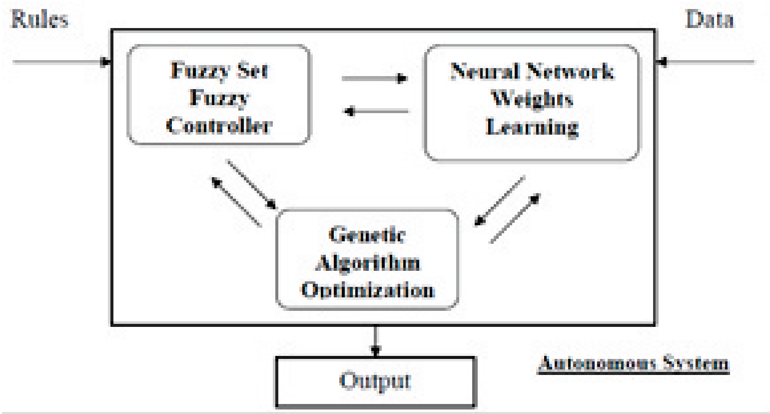
Realization of Autonomous Soft Computing System Using Computational Intelligence Methods

Priyanka Sethi
Sri Guru Tegh Bahadur Institute of Management and Information Technology, Near Nanak Ji piao, GT Karnal Road, Delhi
E-mail: priyanka.sethi@ymail.com

Abstract

Computational Intelligence or Soft Computing methods refer to three concepts viz. Artificial Neural Network, Fuzzy Systems and Evolutionary Algorithm (EA) or Genetic Algorithm (GA). The Computational Intelligence methodologies facilitate a better data acquisition from our everyday environment, and thus help in extending the capabilities of conventional computer systems.



The fuzzy controller is realized with the application of fuzzy logic concepts to an information processing system. The controller is based on linguistic rules, so that human experts can insert knowledge into the system. The behaviour of neural networks is adapted by input data supplied by the environment. These networks learn by adjusting the strength of connections (weights) between the neurons. The entire system turns out to be complex and is optimized by application of GA.

Keywords: Computational Intelligence, Soft Computing, Artificial Neural Network, Fuzzy Systems, Evolutionary Algorithm, Genetic Algorithm.

1 Computational Intelligence

The three Soft Computing methods operating in synchronization with each other for real time learning: fuzzy systems, artificial neural networks and genetic algorithms are referred as Computational Intelligence (CI) methods. Unlike classical reasoning and modeling approaches, these methods provide a set of more flexible computing technique for solving real - world problems involving incomplete, uncertain and imprecise input-output information. There are two well established approaches to handle this imperfect information. One is the *probabilistic* reasoning approach, based on probability theory. The other is *possibilistic* reasoning approach, based on the semantics of fuzzy sets and many-valued logics. Both disciplines differ in terminology and apparatus. The theory of probability serves to determine a degree of uncertainty, in which any of the presumed effects occurs. In the contrary fuzzy logic helps to determine which one of many effects defined in different way have occurred. The aim of the implementation of both methods is to obtain better control and prediction of highly non-linear systems behavior.

1.1 Computational Intelligence Methods in the Control of Complex Technological Systems

In the operation of large and complicated systems, the ensuring a limitation of errors becomes more significant. Two factors viz. safety and operation economy, influence the development and use of new procedures. Both factors induce the elimination of risk and optimal system performance. Since the conventional automated controllers are not completely suitable, their operations are primarily dependent on human operators. Among many controllers proposed to replace the manual operations, the neuro-fuzzy control method is generally regarded as a suitable control method due to its human-like characteristics. As such these methods are discussed in the following sections.

1.1.1 Fuzzy Logic

Fuzzy logic is a mathematical discipline forming the basis for successful applications in the area of regulation and control. Its structure corresponds to the natural human thinking and is even able to work with the normal usage of words. The real world problems carry the aspiration levels and priority factors of the decision maker, as such the weights to be assigned to the goals, are imprecise in nature. In such situations it is necessary to use fuzzy set theory.

Fuzzy set, introduced by **Zadeh [3]** as a new way to represent vagueness in everyday life. It is a generalization of conventional set theory, one of the basic structures underlying computational mathematics and models. A fuzzy set is a class of objects in which there is no sharp boundary between those objects that belong to the class and those that do not. Each object has a number associated with it that represents its grade of membership in the fuzzy set. The linguistic terms Positive, very positive, and negative can be represented by fuzzy sets based on appropriate membership functions.

The main difference between the approaches of fuzzy logic and the classical technique is that fuzzy logic uses even uncertain information to describe the system, whereas the classical technique requires exact analytical relations. Therefore fuzzy logic offers the mathematical language appropriate for the description of the control of a system, the behavior of which may be described in form of the 'if-then' rules.

Fuzzy Control is a powerful control synthesis technology that complements other analytical / data-driven techniques. A fundamental characteristic of fuzzy logic is the linguistic, often unsharp presentation of quantities. This unsharpness should not be interpreted as inaccuracy. Fuzzy logic gives accurate result, though the computation is based on uncertain data.

The fuzzy controllers incorporate the experience of human operators into their design in form of fuzzy Rules and Fuzzy Membership Functions. A fuzzy control algorithm is designed from a set of linguistic rules, which describe the operator's control strategy. The main advantages of the fuzzy controller are the possibility of implementing *rule of thumb* experience, intuition, heuristics and the fact that it does not need a mathematical model of the process as explained in [1].

Fuzzy logic is applicable to a vast gamut of applications, spanning over large range of cost, performance, complexity and throughput requirements. It makes high demands on microelectronics and offers promising opportunities. The combination of FC with other emerging technologies, such as neural network and genetic algorithms could further improve the cost/benefit ratio when applied to a large spectrum of complex control problems.

1.1.2 Artificial Neural Networks (ANN)

Neural networks attracted considerable attention in many disciplines and their implementation has proven to be useful. ANN solve problem situations by comparing them to information prepared in advance. ANN is being created using a special procedure, called training, where information about possible situations and the corresponding solutions are processed. Therefore, the knowledge of the specific dependencies between the input parameters and their solution is not needed for the ANN creation. ANN methods are particularly useful for the solution of situations, described by a great number of parameters with ambiguous dependencies.

The neuro-fuzzy method can replace manual operations in large systems due to its human-like characteristics. A neuro-fuzzy system is a fuzzy logic system equipped with a training algorithm. The fuzzy logic system is constructed from a collection of fuzzy *if-then* rules and the training algorithm adjusts the parameters of the fuzzy logic system based on numerical information (mainly input-output pairs). Neuro-fuzzy systems combine linguistic and numerical information. Because fuzzy logic systems are constructed from fuzzy *if-then* rules, linguistic information can be directly incorporated. On the other hand, numerical information is incorporated by training the fuzzy logic system to match the input-output pairs.

One can use two or more learning algorithm (hybrid learning algorithm) to train the parameters of neuro-fuzzy inference systems. For example, one can train the antecedent parameters by using a back-propagation method and the consequent parameters using a least-squares method as explained in [11]. The consequent parameters are updated first using a least-squares method and then the antecedent parameters are updated by back-propagating the errors that still exist [7]. Alternatively one can train the antecedent parameters by using a genetic algorithm and the consequent parameters using a least-squares method.

It is difficult to create a complete kit of training sets and the generalizing ability of ANN is limited. Training sets are obtained by capturing directly from the pertinent real process. The function of neural network is to approximate the function of the process even in extreme situations. It is possible to obtain process functions representative enough for such situations by simulating extreme situations using a mathematical model of the process.

1.1.3 Genetic Algorithm

The term *genetic algorithm* is derived from the fact that its operations are based on the mechanics of genetic adaptation in biological systems. Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. Genetic algorithms for optimization were formally introduced in the 1970s by Holland (1975) in [8]. More details about genetic algorithms can be found in [9] by Goldberg (1989) and [10] by Mitchell (1996). Genetic algorithms have been proven to be efficient in many different areas, such as nuclear fuel loading optimization [explained by DeChaine and Feltus 1996; Parks 1996]

The genetic algorithm is a method for moving from one population of chromosomes to a new population by using a kind of natural selection together with the genetics-inspired operators of selection, crossover, and mutation. Many optimization methods move from a single point in the decision space to the next using some transition rule to determine the next point. This point-to-point method is dangerous because it is a perfect prescription for locating false peaks in many peaked search spaces. By contrast, genetic algorithms work from many points simultaneously climbing many peaks in parallel. Thus, the probability of finding a false peak is reduced over methods that go point to point. Therefore, genetic algorithms are less susceptible to being stuck at local optima than conventional search methods.

Many search techniques require much auxiliary information in order to work properly. For example, gradient techniques need derivatives in order to climb the current peak. By contrast, genetic algorithms do not need all this auxiliary information. Also, genetic algorithms use probabilistic transition rules to guide their search but the use of probability does not suggest that the method is some simple random search. Genetic algorithms use random choice as a tool to guide a search toward regions of the search space with likely improvement [9]. Despite of these advantages, however, genetic algorithms tend to be computationally expensive.

In genetic algorithms, the term chromosome typically refers to a candidate solution to a problem, generally encoded as a bit string. Each chromosome can be thought of as a point in the search space of candidate solutions. The genetic algorithms process populations of chromosomes, successively replacing one such population with another. The genetic algorithms require a fitness function that assigns a score to each chromosome in the current population. The fitness of a chromosome depends on how well that chromosome solves the problem at hand (e.g. [10]). The fitness function should be defined properly according to the given problem (e.g. [12]).

Initially, after an initial population of chromosomes is randomly generated, then the typical genetic algorithm evolves the population through the following three operators.

<u>Selection Operator:</u> This operator selects individuals (chromosomes) in the population for reproduction. The goodness of each individual depends on its fitness. Fitness may be determined by an objective function. The fitter the chromosome, the more times it is likely to be selected to be reproduced.

<u>Crossover Operator:</u> After two individuals are chosen from the population using the selection operator, the crossover operator randomly chooses a crossover site along the bit strings and exchanges the subsequences before and after that crossover site between the two individuals to create two offspring. For example, the strings *000000* and 111111 could be crossed over after the second locus in each to produce the two offspring *110000* and *001111*. The two new offspring created from this mating are put into the next generation of the population. By recombining portions of good individuals, this process is likely to create even better individuals.

<u>Mutation Operator:</u> With some low probability, a portion of the new individuals will have some of their bits flipped. Mutation can occur at each bit position in a string with some small probability. Its purpose is to maintain diversity within the population and inhibit premature convergence.

Most genetic algorithms follow the procedures as explained above.
To use a genetic algorithm, one must represent a solution to a given problem as a chromosome. The genetic algorithm then creates a population of solutions (chromosomes) and applies genetic operators such as selection, crossover and mutation to evolve the solutions in order to find the best one. The three most important aspects of using genetic algorithms are (1) definition of the objective function, (2) definition and implementation of the genetic representation, and (3) definition and implementation of the genetic operators.

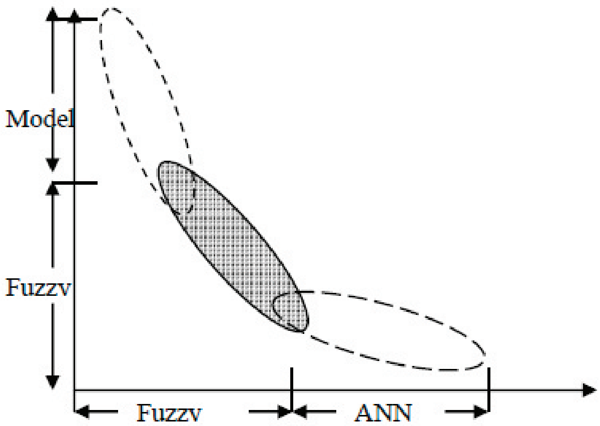
To increase the efficiency of the conventional genetic algorithm [11], the proposed genetic algorithm has initial coarse tuning characteristics by initially representing each parameter in a chromosome by a small bit number. If the parameters in a chromosome are represented by big bit numbers, the genetic algorithm can find the accurate optimal points in a limit of resolution but needs much more time to reach a convergence point. The genetic algorithm is a time-consuming algorithm. To represent the parameters in chromosomes by a big bit number from the beginning is not judiciable. However, representation of chromosomes (solutions) by a big bit number becomes necessary as they gradually approach the optimal points. Therefore, when the simulation generation reaches one third of the maximum generation, the bit number is increased by one third of its initial bit number. And then, when the simulation generation reaches two thirds of the maximum generation, the bit number is increased by two thirds of its initial bit number. By this method, the genetic algorithm has initial coarse tuning and final fine tuning characteristics.

The crossover site is selected by two ways when a chromosome contains many parameters. The first is that the crossover site is randomly selected anywhere in a chromosome. The second is that the crossover site is randomly selected between only parameters in a chromosome. This method slows an initial premature convergence without reaching optimal solutions and speeds up a final convergence. The first way increases diversity since the crossover site is randomly selected anywhere in a chromosome. Therefore, this way prevents a premature convergence without reaching optimal solutions. However, this way slows the final convergence due to excessive diversity in a final stage.

2 Structure of an Early Failure Detection System

The first technical method of failure detection signaled parameters that exceed the limit. To detect a failure before dangerous areas are reached. Simultaneously, locating it precisely, including the initial cause of the failure condition is the main concern nowadays.

Every time a problem is solved, knowledge about the given problem in form of measured data of correct or incorrect behavior or in the form of an analytical description of the system is used. An effort is required to obtain a sufficient amount of knowledge about the problem. The effort is approximately the same for all possible solutions. CI methods can be divided into the levels of the necessary amount of data and the necessary amount of analytical knowledge (as depicted in Figure), allowing selection of the most suitable method for the solution according to the specificity of the solved problem.



3. Conclusion

Soft computing offers interesting opportunities for automation, e.g. fault tolerance, association, self-organization, local adaptation, etc. Autonomy, an important feature for nanoelectronic components allows for the possibility of realizing autonomous soft computing systems that are self-testable, fault tolerant, adaptive, self-optimizing and self-programmable in order to auto adapt themselves to the environment, i.e. systems programmed by their outside world. The hardware requirements for such real intelligent systems are higher by several orders of magnitude. Only networked nanoelectronic systems with a high degree of parallelism offer a promising way to implement such complex applications. CI methodologies enable nanoelectronic systems with new and very special characteristics. Therefore the development of new systems is a joint task between technology development and system engineering. The best strategies meet in the middle, and the goal is product-oriented development.

References

A. Satyadas and K. Krishna Kumar, “GA-optimized fuzzy controller for spacecraft control,” *WCCI’94, Proc. of 3rd IEEE International Conf. on Fuzzy Systems*, 1994, pp. 1979-1984

D. Goldberg, “Genetic Algorithms in Search, Optimization & Machine Learning,” *Addison-Wesley, Reading* 1989

H.Schwefel, I.Wegner, K.Weinert (Eds.), “Advances in CI,” *Springer, Berlin Heidelberg NY,* 2002

J.H. Holland, **“**Adaptation in Natural and Artificial Systems,**”** *The Univ. Of Michigan Press, Ann Arbor,* 1975

J.S. Jang, “Self-Learning Fuzzy Controllers Based on Temporal Back Propagation,**”** *IEEE Transactions on Neural Networks,* Vol 3, No 5, 1992, pp. 714-723

Jin Yeong Yang and Kun Jai Lee, “Optimal Operation Planning of Radioactive Waste Processing System by Fuzzy Theory,” *Fuzzy Systems and Soft Computing in Nuclear Engineering,* Da Ruan (Ed.), Physica-Verlag, Springer

K. Goser, P. Glosekotter, J.Dienstuhl (Eds.), “Nanoelectronics & Nanosystems from Transistors to Molecular & Quantum Devices**,”** *Springer, Berlin Heidelberg NY,* 2004

L.A.Zadeh, **“**Fuzzy Logic, Neural Networks and Soft Computing,” *Communications of ACM,* Vol.37,No 3,1994, pp. 77-84

Mitchell, Melanie, “An Introduction to Genetic Algorithms,” *MIT Press, 1996*.

Na, Man Gyun **, “**A neuro-fuzzy inference system for sensor monitoring,**”** *Chosun Univ., Kwangju (Korea, Republic of)*

Priyanka Sethi, **“**A Review of Genetic Algorithm and its application for solving optimization problem in Power System,**”** *Proc. National Conference ECTTA-2005,* Oct 22-23, pp. 51-56

W.Kickert and E.H.Mamdani, “Analysis of a Fuzzy Logic Controller,” *Fuzzy Sets and Systems,* 12: 29-44, 1978

Miss Priyanka Sethi (M’13) The author became a Member (M) of Association of IISTE in 2013. The author was born in Jodhpur, Rajasthan (India) on 6 Nov, 1977. She completed Engineering from a reputed State Government University. She completed B.E. in Electrical in 2001 and M.E.in Electronics & Communications in 2007 from MBM Engineering College, JNV University, Jodhpur. She is specialised in Digital Communications. She is hard-working and pursuing career in teaching. She has taught more than 18 subjects in Electrical, Electronics & Communication and Computer Science & Engineering in a span of 12 years. She has 14 research paper publications to her credit. Her aim is to devise an autonomous system by application of Mathematical modeling and generic Computational methods. Her research interests include *Mathematical Modeling, Simulation using Soft Computing Methods and Signal Processing..*

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Recent conferences: http://www.iiste.org/conference/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar