids most of the empirical shortcomings of other multiobjective GAs by the adaptive nature of the procedure. Other models of multiobjective GA based on the fuzzy logic tools are found in [44, 52, 98, 119].

Fuzzy Parallel EAs

The availability, over the last few years, of fast and cheap parallel hardware has favoured research into possible ways for implementing parallel versions of EAs [20]. EAs are good candidates for effective parallelization, since they are inspired on the principles of parallel evolution, for a population of individuals. Among the many types of parallel EAs, *distributed* and *cellular* algorithms are two popular optimization tools. The basic idea of the distributed EAs lies in the partition of the population into several subpopulations, each one of them being processed by an EA, independently from the others. Furthermore, a migration process produces a chromosome exchange between the subpopulations. An important control parameter that determines the operation of this process is the migration rate, which controls how many chromosomes migrate. Maeda et al. [131, 132] propose an adaptive search method for distributed EAs. Its main characteristic feature is the fuzzy adaptive control of the migration rate by evaluating the evolutionary degree for each subpopulation. Simulations were performed to confirm the efficiency of this method, which was shown to be superior to both ordinary and parallel EAs. In a cellular EA, the concept of (small) neighbourhood is intensively used; this means that an individual may only interact with its nearby neighbours in the breeding loop [3]. In [156], the fuzzy adaptive mechanism proposed by Shi et al. [168] was considered to adapt parameters associated with cellular EAs, obtaining a fuzzy cellular EA model.

Fuzzy Genetic Programming

Genetic Programming's [111] basic distinction from GAs is the evolution of dynamic tree structures, often interpreted as programs, rather than fixed-length vectors. In [10], it is claimed that genetic programming requires human supervision during their routine use as practical tools for the following reasons: 1) to detect the emergence of a solution, 2) to tune algorithm parameters and 3) to monitor the evolution process in order to avoid undesirable behaviour such as premature convergence. It is also advised that any attempt to develop artificial intelligence tools based on genetic programming should take these issues into account. The authors proposed FLCs for this task. They called the collection of fuzzy rules and routines in charge of controlling the evolution of the GA population "fuzzy government". Fuzzy government was applied to the symbolic inference of the formulae problem. Genetic programming was used to solve the problem along with different FLCs, which dynamically adjusted the maximum length for genotypes, acted on the mutation probability, detected the emergence of a solution, and stopped the process. The results showed that the performance of the fuzzy governed GA was almost impossible to distinguish from the performance of the same algorithm operated directly with human supervision. Other work on fuzzy adaptive search methods for genetic programming is [130].

Fuzzy Cultural Algorithms

Cultural algorithms (CAs) [154] are dual inheritance systems that consist of a social population and a belief space. The problem solving experience of individuals selected from the population space by an acceptance function is used to generate problem solving knowledge that resides in the belief space. This knowledge can be viewed as a set of beacons that can control the evolution of the population component by means of an influence function. The influence function can use the knowledge in the belief space to modify any aspect of the population component. Various evolutionary models have been used for the population component of CAs, including GAs, genetic programming, evolution strategies, and evolutionary programming. In [155], a fuzzy approach to CAs is presented in which an FLC regulates the amount of information to be transferred to the belief space used by the CA over time. In particular, the FLC determines the number of individuals which shall impact the current beliefs. Its inputs are the individual success ratio (ratio of the number of successes to the total number of mutations) and the current generation. A comparison was made between the fuzzy version of a CA (that used evolutionary programming as the population component) and its non fuzzy version on 34 optimization functions. The conclusions were: 1) the fuzzy interface between the population and belief space outperformed the non fuzzy version in general, and 2) the use of a fuzzy acceptance function significantly improved the success ratio and reduced CPU time.

Fuzzy Ant System

Ant Colony Optimization (ACO) [51] is a population-based metaheuristic approach for solving hard combinatorial optimization problems. The inspiring source of ACO is the foraging behavior of real ants which enables them to find shortest paths between a food source and their nest. They are based on a colony of artificial ants, that is, simple computational agents that work cooperatively and communicate through artificial pheromone trails. ACO algorithms are essentially construction algorithms: every ant constructs a solution to the problem by travelling on a construction graph. The edges of the graph, representing the possible steps the ant can make, have two kinds of associated information (heuristic information and artificial pheromone trail information), which are used to define transition probabilities of moving from one node to other, guiding ant movement. This information is modified during the algorithm run, depending on the solutions found by the ants. In [181], a fuzzy ACO approach is presented, which uses fuzzy logic to calculate an ant's utility to visit the next node. In particular, transition probabilities (usually given in a classical ACO in closed form) are computed by a fuzzy rule-based system. Their authors claim that when using fuzzy logic as a separate module within the ACO, it is possible to handle the uncertainty that sometimes exists in some complex combinatorial optimization problems. The control strategies of an ant can also be formulated in terms of descriptive fuzzy rules. Other ACO models based on fuzzy logic are presented in [104, 142].

Fuzzy Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm [103] is inspired by social behaviour patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviours observed in flocks of birds, schools of fish, or swarms of bees, and even human social behaviour. The standard PSO model consists of a swarm of particles, which are initialized with a population of random candidate solutions. Each particle has a position represented by a position-vector, and a velocity represented by a velocity vector. The particles move iteratively through the d-dimension problem space to search new solutions, where the fitness can be calculated as a quality measure. A particle decides where to move next, considering its own experience, which is the memory of its best past position, and the experience of the most successful particle in the swarm. It has been shown that the trajectories of the particles oscillate in different sinusoidal waves and converge quickly, sometimes prematurely. Liu and Abraham [124] proposed an adaptive mechanism based on FLCs to control the velocity of particles in order to avoid premature convergence in PSO. Empirical results demonstrated that the performance of standard PSO degrades remarkably with the increase in the dimension of the problem, while the influence is very little in the fuzzy PSO approach. Another instance of a PSO model tuned by FLCs may be found in [99]. Finally, we should point out that a fuzzy version of PSO specifically designed to tackle the quadratic assignment problem was presented in [125].

Fuzzy Differential Evolution

The differential evolution algorithm (DE) is one of the most recent EAs for solving real-parameter optimization problems [151]. Like other EAs, DE is a population-based, stochastic global optimizer capable of working reliably in nonlinear and multimodal environments. DE has few control parameters. However, choosing the best parameter setting for a particular problem is not easy [129]. Liu and Lampinen [127, 128, 129] present the fuzzy adaptive differential evolution algorithm, which uses FLCs controllers to adapt the search parameters for the DE mutation operation and crossover operation. These two parameters were adapted individually for each generation. Parameter vector change and function value change over the whole population members between the last two generations were nonlinearly depressed and then used as the inputs for both FLCs. Experimental results, provided by the proposed algorithm for a set of standard test functions, outperformed those of the standard differential evolution algorithm for optimization problems with higher dimensionality.

4.4 Future Work on Fuzzy EAs

Despite the recent activity and the associated progress in fuzzy EAs research, there remain many directions in which the work may be improved or extended. Next, we report on some of these.

4.4.1 Improvements for FAGAs

Future research may take into account the following issues in order to produce effective FAGAs.

Relevant Inputs for the FLCs

Research on determining relevant input variables for the FLCs controlling GA behaviour should be studied in greater depth. These variables should describe either states of the population or features of the chromosomes, so that control parameters may be adapted on the basis thereof to introduce real performance improvements. In this vein, Boulif and Karim [18] claimed recently that previous researches on

FAGAs allowed building FAGA systems that outperform significantly conventional GAs. However, these works, albeit interesting, do not consider the causes that propel the GA in its search for good solutions but rather their effects. Indeed, the fuzzy models use as input either convergence speed, the population diversity or its average fitness (see Table 1). These authors think that it will be more interesting to deal with *cause* inputs first, admitting that nothing forbids complementing them by *effect* inputs. This perspective may prove useful for detecting relevant inputs and determining how to exploit them to adequately tune GA parameters.

Adaptation by Coevolution with Fuzzy Behaviors

The adaptation of GA parameters by coevolution with fuzzy behaviours (FBs) becomes a prospective way for future FAGA works, mainly for two reasons: 1) The use of online learning techniques to derive rule-bases for FAGA has been little explored (see Table 1) and 2) Adaptation of EAs by means of the coevolutionary model is, nowadays, a topic of high interest [171].

Different types of parameter settings were associated with genetic operators, which could be adapted by means of coevolution with FBs. These include the following:

- *Operator probabilities*. There is a type of GA that does not apply both crossover and mutation to the selected solutions. Instead, a set of operators is available, each with a probability of being used, and only one of these is selected to produce offspring. Many adaptive GAs have been designed starting from this GA approach, which adjust the operator probabilities throughout the run [185].
- *Operator parameters*. These parameters determine the way in which genetic operators work. Examples include: 1) the step size of mutation operators for real-coded GAs, which determines the strength in which real genes are mutated, 2) parameters associated with crossover operators for real-coded GAs (see [84]) and dynamic FCB-crossovers [81], 3) the number of parents involved in multi-parent recombination operators, and 4) parameters associated with crossover operators for binary-coded GAs, such as *n*-point crossover and uniform crossover.

  The adaptation at individual-level of operator probabilities and operator parameters by coevolution with FBs may be carried out by considering these variables as a consequence of the fuzzy rules represented in the FBs. Furthermore, appropriate features of the parents should be chosen, as a basis oo which the adjustment of these variable is expressed. On the other hand, hybrid models may be built, in such a way that FBs include information for both the adaptation of operator probabilities and operator parameters. In this case, the model shall detect the operators that should be applied more frequently, along with favourable operator parameter values for them.
- *Mate selection parameters*. In mate selection mechanisms [158], chromosomes carry out the choice of mates for crossover on the basis of their own preferences (which are formulated in terms of different chromosome characteristics, such as the phenotypical distance between individuals).

> Mate selection strategies may be expressed by means of FBs. In particular, given two chromosomes, an FB may induce a probability of mating depending on their characteristics. This probability determines whether or not they are crossed. Then, the process of coevolution with FBs shall discover FBs containing mate selection strategies that encourage recombination between chromosomes that have useful information (characteristics) to exchange.

The adaptive mechanism by coevolution with FBs may also be used for problems where we intuit that particular features of the parents may be taken into account to allow the crossover operator behaviour to be more effective, but we do not know the precise fuzzy rules determining the relation between these features and the appropriate control actions for the operator. In this fashion, this approach allows particular knowledge about the problem to be integrated in the EA in order to improve its behaviour.

4.4.2 Applications and Extensions of Fuzzy EAs

Fuzzy EAs may be defined to tackle particular problems such as multimodal optimisation problems. In addition, fuzzy logic may help modern hybrid metaheuristics to improve their behaviour, obtaining fuzzy hybrid metaheuristics.

Multimodal Optimisation Problems

Given a problem with multiple solutions, a simple EA will tend to converge to a single solution. As a result, various mechanisms have been proposed to stably maintain a diverse population throughout the search, thereby allowing EAs to identify multiple optima reliably. Many of these methods work by encouraging artificial niche formation through sharing and crowding [169], but these methods introduce one or more parameters that affect algorithm performance, parameters such as the sharing radius in fitness sharing or the crowding factor in crowding. In many problems, the uniform specification of niche size is inadequate to capture solutions of varying location and extent without also increasing the population size beyond reasonable bounds. Therefore, there remains a need to develop niching methods that stably and economically find the best niches, regardless of their spacing and extent. FLCs may be useful for the adaptation of parameters associated with sharing and crowding methods. Possible inputs may be: diversity measures, the number of niches that are currently in the population, etc.

Application of Fuzzy Tools to Improve Hybrid Metaheuristics

Over the last years, a large number of search algorithms were reported that do not purely follow the concepts of one single classical metaheuristic, but they attempt to obtain the best from a set of metaheuristics (and even other kinds of optimization methods) that perform together and complement each other to produce a profitable synergy from their combination. These approaches are commonly referred to as hybrid metaheuristics [178]. Memetic algorithms (MAs) [112] are well-known instances of this class of algorithms. They combine an EA in charge of the global search with a local search (LS) procedure, which is executed within the EA run, looking for a synergy that takes benefits from both. The classic scheme of MAs applies the local search procedure on the solutions obtained by the EA with the aim of improving the accuracy of the population members. An important aspect in MAs is the number of fitness function evaluations required by the LS algorithm during their operation (LS intensity). It is fundamental to identify a proper intensity for the LS, because a LS that is too short may be unsuccessful at exploring the neighbourhood of the solution and therefore unsuccessful at improving the search quality. On the other hand, too long LS may backfire by consuming additional fitness evaluations unnecessarily.

A great part of the experience acquired about the application of fuzzy logic to improve EAs may be reused to enhance the behaviour of these innovative search algorithms. For example, FLCs may be designed with the aim of coordinating the different components in a hybrid metaheuristic, assigning different fitness function evaluations to them depending on their specific exploration and/or exploitation features. In particular, for the case of MAs, the adaptation of the LS intensity by FLCs becomes a prospective line of research for obtaining effective MAs.

5 Concluding Remarks

In this chapter, we painted a complete picture of GFSs and fuzzy EAs. In particular, we overviewed important design principles for these algorithms, cited existing literature whenever relevant, provided a taxonomy for each one of them, and discussed future directions and some challenges for these two lines of research. Mainly, this work reveals that GFSs and fuzzy EAs have consolidated backgrounds of knowledge, and therefore, they are two outstanding examples of positive collaboration between soft computing technologies. In addition, it shows that there still remain many exciting research issues connected with these two topics.

References

1. Ah King, R.T.F., Radha, B., Rughooputh, H.C.S.: A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In: Proc of the 2004 IEEE International Conference on Networking, Sensing and Control, pp. 577–582 (2004)
2. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation 6, 443–462 (2002)
3. Alba, E., Dorronsoro, B.: The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Transactions on Evolutionary Computation 9(2), 126–142 (2005)
4. Alcalá, R., Casillas, J., Cordón, O., Herrera, F.: Building fuzzy graphs: features and taxonomy of learning non-grid-oriented fuzzy rule-based systems. International Journal of Intelligent Fuzzy Systems 11, 99–119 (2001)

5. Alcalá, R., Alcalá-Fdez, J., Herrera, F.: A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. IEEE Transactions on Fuzzy Systems 15(4), 616–635 (2007)
6. Alcalá, R., Alcalá-Fdez, R., Herrera, F., Otero, J.: Genetic learning of accurate and compact fuzzy rule based systems based on the 2-Tuples linguistic representation. International Journal of Approximate Reasoning 44, 45–64 (2007)
7. Alcalá, R., Gacto, M.J., Herrera, F., Alcalá-Fdez, J.: A multi-objective genetic algorithm for tuning and rule selection to obtain accurate and compact linguistic fuzzy rule-based systems. International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems 15(5), 521–537 (2007)
8. Alcalá-Fdez, J., Herrera, F., Marquez, F., Peregrin, A.: Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems. International Journal of Intelligent Systems 22(9), 1035–1064 (2007)
9. Alcalá-Fdez, J., Sánchez, L., García, S., del Jesús, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F.: KEEL: A software tool to assess evolutionary algorithms for data mining problems. In: Soft Computing (in press)
10. Arnone, S., Dell'Orto, M., Tettamanzi, A.: Toward a fuzzy government of genetic populations. In: Proc. of the 6th IEEE Conference on Tools with Artificial Intelligence, pp. 585–591. IEEE Computer Society Press, Los Alamitos (1994)
11. Au, W.-H., Chan, K.C.C., Wong, A.K.C.: A fuzzy approach to partitioning continuous attributes for classification. IEEE Transactions on Knowledge and Data Engineering 18(5), 715–719 (2006)
12. Berlanga, F.J., del Jesus, M.J., González, P., Herrera, F., Mesonero, M.: Multiobjective evolutionary induction of subgroup discovery fuzzy rules: A case study in marketing. In: Perner, P. (ed.) ICDM 2006. LNCS, vol. 4065, pp. 337–349. Springer, Heidelberg (2006)
13. Bergmann, A., Burgard, W., Hemker, A.: Adjusting parameters of genetic algorithms by fuzzy control rules. In: Becks, K.-H., Perret-Gallix, D. (eds.) New Computing Techniques in Physics Research III, pp. 235–240. World Scientific Press, Singapore (1994)
14. Bernadó-Mansilla, E., Garrell-Guiu, J.M.: Accuracy-based learning classifier systems: models, analysis and applications to classification tasks. Evolutionary Computation 11(3), 209–238 (2003)
15. Botta, A., Lazzerini, B., Marcelloni, F.: Context adaptation of Mamdani fuzzy systems through new operators tuned by a genetic algorithm. In: Proceedings of the 2006 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2006), Vancouver, Canada, pp. 7832–7839 (2006)
16. Botta, A., Lazzerini, B., Marcelloni, F., Stefanescu, D.C.: Exploiting fuzzy ordering relations to preserve interpretability in context adaptation of fuzzy systems. In: Proceedings of the 2007 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), London, UK, pp. 1137–1142 (2007)
17. Botta, A., Lazzerini, B., Marcelloni, F., Stefanescu, D.C.: Context Adaptation of Fuzzy Systems Through a Multi-objective Evolutionary Approach Based on a Novel Interpretability Index. Soft Computing 13(3), 437–449 (2009)
18. Boulif, M., Atif, K.: A new fuzzy genetic algorithm for the dynamic bi-objective cell formation problem considering passive and active strategies. International Journal of Approximate Reasoning 47, 141–165 (2008)
19. Cano, J.R., Herrera, F., Lozano, M.: Evolutionary stratified training set selection for extracting classification rules with trade-off precision-interpretability. Data and Knowledge Engineering 60, 90–108 (2007)

20. Cantú-Paz, E.: Efficient and accurate parallel genetic algorithms. Book Series on Genetic Algorithms and Evolutionary Computation. Kluwer, Norwell (2000)
21. Carse, B., Fogarty, T.C., Munro, A.: Evolving fuzzy rule based controllers using genetic algorithms. Fuzzy Sets and Systems 80(3), 273–293 (1996)
22. Casillas, J., Carse, B., Bull, L.: Fuzzy-XCS: A Michigan genetic fuzzy system. IEEE Transactions on Fuzzy Systems 15(4), 536–550 (2007)
23. Casillas, J., Cordón, O., Herrera, F., del Jesus, M.J.: Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems. Information Sciences 136(1-4), 135–157 (2001)
24. Casillas, J., Cordón, O., del Jesus, M.J., Herrera, F.: Genetic tuning of fuzzy rule deep structures preserving interpretability for linguistic modeling. IEEE Trans. on Fuzzy Systems 13(1), 13–29 (2005)
25. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Accuracy improvements in linguistic fuzzy modeling. Springer, Berlin (2003)
26. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Interpretability issues in fuzzy modeling. Springer, Berlin (2003)
27. Casillas, J., Martínez, P.: Consistent, complete and compact generation of DNF-type fuzzy rules by a Pittsburgh-style genetic algorithm. In: Proceedings of the 2007 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), London, UK, pp. 1745–1750 (2007)
28. Casillas, J., Martínez, P., Benítez, A.D.: Learning consistent, complete and compact fuzzy rules sets in conjunctive normal form for system identification. Soft Computing 13(3), 451–465 (2009)
29. Chen, C.-H., Hong, T.-P., Tseng, V.S., Lee, C.-S.: A genetic-fuzzy mining approach for items with multiple minimum supports. Soft Computing 13(3), 521–533 (2009)
30. Cherkassky, V., Mulier, F.: Learning from data: concepts, theory and methods. John Wiley & Sons, New York (1998)
31. Mc Clintock, S., Lunney, T., Hashim, A.: Using fuzzy logic to optimize genetic algorithm performance. In: Proceedings of 1997 IEEE International Conference on Intelligent Engineering Systems, Budapest, Hungary, pp. 271–275 (1997)
32. Mc Clintock, S., Lunney, T., Hashim, A.: A fuzzy logic controlled genetic algorithm environment. In: Proceedings of 1997 IEEE International Conference on Systems, Man, and Cybernetics, Orlando, Florida, USA, pp. 2181–2186 (1997)
33. Cococcioni, M., Ducange, P., Lazzerini, B., Marcelloni, F.: A Pareto-based multi-objective evolutionary approach to the identification of Mamdani fuzzy systems. Soft Computing 11(11), 1013–1031 (2007)
34. Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers, Dordrecht (2002)
35. Cordón, O., del Jesús, M.J., Herrera, F., Lozano, M.: MOGUL: A Methodology to Obtain Genetic fuzzy rule-based systems Under the iterative rule Learning approach. International Journal of Intelligent Systems 14, 1123–1153 (1999)
36. Cordón, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets and Systems 141, 5–31 (2004)
37. Cordón, O., Herrera, F.: A three-stage evolutionary process for learning descriptive and approximate fuzzy-logic-controller knowledge bases from examples. International Journal of Approximate Reasoning 17(4), 369–407 (1997)
38. Cordón, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic fuzzy systems. In: Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific, Singapore (2001)

39. Cordón, O., Herrera, F., Villar, P.: Analysis and guidelines to obtain a good fuzzy partition granularity for fuzzy rule-based systems using simulated annealing. International Journal of Approximate Reasoning 25(3), 187–215 (2000)
40. Cordón, O., Herrera, F., Magdalena, L., Villar, P.: A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. Information Sciences 136, 85–107 (2001)
41. Cordón, O., Herrera, F., Villar, P.: Generating the knowledge base of a fuzzy rule-based system by the genetic learning of data base. IEEE Transactions on Fuzzy Systems 9(4), 667–674 (2001)
42. Crockett, K.A., Bandar, Z., Fowdar, J., O'Shea, J.: Genetic tuning of fuzzy inference within fuzzy classifier systems. Expert Systems with Applications 23, 63–82 (2006)
43. Crockett, K., Bandar, Z., Mclean, D.: On the optimization of T-norm parameters within fuzzy decision trees. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), London, UK, pp. 103–108 (2007)
44. Das, D.: Optimal placement of capacitors in radial distribution system using a Fuzzy-GA method. International Journal of Electrical Power & Energy Systems (in press)
45. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, Chichester (2001)
46. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6(2), 182–197 (2002)
47. De Jong, K.A., Spears, W.M., Gordon, D.F.: Using genetic algorithms for concept learning. Machine Learning 13, 161–188 (1993)
48. del Jesus, M.J., González, P., Herrera, F., Mesonero, M.: Evolutionary fuzzy rule induction process for subgroup discovery: A case study in marketing. IEEE Transactions on Fuzzy Systems 15(4), 578–592 (2007)
49. Demsar, J.: Statistical comparison of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
50. Diettereich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1924 (1998)
51. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press, Cambridge (2004)
52. Dozier, G.V., McCullough, S., Homaifar, A., Moore, L.: Multiobjective evolutionary path planning via fuzzy tournament selection. In: IEEE International Conference on Evolutionary Computation (ICEC 1998), pp. 684–689. IEEE Press, Piscataway (1998)
53. Driankow, D., Hellendoorn, H., Reinfrank, M.: An introduction to fuzzy control. Springer, Berlin (1993)
54. Dubois, D., Prade, H., Sudkamp, T.: On the representation, measurement, and discovery of fuzzy associations. IEEE Trans. on Fuzzy Systems 13, 250–262 (2005)
55. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. Evolutionary Computation 3(2), 124–141 (1999)
56. Eiben, A.E., Smith, J.E.: Introduction to evolutionary computation. Springer, Berlin (2003)
57. Fayyad, U., Piatesky-Shapiro, G., Smyth, P.: From data mining from knowledge discovery in databases. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery & Data Mining, pp. 1–34. AAAI/MIT (1996)
58. Fogel, D.B.: An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks 5(1), 3–14 (1994)
59. Freitas, A.A.: Data mining and knowledge discovery with evolutionary algorithms. Springer, Berlin (2002)
60. Gacto, M.J., Alcalá, R., Herrera, F.: Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. Soft Computing 13(3), 419–436 (2009)

61. Geyer-Schulz, A.: Fuzzy rule-based expert systems and genetic machine learning. Physica-Verlag, Berlin (1995)
62. Giordana, A., Neri, F.: Search-intensive concept induction. Evolutionary Computation 3, 375–416 (1995)
63. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1989)
64. González, A., Pérez, R.: SLAVE: A genetic learning system based on an iterative approach. IEEE Transactions on Fuzzy Systems 27, 176–191 (1999)
65. González, A., Pérez, R.: Selection of relevant features in a fuzzy genetic learning algorithm. IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics 31(3), 417–425 (2001)
66. González, A., Pérez, R.: An analysis of the scalability of an embedded feature selection model for classification problems. In: Proc. Eleventh Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU 2006), Paris, pp. 1949–1956 (2006)
67. Greene, D.P., Smith, S.F.: Competition-based induction of decision models from examples. Machine Learning 3, 229–257 (1993)
68. Grefenstette, J.J.: Optimization of control parameters for genetic algorithms. IEEE Trans Systems, Man, and Cybernetics 16, 122–128 (1986)
69. Gudwin, R.R., Gomide, F.A.C., Pedrycz, W.: Context adaptation in fuzzy processing and genetic algorithms. International Journal of Intelligent Systems 13(10-11), 929–948 (1998)
70. Hamzeh, A., Rahmani, A., Parsa, N.: Intelligent exploration method to adapt exploration rate in XCS, based on adaptive fuzzy genetic algorithm. In: Proc. of the 2006 IEEE Conference on Cybernetics and Intelligent Systems, pp. 1–6 (2006)
71. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Mining & Knowledge Discovery 15(1), 55–86 (2007)
72. Herrera, F.: Genetic fuzzy systems: Status, critical considerations and future directions. International Journal of Computational Intelligence Research 1(1), 59–67 (2005)
73. Herrera, F.: Genetic fuzzy systems: taxonomy, current research trends and prospects. Evolutionary Intelligence 1, 27–46 (2008)
74. Herrera, F., Lozano, M.: Adaptation of genetic algorithm parameters based on fuzzy logic controllers. In: Herrera, F., Verdegay, J.L. (eds.) Genetic Algorithms and Soft Computing, pp. 95–125. Physica-Verlag (1996)
75. Herrera, F., Lozano, M.: Heuristic crossover for real-coded genetic algorithms based on fuzzy connectives. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 336–345. Springer, Heidelberg (1996)
76. Herrera, F., Lozano, M.: Adaptive control of the mutation probability by fuzzy logic controllers. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 335–344. Springer, Heidelberg (2000)
77. Herrera, F., Lozano, M.: Adaptive genetic operators based on coevolution with fuzzy behaviours. IEEE Trans. on Evolut. Comput. 5(2), 1–18 (2001)
78. Herrera, F., Lozano, M.: Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. Soft Computing 7, 545–562 (2003)
79. Herrera, F., Lozano, M., Verdegay, J.L.: Tackling fuzzy genetic algorithms. In: Genetic Algorithms in Engineering and Computer Science, pp. 167–189. John Wiley, New York (1995)
80. Herrera, F., Lozano, M., Verdegay, J.L.: Tuning fuzzy-logic controllers by genetic algorithms. International Journal of Approximate Reasoning 12(3-4), 299–315 (1995)

81. Herrera, F., Lozano, M., Verdegay, J.L.: Dynamic and heuristic fuzzy connectives-based crossover operators for controlling the diversity and convengence of real-coded genetic algorithms. Int. Journal of Intelligent Systems 11, 1013–1041 (1996)
82. Herrera, F., Lozano, M., Verdegay, J.L.: Fuzzy connectives based crossover operators to model genetic algorithms population diversity. Fuzzy Sets and Systems 92(1), 21–30 (1997)
83. Herrera, F., Lozano, M., Verdegay, J.L.: A learning process for fuzzy control rules using genetic algorithms. Fuzzy Sets and Systems 100, 143–151 (1998)
84. Herrera, F., Lozano, M., Sánchez, A.M.: A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study. International Journal of Intelligent Systems 18, 309–338 (2003)
85. Homaifar, A., Mccormick, E.: Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Transactions on Fuzzy Systems 3(2), 129–139 (1995)
86. Hoffmann, F., Schauten, D., Hölemann, S.: Incremental evolutionary design of TSK fuzzy controllers. IEEE Transactions on Fuzzy Systems 15(4), 563–577 (2007)
87. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
88. Holland, J.H., Reitman, J.S.: Cognitive systems based on adaptive algorithms. In: Waterman, D.A., Hayes-Roth, F. (eds.) Patter-Directed Inference Systems. Academic Press, London (1978)
89. Hong, T.P., Chen, C.H., Wu, Y.L., et al.: A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. Soft Computing 10(11), 1091–1101 (2006)
90. Hüllermeier, E.: Fuzzy methods in machine learning and data mining: Status and prospects. Fuzzy Sets and Systems 156(3), 387–406 (2005)
91. Ishibuchi, H.: Multiobjective genetic fuzzy systems: review and future research directions. In: Proceedings of the 2007 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), London, UK, pp. 913–918 (2007)
92. Ishibuchi, H., Murata, T., Turksen, I.B.: Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. Fuzzy Sets and Systems 8(2), 135–150 (1997)
93. Ishibuchi, H., Nakashima, T., Murata, T.: Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. IEEE Transactions on Systems, Man and Cybernetics. Part B-Cybernetics 29(5), 601–618 (1999)
94. Ishibuchi, H., Nakashima, T., Nii, M.: Classification and modeling with linguistic information granules: Advanced approaches to linguistic data mining. Springer, Berlin (2004)
95. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selection fuzzy IF-THEN rules for classification problems using genetic algorithms. IEEE Transactions on Fuzzy Systems 3(3), 260–270 (1995)
96. Ishibuchi, H., Yamamoto, T.: Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy Sets and Systems 141(1), 59–88 (2004)
97. Juang, C.F., Lin, J.Y., Lin, C.T.: Genetic reinforcement learning through symbiotic evolution for fuzzy controller design. IEEE Transactions on Systems, Man and Cybernetics. Part B-Cybernetics 30(2), 290–302 (2000)
98. Kacem, I., Hammadi, S., Borne, P.: Pareto-optimality approach based on uniform design and fuzzy evolutionary algorithms for flexible job-shop scheduling problems (FJSPs). In: 2002 IEEE International Conference on Systems, Man and Cybernetics, p. 7 (2002)

99. Kang, Q., Wang, L., Wu, Q.: Research on fuzzy adaptive optimization strategy of particle swarm algorithm. International Journal of Information Technology 12(3), 65–77 (2006)
100. Karr, C.: Genetic algorithms for fuzzy controllers. AI Expert 6(2), 26–33 (1991)
101. Kaya, M.: Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. Soft Computing 10(7), 578–586 (2006)
102. Kaya, M., Alhajj, R.: Genetic algorithm based framework for mining fuzzy association rules. Fuzzy Sets and Systems 152(3), 587–601 (2005)
103. Kennedy, J., Eberhart, R.C.: Swarm intelligence. Morgan Kauffmann, San Francisco (2001)
104. Kiliç, S., Kahraman, C.: Metaheuristic techniques for job shop scheduling problem and a fuzzy ant colony optimization algorithm. Studies in Fuzziness and Soft Computing 201, 401–425 (2006)
105. Kim, D., Choi, Y., Lee, S.: An accurate COG defuzzifier design using Lamarckian coadaptation of learning and evolution. Fuzzy Sets Syst. 130(2), 207–225 (2002)
106. King, R.T.F.A., Radha, B., Rughooputh, H.C.S.: A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In: Proc. of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, pp. 577–582 (2004)
107. Klir, G., Yuan, B.: Fuzzy sets and fuzzy logic; theory and applications. Prentice-Hall, Englewood Cliffs (1995)
108. Klösgen, W.: EXPLORA: a multipattern and multistrategy discovery assistant. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 249–271. MIT Press, Cambridge (1996)
109. Konar, A.: Computational intelligence: principles, techniques and applications. Springer, Berlin (2005)
110. Kovacs, T.: Strength or accuracy: credit assignment in learning classifier systems. Springer, Berlin (2004)
111. Koza, J.R.: Genetic programing: on the programming of computers by means of natural selection. The MIT Press, Cambridge (1992)
112. Krasnogor, N., Smith, J.E.: A tutorial for competent memetic algorithms: model, taxonomy, and design issue. IEEE Trans. Evol. Comput. 9(5), 474–488 (2005)
113. Kuncheva, L.: Fuzzy classifier design. Springer, Berlin (2000)
114. Kweku-Muata, Osey-Bryson: Evaluation of decision trees: a multicriteria approach. Computers and Operations Research 31, 1933–1945 (2004)
115. Last, M., Eyal, S.: A fuzzy-based lifetime extension of genetic algorithms. Fuzzy Sets and Systems 149, 131–147 (2005)
116. Last, M., Eyal, S., Kandel, A.: Effective black-box testing with genetic algorithms. In: Ur, S., Bin, E., Wolfsthal, Y. (eds.) HVC 2005. LNCS, vol. 3875, pp. 134–148. Springer, Heidelberg (2006)
117. Lau, H.C.W., Chan, T.M., Tsui, W.T.: Fuzzy logic guided genetic algorithms for the location assignment of items. In: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 4281–4288 (2007)
118. Lavra, N., Cestnik, B., Gamberger, D., Flach, P.: Decision support through subgroup discovery: three case studies and the lessons learned. Machine Learning 57, 115–143 (2004)
119. Lee, M.A., Esbensen, H.: Fuzzy/multiobjective genetic systems for intelligent systems design tools and components. In: Fuzzy Evolutionary Computation, pp. 57–80. Kluwer Academic Publishers, Dordrecht (1997)

120. Lee, M.A., Takagi, H.: Dynamic control of genetic algorithms using fuzzy logic techniques. In: Proc of the Fifth Int Conf on Genetic Algorithms, pp. 76–83. Morgan Kaufmann, San Francisco (1993)
121. Lee, M.A., Takagi, H.: A framework for studying the effects of dynamic crossover, mutation, and population sizing in genetic algorithms. In: Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms. LNCS, vol. 1011, pp. 111–126. Springer, Heidelberg (1994)
122. Li, Q., Tong, X., Xie, S., Liu, G.: An improved adaptive algorithm for controlling the probabilities of crossover and mutation based on a fuzzy control strategy. In: Proc. of the 6th International Conference on Hybrid Intelligent Systems and 4th Conference on Neuro-Computing and Evolving Intelligence, p. 50. IEEE Computer Society, Los Alamitos (2006)
123. Li, Q., Yin, Y., Wang, Z., Liu, G.: Comparative studies of fuzzy genetic algorithms. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 251–256. Springer, Heidelberg (2007)
124. Hongbo Liu, H., Abraham, A.: A Fuzzy adaptive turbulent particle swarm optimization. In: Proc. Fifth International Conference on Hybrid Intelligent Systems, pp. 445–450 (2005)
125. Liu, H., Abraham, A.: An hybrid fuzzy variable neighborhood particle swarm optimization algorithm for solving quadratic assignment problems. Journal of Universal Computer Science 13(9), 1309–1331 (2007)
126. Liu, H., Xu, Z., Abraham, A.: Hybrid fuzzy-genetic algorithm approach for crew grouping. In: Proceedings of the 2005, 5th International Conference on Intelligent Systems Design and Applications (ISDA 2005), pp. 332–337 (2005)
127. Liu, J., Lampinen, J.: Adaptive parameter control of differential evolution. In: Proceedings of the 8th International Mendel Conference on soft computing, pp. 19–26 (2002)
128. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. In: Proceedings of the 17th IEEE region 10th International Conference on computer, communications, control and power engineering, pp. 606–611 (2002)
129. Liu, J., Lampinen, J.: A fuzzy adaptive differential evolution algorithm. Soft Comput. 9, 448–462 (2005)
130. Maeda, Y.: Fuzzy adaptive search method for genetic programming. International Journal of Advanced Computational Intelligence 3(2), 131–135 (1999)
131. Maeda, Y., Ishita, M., Li, Q.: Fuzzy adaptive search method for parallel genetic algorithm with island combination process. International Journal of Approximate Reasoning 41, 59–73 (2006)
132. Maeda, Y., Li, Q.: Fuzzy adaptive search method for parallel genetic algorithm tuned by evolution degree based on diversity measure. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 677–687. Springer, Heidelberg (2007)
133. Magdalena, L.: Adapting the gain of an FLC with genetic algorithms. International Journal of Approximate Reasoning 17(4), 327–349 (1997)
134. Mamdani, E.H.: Applications of fuzzy algorithm for control a simple dynamic plant. Proceedings of the IEEE 121(12), 1585–1588 (1974)
135. Márquez, F.A., Peregrín, A., Herrera, F.: Cooperative evolutionary learning of linguistic fuzzy rules and parametric aggregation connectors for Mamdani fuzzy systems. IEEE Trans. on Fuzzy Systems 15(6), 1162–1178 (2008)
136. Matousek, R., Osmera, P., Roupec, J.: GA with fuzzy inference system. In: Proceedings of the 2000 Congress on Evolutionary Computation, pp. 646–651 (2000)

137. Meyer, L., Feng, X.: A fuzzy stop criterion for genetic algorithms using performance estimation. In: Proc. Third IEEE Int. Conf. on Fuzzy Systems, pp. 1990–1995 (1994)
138. Mikut, R., Jäkel, J., Gröll, L.: Interpretability issues in data-based learning of fuzzy systems. Fuzzy Sets and Systems 150, 179–197 (2005)
139. Moriarty, D.E., Miikkulainen, R.: Efficient reinforcement learning through symbiotic evolution. Machine Learning 22, 11–32 (1996)
140. Mucientes, M., Vidal, J.C., Bugarín, A., Lama, M.: Processing time estimations by variable structure TSK rules learned through genetic programming. Soft Computing 13(3), 497–509 (2009)
141. Mühlenbein, H., Schlierkamp-Voosen, D.: Predictive models for the breeder genetic algorithm I. continuous parameter optimization. Evolutionary Computation 1, 25–49 (1993)
142. Mirabedini, S.J., Teshnehlab, M.: Performance evaluation of fuzzy ant based routing method for connectionless networks. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4488, pp. 960–965. Springer, Heidelberg (2007)
143. Nojima, Y., Kuwajima, I., Ishibuchi, H.: Data set subdivision for parallel distribution implementation of genetic fuzzy rule selection. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007), London, UK, pp. 2006–2011 (2007)
144. Nojima, Y., Ishibuchi, H., Kuwajima, I.: Parallel distributed genetic fuzzy rule selection. Soft Computing 13(3), 511–519 (2009)
145. Orriols-Puig, A., Casillas, J., Bernadó-Mansilla, E.: Fuzzy-UCS: preliminary results. In: 10th International Workshop on Learning Classifier Systems (IWLCS 2007), London, UK, pp. 2871–2874 (2007)
146. Palm, R., Driankov, D.: Model based fuzzy control. Springer, Berlin (1997)
147. Park, D., Kandel, A., Langholz, G.: Genetic-based new fuzzy-reasoning models with applications to fuzzy control. IEEE Transactions on Systems, Man and Cybernetics 24(1), 39–47 (1994)
148. Pedrycz, W. (ed.): Fuzzy modelling: paradigms and practice. Kluwer Academic Press, Dordrecht (1996)
149. Pedrycz, W.: Fuzzy evolutionary computing. Soft Computing 2, 61–72 (1998)
150. Pham, D.T., Karaboga, D.: Optimum design of fuzzy logic controllers using genetic algorithms. Journal of Systems Engineering 1, 114–118 (1991)
151. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential evolution: a practical approach to global optimization. Springer, Heidelberg (2005)
152. Rachmawati, L., Srinivasan, D.: A hybrid fuzzy evolutionary algorithm for a multi-objective resource allocation problem. In: Proc. of the Fifth International Conference on Hybrid Intelligent Systems (HIS 2005), pp. 55–60 (2005)
153. Regattieri-Delgado, M., Yassue-Nagai, E., Ramos de Arruda, L.V.: A neuro-coevolutionary GFS to build soft sensors. Soft Computing 13(3), 481–495 (2009)
154. Reynolds, R.G.: An introduction to cultural algorithms. In: Proc. of the 3rd Annual Conference on Evolutionary Programming, pp. 131–139. World Scientific, Singapore (1994)
155. Reynolds, R.G., Chung, C.J.: Regulating the amount of information used for self-adaptation in cultural algorithms. In: Proc. of the Seventh Int. Conf. on Genetic Algorithms, pp. 401–408. Morgan Kaufmann Publishers, San Francisco (1997)
156. Richter, J.N.: Fuzzy evolutionary cellular automata. Master thesis, UT State University, Logan, Utah (2003)
157. Rojas, R.: Neural networks: a systematic introduction. Springer, Berlin (1996)
158. Ronald, E.: When selection meets seduction. In: Proc of the Fifth Int Conf on Genetic Algorithms, pp. 167–173. Morgan Kaufmann, San Francisco (1993)

159. Sahoo, N.C., Ranjan, R., Prasad, K., Chaturvedi, A.: A fuzzy-tuned genetic algorithm for optimal reconfigurations of radial distribution network. European Trans. Electr. Power 17, 97–111 (2006)
160. Sahoo, N.C., Prasad, K.: A fuzzy genetic approach for network reconfiguration to enhance voltage stability in radial distribution systems. Energy Conversion and Management 47, 3288–3306 (2006)
161. Sánchez, L., Casillas, J., Cordón, O., del Jesus, M.J.: Some relationships between fuzzy and random classifiers and models. International Journal of Approximate Reasoning 29, 175–213 (2001)
162. Sánchez, L., Couso, I.: Advocating the use of imprecisely observed data in genetic fuzzy systems. IEEE Transactions on Fuzzy Systems 15(4), 551–562 (2007)
163. Sánchez, L., Otero, J., Couso, I.: Obtaining Linguistic Fuzzy Rule-based Regression Models from Imprecise Data with Multiobjective Genetic Algorithms. Soft Computing 13(3), 467–479 (2009)
164. Sebban, M., Nock, R., Cahuchat, J.H., Rakotomalala, R.: Impact of learning set quality and size on decision tree performance. Int. J. of Computers, Syst. and Signals 1, 85–105 (2000)
165. Setnes, M., Roubos, H.: GA-fuzzy modeling and classification: complexity and performance. IEEE Transactions on Fuzzy Systems 8(5), 509–522 (2000)
166. Setzkorn, C., Paton, R.C.: On the use of multi-objective evolutionary algorithms for the induction of fuzzy classification rule systems. BioSystems 81, 101–112 (2005)
167. Sharma, S.K., Irwin, G.W.: Fuzzy coding of genetic algorithms. IEEE Trans. on Evolutionary Computation 7(4), 344–355 (2003)
168. Shi, Y., Eberhart, R., Chen, Y.: Implementation of evolutionary fuzzy systems. IEEE Trans Fuzzy Systems 7(2), 109–119 (1999)
169. Gulshan, S., Kalyanmoy, D.: Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In: Proc. of the 8th annual conference on Genetic and evolutionary computation, pp. 1305–1312 (2006)
170. Smith, S.: A learning system based on genetic algorithms. Ph.D. thesis. Unversity of Pittsburgh (1980)
171. Smith, J.E.: Coevolving memetic algorithms: a review and progress report. IEEE Transaction on Systems, Man, and Cybernetics Part B: Cybernetics 37(1), 6–17 (2007)
172. Song, Y.H., Wang, G.S., Johns, A.T., Wang, P.Y.: Improved genetic algorithms with fuzzy logic controlled crossover and mutation. In: UKACC International Conference on CONTROL 1996, pp. 140–144 (1996)
173. Song, Y.H., Wang, G.S., Wang, P.T., Johns, A.T.: Environmental/economic dispatch using fuzzy logic controlled genetic algorithms. IEEE Proc. on Generation, Transmission and Distribution 144(4), 377–382 (1997)
174. Streifel, R.J., Marks II, R.J., Reed, R., Choi, J.J., Healy, M.: Dynamic fuzzy control of genetic algorithm parameter coding. IEEE Trans Systems, Man, and Cybernetics - Part B: Cybernetics 29(3), 426–433 (1999)
175. Subbu, R., Sanderson, A.C., Bonissone, P.: Fuzzy logic controlled genetic algorithms versus tuned genetic algorithms: An agile manufacturing application. In: Proc. ISIC/CIRA/ISAS Conf., pp. 434–440 (1998)
176. Subbu, R., Bonissone, P.: A retrospective view of fuzzy control of evolutionary algorithm resources. In: Proc. IEEE Int. Conf. Fuzzy Syst., pp. 143–148 (2003)
177. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modelling and control. IEEE Transactions on Systems, Man and Cybernetics 15(1), 116–132 (1985)
178. Talbi, E.-G.: A taxonomy of hybrid metaheuristics. J. Heuristics 8(5), 541–565 (2002)

179. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson, Boston (2006)
180. Thrift, P.: Fuzzy logic synthesis with genetic algorithms. In: Proc. of 4th International Conference on Genetic Algorithms (ICGA 1991), pp. 509–513 (1991)
181. Teodorovic, D., Lucic, P.: Schedule synchronization in public transit using the fuzzy ant system. Transportation Planning and Technology 28(1), 47–76 (2005)
182. Tettamanzi, A.G.: Evolutionary algorithms and fuzzy logic: a two-way integration. In: 2nd Joint Conference on Information Sciences, pp. 464–467 (1995)
183. Tettamanzi, A., Tomassini, M.: Fuzzy evolutionary algorithms. In: Soft Computing: Integrating Evolutionary, Neural, and Fuzzy Systems, pp. 233–248. Springer, Heidelberg (2001)
184. Tsang, C.-H., Tsai, J.H., Wang, H.: Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recognition 40(9), 2373–2391 (2007)
185. Tuson, A.L., Ross, P.: Adapting operator settings in genetic algorithms. Evolut. Comput. 6(2), 161–184 (1998)
186. Valenzuela-Rendon, M.: The fuzzy classifier system: A classifier system for continuously varying variables. In: Proc. of 4th International Conference on Genetic Algorithms (ICGA 1991), pp. 346–353 (1991)
187. Valenzuela-Rendon, M.: Reinforcement learning in the fuzzy classifier system. Expert Systems with Applications 14, 237–247 (1998)
188. Venturini, G.: SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 280–296. Springer, Heidelberg (1993)
189. Voget, S.: Multiobjective optimization with genetic algorithms and fuzzy-control. In: Proc of the Fourth European Congress on Intelligent Techniques and Soft Computing, pp. 391–394 (1996)
190. Voget, S., Kolonko, M.: Multidimensional optimization with a fuzzy genetic algorithm. Journal of Heuristic 4(3), 221–244 (1998)
191. Voigt, H.M.: Fuzzy evolutionary algorithms. Technical Report tr-92-038, International Computer Science Institute (ICSI), Berkeley (1992)
192. Voigt, H.M.: Soft genetic operators in evolutionary algorithms. In: Banzhaf, W., Eckman, F.H. (eds.) Evolution as a Computational Process 1992. LNCS, vol. 899, pp. 123–141. Springer, Heidelberg (1995)
193. Voigt, H.M., Anheyer, T.: Modal mutations in evolutionary algorithms. In: Proceeding of the First IEEE International Conference on Evolutionary Computation, pp. 88–92. IEEE Press, Los Alamitos (1994)
194. Voigt, H.M., Born, J., Santibáñez-Koref, I.: A multivalued evolutionary algorithms. Technical Report tr-93-022, International Computer Science Institute (ICSI), Berkeley (1993)
195. Voigt, H.M., Mühlenbein, H., Cvetković, D.: Fuzzy recombination for the breeder genetic algorithm. In: Proc. of the Sixth Int. Conf. on Genetic Algorithms, pp. 104–111. Morgan Kaufmann Publishers, San Francisco (1995)
196. Wang, K.: A new fuzzy genetic algorithms based on population diversity. In: Proc. of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 108–112 (2001)
197. Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K.F.: Multiobjective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. Fuzzy Sets and Systems 149, 49–186 (2005)

198. Wang, P.Y., Wang, G.S., Song, Y.H., Johns, A.T.: Fuzzy logic controlled genetic algorithms. In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, pp. 972–979 (1996)
199. Wilson, S.: Classifier fitness based on accuracy. Evol. Comput. 3(2), 149–175 (1995)
200. Wong, M.L., Leung, K.S.: Data mining using grammar based genetic programming and applications. Kluwer Academic Publishers, Dordrecht (2000)
201. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
202. Xu, H.Y., Vukovich, G.: A fuzzy genetic algorithm with effective search and optimization. In: Proc. of 1993 International Joint Conference on Neural Networks, pp. 2967–2970 (1993)
203. Xu, H.Y., Vukovich, G., Ichikawa, Y., Ishii, Y.: Fuzzy evolutionary algorithms and automatic robot trajectory generation. In: Proceeding of the First IEEE International Conference on Evolutionary Computation, pp. 595–600. IEEE Press, Los Alamitos (1994)
204. Yager, R.R., Filev, D.P.: Essentials of fuzzy modeling and control. John Wiley & Sons, Chichester (1994)
205. Yamakawa, T.: Stabilization of an inverted pendulum by a high-speed fuzzy logic controller hardware system. Fuzzy Sets and Systems 32, 161–180 (1989)
206. Yang, Q., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making 5(4), 597–604 (2006)
207. Yun, Y., Gen, M.: Performance analysis of adaptive genetic algorithm with fuzzy logic and heuristics. Fuzzy Optimization and Decision Making 2, 161–175 (2003)
208. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
209. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Proc. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (EUROGEN 2001), Barcelona, Spain, pp. 95–100 (2001)
210. Zeng, X., Rabenasolo, B.: A fuzzy logic based design for adaptive genetic algorithms. In: Proc. of the European Congress on Intelligent Techniques and Soft Computing, pp. 660–664 (1997)
211. Zhu, L., Zhang, H., Jing, Y.: A new neuro-fuzzy adaptive genetic algorithm. Journal of Electronic Science and Technology of China 1(1) (2003)