PLEASE READ ALL 'COMMENT' TEXT BEFORE PREPARING YOUR ARTICLE. If you do not see the Comments, select View > Print Layout. Please delete them before submitting (Review > Delete All Comments in Document) so that peer reviewers see a clean copy of the manuscript.

Remember that you are writing for an interdisciplinary audience. Please be sure to discuss interdisciplinary themes, issues, debates, etc. where appropriate. Note that the WIREs are forums for review articles, rather than primary literature describing the results of original research.

If you have any questions, contact your editorial office.

WIREs Wiley Interdisciplinary Reviews

Article Title: Automatic Synthesis of Fuzzy Systems: an Evolutionary Overview with a Genetic Programming Perspective

Article Type:

○ OPINION ○ PRIMER ◉ OVERVIEW

○ ADVANCED REVIEW ○ FOCUS ARTICLE ○ SOFTWARE FOCUS

Authors:

[**List each person's full name, ORCID iD, affiliation, email address, and any conflicts of interest.** Copy rows as necessary for additional authors. Please use an asterisk (*) to indicate the corresponding author.]

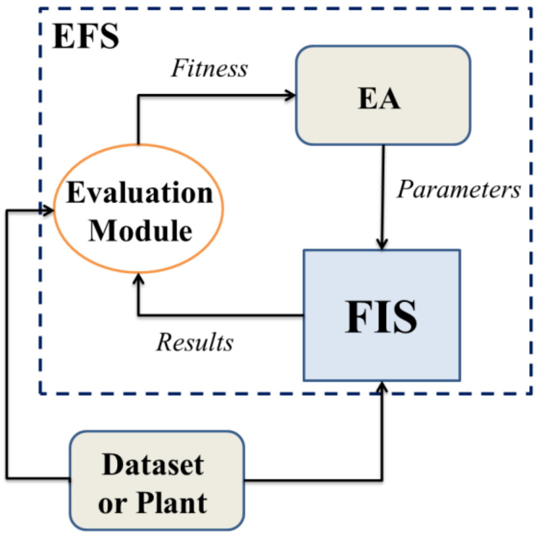
First author Adriano S. Koshiyama, http://orcid.org/0000-0001-7536-1503, University College London (UCL), a.koshiyama@cs.ucl.ac.uk
Second author Ricardo Tanscheit, http://0000-0002-8688-9095, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), ricardo@ele.puc-rio.br
Third author Marley M.B.R. Vellasco*, http://0000-0002-9790-1328, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), marley@ele.puc-rio.br

Abstract

Studies in Evolutionary Fuzzy Systems (EFS) began in the 90s and have experienced a fast development since then, with applications in areas such as pattern recognition, curve-fitting and regression, forecasting and control. An EFS results from the combination of a Fuzzy Inference System (FIS) with an Evolutionary Algorithm (EA). This relationship can be established for multiple purposes: fine-tuning of FIS's parameters, selection of fuzzy rules, learning of a rule base or membership

functions from scratch, etc. Each facet of this relationship creates a strand in the literature, as membership function fine-tuning, fuzzy rule-based learning, etc. and the purpose here is to outline some of what has been done in each aspect. Special focus is given to Genetic Programming-based EFSs by providing a taxonomy of the main architectures available, as well as by pointing out the gaps that still prevail in the literature. The concluding remarks address some further topics of current research and trends, such as interpretability analysis, multi-objective optimisation, and synthesis of a FIS through Evolving methods.

Graphical/Visual Abstract and Caption



Introduction

Fuzzy Logic emerged in the 60's with the pioneering work of Lotfi Zadeh (Zadeh, 1965). Despite its troublesome beginning, as described by Zadeh (2008), mainly originated by a group rooted in probability theory, Fuzzy Logic was able to blossom and yield fruits in the most different areas of scientific knowledge. One of the main works in this period, the Fuzzy Inference System (FIS) proposed by Mamdani (1974) in the 70's, was used not only to devise plant controllers but also in medical applications and pattern recognition problems (Mamdani & Assilian, 1975; Wechsler, 1976; Kickert & Koppelaar, 1976).

This start propelled not only new applications but the development of new FISs (Tsukamoto, 1979; Takagi & Sugeno, 1985). These new modelling methodologies addressed more sophisticated and complex demands; at the same time, a lack of experts to elicit a fuzzy system parametrisation grew. The improvement of computers and database systems allowed large volumes of data and processing to be performed faster than ever before. In this context, in the 90s arose the firsts Hybrid FISs as a viable alternative to model phenomena only analysed until then through statistical methods. Hybrid

FISs were part of a trend moving from expert to data-driven modelling by using information stored in a database to learn membership function parameters as well as the rule base.

In the particular area of Hybrid FISs, two main approaches can be highlighted: Neuro-Fuzzy Systems (Jang, Sun, & Mizutani, 1997; Nauck, Klawoon, & Kruse, 1997; Abraham, 2001; De Souza, Vellasco, & Pacheco, 2002) and Evolutionary Fuzzy Systems (Ishibuchi, Nozaki, Yamamoto, & Tanaka, 1995; Carse, Fogarty, & Munro, 1996). Neuro-Fuzzy Systems are attractive due to the intrinsic characteristic of Neural Networks: automatic adjustment of parameters and universal approximation (Nauck et al., 1997; Abraham, 2001). These two elements combined underpin the building of an accurate, adaptive and robust FIS. However, Evolutionary Fuzzy Systems are more flexible than Neuro-Fuzzy Systems because they allow not only the maximisation of accuracy, but also the inclusion of other subjective criteria as minimisation of a rule base size and length format, overlap degree of membership functions, choice of fuzzy operators, etc. (Pedrycz, 1997; Cordon, Herrera, Hoffmann, & Magdalena, 2001). Additionally, the implementation cost is more affordable due to the lower degree of hybridisation between the techniques; given a FIS and an Evolutionary Algorithm (e.g., Genetic Algorithm) an Evolutionary Fuzzy System can be easily set up.

Since its inception, Evolutionary Fuzzy Systems have experienced a fast development, spawning to different applications, evolutionary schemes and approaches. Hence, this overview aims to provide a map for navigating across the terminology, methodologies and main works presented in the literature. More specifically, this paper puts more emphasis on Genetic Programming-based EFSs, to complement previous reviews on EFSs based on other evolutionary algorithms (Fernandez, Lopez, Del Jesus, & Herrera, 2015). In this respect, this work is structured as follows: the next section presents in more details the organisation, taxonomy and some of the literature in the area. The third section provides an overview of Genetic Programming-based EFSs literature, composed of a categorisation and a description of the main architectures available. The last section consists of concluding remarks and suggestions of other interesting topics for further reading: interpretability, big data problems, multi-objective systems, software and Evolving Fuzzy Systems.

EVOLUTIONARY FUZZY SYSTEMS

The pioneering works on Evolutionary Fuzzy Systems (EFS) where those of Karr (1991), Valenzuela-Rendón (1991), and Thrift (1991). Each author opened the way for different approaches still in use nowadays: parameter optimisation of membership functions (Karr, 1991) and rule base discovery through one individual as one rule approach (Michigan) (Valenzuela-Rendón, 1991) or by treating an individual as a rule base (Pittsburgh) (Thrift, 1991). In general, each work tried to synthesise an EFS through a Genetic Algorithm (GA), by that time the well-known Evolutionary Algorithm (EA).

After those works, a complete area has developed, with new applications other than Control (e.g., pattern recognition, regression, etc.), by using novel EAs (like Genetic Programming - GP), and considering other mechanisms to evaluate, synthesise and build a FIS (Cordón, Gomide, Herrera, Hoffmann, & Magdalena, 2001; Fernández, García, Luengo, Bernadó-Mansilla, & Herrera, 2010). A complete new literature on EFS was and still is being elaborated, such as books and chapters (Geyer-Schulz, 1997; Cordón et al., 2001; Ishibuchi, Nakashima, & Nii, 2005; Ishibuchi & Nojima, 2015), special editions in journals (Nojima, Alcalá, Ishibuchi, & Herrera, 2011; Alcalá, Nojima, Ishibuchi, &

Herrera, 2012; Alcalá, Nojima, Ishibuchi, & Herrera, 2013; Tao, Chuang, & Huang, 2016), and many reviews (Cordón et al., 2004; Herrera, 2008; Fernández et al., 2010; Fazzolari, Alcalá, Nojima, Ishibuchi, & Herrera, 2013; Fernandez, Lopez, Del Jesus, & Herrera, 2015). This demonstrates the relevance and the growth of this family of Hybrid Systems.

There is no intention here to cover all works in the area and its subareas but to outline the main definitions and concepts that underlines an EFS. In this sense, the next subsection deals with the central terminology and formulation of a generic EFS. The second subsection addresses one of the most common applications of an EFS: rule base creation. This topic deserves a particular treatment, due to the different approaches that have been developed so far. The third subsection consists of a summary of the whole section.

General Guidelines

The most common architecture of an EFS is shown in Figure 1: a result of the interaction between a Fuzzy Inference System (FIS) and an evolutionary algorithm (EA). Many EAs may be considered, such as GA, GP, Evolutionary Strategies and Non-Dominated Sorting GA. The usual relationship is the FIS taking advantage of EA's capability to optimise membership functions, uncover rules, etc.

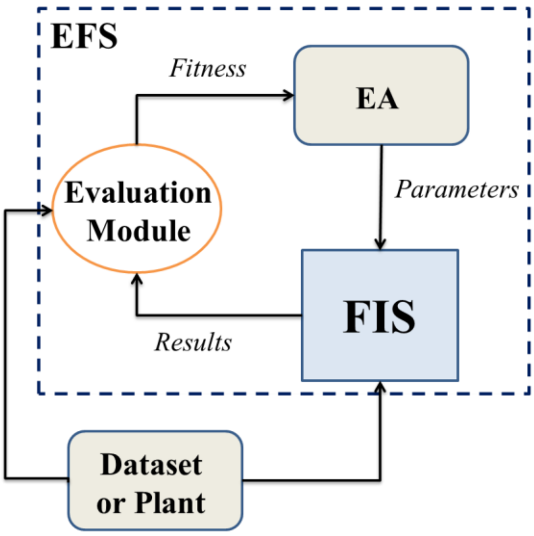


Figure 1. Generic Diagram of an EFS.

Usually, a FIS is made of a Reasoning Method or Inference Method and of a Knowledge Base. The latter can be broken down into two other components: Parameter and Rule base. Their definitions are:

- **Rule Base**: Composed of the rules that underpin a FIS.

- **Parameter Base**: Made of the remaining parameters that compose a FIS, such as membership functions, t-norms, aggregation operators and defuzzification methods.

The Fuzzy Inference is the mechanism that uses the information contained in a Knowledge Base and performs pattern recognition, generates control signals and computes expected values of a forecaster. Hence, when the Fuzzy Inference Method is established, the role of an EA is to act over some component of the Knowledge Base as, for example, update components of the Parameter Base or make the Rule Base more concise. The role of an EA in an EFS is illustrated in Figure 2, adapted from Córdon et al. (2004).

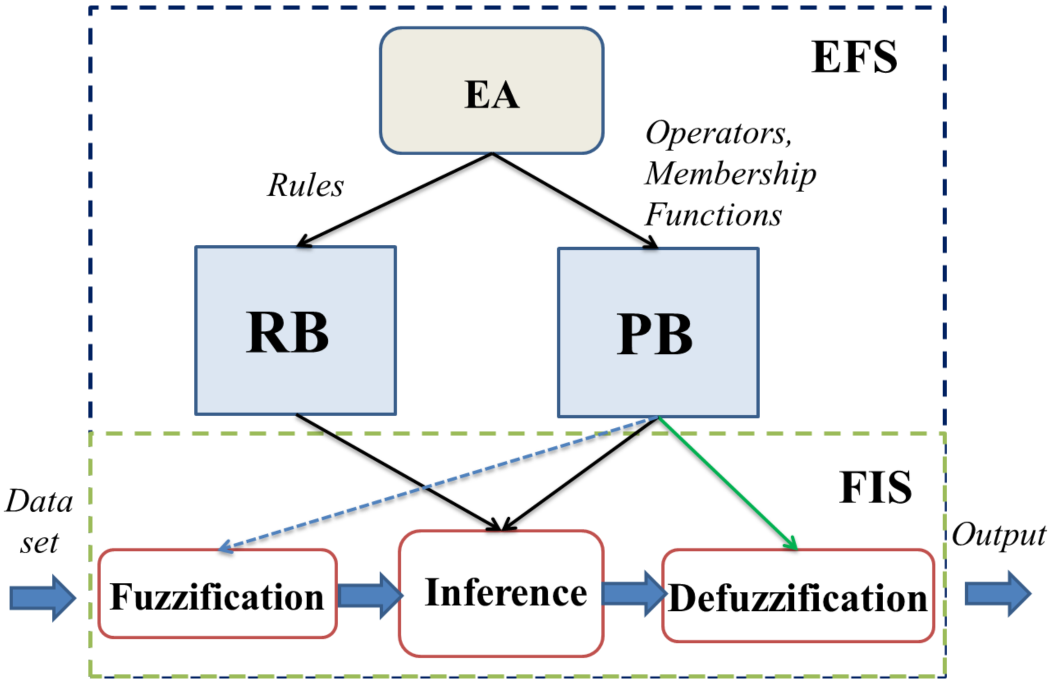


Figure 2. Components of an EFS structure that an EA can act over.

Another relevant definition regards the aims of an EFS. In general, an EFS has two objectives: Evolutionary Parameter Tuning and Knowledge Discovery. Both are briefly described below:

- **Evolutionary Parameter Tuning**: given a Knowledge Base – expert-driven, or extracted via the Wang & Mendel method (Wang & Mendel, 1992) –, an EA is employed to fine-tune its parameters or simplify its structure.

- **Knowledge Discovery**: the EA is used for learning a component (or the whole) of the Knowledge Base.

Evolutionary Parameter Tuning

With the aim to improve a pre-elaborated FIS, many authors have employed an EA to increase accuracy or to reduce the number of rules. In general, there are four unfolding directions of the Evolutionary Parameter Tuning objective: membership function fine-tuning, fuzzy operators adaptation, fuzzy rules selection and defuzzification improvement. Each of these is described below:

- **Membership function fine-tuning**: from a pre-elaborated Knowledge Base, fine-tuning is developed by encoding in an EA's individual the parameters that represent the support of the membership functions (Figure 3). The adjustment can take place by shifting the membership function (Figure 3a), narrowing its support (Figure 3b), or by making them asymmetrical (Figure 3c). Córdon et al. (2001, p.111) present different forms to codify these structures in an EA (in this case a GA). Arslan and Kaya (2001) determine not only the parameters but also the function type; Esmin, Aoki, & Lambert-Torres (2002) adjust the membership functions through Particle Swarm Optimisation, obtaining better results than by using a GA. Casillas, Cordón, Del Jesus, & Herrera (2005) apply linguistic modifiers to optimise the membership functions. Fine-tuning after the FIS construction is also a possibility (Alcalá, Gacto, & Herrera, 2011; Alcalá, Nojima, Herrera, & Ishibuchi, 2011; Brito, Vellasco, & Tanscheit (2012); Sanz, Fernández, Bustince, & Herrera, 2013; Fernández, Del Río, & Herrera, 2016). However, very few works follow the criteria outlined by De Oliveira (1999) to constrain the adjustment of membership functions so that the semantic is kept intact (distinguishability, the universe of discourse coverage, normalisation, etc.).

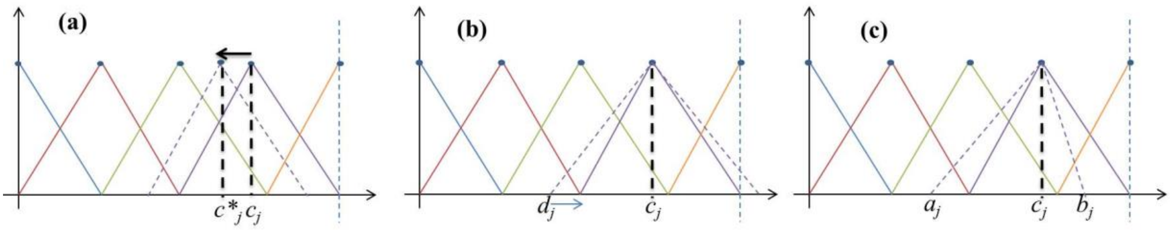


Figure 3. Different ways to manipulate the shape and location of a membership function.

- **Fuzzy operators adaptation**: the central principle of this approach is to improve the FIS by swapping or tuning t-norms, t-conorms and aggregation operators. Hence, most of the applications reside on the use of parameterizable fuzzy operators as Schweizer-Sklar, Hamacher, Frank, Yager, etc. A compilation and theoretical analysis of that can be found in (Klement, Mesiar, & Rap, 2000). The work of Alcalá-Fdez, Herrera, Márquez, & Peregrín (2007) is the archetypical example of this movement, as well as those of Crockett, Bandar, Fowdar, & O'Shea (2006), and Crockett, Bandar, & McLean (2007), which also consider the modification of membership functions.

- **Fuzzy rules selection**: from an already established Knowledge Base, it is possible to make use of an EA to simplify and reduce fuzzy rules. In this sense, it is possible to improve the rule base interpretability (making it more compact and less conflicting) and the system's overall accuracy. The use of an EA is reasonably adequate, since the search space grows exponentially as the number of rules increases. A brute-force approach is infeasible even for small problems. Figure 4 presents a possible representation of a GA for this sort of task. Among several approaches, the most common ones freeze the Parameter Base and generate the Rule Base via the Wang & Mendel approach. Casillas, Cordón, Del Jesus, & Herrera (2005) applied this scheme to regression problems, with the addition of fine-tuning the membership functions by applying drifts and linguistic modifiers. Fernández, Del Jesus, & Herrera (2010) used this approach to deal with imbalanced classification tasks. Pulkinnen and Koivisto (2010) employed the Wang and Mendel method with a Decision Tree to compose an EFS for regression tasks. Sánz et al. (2013) used a similar approach but devoted to a Type-2 FIS to evaluate and classify heart diseases. Cintra, Camargo, & Monard (2016) exhibit a proposal for the automatic generation of fuzzy rule bases, which extracts a set of rules using the formal concept analysis theory directly from data. After extracting the rules forming the genetic search space, their methodology involves the use of a GA to select the final rule base. Rey, Galende, Fuente, & Sainz-Palmero (2017) introduce a Multi-Objective EFS where Relevance is added to Accuracy and Interpretability for a better trade-off in the rule selection process.

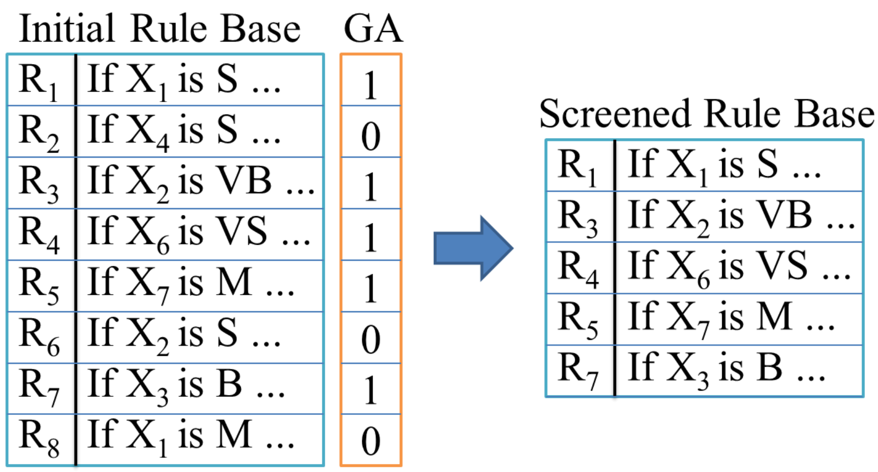


Figure 4. Example of rule base screening through a binary GA individual –only the rules for which the values are 1 are kept in the screened rule base.

- **Defuzzification improvement**: a practical approach to refine a FIS is to revamp the defuzzification process. Given a set of inputs and targets, a simple procedure would be: (*i*) define a set of defuzzification methods (centre of area, height, etc.); (*ii*) propagate the inputs through the reasoning method and apply the different defuzzification methods; (*iii*) combine the alternatives to minimise the total difference between the predictions and the target. A work in this direction is that of Kim, Choi, & Lee (2002). Márquez, Márquez, & Peregrín (2012) point out that an adaptive defuzzification mechanism improves the system accuracy, but the weights introduced tend to decrease the system overall interpretability. In this sense, their work aims to use a Multi-Objective EA to increase accuracy, but preserves the interpretability with three goals in mind: 1) reduce the number of total rules by considering that a rule with weight close to zero can be removed; 2) disregard weights that are close to

1, and 3) reduce the average number of rules triggered at the same time. A more recent version of this work is devoted to Big Data problems (Márquez, Márquez, & Peregrín, 2017).

In general, it is hard to classify a certain work as belonging exclusively to one of the four subcategories above. The most common approach consists of a hybrid scheme that usually blends two of the four procedures. Also, all those approaches aim at improving an already existing Knowledge Base. The next topic considers the occasions in which Knowledge Base elaboration is a pending issue.

Knowledge Discovery

Regarding Knowledge Discovery, there are two main lines that can be followed to set up an EFS: Granularity Learning (number and format of membership functions), and Rule Base Learning. The prevailing forms of these approaches are described below.

- **Granularity Learning**: a typical approach codifies in the structure of an EA's individual the number of membership functions that each variable can assume. Given an encoding structure (chromosome, tree, etc.) defined as C, the number of membership functions is defined by the user, usually within the set {1,2,...,7} (Cordón, Herrera, Magdalena, & Villar, 2001; Alcalá et al., 2011). In this case, the value 1 represents the removal of a feature from the rule learning process (also known as don't care operator) and 2, 3, ...,7 indicates the number of membership functions. These tend to be triangular shaped, uniformly divided and doubly overlapped (sometimes called strong partition). Figure 5 shows an example of this codification. Given the number and position of membership functions, the Wang & Mendel method is generally used for rule generation. Besides, it is common to displace the functions during the evolutionary process. Using the previous encoding C, it is possible to break it down in $C = [C_1, C_2]$, with C_1 indicating the granularity and C_2 establishing the shifting degree (including restrictions) of all membership functions in the corresponding universe of discourse. Hence, it is possible to adapt not only the granularity but also the disposition of the functions. This sort of approach is a typical example of a 2-tuple approach (Herrera & Martinez, 2002), which has often been used in recent literature (Gacto, Alcalá, & Herrera, 2009; Alcalá et al., 2011; Palacios, Palacios, Sánchez, & Alcalá-Fdez, 2015; Rodríguez-Fdez, Mucientes, & Bugarín, 2016a; Rodríguez-Fdez, Mucientes, & Bugarín, 2016b).

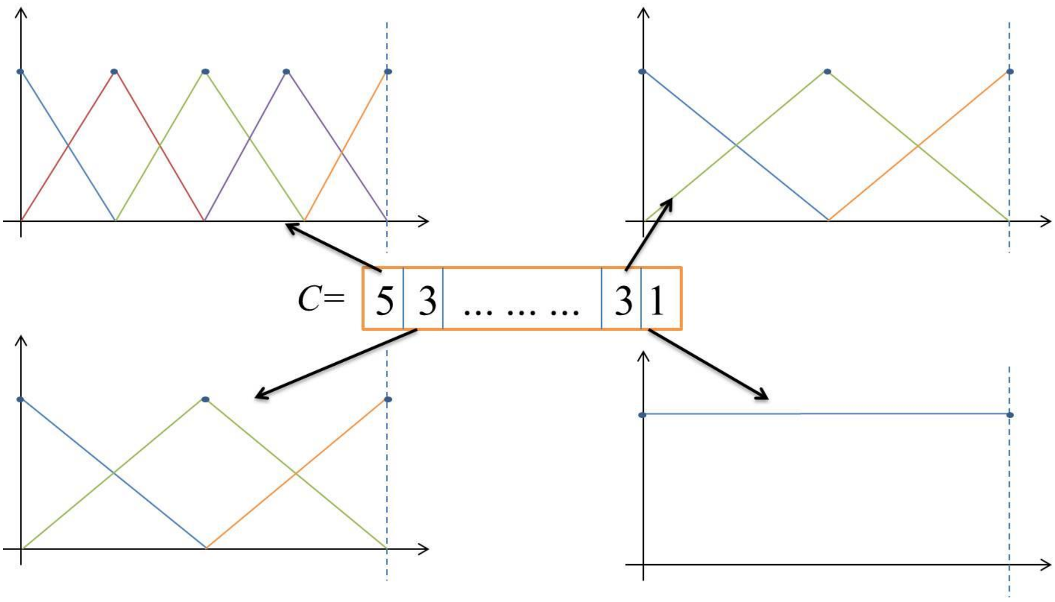


Figure 5. Example of granularity learning.

- **Rule Base Learning**: after the definition of the Parameter Base, an EA is responsible for extracting rules from a dataset. There are many approaches to codify a set of rules; in general, they tend to concentrate exclusively on the learning process of a rule base (Herrera, 2008). This particular subject will be dealt with in the next subsection.

Figure 6 displays an overall view of the areas covered by an EFS: the two primary objectives – Evolutionary Parameter Tuning and Knowledge Discovery – and possible actions over different parts of a typical FIS.

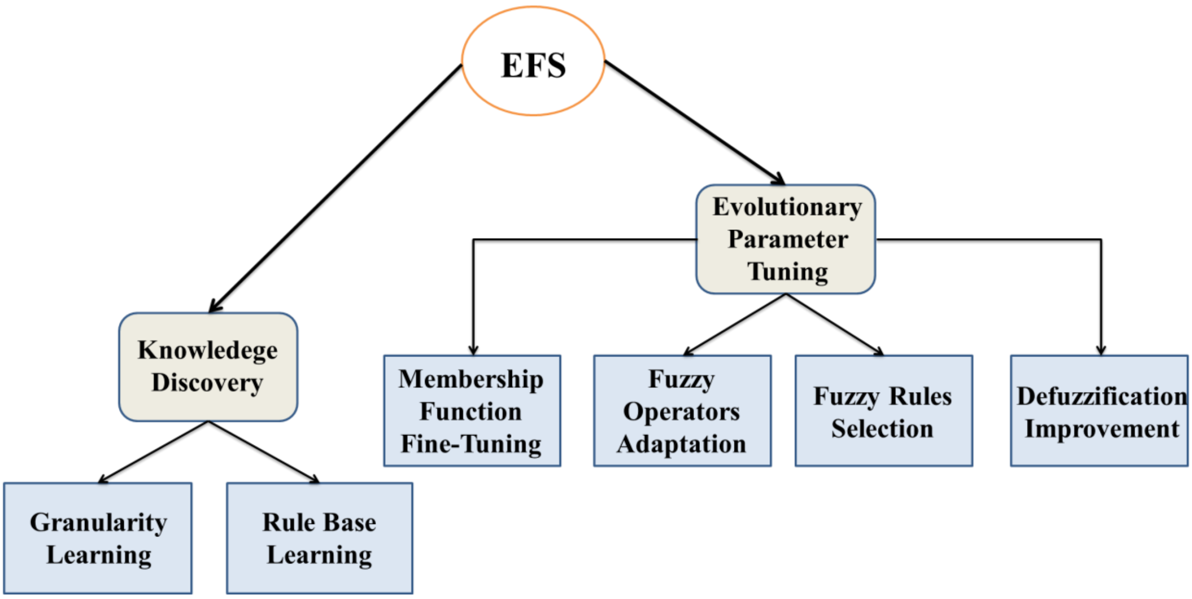


Figure 6. Global perspective of the areas in which an EA can operate over a FIS.

Evolutionary Fuzzy Systems for Rule Base Learning

The works on EFS for Rule Base generation explore ways to codify a rule base in an individual or a population of an EA, in such a way that the rule learning process can address the user-set criteria. In the literature, it is possible to find book chapters on this topic (Cordón et al., 2001), as well as publications that extrapolate the range of the EFSs (Fernández et al., 2010). The four primary forms of rule base encoding are presented below: Pittsburgh, Michigan, Iterative Rule Learning and Genetic Cooperative-Competitive Learning. Each of them not only distinguishes itself by the encoding process but also by the way the quality of a fuzzy rule base is evaluated.

Pittsburgh Approach

From all the potential approaches, the Pittsburgh-type might be the most intuitive and straightforward to conceive. This approach considers that each EA's individual is a rule base (Figure 7). The user initially sets the Parameter Base, that is, t-conorms and t-norms operators, granularity, membership functions shapes and the defuzzification method. This component usually remains fixed and each individual proposes a potential rule base; the best individual is the one that satisfies some user-defined criteria such as accuracy, interpretability, response time, etc.

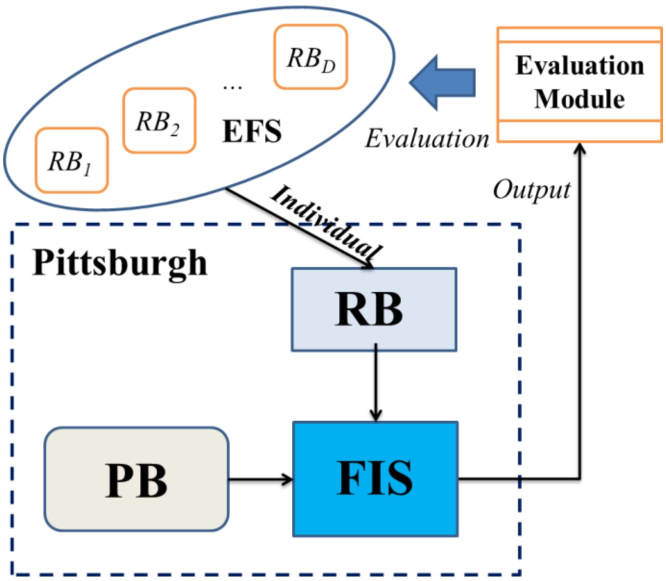


Figure 7. Macro level description of the synthesising process of a Pittsburgh-type EFS.

Regarding applications, a Pittsburgh-type EFS appears in practically all areas. One of the pioneers in the EFS area, Thrift (1991), and also Alba, Cotta, & Troya (1996) encoded in an individual a rule base to build a fuzzy controller. Córdon et al. (2001, p.146-147) presents a series of applications of Pittsburgh-type EFSs. The works of Sánchez, Couso, & Corrales (2001) and Tsakonas (2006) used GP to elaborate a FIS for pattern recognition. Gorzalczany and Rudzinski (2012) describe an application for time series forecasting; Casillas, Martínez, & Benítez (2009) apply the Pittsburgh-type EFS to build compact and consistent fuzzy rule bases for regression tasks.

The critics of this approach turn their batteries mainly to the computational time that a Pittsburgh-type EFS takes to synthesise a rule base and some even highlight the reduced effectiveness of the recombination operators in generating diverse populations (Cordón et al., 2004). An alternative explored by many authors is the computational parallelisation of the EFS (Rojas et al., 2001) or the hybridisation with other less demanding rule-based learning strategies (Ishibuchi, Yamamoto, & Nakashima, 2005), or even exploring at the same time hybridisation and parallelisation (Ishibuchi, Yamane, Nojima, 2013). Rudziński (2016) proposes a multi-objective genetic approach to design interpretability-oriented fuzzy rule-based classifiers from data. The proposed approach allows to obtain systems with various levels of compromise between their accuracy and interpretability. Original crossover and mutation operators, as well as chromosome-repairing technique to directly transform the rules are also proposed. The interpretability measure is based on the arithmetic mean of three components: the average length of rules, the number of active fuzzy sets, and the number of active inputs of the system. This same methodology is applied to rule-based credit classification (Gorzałczany & Rudziński, 2016).

Although gains have been reported with the use of these mixed approaches, many authors prefer to follow other forms of rule base encoding. In this sense, mapping only one rule per individual can drastically reduce machine time and increase the impact of an EA's recombination operators. This type of codification, denoted as Michigan, is discussed in the next topic.

Michigan Approach: General Outlook

The Michigan approach aims to generate a concise rule base with low computation cost. In this representation, an EA's individual codifies only one rule, such that the final rule base is the concatenation of the individuals nurtured by the EA. The consequence is a dramatic reduction in computation overhead (memory and time), evaluation and recombination time. On the other hand, the evaluation process is more complicated and indirect, since one rule can represent well the process when isolated, but, when inserted in a rule base, may cause conflicts in the fuzzy reasoning process with a direct effect in the prediction phase.

Therefore, any Michigan approach includes, beyond the rule base evaluation, a method to assign credit to each rule; this is usually linked with the rule contribution to the final FIS performance. From the credit attributed to each rule it is possible to rank, select, and apply the recombination operators intrinsically to the EA. Here is where the Michigan approaches -- Michigan, Iterative Rule Learning, and Genetic Cooperative-Competitive Learning – differ from each other: in a lesser degree, due to the way each rule/individual is given credit to; in a higher degree, due to different mechanisms used to adapt and synthesise the rule base.

Final note: the most common metrics to evaluate a rule quality are the Confidence Degree (or Fuzzy Accuracy), and the Support (Cordón et al., 2001). The Confidence Degree, which applies exclusively to classification tasks, measures the intensity of the relation between an individual rule and the class, while the Support reflects the matching degree between the rule and the database for any sort of application. Finally, these approaches require the inclusion of mechanisms to avoid generating rules too similar in the same or the future generations in an EA run. In this sense, it is possible to penalise rules with the same activation profile or that classify the same patterns, or even to remove from the dataset already classified patterns (an example is provided in Berlanga, Rivera, Del Jesús, & Herrera (2010)).

Michigan Approach: Specific Aspects

In general, a Michigan-type EFS employs a reinforcement learning scheme, such that the EA is used for adapting the rules along the evolutionary process (Herrera, 2008; Kovacs, 2012). When the application requires a supervised learning process (e.g., classification, regression, and forecasting) it is necessary to replace the original form of the dataset that is supplied to the EFS: from the traditional batch-style to incremental feeding. Then, the rule base is randomly initiated and a pattern (or a small subset) is presented to the rule base, with the aim to identify which rule has promoted the correct classification or closest prediction. Since most applications reside in classification (Herrera, 2008; Fernández et al., 2010), the Support and Confidence Degree are also computed. With these metrics, it is possible to rank the individuals, select and apply the recombination operators. The synthesis of a FIS by using the Michigan-type approach is depicted in Figure 8.

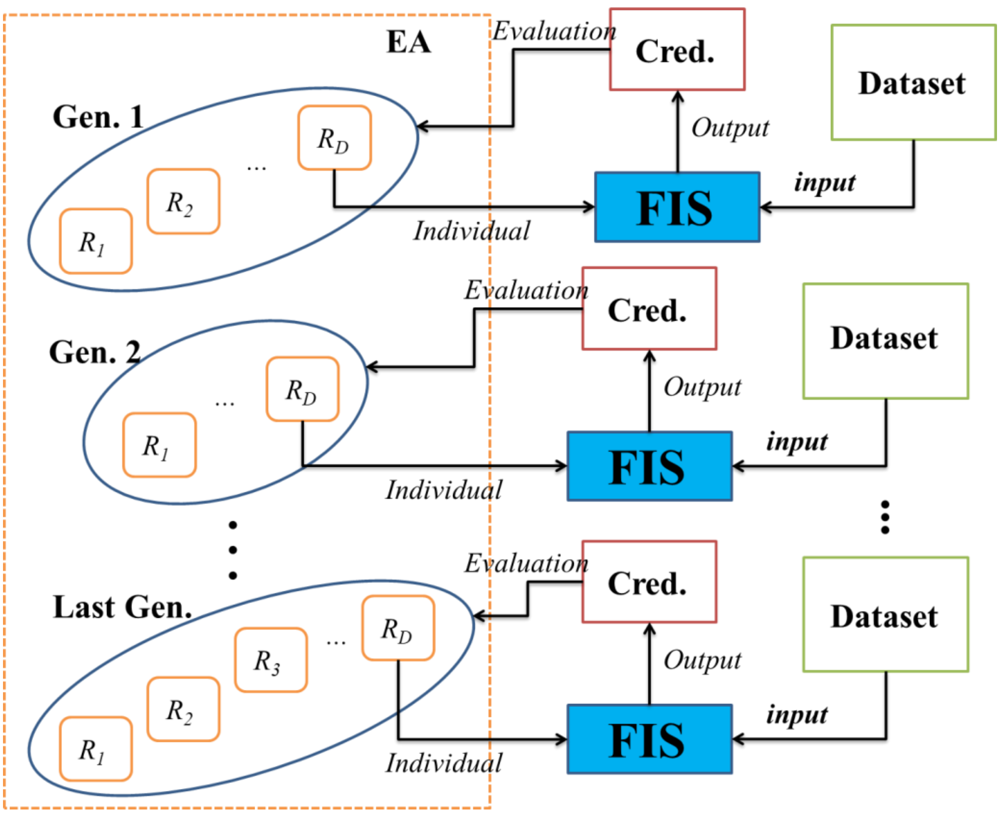


Figure 8. Global level description of the synthesising process of a Michigan-type EFS.

As new patterns are offered, rules compete between each other in such a manner that the most qualified – high confidence, support, and accuracy – remains during the evolutionary process. In the last generation, when all patterns have been shown, the final population forms the FIS's rule base. One of the pioneers in the EFS area, Valenzuela-Rendón (1991), presents a Michigan-type EFS devoted to classification tasks. One of the first books in EFS (Geyer-Schulz, 1997) uses the Michigan approach extensively to build fuzzy rule bases through the interplay with GP as the EA. One of the most prominent is from Casillas, Carse, & Bull (2007) that present the Fuzzy-XCS model, a fuzzy classifier that makes use of the Michigan approach to sets its structure. In this work the authors have revamped the Michigan-type methodology, showing new applications and directions, but keeping the reinforcement learning style. Marín-Blázquez and Pérez (2009) exhibits an application of the Fuzzy-XCS model for intrusion detection. Nojima, Watanabe, & Ishibuchi (2015) introduce two simple modifications, one related to rule generation, where each rule is generated from some multiple misclassified patterns to generate each fuzzy if-then rule. The other is related to the fitness calculation; they incorporate a penalty term into the fitness function based on the number of misclassified patterns. A similar principle is applied in Nojima, Takemura, Watanable, & Ishibuchi (2017), but using an (1+1)-Evolutionary Strategy for performing rule set optimization without losing their high computational efficiency.

In the literature review realised by Herrera (2008), the author verifies the necessity of the conception of new works in this area. However, in parallel, two new approaches derived from Michigan appeared: Iterative Rule Learning and Genetic Cooperative-Competitive Learning. Both do not follow the reinforcement learning approach, as well as some other minor elements. Next topics delve into the Iterative Rule Learning EFSs and Genetic Cooperative-Competitive Learning approaches.

Iterative Rule Learning Approach

In comparison with the Michigan-type, Iterative Rule Learning (IRL) is a more recent procedure (Cordón, & Herrera, 1997); the main idea is to execute the EA many times, aiming to obtain and store the best rule at each execution. A diagram of the whole process to obtain a full FIS rule base is shown in Figure 9.

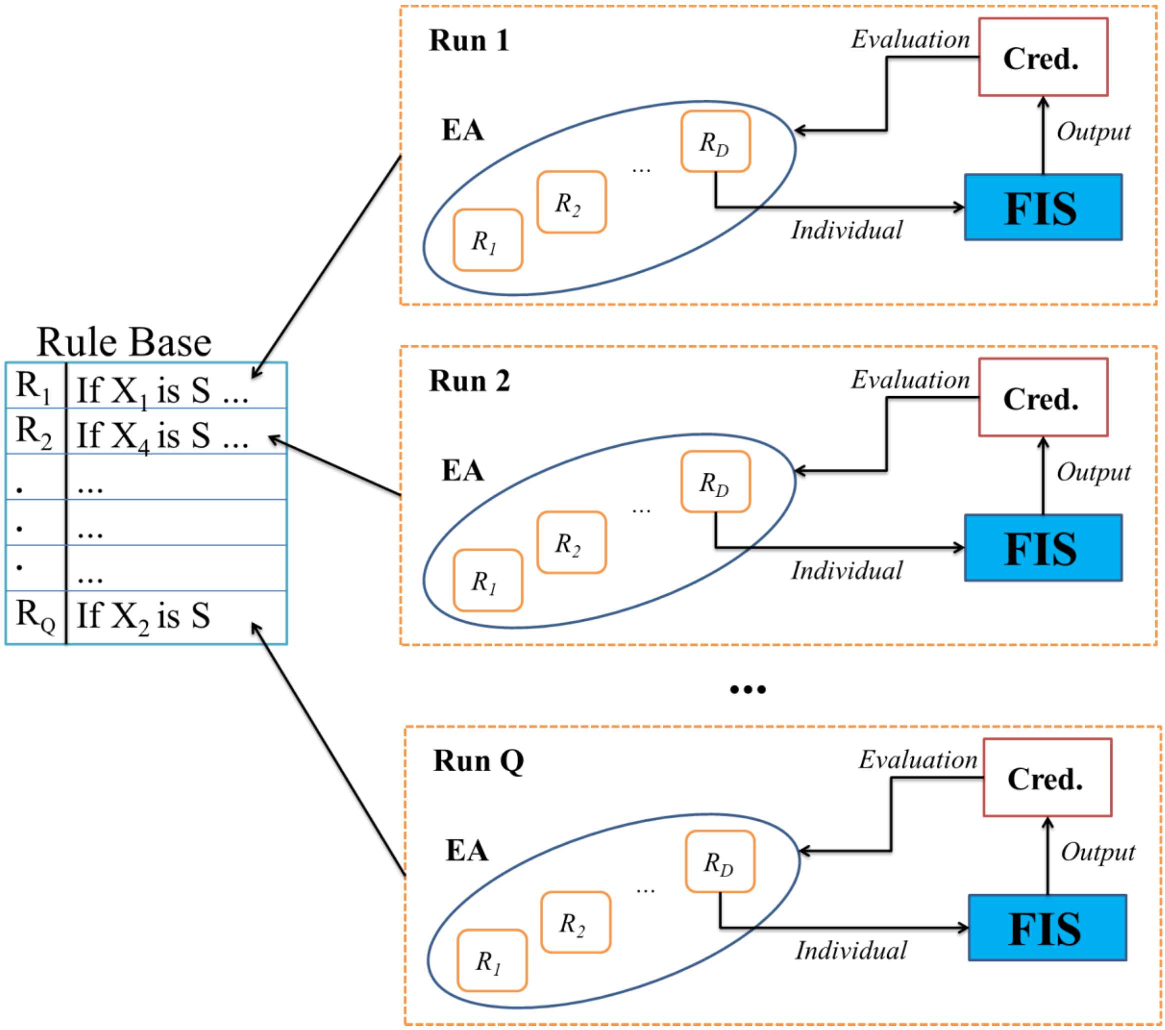


Figure 9. Synthesis of an IRL-type EFS.

The definition of the best rule maybe based on the Confidence Degree, Support, Coverage or other methods extensively presented in Cordón et al. (2001, p.226-231). There are many schemes available to set the stopping criteria, such as: generate as many rules such that they cover all available patterns or generate a minimum set of rules that covers a specific class (Cordón, 2001, p.232).

One of the first architectures found in the literature is the MOGUL (Methodology to Obtain Genetic fuzzy rule-based systems Under the iterative rule Learning approach), proposed by Cordón, Del Jesús, Herrera, & Lozano (1999). This work provided the basis for subsequent research following the IRL principles. The SLAVE (Structural learning algorithm on vague environment) model, developed by González and Pérez (1999) is capable of dealing with high-dimensional datasets. It employs a GA with a structure broken down in two parts: the first sets the relevance of a specific feature, while the second associates some characteristics (rule, granularity, etc.) with the first. Recently, Gárcia, González, & Pérez (2014a) have presented an improvement of González and Pérez (1999, 2009) work. This new model includes the possibility of generating fuzzy relational rules (joint modelling of antecedent terms), and creating new features through a linear combination of the original ones. As an assessment of this new model, the authors perform a study with 27 benchmark datasets for classification. An overview and review of the SLAVE methodologies can be found in García, González, & Pérez (2014b).

Genetic Cooperative-Competitive Learning

EFSs that abide by the Genetic Cooperative-Competitive Learning (GCCL) principles are more recent than the two others individual/rule type (Ishibuchi, Nakashima, & Murata, 1999). The core of this approach consists of a mechanism by which the rules compete and cooperate with each other; in consequence, this demands new ways to assess the rules quality, both individually and collectively (Fernández et al., 2010). Therefore, the GCCL style requires two fitness functions: a local one, that is used for selection and recombination purposes (competition), and a global function, which is used for evaluating the whole population across generations. The best population is always stored (cooperation). The synthesising process undergone by an EFS using the GCCL principles is outlined in Figure 10.

As in other approaches, some penalisation method concerning rule similarity is required to keep a diverse population. Another common trace is the variation of the population size in the evolutionary process. Berlanga et al. (2010) present a mechanism to expand/shrink the population size, based on the similarity level of rules, low Support and with the elaboration of auxiliary rules for patterns that were not covered by the current rules of the population.

From a historical perspective, the first work using a GCCL approach was that of Ishibuchi et al. (1999). It made use of a GA with fixed length representation for pattern recognition. A GCCL-type EFS for control tasks was developed by Juang, Lin, & Lin (2000), where rules are learned and membership functions are tuned; the consequent term can be of Mamdani or Takagi-Sugeno-Kahn types. Mucientes, Vidal, Bugarín, & Lama (2009) present an application of a GCCL-type EFS, with the EA being a context-free GP, to a dataset of machines from a furniture company. The GP-COACH model (Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems) was proposed by Berlanga et al. (2010) to deal with high-dimensional classification problems. It employs GP as the EA that underpins the EFS.

This same model was extended by López, Fernández, Del Jesus, & Herrera (2013) for environments in which the pattern recognition process is undermined due to the imbalance between classes. Palacios, Sánchez, & Couso (2011) proposed an EFS to extract rules from datasets in which the samples are imprecise or defined linguistically. Tsakiridis, Theocharis, & Zalidis proposed de DECO3R method, which uses Differential Evolution as its learning algorithm. In this frame, every chromosome encodes a single fuzzy rule. The proposed AdaBoost-based Fuzzy Token Competition method is employed to deal with the cooperation - competition problem, an integral part to all GCCL algorithms.

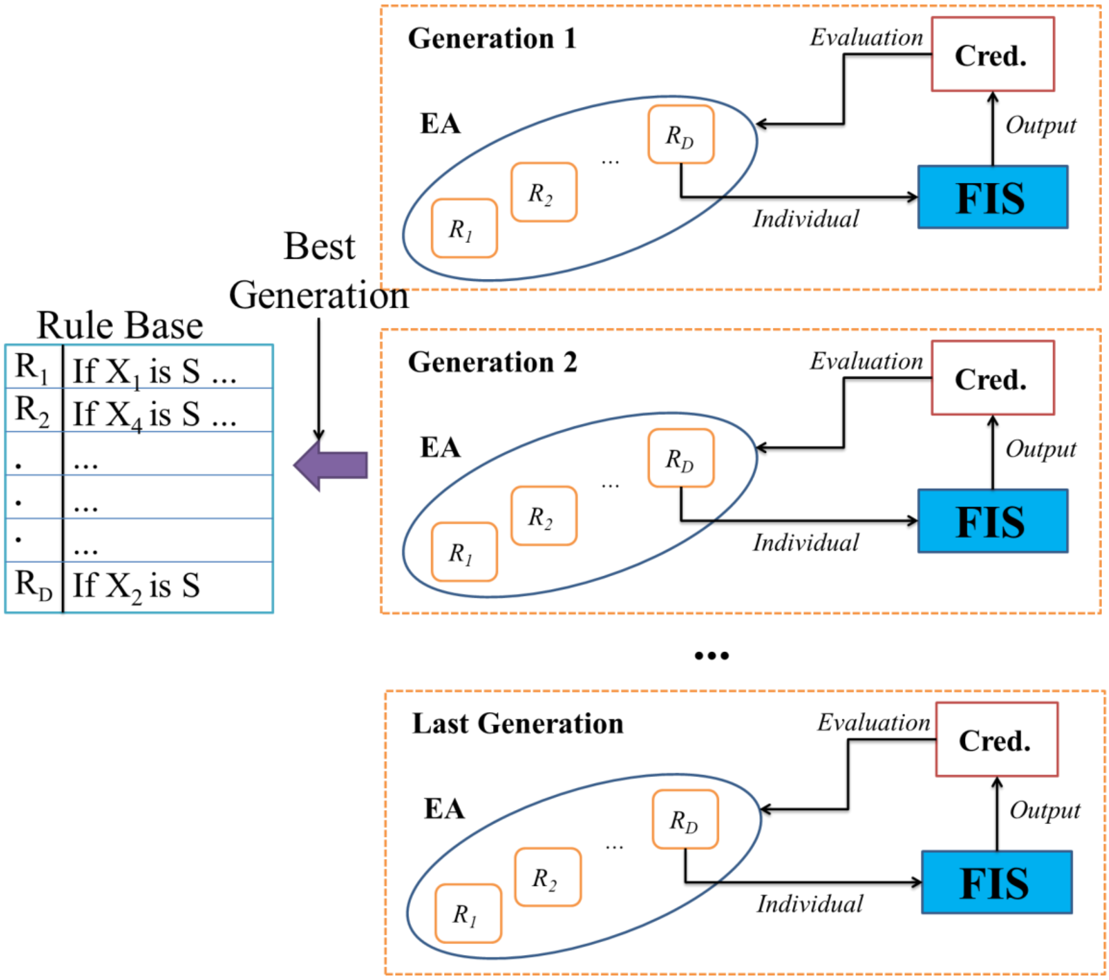


Figure 10. Synthesising process of a GCCL-type EFS.

Summary

Table 1 presents a general summary of the characteristics of each of the four previous Rule Base Learning approaches. The Michigan approach is the only one that taps into reinforcement learning. In a scenario where an EA and a FIS are the only architectures in hand, a Pittsburgh-type EFS tends to have low implementation cost, since the only remaining requirement is to set the evaluation function, while the remaining architectures would require more parameters (mechanism to remove/reduce similar rules, number of executions, etc.).

Table 1. Summary of the pros, cons and characteristics of the rule-based learning methodologies.

Type	Learning	COD	IC	RT	RAcc	RBC	Main App
Pittsburgh	Supervised	Ind=RB	Low	High	High	Low	Generic
Michigan	Reinforce	Ind=R	High	Low	Medium	High	Classification
IRL	Supervised	Ind=R	Medium	Medium	Medium	High	Classification
GCCL	Supervised	Ind=R	Medium	Low	Medium	High	Classification

Caption: COD – Rule base codification, Ind – Individual, RB – Rule Base, R – single rule, IC – Implementation cost, RT – Running time, RAcc – Relative accuracy when compared to other approaches, RBC – Rule base complexity, Main App –Main application.

It should be noted that the Pittsburgh approach requires more processing time than the other ones, while the Michigan approaches demands less, since in average these would only have the same cost per generation of one Pittsburgh-type individual. Ishibuchi et al. (2005) verify, from benchmark datasets, that the Michigan approach tends to generate a more compact rule base, albeit with lower accuracy when compared to Pittsburgh – a more up to date version of this model and results can be found in Lahsana and Seng (2017). The authors argue that this may be caused by the higher diversity of rules, which could be harmful to the joint behaviour of the final rule base. This evidence made the authors to propose a hybrid Pittsburgh-Michigan approach, obtaining good results from this enterprise.

Finally, based on references presented in the previous section as well as on literature reviews (Cordón et al., 2004; Herrera, 2008; Cordón, 2011; Fernández et al., 2010), it is possible to conclude that Michigan approaches are more prevalent in classification problems, with some few exceptions for control tasks. However, Pittsburgh-type EFSs are more generic, with applications in areas as regression, forecasting, control and classification problems.

Next section is devoted to a less explored area of the literature: Genetic Programming-based Evolutionary Fuzzy Systems. In summary, most of these systems are built to learn fuzzy rules; their main difference is centred on the design of the Genetic Programming algorithm, as detailed in the next section.

GENETIC PROGRAMMING-BASED EVOLUTIONARY FUZZY SYSTEMS

This section is devoted to present and discuss systems that use GP as the mean to learn fuzzy rule-based systems. After some initial background information on the historical and motivational elements of this subtopic, we exhibit and discuss the two main strands that we devised in this subarea: Context-free Grammar and Symbolic-centred approaches. Then, we close this section by contrasting both approaches and presenting some available gaps in this specific part of the literature.

Background

Overall, most EFSs are based on GAs (Cordón et al., 2004; Herrera, 2008), due to historical (started in the 70s) and practical reasons (availability of many implementations and abundant literature, a reasonable comprehension of its constituents, broad adoption by many practicioners, etc.). After the seminal work of Koza (1992), slowly, GP started to be adopted by the EFSs' community, due to its capacity to accommodate in a compact structure a dynamical representation such as a fuzzy rule base (Geyer-Schulz, 1997; Cordón et al., 2004). Figure 11 displays a comparative example of a fuzzy rule codified by a GA with integer representation and a Grammar-based GP, for a problem with 10 input variables.

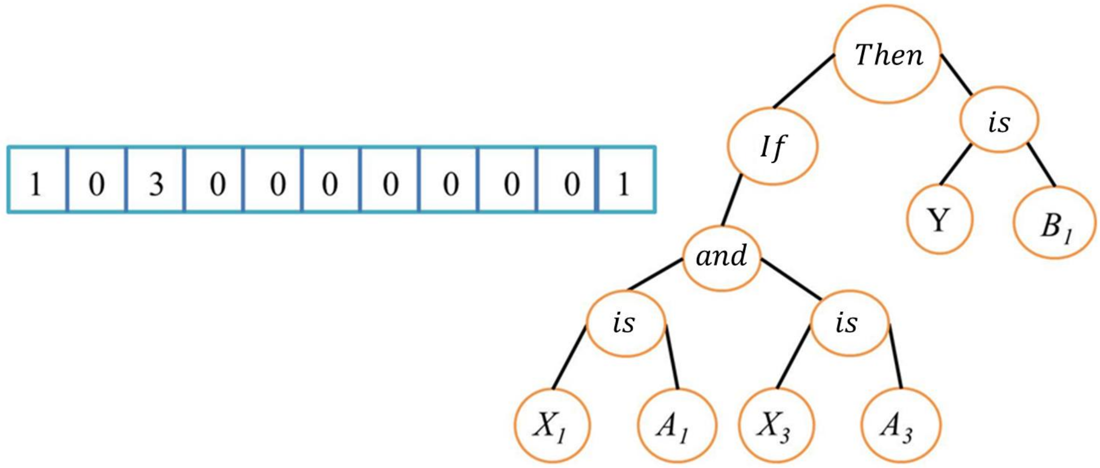


Figure 11. Example of a fuzzy rule created using a GA with its counterpart GP, respectively.

Each representation expresses a fuzzy rule that can be read as "If X_1 is A_1 and X_3 is A_3, then Y is B_1". However, it is clear that the GP yield a simplified representation, with the zeros/don't care symbol being redundant yet important in the GA codification. This difference is magnified in high-dimensional scenarios with its implication in higher processing time for evaluation, selection and recombination. Beyond this advantage in expressing and evolving a fuzzy rule, many empirical studies suggests that GP-based EFSs outperforms other EFSs in terms of accuracy and interpretability metrics (Berlanga et al., 2010; Tsakonas, 2013; Koshiyama, Vellasco, & Tanscheit, 2016}.

Scanning the literature of GP-based EFSs it is possible to group the approaches in two categories: Context-Free Grammar and Symbolic-centred. Both are grounded on which type of GP is being used, or more specifically, how the fuzzy rules are being codified and post-processed to elaborate a fuzzy rule-based system. Our previously example (Figure 11) is rooted with the preferred route in the literature, that is, the Context-free Grammar based approach; the seminal work of Koza (1992) and many other Symbolic Regression works represent the solution based on the second category (Symbolic-centred). The next two subsections describe in detail both categories.

Context-free Grammar approaches

The Context-free Grammar scheme uses the GP as a means to manufacture fuzzy rules that follow a certain structure and comply to a set of constraints. In general, the rules generated by this approach are the most straightforward to analyse and to incorporate into a fuzzy inference system – since very little, or even nothing, is changed in its reasoning method when models following this procedure are used. The main component that uniquely identify the members of this family is the Context-free Grammar that is passed to the GP, and its capacity to use it in order to create rules following a certain pattern. Table 2 presents an example of Context-free Grammar that can be used by the GP to manufacture fuzzy rules.

Table 2. Example of Context-free Grammar, similar to that of Berlanga et al. (2010).

➢ Start -> [If], antec, [then], conseq, [.]
➢ antec -> descriptor 1, [and], descriptor 2
➢ descriptor 1 -> [any]
➢ descriptor 1 -> [X_i is] label
➢ descriptor 2 -> [any]
➢ descriptor 2 -> [X_j is] label
➢ label -> {member(?a, [L, M, H, L or M, L or H, M or H, L or M or H])}, [?a]
➢ conseq -> [Class is] descriptorClass
➢ descriptorClass -> {member(?a, [C_1, C_2, ..., C_k])}, [?a]

Though we could cite some previous works (Sánchez et al., 2001; Tsakonas, 2006), the most illustrative model in this subarea is the so-called GP-COACH (Genetic Programming Based Learning of Compact and Accurate Fuzzy Rule-Based Classification System for High Dimensional Problems). This GP-based EFS was proposed by Berlanga et al. (2010) in an extension of a previous research (Berlanga, Del Jesus, & Herrera, 2005). This EFS is oriented towards rule base generation following a GCCL-scheme. More specifically, this model implements the following elements in addition to the ordinary components of an EFS:

- GP-COACH allows the creation of rules in a DNF (Disjunctive Normal Form) format – an example is: "If X_1 is A_1 or A_2 and X_2 is A_3 or A_4 and ...", with t-conorms joining multiple linguistic terms represented by triangular-shaped membership functions. It implements the Context-free Grammar displayed in Table 2, generating rules akin to Figure12.

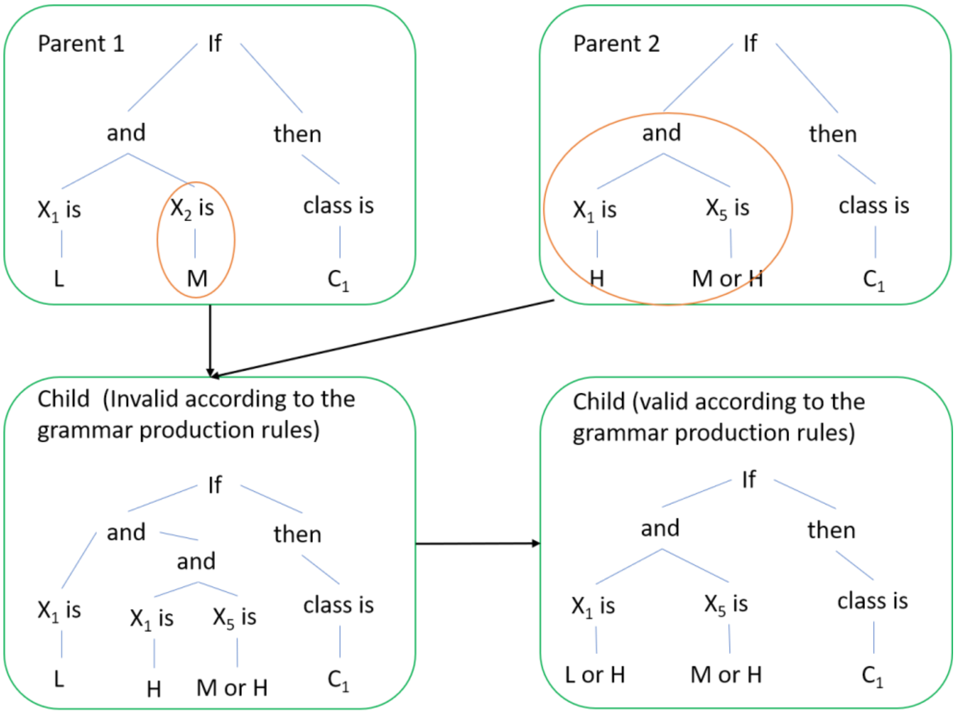


Figure 12. Example of GP-COACH DNF fuzzy rule type.

- Since a GCCL-scheme is being used, two evaluation functions are setup:
    - Local: evaluate the quality of rule (individual) through the Support and Confidence metrics.
    - Global: evaluate the rule base (population) performance, taking into account the accuracy and complexity of the fuzzy rules.
- It employs new mechanisms for: (i) evaluation — adding diversity criteria during the assessment process; (ii) recombination — generating secondary rules to cover unclassified patterns; and (iii) selection — adding intergenerational competition to amplify/reduce certain genotype in a population. In total, 24 benchmark classification datasets were used; the results suggest that this model outperforms other four EFSs in accuracy and complexity terms. This model was also used by López et al. (2013), with the inclusion of an oversampling procedure and a 2-tuple representation and fine-tuning in a post-processing stage, to solve highly imbalanced classification tasks.

Beyond GP-COACH, some other proposals have been developed in this subarea, like:

- Tsakonas (2006): the author investigates the effectiveness of GP-generated intelligent structures in classification tasks. More importantly, the author implemented a Grammar Guided GP-based EFS, following a Pittsburgh-style structure. Nothing in special was added in terms of diversity increasing operators, or different manners to evaluate an individual. The models were evaluated thoroughly in six well-known real world data sets.
- Muni and Pal (2012): the authors propose a Pittsburgh-style Multigene/Multi-tree GP-based EFS. Each individual is composed of *K* trees, where *K* represents the number of classes. Each tree expresses a set of fuzzy rules linked with a specific class, merging this spontaneous rule generating scheme with some created via a C-Means algorithm. They have used Gaussian-type membership functions, and have included new recombination operators to synthesise solutions. To evaluate their method, they have used five datasets and compared the results found with Lim, Loh, & Shih (2000). In a way, this work is a Context-free Grammar version of a later work done by Koshiyama et al. (2013), but without the addition of the C-Means algorithm, the exclusion of some operators and the inclusion of more benchmark datasets.
- Tsakonas (2013): presents a method named MEMFIS (MEMetic genetic programming Fuzzy Inference System) to incorporate standard Neuro-Fuzzy learning for Takagi–Sugeno fuzzy systems that evolve under a Grammar-driven GP framework. MEMFIS incorporates local-search training methods for fuzzy systems, namely backpropagation gradient descent for the linguistic variables and recursive least squares for the coefficients of the linear functions. In addition to local search, complete fuzzy rule bases evolve under a GP framework with a context-free grammar guidance. MEMFIS effectively generates and trains first-order Takagi–Sugeno fuzzy systems for regression and control. This new approach is used in regression and control benchmarks, such as the inverted pendullum, comparing favourably to its peers.
- Carmona et al. (2015): presents a new approach named Fuzzy Genetic Programming-based for Subgroup Discovery: FuGePSD. This algorithm represents an EFS based on GP, employing

the GCCL approach where rules of the population cooperate and compete between them to obtain optimal solution. FuGePSD employs several genetic (mutation, crossover, insertion and drop) and selection (token competition and screening) operators in order to obtain rules that are as general and precise as possible in describing new information of the search space. In this way, FuGePSD includes an operator to promote the diversity at genotype level, where rules describing the same examples are penalised. FuGePSD displays its potential with high-quality results in a wide experimental study performed with respect to others evolutionary algorithms for subgroup discovery. Moreover, the quality of this proposal is applied to a case study related to acute sore throat problems.

Symbolic-centred approaches

The Symbolic-centred approaches are marked by the use of the Koza-style GP, very similar to the way other researchers in the area perform Symbolic Regression. Instead of defining a Context-free grammar that will be manipulated and enhanced along the evolutionary process, the user have to set the Terminals and Functions before starting the evolution (McPhee, Poli, & Langdon, 2008). The Terminals are the set of features that have been already mapped into fuzzy sets – differing from the traditional form where the features are included in their raw version. The Functions are, in some implementations, constrained to certain mathematical operations that guarantee interpretability and act as t-norms, t-conorms and linguistic modifiers – like product, minimum, square-root, etc.

By establishing both sets, it is possible to execute the GP similarly to another Symbolic Regression run, with some additional components depending on the type of the fuzzy system being optimised. Before outlining the major representant of the Symbolic-centred family -- the Genetic Programing Fuzzy Inference System (GPFIS) model --, we first highlight some previous works in the literature:

- Chien et al. (2002): The authors propose a new learning approach based on genetic programming to generate discriminant functions for classifying data. An adaptable incremental learning strategy and a distance-based fitness function are developed to improve the efficiency of genetic programming-based learning process. They first transform attributes into membership degrees of fuzzy sets and then a set of discriminant functions is generated based on the proposed learning procedure. The set of derived functions with fuzzy attributes gives high accuracy of classification and presents a linear form (such as the one exemplified below for the Iris dataset).

  $\mathrm{IF}(\mathrm{sl\_S}+\mathrm{sw\_L}\geq\mathrm{pl\_M}+\mathrm{pl\_L})$

  THEN setosa.

  $\mathrm{IF}(29\times\mathrm{pw\_M}+71\times\mathrm{pl\_M}\geq 67+\mathrm{sl\_M})$

  THEN versicolor.

  $\mathrm{IF}(2\times\mathrm{pw\_L}+\mathrm{pl\_L}\geq\mathrm{sw\_L}+\mathrm{sl\_L}+\mathrm{pl\_M})$

  THEN virginica.

The first expression can be read as: "If Sepal Length is Small or Large enough in relation to the size of the Petal Length, then the plant is from the Setosa species". These rules are more challenging to read in comparison to the one from GPFIS model, as will be described below, nonetheless they can be transformed into inference rules in an expert system.

- Koshiyama et al. (2013): the authors present the so-called Genetic Programming Fuzzy Classification System (GPF-CLASS). This model differs from the traditional approach of GP-based EFSs, which uses the metaheuristic to learn "if-then" fuzzy rules (Context-free Grammar). This "if-then" classical approach needs several changes and constraints on the use of genetic operators, evaluation and selection, which depends primarily on the metaheuristic used. Genetic Programming makes this implementation costly and explores few of its characteristics and potentialities. The GPF-CLASS model seeks for a greater integration with the metaheuristic: Multi-Gene Genetic Programming, exploring its potential of terminals selection (input features) and functional form and, at the same time, aims to provide the user with a more straightforward comprehension of the classification solution. Below it is possible to identify an example of a fuzzy rule induced by the GPF-Class for the Iris dataset:

  $$Setosa = \mu_M(SL)$$

  $$Versicolor = \mu_{VL}(PL)$$

  $$Virginica = 2.0\mu_{VH}(PL) + \mu_H(PW) - \mu_L(PL) - \mu_M(PL) - \mu_{VL}(PL) - 2.0\mu_M(SL) - 2.0\mu_H(SL)$$

  For the first equation, the more Medium the sepal length is for a given pattern, the higher is its membership degree to the Setosa species. The second equation expresses that: If a plant has a Very Little Petal Length, then its compatibility with Versicolor species is higher. Then, for these two equations, the discriminant function generate by GPF-CLASS is very simple and informative. The third equation offsets the observed simplicity, providing a discrimination function harder to interpret and build an explanation for.

A model that can may be regarded as an archetype of the Symbolic-centred approach is the Pittsburgh-type EFS called Genetic Programing Fuzzy Inference System (GPFIS). Due to the addition of new components to post-process a set of fuzzy antecedents in a fuzzy rule base, its implementation is more complex than that of a typical Pittsburgh-style EFS (contrasting with the characteristics enunciated in Table 1). Another characteristic that contributes to a higher complexity is its generality, since it can be used for different tasks: GPFIS-CLASS for pattern recognition (Koshiyama, Vellasco, & Tanscheit, 2015a), GPFIS-Regress for regression tasks (Koshiyama, Vellasco, & Tanscheit, 2016), GPFIS-Forecast for time-series modelling (Koshiyama, Vellasco, & Tanscheit, 2015b) and GPFIS-Control for the automatic design of fuzzy controllers (Koshiyama, Vellasco, & Tanscheit, 2014). The main modules of the general GPFIS model are shown in Figure 13.

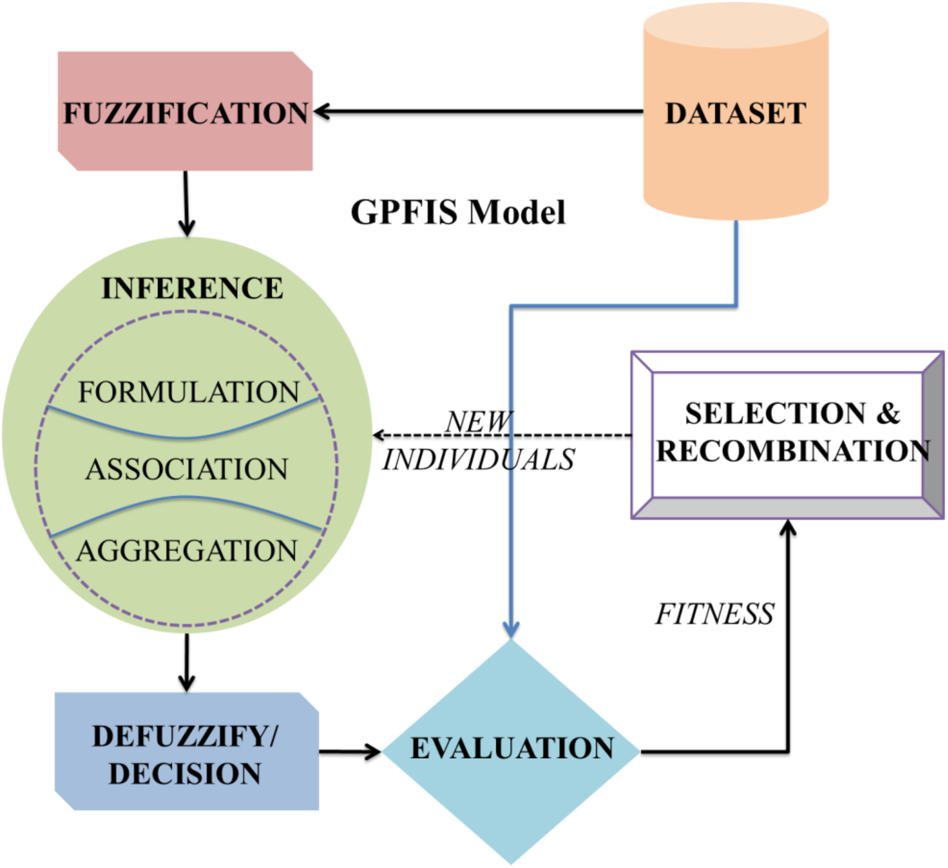


Figure 13. Main stages of the GPFIS model.

Modelling begins by mapping crisp values into membership degrees of fuzzy sets (Fuzzification). Then, a fuzzy inference procedure is performed in three subparts: (*i*) generation of fuzzy rules premises (Formulation); (*ii*) assignment of the best suited consequent term to each premise (Association) and (*iii*) aggregation of activated fuzzy rules (Aggregation). Finally, Decision, Evaluation and Selection & Recombination are performed. The following steps briefly summarises these components:

- **Fuzzification:** The specification of fuzzy sets involves the definition of three factors: (*i*) functional description (triangular, trapezoidal, etc.); (*ii*) support and granularity of membership functions; and (*iii*) linguistic terms, to qualify the subspace defined by the membership function with an appropriate label. In theory, this should be specified by an expert. In practice, however, due to the difficulty of having an expert available, membership functions are usually defined as strongly partitioned.

- **Fuzzy Inference:** The inference process in the GPFIS model is subdivided into 3 steps: (*i*) Formulation — responsible for combining the linguistic terms of each feature to build a fuzzy rule premise (antecedent creation, using Multi-Gene Genetic Programming); (*ii*) Association — given a set of premises, this step verifies the consequent term that is most suited to each premise (fuzzy rule creation and screening); and (*iii*) Aggregation — receives as input the activated fuzzy rules and computes a consensual value for each consequent output. Figure 14 illustrates this whole process described in detail in the following sub-sections.

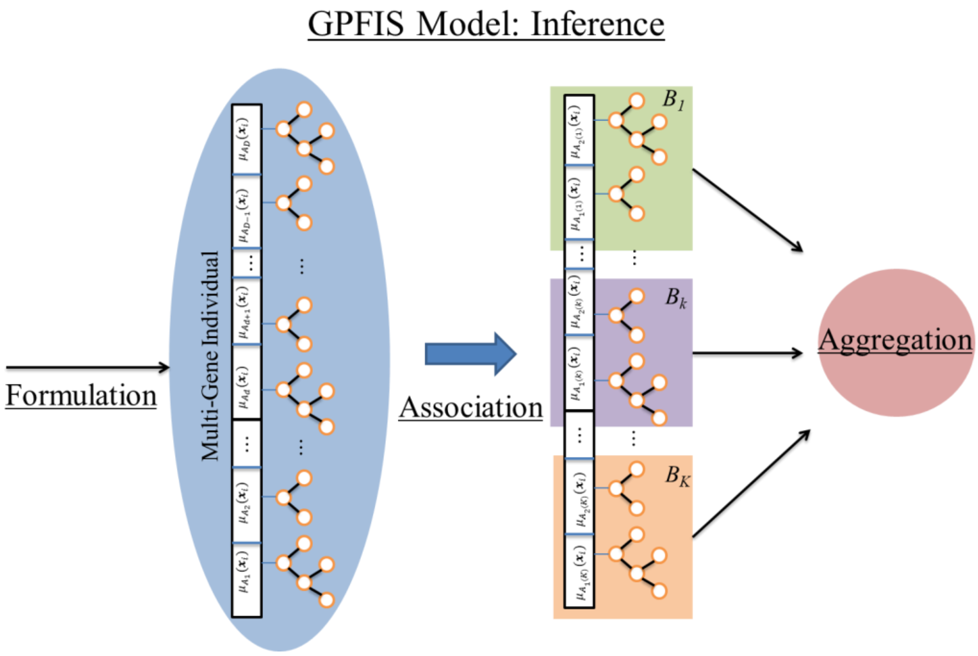


Figure 14. Diagram describing the main stages of GPFIS model Inference process.

i. *Formulation:* The GPFIS model makes use of Multi-Gene Genetic Programming (Hinchliffe et al., 1996; Gandomi & Alavi, 2012) to obtain a set of fuzzy rules premises. A fuzzy rule premise is commonly defined as: "If X_1 is A_{l1}, and ..., and X_j is A_{lj}, and ..., and X_J is A_{lJ}". Alternatively, in mathematical terms: μ_{Ad} ($\mathbf{x}_i$) = μ_{Al1} (x_{i1}) * ... * μ_{Alj} (x_{ij}) * ... * μ_{AlJ} (x_{iJ}), where μ_{Ad} ($\mathbf{x}_i$) is the joint membership degree of pattern $\mathbf{x}_i$ concerning the d-th premise (d=1,...,D), computed through a t-norm "*" that combines each membership degree μ_{Alj} (x_{ij}). Table 3 present the Terminals and Functions that characterises this Symbolic-centred approach.

Table 3. Terminals and Functions for a generic GPFIS model.

Terminals (Fuzzy Sets)	Functions (Fuzzy Operators)
μ_{A11} (x_{i1}), μ_{A21} (x_{i1}), …, μ_{AL1} (x_{i1}), …, μ_{ALJ} (x_{iJ})	Product, minimum, maximum, square-root, etc.

In general terms, a fuzzy rule premise can be described as a combination of each μ_{Alj} (x_{ij}) (Terminals) by using t-norms, t-conorms, negation and linguistic hedge operators (Functions). However, the search space grows exponentially as the number of features and available operators increases. Multi-Gene Genetic Programming (MGGP) is employed to deal with this large search space. Figure 15 presents an example of a solution provided by an individual of an MGGP population; premise 1 is analytically represented by: μ_{A1} ($\mathbf{x}_i$) = μ_{A21} (x_{i1}) * μ_{A32} (x_{i2}), which denotes, in linguistic terms: "If X_1 is A_{21}, and X_2 is A_{32}".

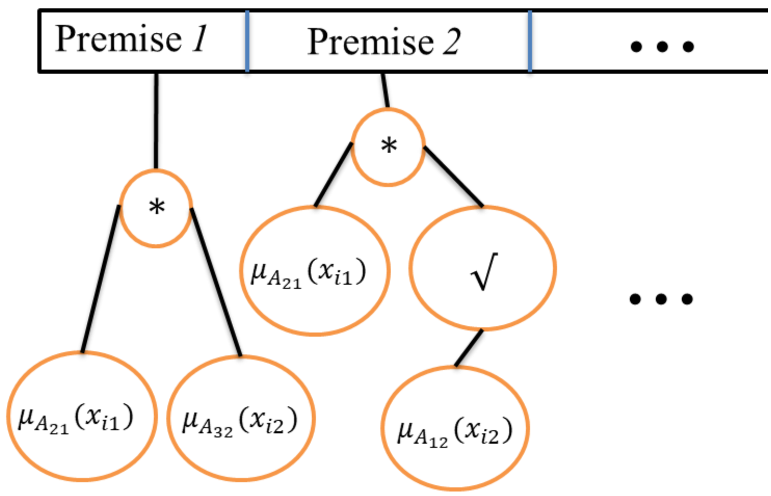


Figure 15. Set of premises encoded by a Multi-Gene Genetic Programming individual.

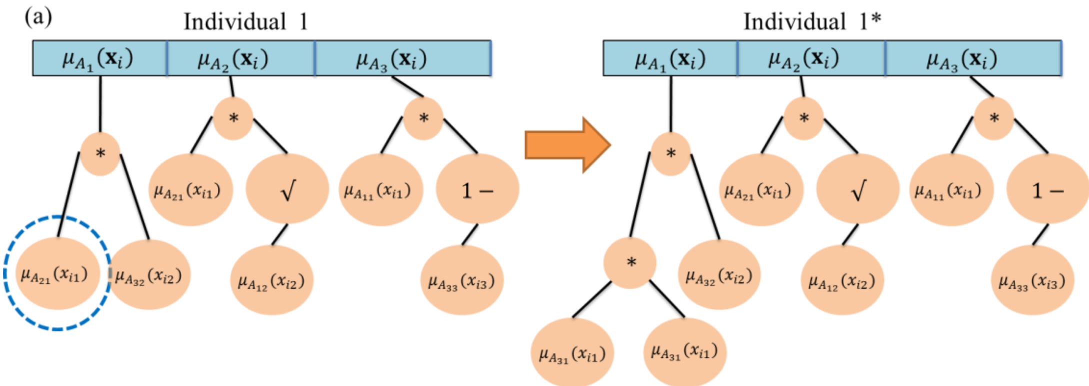
ii. *Association*: In the association step, the most compatible consequent term with a given premise is determined. A simple and intuitive association technique is the so-called Uniform Division. This method involves splitting the premises in *K* slots of equal size and associating them to a specific *k*-th consequent term; this approach is widely used in GPFIS-Control. The use of this simple method, however, may imply in the association of a premise to an incorrect consequent term. To avoid this and promote the reduction of the search space, each premise should be assigned to the k-th consequent term that maximises a particular compatibility measure between the specific premise and a consequent term. In classification, a straightforward and widely used compatibility measure is the Confidence (Certainty) Degree (Berlanga et al., 2010), while for regression and forecasting the Fuzzy Confidence Degree (Koshiyama, Vellasco, & Tanscheit, 2016) may be used.

iii. *Aggregation:* In a fuzzy inference system, an input pattern may activate several rules related to different consequents. The Aggregation step aims to merge the activation degrees over fuzzy rules related to a same consequent to generate a consensual output. The most commonly used aggregation operator is the maximum t-conorm (Pedrycz & Gomide, 1998) – extensively employed in GPFIS-Control. An alternative would be the weighted mean (Beliakov & Warren, 2001), where the weights are computed by solving a Restricted Least Squares problem – used more for classification, regression and forecasting tasks. Once the merged membership degree for each consequent term has been computed, it is necessary to verify which class a pattern belongs to or apply a defuzzification method to generate an output of a controller or a forecasting system.

- **Decision/Defuzzification***:* This step converts a set of fuzzy activations in the consequent terms into a single and crisp output. In classification problems, this may be called decision; in the remaining problems, defuzzification is a better term. For classification problems, the decision is made by the class to which the input pattern has maximum expected membership; when a tie occurs, either a heuristic can be applied (the class that has more patterns in the dataset), or no class is assigned at all. For the remaining tasks, defuzzification is performed through the Height Method (Roychowdhury & Pedrycz, 2001), due to the widespread use of strongly partitioned fuzzy sets in the experiments with GPFIS models. For control tasks, sometimes the Mean of Maximum or the Center of Gravity defuzzification methods may provide a better performance (Roychowdhury & Pedrycz, 2001).

- **Evaluation:** Evaluation in GPFIS model involves two objectives: 1) maximising performance (accuracy in classification, the difference to the setpoint in control, etc.); and 2) reducing the rule base complexity. The first one is responsible for ranking the individuals in the population, while the second objective is used as a tiebreaker. Reduction of rule base complexity is performed through a simple heuristic called Lexicographic Parsimony Pressure (Luke & Panait, 2002). This is employed only when two individuals have the same fitness. Therefore, given two individuals with the same fitness, the best one is the individual with fewer nodes. Fewer nodes indicate rules with fewer antecedent elements, linguistic hedges and negation operators, as well as few premises, resulting in a more compact fuzzy rule set.

- **Selection & Recombination:** After the evaluation process, a set of individuals is selected (through a tournament procedure) and recombined. Mutation (Figure 16a), low-level crossover (Figure 16b) or high-level crossover (Figure 16c) are applied to some subset of individuals. Finally, the new population is generated. This process continues until a stopping criterion – the number of evaluations, in the context of GPFIS -- is met; then, the last population is returned.



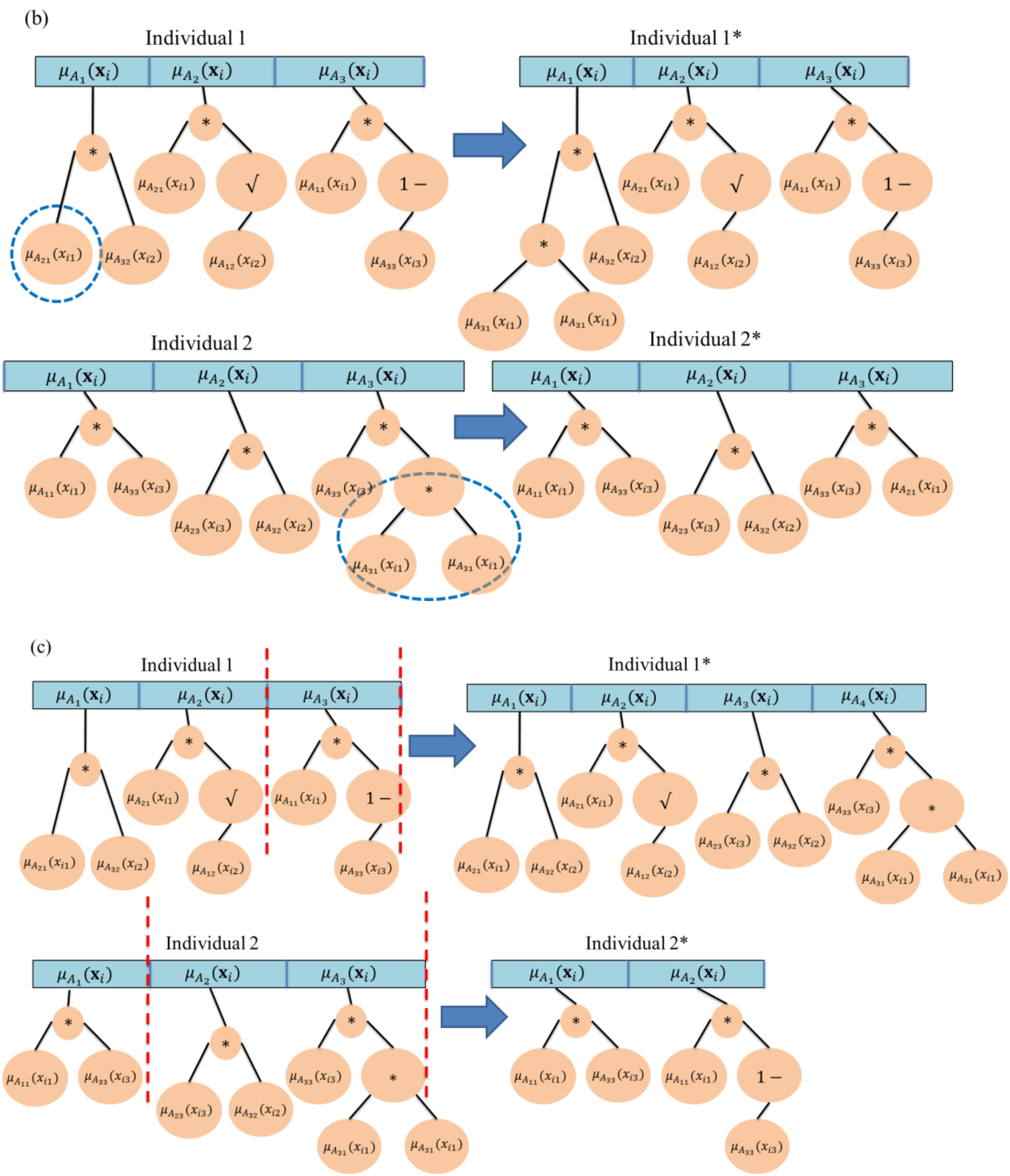


Figure 16. Example of recombination operators applied in GPFIS model solutions.

Summary

This section presented an overview of GP-based EFS, focusing on the historical evolution and the two main strands of research: Context-free Grammar and Symbolic-centred approaches. Context-free Grammar systems make better use of some inherent flexibility of the way GP codifies a solution, being more common in the literature. However, as pointed out in the example of GP-COACH, the proposed model might generate some invalid solutions according to the grammar established, requiring some specific post-operation to fix them.

The main characteristics of the Symbolic-centred approach when compared to the Context-free are: (i) the way the elements used to build the fuzzy rules are declared, based on terminals and functions instead of specifying a grammar; (ii) the constraints added to enforce the breeding of rules following a specific structure; (iii) the GP implementation is simplified because the Symbolic-centred approach uses the Koza-style GP; and (iv) the post-processing that is required to build up the inference system and analyse the rules. The GPFIS model is the most interpretable one among all Symbolic-centred approaches, though its rules may not be as straightforward to read as in the Context-free Grammar methods. On the other hand, the performance of Symbolic-centred approaches tends to be superior to that of Context-free approaches. The results obtained with GPFIS corroborate this.

Overall, GP-based EFSs have focused primarily on learning fuzzy rules and classification tasks. Future research could concentrate on other areas of an EFS modelling other than Rule Base Learning (Figure 6), such as Granularity Learning, Membership Functions Fine-Tuning, etc. GP-based EFSs' research could devote more time to solve forecasting and regression problems.

Conclusion

This overview has presented the main concepts of Evolutionary Fuzzy Systems, such as terminologies, application realms, different facets of an EFS (Parameter Tuning, Knowledge Discovery) and usual rule base learning schemes. We give a special focus on Genetic Programming-based EFSs, by providing a categorisation that can cluster the main architectures available, as well as by pointing out the gaps that still exist in the literature. It is hoped that this will help readers to navigate through the area and find the works most linked to this particular research topic.

Though not specifically covered in this overview, the reader may be interested in some other topics around automatic synthesis of Fuzzy Inference System, such as:

- Interpretability: this is an important aspect of any FIS, since the fuzzy rules can help understand the relation between the inputs and the final inference result (e.g. the output class). Some special journal editions and book chapters are dedicated to debating this theme, aiming to clarify the notion of interpretable rule, touching concepts like complexity, semantics, constraints, etc., as well as proposing new metrics to assess the interpretability degree of a Fuzzy Inference System (Alonso & Magdalena, 2011; Casillas, Cordón, Triguero, & Magdalena, 2013; Cpalka, 2017).

- Multi-Objective Evolutionary Fuzzy Systems: although this topic is not new (Ishibuchi et al., 1995), this area has been experiencing a steady revival, perhaps due to the many available implementations of Multi-Objective Evolutionary Algorithms (e.g., NSGA-II, SPEA2, MOEA-D, etc.), as well as the growth of improved ways to assess interpretability (Fazzolari et al., 2013; Ishibuchi & Nojima, 2015).

- Evolving Fuzzy Inference Systems: this is another whole strand for automatic synthesis of Fuzzy Inference Systems (Lughofer, 2011). In general, these systems are designed to learn fuzzy rules, as well as the right granularity, based on recursive rules and non-evolutionary optimisation procedures. This area can be roughly broken down into two threads: those clustering-based procedures to elaborate these systems (Angelov & Zhou, 2008; Lima, Hell, Balini, & Gomide, 2010) and partitioning-combination techniques to elaborate fuzzy rule bases (Coutinho, Vellasco, Tanscheit, & Koshiyama, 2016; Paredes, Vellasco, Tanscheit, & Koshiyama, 2016), some of them dedicated to Big Data Problems (Samudio, Vellasco, Tanscheit, & Koshiyama, 2016).

Acknowledgments

The authors would like to thank the Brazilian agencies, CNPq and FAPERJ, for the financial support to develop this work.

References

Abraham, A. (2001). Neuro fuzzy systems: State-of-the-art modeling techniques. *Connectionist models of neurons, learning processes, and artificial intelligence*, 269-276.

Alba, E., Cotta, C., & Troya, J. M. (1996, July). Type-constrained genetic programming for rule-base definition in fuzzy logic controllers. In *Proceedings of the 1st annual conference on genetic programming* (pp. 255-260). MIT Press.

Alcalá, R., Gacto, M. J., & Herrera, F. (2011). A fast and scalable multiobjective genetic fuzzy system for linguistic fuzzy modeling in high-dimensional regression problems. *IEEE Transactions on Fuzzy Systems*, *19*(4), 666-681.

Alcalá-Fdez, J., Herrera, F., Márquez, F., & Peregrín, A. (2007). Increasing fuzzy rules cooperation based on evolutionary adaptive inference systems. *International Journal of Intelligent Systems*, *22*(9), 1035-1064.

Alcalá, R., Nojima, Y., Herrera, F., & Ishibuchi, H. (2011). Multiobjective genetic fuzzy rule selection of single granularity-based fuzzy classification rules and its interaction with the lateral tuning of membership functions. *Soft Computing*, *15*(12), 2303-2318.

Alcalá, R., Nojima, Y., Ishibuchi, H., & Herrera, F. (2012). Special issue on evolutionary fuzzy systems. *International Journal of Computational Intelligence Systems*, *5*(2), 209-211.

Alcalá, R., Nojima, Y., Ishibuchi, H., & Herrera, F. (2013). Special Issue on Evolutionary Fuzzy Systems EFSs. *Knowledge-Based Systems*, 54(C), 1-2.

Alonso, J., Magdalena, L. (2011). Special issue on interpretable fuzzy systems. *Information Sciences*, 181(20), 4331-4339.

Angelov, P. P., & Zhou, X. (2008). Evolving fuzzy-rule-based classifiers from data streams. *IEEE Transactions on Fuzzy Systems*, 16(6), 1462-1475.

Arslan, A., & Kaya, M. (2001). Determination of fuzzy logic membership functions using genetic algorithms. *Fuzzy sets and systems*, *118*(2), 297-306.

Beliakov, G., & Warren, J. (2001). Appropriate choice of aggregation operators in fuzzy decision support systems. *IEEE Transactions on Fuzzy Systems*, *9*(6), 773-784.

Berlanga, F., Del Jesus, M. J., & Herrera, F. (2005). Learning fuzzy rules using genetic programming: context-free grammar definition for high-dimensionality problems. In *Proc. of the I Workshop on Genetic Fuzzy Systems* (pp. 136-141).

Berlanga, F. J., Rivera, A. J., del Jesús, M. J., & Herrera, F. (2010). GP-COACH: Genetic Programming-based learning of COmpact and ACcurate fuzzy rule-based classification systems for High-dimensional problems. *Information Sciences*, *180*(8), 1183-1200.

Marín-Blázquez, J. G., & Pérez, G. M. (2009). Intrusion detection using a linguistic hedged fuzzy-XCS classifier system. *Soft computing*, *13*(3), 273-290.

Brito, V., Vellasco, M., & Tanscheit, R. (2012). Fuzzyfuture: ferramenta de previsão de séries temporais baseada em sistema híbrido fuzzy-genético. *Learning and Nonlinear Models*, 10, 99-108.

Carmona, C. J., Ruiz-Rodado, V., del Jesús, M. J., Weber, A., Grootveld, M., González, P., & Elizondo, D. (2015). A fuzzy genetic programming-based algorithm for subgroup discovery and the application to one problem of pathogenesis of acute sore throat conditions in humans. *Information Sciences*, 298, 180-197.

Carse, B., Fogarty, T. C., & Munro, A. (1996). Evolving fuzzy rule based controllers using genetic algorithms. *Fuzzy sets and systems*, *80*(3), 273-293.

Casillas, J., Cordón, O., Del Jesus, M. J., & Herrera, F. (2005). Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Transactions on Fuzzy Systems*, *13*(1), 13-29.

Casillas, J., Martínez, P., & Benítez, A. D. (2009). Learning consistent, complete and compact sets of fuzzy rules in conjunctive normal form for regression problems. *Soft Computing*, *13*(5), 451-465.

Casillas, J., Cordón, O., Del Jesus, M. J., & Herrera, F. (2005). Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction. *IEEE Transactions on Fuzzy Systems*, *13*(1), 13-29.

Casillas, J., Carse, B., & Bull, L. (2007). Fuzzy-XCS: A Michigan genetic fuzzy system. *IEEE Transactions on Fuzzy Systems*, *15*(4), 536-550.

Casillas, J., Cordón, O., Triguero, F. H., & Magdalena, L. (Eds.). (2013). *Interpretability issues in fuzzy modeling* (Vol. 128). Springer.

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

Cintra, M. E., Camargo, H. A., & Monard, M. C. (2016). Genetic generation of fuzzy systems with rule extraction using formal concept analysis. *Information Sciences*, 349, 199-215.

Cordón, O., & Herrera, F. (1997). 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.

Cordón, O., del Jesús, M. J., Herrera, F., & Lozano, M. (1999). MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach. *International Journal of Intelligent Systems*, *14*(11), 1123-1153.

Cordón, O., Herrera, F., Magdalena, L., & Villar, P. (2001). A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. *Information Sciences*, *136*(1), 85-107.

Cordón, O. (2011). A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning*, *52*(6), 894-913.

Cordon, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001). *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. World Scientific Publishing Company.

Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., & Magdalena, L. (2004). Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy sets and systems (Print)*, *141*(1), 5-31.

Coutinho, J. R., Tanscheit, R., Vellasco, M., & Koshiyama, A. (2016, July). AutoMFIS: Fuzzy Inference System for multivariate time series forecasting. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on* (pp. 2120-2127).

Cpałka, K. (2017). Introduction to Fuzzy System Interpretability. In *Design of Interpretable Fuzzy Systems* (pp. 27-36). Springer International Publishing.

Crockett, K. A., Bandar, Z., Fowdar, J., & O'Shea, J. (2006). Genetic tuning of fuzzy inference within fuzzy classifier systems. *Expert Systems*, *23*(2), 63-82.

Crockett, K., Bandar, Z., & Mclean, D. (2007, July). On the optimization of T-norm parameters within fuzzy decision trees. In *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International* (pp. 1-6). IEEE.

De Oliveira, J. V. (1999). Semantic constraints for membership function optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *29*(1), 128-138.

De Souza, F. J., Vellasco, M. M., & Pacheco, M. A. (2002). Hierarchical neuro-fuzzy quadtree models. *Fuzzy sets and systems*, *130*(2), 189-205.

Esmin, A. A. A., Aoki, A. R., & Lambert-Torres, G. (2002, October). Particle swarm optimization for fuzzy membership functions optimization. In *Systems, Man and Cybernetics, 2002 IEEE International Conference on* (Vol. 3, pp. 6-pp). IEEE.

Fazzolari, M., Alcala, R., Nojima, Y., Ishibuchi, H., & Herrera, F. (2013). A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions. *IEEE Transactions on Fuzzy systems*, *21*(1), 45-65.

Fernández, A., García, S., Luengo, J., Bernadó-Mansilla, E., & Herrera, F. (2010). Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study. *IEEE Transactions on Evolutionary Computation*, *14*(6), 913-941.

Fernández, A., del Jesus, M. J., & Herrera, F. (2010). On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets. *Information Sciences*, *180*(8), 1268-1291.

Fernández, A., del Río, S., & Herrera, F. (2016, July). A First Approach in Evolutionary Fuzzy Systems based on the lateral tuning of the linguistic labels for Big Data classification. In *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on* (pp. 1437-1444). IEEE.

Fernandez, A., Lopez, V., del Jesus, M. J., & Herrera, F. (2015). Revisiting evolutionary fuzzy systems: Taxonomy, applications, new trends and challenges. *Knowledge-Based Systems*, 80, 109-121.

Gacto, M. J., Alcalá, R., & Herrera, F. (2009). Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing*, *13*(5), 419-436.

Gandomi, A. H., & Alavi, A. H. (2012). A new multi-gene genetic programming approach to nonlinear system modeling. Part I: materials and structural engineering problems. *Neural Computing and Applications*, *21*(1), 171

García, D., González, A., & Pérez, R. (2014a). A feature construction approach for genetic iterative rule learning algorithm. *Journal of Computer and System Sciences*, 80(1), 101-117.

García, D., González, A., & Pérez, R. (2014b). Overview of the SLAVE learning algorithm: A review of its evolution and prospects. *International Journal of Computational Intelligence Systems*, 7(6), 1194-1221.

Geyer-Schulz, A. (1997). *Fuzzy rule-based expert systems and genetic machine learning* (Vol. 3). Physica Verlag.

Gonzalez, A., & Pérez, R. (1999). SLAVE: A genetic learning system based on an iterative approach. *IEEE Transactions on Fuzzy Systems*, *7*(2), 176-191.

González, A., & Pérez, R. (2009). Improving the genetic algorithm of SLAVE. *Mathware & Soft Computing*, *16*(1), 59-70.

Gorzałczany, M. B., & Rudziński, F. (2012). Genetic fuzzy rule-based modelling of dynamic systems using time series. In *Swarm and Evolutionary Computation* (pp. 231-239). Springer, Berlin, Heidelberg.

Gorzałczany, M. B., & Rudziński, F. (2016). A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability. *Applied Soft Computing*, 40, 206-220.

Herrera, F., & Martinez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on fuzzy systems*, *8*(6), 746-752.

Herrera, F. (2008). Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, *1*(1), 27-46.

Hinchliffe, M., Hiden, H., McKay, B., Willis, M., Tham, M., & Barton, G. (1996). Modelling chemical process systems using a multi-gene. *Late Breaking Papers at the Genetic Programming*, 56-65.

Ishibuchi, H., Nozaki, K., Yamamoto, N., & Tanaka, H. (1995). Selecting fuzzy if-then rules for classification problems using genetic algorithms. *IEEE Transactions on fuzzy systems*, *3*(3), 260-270.

Ishibuchi, H., Nakashima, T., & Nii, M. (2006). *Classification and modeling with linguistic information granules: Advanced approaches to linguistic Data Mining*. Springer Science & Business Media.

Ishibuchi, H., Nakashima, T., & Murata, T. (1999). Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, *29*(5), 601-618.

Ishibuchi, H., Yamamoto, T., & Nakashima, T. (2005). Hybridization of fuzzy GBML approaches for pattern classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(2), 359-365.

Ishibuchi, H., Yamane, M., & Nojima, Y. (2013, July). Rule weight update in parallel distributed fuzzy genetics-based machine learning with data rotation. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on* (pp. 1-8). IEEE.

Ishibuchi, H., & Nojima, Y. (2015). Multiobjective Genetic Fuzzy Systems. In *Springer Handbook of Computational Intelligence* (pp. 1479-1498). Springer Berlin Heidelberg.

Jang, J. S. R., Sun, C. T., & Mizutani, E. (1997). Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [Book Review]. *IEEE Transactions on automatic control*, *42*(10), 1482-1484.

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

Karr, C. (1991). Genetic algorithms for fuzzy controllers. *AI Expert*, *6*(2), 26-33.

Kickert, W. J., & Koppelaar, H. (1976). Application of fuzzy set theory to syntactic pattern recognition of handwritten capitals. *IEEE transactions on Systems, Man, and Cybernetics*, (2), 148-151.

Kim, D., Choi, Y. S., & Lee, S. Y. (2002). An accurate COG defuzzifier design using Lamarckian co-adaptation of learning and evolution. *Fuzzy sets and systems*, *130*(2), 207-225.

Klement, E. P., Mesiar, R., & Pap, E. (2000). Triangular norms. Kluwer Academic Publishers.

Koshiyama, A. S., Escovedo, T., Dias, D. M., Vellasco, M. M., & Tanscheit, R. (2013). GPF-CLASS: A Genetic Fuzzy model for classification. In *Evolutionary Computation (CEC), 2013 IEEE Congress on* (pp. 3275-3282).

Koshiyama, A. S., Vellasco, M. M., & Tanscheit, R. (2016). A Novel Genetic Fuzzy System for Regression Problems. In *Fuzzy Technology* (pp. 85-101). Springer International Publishing.

Koshiyama, A. S., Vellasco, M. M., & Tanscheit, R. (2015a). GPFIS-CLASS: A Genetic Fuzzy System based on Genetic Programming for classification problems. *Applied Soft Computing*, *37*, 561-571.

Koshiyama, A. S., Vellasco, M. M., & Tanscheit, R. (2015b). Development of a fuzzy rule-based system using Genetic Programming for Forecasting problems. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on* (pp. 1-7). IEEE.

Koshiyama, A. S., Vellasco, M. M., & Tanscheit, R. (2014). GPFIS-control: a genetic fuzzy system for control tasks. *Journal of Artificial Intelligence and Soft Computing Research*, *4*(3), 167-179.

Kovacs, T. (2012). *Strength or accuracy: credit assignment in learning classifier systems*. Springer Science & Business Media.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* (Vol. 1). MIT press.

Lahsasna, A., & Seng, W. C. (2017). An improved genetic-fuzzy system for classification and data analysis. *Expert Systems with Applications*, 83, 49-62.

Lim, T. S., Loh, W. Y., & Shih, Y. S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning*, 40(3), 203-228.

Lima, E., Hell, M., Ballini, R., & Gomide, F. (2010). Evolving fuzzy modeling using participatory learning. *Evolving intelligent systems: methodology and applications*, 67-86.

López, V., Fernández, A., Del Jesus, M. J., & Herrera, F. (2013). A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. *Knowledge-Based Systems*, *38*, 85-104.

Lughofer, E. (2011). *Evolving fuzzy systems-methodologies, advanced concepts and applications* (Vol. 53). Berlin: Springer.

Luke, S., & Panait, L. (2002, July). Lexicographic parsimony pressure. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation* (pp. 829-836). Morgan Kaufmann Publishers.

Mamdani, E. H. (1974, December). Application of fuzzy algorithms for control of simple dynamic plant. In *Proceedings of the institution of electrical engineers* (Vol. 121, No. 12, pp. 1585-1588). IET.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, *7*(1), 1-13.

Márquez, A. A., Márquez, F. A., & Peregrín, A. (2012). A mechanism to improve the interpretability of linguistic fuzzy systems with adaptive defuzzification based on the use of a multi-objective evolutionary algorithm. *International Journal of Computational Intelligence Systems*, 5(2), 297-321.

Márquez, A. A., Márquez, F. A., & Peregrín, A. (2017, July). A scalable evolutionary linguistic fuzzy system with adaptive defuzzification in big data. In *Fuzzy Systems (FUZZ-IEEE), 2017 IEEE International Conference on* (pp. 1-6).

McPhee, N. F., Poli, R., & Langdon, W. B. (2008). Field guide to genetic programming. *Lulu press*.

Mucientes, M., Vidal, J. C., Bugarín, A., & Lama, M. (2009). Processing time estimations by variable structure TSK rules learned through genetic programming. *Soft Computing*, *13*(5), 497-509.

Muni, D. P., & Pal, N. R. (2012). Evolution of fuzzy classifiers using genetic programming. *Fuzzy Information and Engineering*, 4(1), 29-49.

Nauck, D., Klawonn, F., & Kruse, R. (1997). *Foundations of neuro-fuzzy systems*. John Wiley & Sons.

Nojima, Y., Alcalá, R., Ishibuchi, H., & Herrera, F. (2011). Special issue on evolutionary fuzzy systems. *Soft Computing*, *15*(12), 2299-2301.

Nojima, Y., Watanabe, K., & Ishibuchi, H. (2015, August). Simple modifications on heuristic rule generation and rule evaluation in Michigan-style fuzzy genetics-based machine learning. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on* (pp. 1-8). IEEE.

Nojima, Y., Takemura, S., Watanabe, K., & Ishibuchi, H. (2017, June). Michigan-style fuzzy GBML with (1+ 1)-ES generation update and multi-pattern rule generation. In *Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems* (IFSA-SCIS), 2017 Joint 17th World Congress of International (pp. 1-6). IEEE.

Palacios, A. M., Sánchez, L., & Couso, I. (2011). Linguistic cost-sensitive learning of genetic fuzzy classifiers for imprecise data. *International Journal of Approximate Reasoning*, *52*(6), 841-862.

Palacios, A. M., Palacios, J. L., Sánchez, L., & Alcalá-Fdez, J. (2015). Genetic learning of the membership functions for mining fuzzy association rules from low quality data. *Information Sciences*, 295, 358-378.

Paredes, J., Tanscheit, R., Vellasco, M., & Koshiyama, A. (2016, June). Automatic synthesis of fuzzy inference systems for classification. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (pp. 486-497). Springer International Publishing.

Pedrycz, W. (Ed.). (1997). *Fuzzy evolutionary computation* (Vol. 1197). Dordrecht: Kluwer Academic Publishers.

Pedrycz, W., & Gomide, F. (1998). *An introduction to fuzzy sets: analysis and design*. Mit Press.

Pulkkinen, P., & Koivisto, H. (2010). A dynamically constrained multiobjective genetic fuzzy system for regression problems. *IEEE Transactions on Fuzzy Systems*, *18*(1), 161-177.

Rey, M. I., Galende, M., Fuente, M. J., & Sainz-Palmero, G. I. (2017). Multi-objective based Fuzzy Rule Based Systems (FRBSs) for trade-off improvement in accuracy and interpretability: A rule relevance point of view. *Knowledge-Based Systems*, 127, 67-84.

Rodríguez-Fdez, I., Mucientes, M., & Bugarín, A. (2016a). S-FRULER: Scalable fuzzy rule learning through evolution for regression. *Knowledge-Based Systems*, 110, 255-266.

Rodríguez-Fdez, I., Mucientes, M., & Bugarín, A. (2016b). FRULER: Fuzzy rule learning through evolution for regression. *Information Sciences*, 354, 1-18.

Rojas, I., Gonzalez, J., Pomares, H., Rojas, F. J., Fernández, F. J., & Prieto, A. (2001). Multidimensional and multideme genetic algorithms for the construction of fuzzy systems. *International Journal of Approximate Reasoning*, *26*(3), 179-210.

Roychowdhury, S., & Pedrycz, W. (2001). A survey of defuzzification strategies. *International Journal of intelligent systems*, *16*(6), 679-695.

Rudziński, F. (2016). A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers. *Applied Soft Computing*, 38, 118-133.

Sánchez, L., Couso, I., & Corrales, J. A. (2001). Combining GP operators with SA search to evolve fuzzy rule based classifiers. *Information Sciences*, *136*(1), 175-191.

Samudio, O., Vellasco, M., Tanscheit, R., & Koshiyama, A. (2016, October). RandomFIS: A Fuzzy Classification System for Big Datasets. In *INNS Conference on Big Data* (pp. 262-272). Springer International Publishing.

Sanz, J. A., Fernandez, A., Bustince, H., & Herrera, F. (2013). IVTURS: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection. *IEEE Transactions on Fuzzy Systems*, *21*(3), 399-411.

Sanz, J. A., Galar, M., Jurio, A., Brugos, A., Pagola, M., & Bustince, H. (2014). Medical diagnosis of cardiovascular diseases using an interval-valued fuzzy rule-based classification system. *Applied Soft Computing*, *20*, 103-111.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1), 116-132.

Tao, C. W., Chuang, C. C., & Huang, H. C. (2016). Editorial Message: Special Issue on Advances in Evolutionary Fuzzy Systems. *International Journal of Fuzzy Systems*, 18(6), 939-939.

Thrift, P. R. (1991, July). Fuzzy Logic Synthesis with Genetic Algorithms. In *ICGA* (pp. 509-513).

Tsakiridis, N. L., Theocharis, J. B., & Zalidis, G. C. (2016). DECO 3 R: A Differential Evolution-based algorithm for generating compact Fuzzy Rule-based classification systems. *Knowledge-Based Systems*, 105, 160-174.

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

Tsakonas, A. (2013). Local and global optimization for Takagi–Sugeno fuzzy system by memetic genetic programming. *Expert Systems with Applications*, 40(8), 3282-3298.

Tsukamoto, Y. (1979). An approach to fuzzy reasoning method. *Advances in fuzzy set theory and applications*.

Valenzuela-Rendón, M. (1991). The fuzzy classifier system: A classifier system for continuously varying variables. In *Proc. 4th International Conference on Genetic Algorithms (ICGA'91)* (pp. 346-353).

Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man, and cybernetics*, *22*(6), 1414-1427.

Wechsler, H. (1976). A fuzzy approach to medical diagnosis. *International journal of bio-medical computing*, *7*(3), 191-203.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, *8*(3), 338-353.

Zadeh, L. A. (2008). Is there a need for fuzzy logic?. *Information sciences*, *178*(13), 2751-2779.