Ten Years of Genetic Fuzzy Systems: Current Framework and New Trends

O. Cordón, F. Herrera
Dept. Computer Science and A.I.
Universidad de Granada
Granada 18071, Spain
ocordon,herrera@decsai.ugr.es

F. Gomide
Faculty of Electrical and Computer Eng.
State University of Campinas
Campinas–SP–Brazil
gomide@dca.fee.unicamp.br

F. Hoffmann
Royal Institute of Technology
Center for Autonomous Systems
SE-10044 Stockholm, Sweden
hoffmann@nada.kth.se

L. Magdalena
E.T.S. Ingenieros de Telecomunicación
Universidad Politécnica de Madrid
Madrid 28040, Spain
llayos@mat.upm.es

Abstract

Although fuzzy systems demonstrated their ability to solve different kinds of problems in various applications, there is an increasing interest on augmenting them with learning capabilities. Two of the most successful approaches to hybridise fuzzy systems with adaptation methods have been made in the realm of soft computing: neuro-fuzzy systems and genetic fuzzy systems hybridise the approximate reasoning method of fuzzy systems with the learning capabilities of neural networks and evolutionary algorithms. This contribution focus on genetic fuzzy systems, paying special attention to genetic fuzzy rule based systems, giving a brief overview of the field.

1 Introduction

Fuzzy systems have demonstrated their ability for classification [6], modelling [54] or control [16], in a huge number of applications. In most of the cases, the key for the success was the ability of fuzzy systems to incorporate human expert knowledge. In the nineties, despite the previous successful history, the lack of learning capabilities characterising most of the works in the field generated a certain interest for the study of fuzzy systems with added learning capabilities. Two of the most successful approaches to that integration of learning capabilities have been the hybridisation attempts made in the framework of soft computing. One of the successful directions of that effort was that of neuro-fuzzy systems [38, 50]. A different approach to hybridisation lead to genetic fuzzy systems (GFS)[30].

A GFS is basically a fuzzy system augmented by a learning process based on a genetic algorithm (GA). Genetic algorithms are search algorithms, based on natural genetics, that provide robust search capabilities in complex spaces, and thereby offer a valid approach to problems requiring efficient and effective search processes [22, 33].

Genetic processes cover different levels of complexity according to the structural changes produced by the algorithm [15], from the simplest case of parameter optimisation to the highest level of complexity of learning the rule set of a rule based system. Parameter optimisation has been the approach utilised to adapt a wide range of different fuzzy systems, as in genetic fuzzy clustering or genetic neuro-fuzzy systems, that will be briefly considered in Section 5. Analysing the literature the most extended GFSs are *genetic fuzzy rule-based systems* (GFRBSs) [13], where the genetic process learns or tunes different components of a fuzzy rule-based system (FRBS). Inside GFRBSs it is possible to find either parameter optimisation or rule generation processes.

The paper briefly introduces GAs (Section 2), offering then a general approach to GFRBSs (Section 3). Later, Section 4 describes the new (or less common) lines of research in the field of GFRBSs. Section 5 offers a general view of GFSs out of the fuzzy rule based approach. Finally, some conclusions are presented.

2 Genetic algorithms

GAs are general purpose search algorithms which use principles inspired by natural genetics to evolve solutions to problems [22, 33]. The basic idea is to maintain a population of chromosomes (representing candidate solutions to the concrete problem being solved) that evolves over time through a process of competition and controlled variation.

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

Although there are many possible variants of the basic GA, the fundamental underlying mechanism consists of three operations: evaluation of individual fitness, formation of a gene pool (intermediate population) through selection mechanism, and recombination through crossover and mutation operators.

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

When considering a rule based system and focusing on learning rules, there are three main approaches that have been applied in the literature: Pittsburgh approach [61], Michigan approach [34] and Iterative rule learning approach [66]. Pittsburgh and Michigan approaches are the most extended methods for rule learning developed in the field of genetic algorithms. The first one is characterised by representing an entire rule set as a genetic code (chromosome), maintaining a population of candidate rule sets and using selection and genetic operators to produce new generations of rule sets. The Michigan approach considers a different model where the members of the population are individual rules and a rule set is represented by the entire population. In the third approach, the iterative one, chromosomes code individual rules, and a new rule is adapted and added to the rule set, in an iterative fashion, in every run of the genetic algorithm.

3 Genetic fuzzy rule-based systems

The most extended GFS type is the *genetic fuzzy rule-based system*, where a GA is employed to learn or tune different components of an FRBS. Several hundred of papers have been devoted to analyse the automatic generation of the knowledge base of a FRBS using GAs. The key point is to employ an evolutionary learning process to automate the knowledge base generation, which can be considered as an optimisation or search problem.

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

The first step in designing a GFRBS is to decide which parts of the KB are subject to optimisation by the GA. The KB of an FRBS does not constitute an homogeneous structure but is rather the union of qualitatively different components. As an example, the KB of a descriptive Mamdani-type FRBS is comprised of two components: a data base (DB), containing the definitions of the scaling factors and the membership functions of the fuzzy sets associated with the linguistic labels, and a rule base (RB), constituted by the collection of fuzzy rules.

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

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

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

3.1 Genetic tuning

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

Tuning the scaling functions Scaling functions applied to the input and output variables of an FRBS normalise the universes of discourse in which the fuzzy membership functions are defined. Usually, the scaling functions are parameterised by a single scaling factor [51] or a lower and upper bound [48] in case of linear scaling, and one or several contraction/dilatation parameters in case of non-linear scaling [25, 47]. These parameters are adapted such that the scaled universe of discourse better matches the underlying variable range.

The usual approach of this kind of process is the adaptation of one to four parameters (defining the scaling function) per variable.

Tuning the membership functions In the case of tuning the membership functions, an individual represents the entire DB as its chromosome encodes the parameterised membership functions associated to the linguistic terms. The most common shapes for the membership functions (in GFRBSs) are triangular (either isosceles [39, 52] or asymmetric [9, 41]), trapezoidal [29, 40] or Gaussian functions [26, 28]. The number of parameters per membership function usually ranges from one to four, where each parameter is binary [58] or real coded [46].

The structure of the chromosome is different for FRBSs of the descriptive (using linguistic variables) or the approximate (using fuzzy variables) type. When tuning the membership functions in a linguistic model [9], the entire fuzzy partitions is encoded into the chromosome and it is globally adapted to maintain the global semantic in the RB. On the other hand, tuning the membership functions of an approximate model [29, 59] is a particular instantiation of KB learning since the rules are completely defined by their membership functions instead of referring to linguistic terms in the DB.

3.2 Genetic learning of the rule base

Genetic learning of the RB assumes a predefined set of fuzzy membership functions in the DB to which the rules refer to by means of linguistic labels. Genetic learning of the RB only applies to descriptive FRBSs, as in the approximate approach adapting rules is equivalent to modify the membership functions.

The three learning approaches described in previous section are considered in learning rule bases: Michigan approach [3, 37, 63], Pittsburgh approach [32, 55, 62], and iterative rule learning approach [9, 24]. The RB is either represented by a relational matrix [62], a decision table [55], or a list of rules [24, 32, 63].

Representations through relational matrix and decision table are only considered when applying the Pittsburgh approach. The list of rules is the most extended representation, applying quite different codes for the individual rules. A common approach to coding individual rules is the use of rules in disjunctive normal form (DNF) represented in the form of a fixed length binary string [24, 63]. DNF rules are also considered when working on variable length codes [32], that are based on messy GAs [22].

3.3 Genetic learning of the knowledge base

As genetic learning of the KB deals with a heterogeneous search space, it encompasses different genetic representations such as variable length chromosomes, multi-chromosome genomes and chromosomes encoding single rules instead of a whole KB. The computational cost of the search grows with the increasing complexity of the search space. An GFRBS that encodes individual rules rather than entire KBs is an option to maintain a flexible, complex rule space in which the search for a solution remains feasible and efficient. Again, the three learning approaches are considered: Michigan approach [53, 65], Pittsburgh approach [5, 44, 48, 52], and iterative rule learning approach [8, 9, 10].

The proposals for learning KBs include systems obtaining approximate Mamdani-type FRBSs [5, 9, 65], descriptive Mamdani-type FRBSs (scaling functions and rules [48] or membership functions and rules [52]), and TSK fuzzy systems [10, 44].

4 New trends in genetic fuzzy rule-based systems

In addition to those presented in previous section, it is possible to explore new directions in applying genetic (evolutionary) techniques to FRBSs. The present section offers some hints to these works.

Designing fuzzy rule-based systems with genetic programming Genetic programming (GP) is concerned with the automatic generation of computer programs [42]. Different proposals can be found for using genetic programming to evolve fuzzy rule sets, internally represented as type-constrained syntactic trees [1, 19, 31]. Fuzzy GP, proposed in [19], combines a simple GA that operates on a context-free language with a context-free fuzzy rule language. Nowadays, it is possible to distinguish among GPs that utilise a grammar for learning linguistic rules [1, 19], and approaches that use domain-specific knowledge to define the function and terminal set which constitute the building blocks for the fuzzy rules to be learned [31].

Genetic selection of fuzzy rule sets In high-dimensional problems, the number of rules in the RB grows exponentially as more inputs are added. Rule reduction methods have been formulated using neural networks, clustering techniques, orthogonal transformation methods, and algorithms based on similarity measures among others. In recent years genetic techniques have been considered to address the problem of high-dimensional spaces in FRBS design [8, 23, 36].

Learning the knowledge base via the genetic derivation of the data base Genetic tuning of the DB usually assumes that a predefined RB is used to evaluate the quality of the overall FRBS. *A priori genetic DB learning* refers to a KB learning process in which a GA adapts the DB components such as scaling functions, membership functions and granularity parameters, whilst an additional fuzzy rule generation method derives the RB from the DB definition encoded in the chromosome [11, 12, 20, 35].

Other genetic-based machine learning approaches This paragraph mentions other approaches found in the literature. A hierarchical evolutionary method to design FRBSs, where a GA works on different populations encoding information items of different levels, to finally evolve a population of complete FRBSs is proposed in [56]. Different genetic schemes as those inspired on the virus theory of evolution applied to learn TSK fuzzy rule sets [60], on the genetic recombination in bacterial genetics [18], or using DNA coding schemes [17] are found in the literature.

5 Other kinds of genetic fuzzy systems

Although the main GFS type is the GFRBS, other kinds of GFSs have been proposed in the literature and have also obtained successful results. The aim of this section is to introduce several of these other GFSs, namely, genetic fuzzy neural networks and genetic fuzzy clustering algorithms.

5.1 Genetic fuzzy neural networks

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

The usual approach of most genetic fuzzy neural networks found in the literature, is that of adding evolutionary learning capabilities to a fuzzy neural network that usually is a feed-forward multilayer-ed network to which, previously, some fuzzy concepts where incorporated. The result is a feed-forward multilayer-ed network having fuzzy and genetic characteristics [7, 43, 45, 57].

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

5.2 Genetic fuzzy clustering algorithms

There are several references in the literature proposing the application of GAs in fuzzy clustering, most of them devoted to improve the results of FCM-type algorithms [2] by using the GA to optimise some parameters of these kinds of algorithms.

The use of GAs to optimise the parameters of an FCM-type algorithm generates two different GFSs. *Prototype-based algorithms* encode the fuzzy cluster prototypes and evolve them by means of a GA guided by any centroid-type objective function [27, 49]. *Fuzzy partition-based algorithms* encode, and evolve, the fuzzy membership matrix [64].

A second possibility is to use the GA to define the distance norm of an FCM-type algorithm. The system considers and adaptive distance function and employs a GA to learn its parameters in order to obtain an optimal behaviour of the FCM-type algorithm [67].

Finally, a third group of genetic approaches are based on directly solving the fuzzy clustering problem without interaction with any FCM-type algorithm [4].

6 Conclusions

The hybridisation of fuzzy logic and evolutionary computation in so called GFSs became an important research area during the last decade, to which several hundred papers, special issues of different journals and edited books have been devoted [14]. Nowadays, as the field of GFSs matures and grows in visibility, there is an increasing concern about the integration of these two topics from a novel more sophisticated perspective.

As David Goldberg stated, the integration of single methods into hybrid intelligent systems goes beyond simple combinations. For him, the future of Computational Intelligence “*lies in the careful integration of the best constituent technologies*” and subtle integration of the abstraction power of fuzzy systems and the innovating power of genetic systems requires a design sophistication that goes further than putting everything together [21].

References

[1] E. Alba, C. Cotta, and J. M. Troya. Evolutionary design of fuzzy logic controllers using strongly-typed GP. *Mathware & Soft Computing*, 6(1):109–124, 1999.

[2] R. Babuska. *Fuzzy Modeling for Control*. Kluwer Academic Press, 1998.

[3] A. Bonarini. Evolutionary learning of fuzzy rules: competition and cooperation. In W. Pedrycz, editor, *Fuzzy Modelling: Paradigms and Practice*, pages 265–284. Kluwer Academic Press, Norwell, MA, 1996.

[4] B. P. Buckles, F. E. Petry, D. Prabhu, R. George, and R. Srikanth. Fuzzy clustering with genetic search. In *Proc. First IEEE Conference on Evolutionary Computation (ICEC'94)*, pages 46–50, Orlando, FL, USA, 1994.

[5] B. Carse, T. C. Fogarty, and A. Munro. Evolving fuzzy rule based controllers using genetic algorithms. *Fuzzy Sets and Systems*, 80:273–294, 1996.

[6] Z. Chi, H. Yan, and T. Pham. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. World Scientific, 1996.

[7] I.-F. Chung, C. J. Lin, and C. T. Lin. A GA-based fuzzy adaptive learning control network. *Fuzzy Sets and Systems*, 112(1):65–84, 2000.

[8] O. Cordón, M. J. del Jesús, and F. Herrera. Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods. *International Journal of Intelligent Systems*, 13(10-11):1025–1053, 1998.

[9] O. Cordón and F. Herrera. 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.

[10] O. Cordón and F. Herrera. A two-stage evolutionary process for designing TSK fuzzy rule-based systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(6):703–715, 1999.

[11] O. Cordón, F. Herrera, L. Magdalena, and P. Villar. A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base. To appear in Information Sciences, 2001.

[12] O. Cordón, F. Herrera, and P. Villar. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. To appear in IEEE Trans. on Fuzzy Systems., 2001.

[13] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems—Evolutionary tuning and learning of fuzzy knowledge bases*, volume 19 of *Advances in Fuzzy Systems – Applications and Theory*. World Scientific, 2001.

[14] O. Cordón, F. Herrera, and M. Lozano. A classified review on the combination fuzzy logic-genetic algorithms bibliography: 1989-1995. In E. Sanchez, T. Shibata, and L. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives*. World Scientific, 1996.

[15] K. DeJong. Learning with genetic algorithms: An overview. *Machine Learning*, 3(3):121–138, 1988.

[16] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, 1993.

[17] T. Furuhashi. Developement of if-then rules with the use of DNA coding. In W. Pedrycz, editor, *Fuzzy Evolutionary Computation*, pages 107–125. Kluwer Academic Publishers, 1997.

[18] T. Furuhashi, K. Nakaoka, and Y. Uchikawa. A new approach to genetic based machine learning and an efficient finding of fuzzy rules. In *Proc. 1994 IEEE/Nagoya University WWW on Fuzzy Logic and Neural Networks/Genetic Algorithms*, pages 114–121, Nagoya, Japan, 1994.

[19] A. Geyer-Schulz. *Fuzzy Rule-Based Expert Systems and Genetic Machine Learning*. Physica-Verlag, Heidelberg, 1995.

[20] P. Y. Glorennec. Constrained optimization of FIS using an evolutionary method. In Herrera and Verdegay [30], pages 349–368.

[21] D. Goldberg. A meditation on the computational intelligence and its future. Technical Report IlliGAL Report #2000019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 2000. Foreword of the Proc. 2000 Intl. Symposium on Computational Intelligence.

[22] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[23] A. F. Gómez-Skarmeta and F. Jiménez. Fuzzy modeling with hybrid systems. *Fuzzy Sets and Systems*, 104:199–208, 1999.

[24] A. González and R. Pérez. SLAVE: a genetic learning system based on an iterative approach. *IEEE Transactions on Fuzzy Systems*, 7(2):176–191, 1999.

[25] R. Gudwin, F. Gomide, and W. Pedrycz. Context adaptation in fuzzy processing and genetic algorithms. *International Journal of Intelligent Systems*, 13(10/11):929–948, 1998.

[26] H. B. Gurocak. A genetic-algorithm-based method for tuning fuzzy logic controllers. *Fuzzy Sets and Systems*, 108(1):39–47, 1999.

[27] L. O. Hall, J. C. Bezdek, S. Boggavarapu, and A. Bensaid. Genetic fuzzy clustering. In *Proc. NAFIPS'94*, pages 411–415, San Antonio, Texas, USA, 1994.

[28] U. D. Hanebeck and G. K. Schmidt. Genetic optimization of fuzzy networks. *Fuzzy Sets and Systems*, 79(1):59–68, 1996.

[29] F. Herrera, M. Lozano, and J. L. Verdegay. Tuning fuzzy controllers by genetic algorithms. *International Journal of Approximate Reasoning*, 12:299–315, 1995.

[30] F. Herrera and J. L. Verdegay, editors. *Genetic Algorithms and Soft Computing*. Number 8 in Studies in Fuzziness and Soft Computing. Physica-Verlag, 1996.

[31] F. Hoffman and O. Nelles. Genetic programming for model selection of tsk-fuzzy systems. To appear in Information Sciences, 2001.

[32] F. Hoffmann and G. Pfister. Evolutionary design of a fuzzy knowledge base for a mobile robot. *International Journal of Approximate Reasoning*, 17(4):447–469, 1997.

[33] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[34] J. H. Holland and J. S. Reitman. Cognitive systems based on adaptive algorithms. In D. A. Waterman and F. Hayes-Roth, editors, *Pattern-Directed Inference Systems*. Academic Press, 1978.

[35] H. Ishibuchi and T. Murata. A genetic-algorithm-based fuzzy partition method for pattern classification problems. In Herrera and Verdegay [30], pages 555–578.

[36] H. Ishibuchi, T. Murata, and I. B. Türksen. Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems. *Fuzzy Sets and Systems*, 89:135–150, 1997.

[37] H. Ishibuchi, T. Nakashima, and T. Murata. Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 29:601–618, 1999.

[38] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. Prentice-Hall, 1997.

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

[40] C. Karr and E. J. Gentry. Fuzzy control of pH using genetic algorithms. *IEEE Transactions on Fuzzy Systems*, 1(1):46–53, 1993.

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

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

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

[44] M. A. Lee and H. Takagi. Integrating design stages of fuzzy systems using genetic algorithms. In *Proc. Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93)*, pages 613–617, San Francisco, USA, 1993.

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

[46] J. Liska and S. S. Melsheimer. Complete design of fuzzy logic system using genetic algorithms. In *Proc. Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, volume 2, pages 1377–1382, Orlando, FL, USA, 1994.

[47] L. Magdalena. Adapting the gain of an FLC with genetic algorithms. *International Journal of Approximate Reasoning*, 17(4):327–349, 1997.

[48] L. Magdalena and F. Monasterio. A fuzzy logic controller with learning through the evolution of its knowledge base. *International Journal of Approximate Reasoning*, 16(3/4):335–358, 1997.

[49] S. Nascimiento and F. Moura-Pires. A genetic approach for fuzzy clustering with a validity measure function. In X. Liu, P. Cohen, and M. R. Berthold, editors, *Proc. Second International Conference on Intelligent Data Analysis (IDA'97).*, number 1280 in Lecture Notes in Computer Science, pages 325–335, Berlin, 1997. Springer-Verlag.

[50] D. Nauck, F. Klawoon, and R. Kruse. *Foundations of Neuro-Fuzzy Systems*. John Willey & Sons, 1997.

[51] K. C. Ng and Y. Li. Design of sophisticated fuzzy logic controllers using genetic algorithms. In *Proc. Third IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'94)*, volume 3, pages 1708–1712, Orlando, FL, USA, 1994.

[52] D. Park, A. Kandel, and G. Langholz. Genetic-based new fuzzy reasoning models with application to fuzzy control. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(1):39–47, 1994.

[53] A. Parodi and P. Bonelli. A new approach to fuzzy classifier systems. In *Proc. Fifth International Conference on Genetic Algorithms (ICGA'93)*, pages 223–230. Morgan Kaufmann, 1993.

[54] W. Pedrycz, editor. *Fuzzy Modelling: Paradigms and Practice*. Kluwer Academic Press, 1996.

[55] D. T. Pham and D. Karaboga. Optimum design of fuzzy logic controllers using genetic algorithms. *Journal of Systems Engineering*, 1:114–118, 1991.

[56] M. Regattieri-Delgado, F. Von-Zuben, and F. Gomide. Hierarchical genetic fuzzy systems. To appear in Information Sciences, 2001.

[57] M. Russo. FuGeNeSys: A fuzzy genetic neural system for fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, 6:373–388, 2000.

[58] A. Satyadas and K. KrishnaKumar. EFM-based controllers for space attitude control: applications and analysis. In Herrera and Verdegay [30], pages 152–171.

[59] M. Setness and H. Roubos. GA-fuzzy modeling and classification: complexity and performance. *IEEE Transactions on Fuzzy Systems*, 8(5):509–522, 2000.

[60] K. Shimojima, N. Kubota, and T. Fukuda. Virus-evolutionary genetic algorithm for fuzzy controller optimization. In Herrera and Verdegay [30], pages 369–388.

[61] S. F. Smith. *A learning system based on genetic adaptive algorithms*. Doctoral dissertation, Department of Computer Science. University of Pittsburgh, 1980.

[62] P. Thrift. Fuzzy logic synthesis with genetic algorithms. In *Proc. Fourth International Conference on Genetic Algorithms (ICGA'91)*, pages 509–513, San Diego, USA, 1991. Morgan Kaufmann.

[63] M. Valenzuela-Rendón. The fuzzy classifier system: Motivations and first results. In H. P. Schwefel and R. Männer, editors, *Proc. First International Conference on Parallel Problem Solving from Nature - PPSN I*, pages 330–334. Springer-Verlag, Berlin, 1991.

[64] T. Van Le. Evolutionary fuzzy clustering. In *Proc. Second IEEE Conference on Evolutionary Computation (ICEC'95)*, volume 2, pages 753–758, Perth, Australia, 1995.

[65] J. R. Velasco. Genetic-based on-line learning for fuzzy process control. *International Journal of Intelligent Systems*, 13(10-11):891–903, 1998.

[66] G. Venturini. SIA: a supervised inductive algorithm with genetic search for learning attribute based concepts. In *Proc. European Conference on Machine Learning*, pages 280–296, Viena, 1993.

[67] B. Yuan, G. J. Klir, and J. F. Swan-Stone. Evolutionary fuzzy c-means clustering algorithm. In *Proc. Fourth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'95)*, pages 2221–2226, Yokohama, Japan, 1995.