Volume-5, Issue-11, November - 2016 • ISSN No 2277 - 8160

Original Research Paper | Engineering

Fuzzy Genetic Algorithms: Fuzzy Logic Controllers and Genetics Algorithms

Antonia Plerou — Department of Informatics, Ionian University, Corfu, Greece. Bioinformatics and Human Electrophysiology Laboratory

Elena Vlamou — Department of Civil Engineering, Democritus University of Thrace Xanthi, Greece

Vasil Papadopoulos — Professor, Department of Civil Engineering, Democritus University of Thrace, Xanthi, Greece

ABSTRACT

The fusion of Genetic Algorithms and Fuzzy Logic Systems allows researchers to model real world problems through the development of intelligent and adaptive systems. Fuzzy Genetic Algorithm (FGA) is a Genetic Algorithm that uses fuzzy logic-based techniques. The objective of this blending is to adjust the system parameters to robust and optimize the performance of the genetic algorithms. This paper provides the basic knowledge of fuzzy systems (FSs) and genetic algorithms (Gas) and provides an overview of how these technologies are cooperatively combined and have been applied in the real world.

Summary This paper deals with the fuzzy logic inference systems and genetic algorithmic fusion and evaluate the applications of fuzzy genetic algorithms.

KEYWORDS : Fuzzy sets, Fuzzy logic controllers, Genetic algorithms, Fuzzy genetic algorithms

Introduction

Fuzzy logic and genetic algorithms during the last few years were rapidly progressed in the industrial world in order to solve effectively real-world problems. Fuzzy logic is applied to several fields like control theory or artificial intelligence The term "fuzzy logic" was introduced with fuzzy set theory proposal by Lotfi A. Zadeh in 1965 (Sanchez, Shibata, & Zadeh, 1997) but the idea had been studied since the 1920s as infinite-valued logics notably by Łukasiewicz and Tarski (Sanchez et al., 1997). A genetic algorithm (or GA) is a search technique used in computing to estimate approximate solutions for optimization and search problems. Genetic algorithms are evolutionary algorithms that use techniques inspired by evolutionary biologies such as inheritance, mutation, selection, and crossover (which is also known as recombination) (Vlahavas, 2008). The new population created is thereafter used in the next iteration of the algorithm. In most of the cases, the algorithm terminates when either a maximum number of generations has been produced, or in the case that the population reached a sufficient fitness level. In the case that the algorithm has terminated due to a maximum number of generations, it not certain that a satisfactory solution has been reached (Sumathi & Kumar, 2016)

Genetic Algorithm Description

The basic genetic algorithm philosophy developed by Goldberg was inspired by Darwin's theory of evolution. This theory suggests that the survival of an organism is affected by rule "the strongest species that survives". Darwin's theory also suggests that the survival of an organism can be maintained through the process of reproduction, crossover, and mutation (Hermawanto, 2013) Darwin's concept of evolution is also used to the computational algorithm in order to find a solution to a problem known as an objective function in a natural fashion. A solution generated by the genetic algorithm is called a chromosome while as a population the collection of the chromosome is referred. A chromosome is composed of genes and its value can be either numerical, binary, symbols or characters depending on the given problem. These chromosomes are to undergo a process called fitness function to measure the suitability of solution generated by the genetic algorithm. Some chromosomes in the population are to mate through a process called crossover. The procedure named offspring is the production of new chromosomes which genes composition are the combination of their parent. In a generation, a few chromosomes are about to go through mutation in their gene. The number of chromosomes which is to undergo crossover and mutation is controlled by crossover rate and mutation rate value. Chromosome in the population that are to maintain for the next generation are to be selected according to the Darwinian evolution rule. In addition, the chromosome which has higher fitness value is having a greater probability of being selected again in the next generation. After several generations, the chromosome value is to converge to a certain value which is the best solution for the problem (Grosan, C., & Abraham, 1996).

Genetic Algorithm Process Basic Steps

The genetic algorithm differs from a classical, derivative-based, optimization algorithm in two essential ways. Namely, a classical algorithm generates a single point at each iteration and the sequence of points approaches an optimal solution. On the other hand, a genetic algorithm generates a population of points at each iteration and the best point in the population approaches an optimal solution. Furthermore, a classical algorithm selects the next point in the sequence by a deterministic computation while a genetic algorithm selects the next population by a computation which uses random number generators (Daras & Rassias, 2015)assessment, and data management are core competencies for operation research analysts. This volume addresses a number of issues and developed methods for improving those skills. It is an outgrowth of a conference held in April 2013 at the Hellenic Military Academy, and brings together a broad variety of mathematical methods and theories with several applications. It discusses directions and pursuits of scientists that pertain to engineering sciences. It is also presents the theoretical background required for algorithms and techniques applied to a large variety of concrete problems. A number of open questions as well as new future areas are also highlighted.This book will appeal to operations research analysts, engineers, community decision makers, academics, the military community, practitioners sharing the current state-of-the-art, and analysts from coalition partners. Topics covered include Operations Research, Games and Control Theory, Computational Number Theory and Information Security, Scientific Computing and Applications, Statistical Modeling and Applications, Systems of Monitoring and Spatial Analysis. (Daras & Rassias, 2015. The basic steps of a genetic algorithm are namely the initialization, the evaluation, the selection, the crossover, the mutation and the repetition. These steps are described in details below:

Initialization is used to create an initial population. This population is usually randomly generated and consists of a few individuals to thousands (Shankar, Mishra, Dehuri, Kim, & Wang, 2016).

Evaluation means that each member of the population is evaluated and a 'fitness' for that individual is calculated. The fitness value is calculated according to the problems' desired requirements. These requirements could be simple due to the fact that faster algorithms are better, or more complex because stronger materials are better as long as they are not too heavy (Garg, 2015).

The selection means that individuals are selected according to specific selection rules, which are called parents and contribute to the population at the next generation. The basic concept of selection operator is to give preference to improved individuals, allowing them to pass on their genes to the next generation. The goodness of each individual depends on its fitness. Fitness could be determined by an objective function or by a subjective judgment. Populations overall fitness should be improving continuously. Selection enhances this procedure by discarding the bad designs and solely keeping the best individuals in the population. There are several selection methods though the basic idea is not altered, therefore, it is more likely that proper individuals are to be selected for the next generation (Vijayarani & Vinupriya, 2015).

Crossover is a process of taking more than one parent solutions and producing a child solution from them. Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next one. This is similar to reproduction and biological crossover, upon which genetic algorithms are based. (Shankar et al., 2016) With crossover operator use factor of GA is prime distinguished from other optimization techniques. Especially, two individuals are chosen from the population using the selection operator and a crossover site along the bit strings chosen in a random way. The values of the two strings are exchanged up to and the two new offspring generated from this mixture are put into the next generation of the population. The procedure recombining portions of good individuals lead to the possible creation of improved individuals (Ruan, 2013).

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next one. It is a similar procedure to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution probably alters completely comparing with the previous solution. Hence, GA can come to a better solution by using mutation. The mutation occurs during evolution according to a user-definable mutation probability. This probability is set low but in the opposite case, the search is about to turn into a primitive random search (Shankar et al., 2016).

The final step is repetition which means that the algorithm starts again and repeats constantly in order to reach a termination condition.

Genetic Algorithm Applications

Genetic algorithms are inspired and based on the process of evolution by natural selection in order to provide solutions to real-world problems. Specifically, the genetic algorithm is applied in order to solve several optimization problems, like problems where the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. The genetic algorithm can address problems of mixed integer programming, where several components are restricted to be integer-valued (Almeida, Oliveira, & Pinto, 2015). Furthermore, genetic algorithm (GA) are used solving complex search problems such as engineering to create incredibly high-quality products due to their ability to search a through a huge combination of parameters to find the best match. For instance, GA searches through different combinations of materials and designs to find the better combination in order to result in an overall enhanced result. Additionally, they are used to design computer algorithms, to schedule tasks, and to solve several optimization problems.

Fuzzy Logic

Fuzzy logic provides an inference morphology in order to enable approximate human reasoning capabilities to be applied to knowledge-based systems. The conventional approaches to knowledge representation lack the means for representation of fuzzy concepts. The fuzzy logic is an approach to computing based on "degrees of truth" rather than the "true or false" (1 or 0) Boolean logic on which the modern computer function is based. Fuzzy logic includes 0 and 1 as extreme cases of truth, nevertheless, the various states of truth in between are also included. Fuzzy logic has been extended to handle the concept of partial truth, where the value is ranged between completely true and completely false. Compared to traditional binary sets (where variables only take on true or false values), fuzzy logic variables have a truth value that ranges in degree between 0 and 1. Fuzzy systems suggest a mathematic model to translate the real processes of human knowledge (Plerou, Vlamou, & Papadopoulos, 2016).

Fuzzy Theory

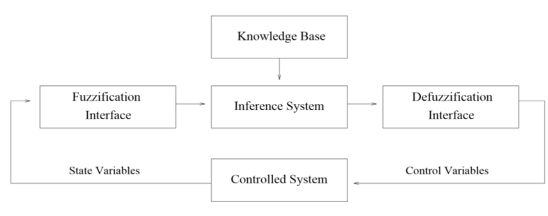
Essential research fields in fuzzy theory are fuzzy sets, fuzzy logic, and fuzzy measure. Fuzzy reasoning is an application of fuzzy logic to knowledge processing. Fuzzy systems have the ability to realize a complex nonlinear input–output relation as a synthesis of multiple simple inputs–output relations, similar to neural networks function. The input–output relation is described in each rule, nevertheless, the boundary of the rule areas is fuzzy. The system output from one rule area to the next rule area gradually changes. This is the essential idea of fuzzy systems and the origin of the term 'fuzzy.' Fuzzy control is an application of fuzzy reasoning to control. It is worth mentioning that even that the majority of fuzzy theory applications were related mostly to engineering, they recently address towards other disciplines, such as medical diagnostics, psychology, education, economy, management, sociology, etc (Ruan, 1997).

Fuzzy Sets

Fuzzy sets are the basic concept of fuzzy logic theory. Let's consider X as a nonempty set. A fuzzy set A in X is characterized by its membership function μ_A: X → [0, 1] and μ_A (x) is interpreted as the degree of membership of element x in fuzzy set A for each x ϵ X. The A set is determined as follows: A = {(u, μ_A (u))/u ϵ X}. The family of all fuzzy sets in X is denoted by F(X) function. When X = {x_1, ……, x_n} is a finite set and A is a fuzzy set in X, the notation A = μ_1 / x_1 + …… + μ_n / x_n is used, where the term μ_i/x_i , i = 1,…, n signifies that μ_i is the grade of membership of x_i in A and the plus sign represents the union of the elements. Each element in the universe of discourse is a member of the fuzzy set to some extent. The set of elements that have a non-zero membership is called the support of the fuzzy set. The function that ties a number to every element of the universe is known as the membership function μ(x) (Ephzibah, 2011).

Fuzzy Logic Controllers

A Fuzzy Logic Controller is a system which consists of a knowledge base, that includes the information given in the form of linguistic control rules, and a fuzzification interface, which has the effect of transforming crisp data into fuzzy sets. Additionally, an Inference System is included which works together with the knowledge base to provide inference with the use of a reasoning method, and a defuzzification interface, which translates the fuzzy control action obtained to a real control action by means of a defuzzification method (Abraham, Hassanie, Siarry, & Engelbrecht, 2009).



Generic structure of an FLC.

Figure 1: Structure of a Fuzzy Logic Controller

The knowledge base encodes the expert knowledge with the use of a set of fuzzy control rules. A fuzzy control rule is a conditional statement which is formed as following: "In the case that a set of conditions are satisfied then a set of consequences could be inferred". The antecedent is a condition in its application domain, the consequent is a control action to be applied to the controlled system or else notion of control rule and both antecedent and consequent are related with fuzzy concepts, namely the linguistic terms or else the notion of the fuzzy rule. The knowledge base consists of two components. Namely: 1) a database, which contains the definitions of the fuzzy control rules linguistic labels, meaning the membership functions of the fuzzy sets specifying the linguistic terms meaning, and 2) a rule base, which is consisting of a collection of fuzzy control rules which represent the expert knowledge (Witold Pedrycz, 2012).

Fuzzy Genetic Algorithms

Essential expertise and knowledge on GAs have resulted from empirical studies conducted over a number of years. This suggests the use of fuzzy logic-based tools for dealing with several problems. An essential application of fuzzy logic that is useful for controlling genetic algorithms within the control strategies underlying based on the human experience and knowledge are fuzzy logic controllers (FLCs). Fuzzy Logic Controllers implement an expert operator's approximate reasoning procedures in the selection of a control action. A Fuzzy Logic Controller allows one to qualitatively express the control strategies based on experience. These control strategies may be expressed in a form that permits both computers and humans to share them efficiently (Herrera, Lozano, & Verdegay, 1995).

Fuzzy Genetic Algorithms (FGA), are Genetic Algorithms integrated with Fuzzy Logic Controllers. Fuzzy Genetic Algorithms may 1) choose control parameters before GA's run, 2) adjust the control parameters on-line to adapt to new situations in a dynamic way and 3) assist the user to access, design, implement and validate the Genetic Algorithm for a specific task (Herrera & Lozano, 1996). The idea is to use a Fuzzy Logic Controllers whose inputs are any combination of Genetic Algorithm performance measures or current control parameters and whose outputs are Genetic Algorithm control parameters. Current performance measures of the Genetic Algorithm are sent to the Fuzzy Logic Controller, which computes new control parameters values that will be used by the Genetic Algorithm.

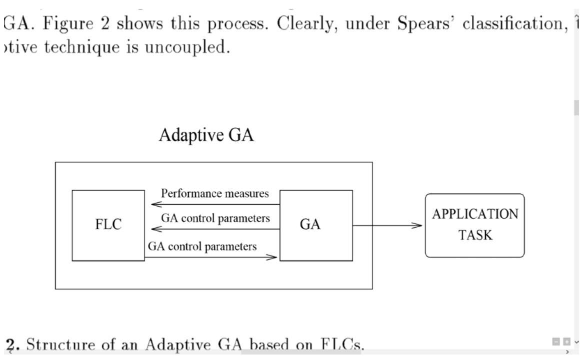


Figure 2: Structure of an Adaptive GA based on Fuzzy Logic Controller

The use of Fuzzy Logic-based techniques for either improving GA behavior and modeling GA components, the results obtained have been called fuzzy genetic algorithms (FGAs). Genetic algorithms are applied in various optimization and search problems involving fuzzy systems. A Fuzzy Genetic Algorithm is defined as an ordering sequence of instructions in which some of the instructions or algorithm components designed with the use of fuzzy logic based tools. A fuzzy fitness finding mechanism guides the GA through the search space, combining the contributions of various criteria identified as the governing factors for the formation of the clusters (Pedrycz, 2012).

Fuzzy Genetic Algorithms Optimization

A single objective optimization model is not able to serve the purpose of a fitness measuring index due to the fact that multiple criteria could be responsible for stringing together data items into clusters. This is valid for the clustering problem and for each problem solving using GA involving multiple criteria. In multi-criteria optimization, the notion of optimality is vaguely defined. The algorithm has two computational elements that work together i.e. the Genetic Algorithm (GA) and the Fuzzy Fitness Finder (FFF). In the simple cases, there is only one criterion for optimization, for instance, maximization or minimization of profit or cost respectively (Jaganathan & Karthikeyan, 2014). Nevertheless, in many real-world decision-making problems, the multiple objectives simultaneous optimization is essential (Michalewicz, 2013)i.e., survival of the fittest. Hence evolution programming techniques, based on genetic algorithms, are applicable to many hard optimization problems, such as optimization of functions with linear and nonlinear constraints, the traveling salesman problem, and problems of scheduling, partitioning, and control. The importance of these techniques is still growing, since evolution programs are parallel in nature, and parallelism is one of the most promising directions in computer science.The book is self-contained and the only prerequisite is basic undergraduate mathematics. This third edition has been substantially revised and extended by three new chapters and by additional appendices containing working material to cover recent developments and a change in the perception of evolutionary computation. (Michalewicz, 2013. An evaluation function to rate solutions in terms of their "fitness" is also necessary. Therefore, genetic operators change the composition of the children in order to make a successful run of a GA. In addition, the values for the parameters of the GA have to be defined as the population size and the parameters for the genetic operators and the terminating condition is evaluated (Michalewicz, 2013)i.e., survival of the fittest. Hence evolution programming techniques, based on genetic algorithms, are applicable to many hard optimization problems, such as optimization of functions with linear and nonlinear constraints, the traveling salesman problem, and problems of scheduling, partitioning, and control. The importance of these techniques is still growing, since evolution programs are parallel in nature, and parallelism is one of the most promising directions in computer science.The book is self-contained and the only prerequisite is basic undergraduate mathematics. This third edition has been substantially revised and extended by three new chapters and by additional appendices containing working material to cover recent developments and a change in the perception of evolutionary computation.", "author" : [{ "dropping-particle" (Michalewicz, 2013. The Fuzzy Genetic Algorithms are mentioned to be most efficient algorithm comparing to standard Genetic Algorithms in solving the traveling salesman and other optimization problems (Herrera & Lozano, 1996).

Limitations of Fuzzy Genetic Algorithm

Considering the enhanced degree of nonlinearity of the output of a fuzzy system, traditional linear optimization tools are noticed to have their limitations. Genetic algorithms are an enhanced and very powerful method to perform tasks namely the generation of the fuzzy rule base, optimization of fuzzy rule bases, generation of membership functions, and adjusting membership functions. These tasks are considered as optimization or search processes within large solution spaces (Bastian & Hayashi, 1995).

Although genetic algorithms are effective tools to identify the fuzzy membership functions of a pre-defined rule base, they demonstrate limitation especially in the case that the input and output variables of a fuzzy system from a given set of data need to be identified. Genetic programming is used to specify the input variables, the rule base and the involved membership functions of a fuzzy model (Bastian & Hayashi, 1995).

The use of Fuzzy Logic Controllers for control Genetic Algorithms is considered for solving two problems to which a standard Genetic Algorithm could expose limited search speed and premature convergence. This due to the fact that 1) control parameters are not well chosen initially for a given task. 2) parameters always being fixed even though the environment in which the GA operates could be variable and 3) problems resulting from the selection of several parameters like population size and in perceiving their influence.

Discussion

The use of genetic algorithms in determining the optimal rules of the fuzzy logic system has been efficient in providing accurate results. Via the operations of selection, crossover, and mutation the GA will converge over successive generations towards the optimum one. These operations provide a fast, useful and robust method because genetic algorithms combine an effective and efficient solution. Since population implicitly contain much more information than solely the individual fitness scores, Genetic Algorithm combines the good information hidden in a solution with good information from another solution in order to produce new solutions with good information inherited from both parents, inevitably leading towards optimality (Sivanandam & Deepa, 2007). The ability of the algorithm to explore and exploit simultaneously, a growing amount of theoretical justification, and the essential application to real-world problems strengthens the conclusion that GAs are an enhanced optimization technique.

References

1. Abraham, A., Hassanie, A.-E., Siarry, P., & Engelbrecht, A. (2009). *Foundations of Computational Intelligence Volume 3: Global Optimization*. Springer Science & Business Media.

2. Almeida, J. P., Oliveira, J. F., & Pinto, A. A. (2015). *Operational Research: IO 2013 - XVI Congress of APDIO, Bragança, Portugal, June 3-5, 2013*. Springer.

3. Bastian, A., & Hayashi, I. (1995). An Anticipating Hybrid Genetic Algorithm for Fuzzy Modeling. *Journal of Japan Society for Fuzzy Theory and Systems*, *10*, 801–810.

4. Daras, N. J., & Rassias, M. T. (2015). *Computation, Cryptography, and Network Security*. Springer.

5. Ephzibah, E. P. (2011). Time complexity analysis of genetic-fuzzy system for disease diagnosis. *Advanced Computing: An International Journal*, *2*(4), 23–31. http://doi.org/10.5121/acij.2011.2403 23

6. Garg, P. B. (2015). Study of Different Selection Strategies for Implementing Optimization Problem. *International Journal for Scientific Research & Development*, *3*(2).

7. Grosan, C., & Abraham, A. (1996). Solving nonlinear equation systems using evolutionary algorithms. In *Proceedings of genetic & evolutionary computation conference, Seattle, USA, Proceedings on CD.*

8. Hermawanto, D. (2013). G. algorithm for solving simple mathematical equality problem. arXiv preprint arXiv:1308. 4675. (2013). Genetic algorithm for solving simple mathematical equality problem. *arXiv Preprint arXiv:1308.4675.*

9. Herrera, F., & Lozano, M. (1996). Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorithms and Soft Computing*, *8*, 95–125.

10. Herrera, F., Lozano, M., & Verdegay, J. L. (1995). Tuning Fuzzy Logic Controllers by Genetic Algorithms. *International Journal of Approximate Reasoning*, *12*, 299–315.

11. Jaganathan, P., & Karthikeyan, T. (2014). An Evolving Approach on Efficient Web Crawler using Fuzzy Genetic Algorithm. *International Journal of Science and Research (IJSR)*, *3*(10).

12. Michalewicz, Z. (2013). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Science & Business Media.

13. Pedrycz, W. (2012). *Fuzzy Evolutionary Computation*. Springer Science & Business Media.

14. Plerou, A., Vlamou, E., & Papadopoulos, B. (2016). Visualization of Neuro-Fuzzy Networks Training Algorithms: The Backpropagation Algorithm Approach. In I. Deliyannis, P. Kostagiolas, & C. Banou (Eds.), *Experimental Multimedia Systems for Interactivity and Strategic Innovation* (pp. 222–257). IGI Global. http://doi.org/10.4018/978-1-4666-8659-5

15. Ruan, D. (Ed.). (1997). *Intelligent Hybrid Systems*. Boston, MA: Springer US. http://doi.org/10.1007/978-1-4615-6191-0

16. Ruan, D. (2013). *Fuzzy Systems and Soft Computing in Nuclear Engineering* (Vol. 21). Physica.

17. Sanchez, E., Shibata, T., & Zadeh, L. A. (1997). *Genetic Algorithms and Fuzzy Logic Systems: Soft Computing Perspectives*. World Scientific.

18. Shankar, B., Mishra, P., Dehuri, S., Kim, E., & Wang, G.-N. (2016). *Techniques and Environments for Big Data Analysis: Parallel, Cloud, and Grid Computing*. Springer.

19. Sivanandam, S. N., & Deepa, S. N. (2007). *Introduction to Genetic Algorithms*. Springer Science & Business Media.

20. Sumathi, S., & Kumar, L. A. (2016). *Computational Intelligence Paradigms for Optimization Problems Using MATLAB®/SIMULINK®*. CRC Press.

21. Vijayarani, S., & Vinupriya, M. (2015). An Efficient Algorithm for Facial Image Classification. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *8*(11), 121–134. http://doi.org/http://dx.doi.org/10.14257/ijsip.2015.8.12.13

22. Vlahavas, I. (Ed.). (2008). Artificial Intelligence for Advanced Problem Solving Techniques. In *IGI Global.* (p. 388). IGI Global.

23. Witold Pedrycz. (2012). *Fuzzy Evolutionary Computation*. Springer Science & Business Media.