# Innovative Design of Adaptive Hierarchical Fuzzy Logic Systems

Masoud Mohammadian
School of Information Sciences & Engineering
University of Canberra, Canberra, Australia
E-mail: Masoud.mohammadian@canberra.edu.au

## Abstract

*In this paper the supervised and unsupervised fuzzy concept learning using evolutionary Algorithms is considered. The paper explores the design and development of hierarchical fuzzy logic systems using an evolutionary algorithm. The development of hierarchical fuzzy logic systems is considered by a new method which determines the number of layers in the hierarchical fuzzy logic system. A hierarchical fuzzy logic system is developed to predict quarterly interest rates in Australia. The advantages and disadvantages of using hierarchical fuzzy logic systems for financial modelling is also considered. Finally evolutionary algorithm is then used to design a fuzzy logic system from a set of data in an unsupervised learning manner. Specifically it's application to urban traffic control is considered.*

## 1. Introduction

Modelling of uncertain dynamic systems, such as that for prediction of interest rates, has often relied on complex mathematical models to describe the dynamic system to be modelled. Mathematical models work well provided the system meets the requirement and assumption of synthesis techniques. There are however some uncertainty in real world applications that make them difficult to model and not easily adaptable to changes in the system which they were not designed for [7, 8, 10; 17]. Computational intelligence techniques such as fuzzy logic, evolutionary algorithms and neural networks have been successfully used in the place of complex mathematical systems [3, 7]. Fuzzy logic is an active research area [1, 2, 3; 7; 9; 10, 17]. Fuzzy logis system modelling or fuzzy logic identification has numerous practical applications in control, prediction and inference and it has been found useful when the system is either difficult to predict and or difficult to model by conventional methods. The majority of fuzzy logic systems have been static and based upon knowledge derived from imprecise heuristic knowledge of experienced operators, and where applicable also upon physical laws that governs the dynamics of the process [3, 16]. It is simply assumed that the fuzzy rules for fuzzy logic system are readily available or can be obtained. This implicit assumption limits the application of fuzzy logic to the cases of the system with a few parameters. The number of parameters of a system could be large. The number of fuzzy rules of a system is directly dependent on these parameters. As the number of parameters increase, the number of fuzzy rules of the system grows exponentially [11, 12, 13]. In fuzzy logic systems, there is a direct relationship between the number of fuzzy sets of input parameters of the system and the size of the fuzzy knowledge base (FKB). Kosko and Isaka [8] call this the "Curse of Dimensionality". The "curse" in this instance is that there is exponential growth in the size of the FKB, $k = m^n$ where $k$ is the number of rules in the FKB, m is the number of fuzzy sets for each input and $n$ is the number of inputs into the fuzzy system.

As the number of fuzzy sets associated with the input parameters increase, the number of rules increases exponentially. There are a number of ways that this exponential growth in the size of the FKB can be contained. The most obvious is to limit the number of inputs to the system. However, this may reduce the accuracy of the system, and in many cases, render the system being modelled unusable. Another approach is to reduce the number of fuzzy sets associate with each input variable. Again, this may reduce the accuracy of the system [7, 8]. The number of rules in the FKB can also be trimmed if it is known that some rules are never used. This can be a time-consuming and tedious task, as every rule in the FKB may need to be examined. Raju [13] suggested using a hierarchical fuzzy logic structure for such fuzzy logic systems to overcome this problem. By using hierarchical fuzzy logic systems the number of fuzzy rules in the system are reduced thereby reducing the computational time while maintaining system robustness and efficiency.

In this paper the design and development of a hierarchical fuzzy logic system using evolutionary algorithms for prediction of interest rate in Australia is

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

considered. Evolutionary algorithms are employed as an adaptive method for design and development of such hierarchical fuzzy logic systems [1, 2; 15]. In this paper we shall show that they can be easily utilized to find the input and layer structure of a "best performing" hierarchical structure for interest rate prediction.

## 2. Hierarchical Fuzzy Logic Systems

A hierarchical fuzzy logic structure is formed by having the most influential inputs as the system input variables to the first level of the hierarchy, the next most important inputs to the second layer, and so on. If the hierarchical fuzzy logic structure contains $n$ system input parameters and $L$ number of hierarchical levels with $n_i$ the number of variables contained in the $i$th level, the total number of rules $k$ is then determined by: $k = \sum_{i=1}^{L} m^{n_i}$, where $m$ is the number of fuzzy sets. This means using a hierarchical fuzzy logic structure, the number of fuzzy rules for the system is reduced to a linear function of the number of system variables $n$, instead of an exponential function of $n$ as is the conventional case [13]. The first level of the hierarchy gives an approximate output, which is then modified by the second level rule set, and so on. This is repeated for all succeeding levels of the hierarchy. One problem occurs when it is not known which inputs to the system have more influence than the others. This is the case in many problems. In some cases statistical analysis could be performed on the inputs to determine which ones have more bearing on the system.

## 3. Learning of Hierarchical Fuzzy Logic Systems

Evolutionary Algorithms (EAs) [19], are powerful search algorithms based on the mechanism of natural selection and use among other operations of reproduction, crossover, mutation and other operations on a population of strings. A set of possible solutions is the population. In this case each solution is a coding of the fuzzy rules of a fuzzy logic system, represented as a string of numbers. New strings are produced every generation by the repetition of a two-step cycle. First, each individual string is decoded and its ability to solve the problem is assessed. Each string is assigned a fitness value, depending on how well it performed. In the second stage the fittest strings are preferentially chosen for recombination to form the next generation. Recombination involves the selection of two strings, the choice of a crossover point in the string, and the switching of the segments to the right of this point, between the two strings (one-point cross-over). For encoding and decoding of the fuzzy rule the output fuzzy sets are encoded. (It is not necessary to encode the input fuzzy sets because the input fuzzy sets are assumed static and do not change.) Assume that the output fuzzy sets are: NB (Negative Big), NS (Negative Small), ZE (Zero), PS (Positive Small), and PB (Positive Big). Then the output fuzzy sets are encoded by assigning 1 = NB (Negative Big), 2 = NS (Negative Small), 3 = ZE (Zero), 4 = PS (Positive Small), and 5 = PB (Positive Big). A string or individual in the population, representing all the fuzzy rules can then be defined as:

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 3 | 5 | 3 | | | 1 |
| 1 | 2 | 3 | 4 | | | 25 |

The initial population of the evolutionary algorithm is obtained by randomly encoding a set of strings. The fitness of each string is obtained by decoding each string into the fuzzy logic structure and determining a value defining its performance on the data given. In developing the population at the next generation, elitism is used with two or more copies of the best performing string from the parent generation included in the next generation to ensure that the best performing strings are not lost. The processes of selection, crossover and mutation are then performed to add further individual strings to complete the population at this next generation. Selection and crossover are the same as in simple genetic algorithms [4] while the mutation operation is modified as explained below. Crossover and mutation take place based on the probability of crossover and mutation respectively. The mutation operator is changed to suit this problem, namely, an allele is selected at random and it is replaced by a random number ranging from 1 to 5. The processes of selection, crossover and mutation are repeated for a number of generations till a satisfactory fuzzy rule base is obtained. A satisfactory rule base is one whose fitness value differs from the desired output of the system by an acceptable small value.

## 4. Hierarchical Fuzzy Logic System Development for Interest Rate Prediction

Economists, and investors, have been unable to find all the factors that influence interest rate fluctuations. It is agreed that there are some major economic indicators released by the Government [5]

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

that are commonly used to look at the current position of the economy. The indicators, see [11, 12] used in this paper are: Interest Rate, Job Vacancies, Unemployment Rate, Gross Domestic Product, Consumer Price Index, Household Saving Ratio (the ratio of household income saved to households disposable income), Home Loans (measure the supply of finance for home loans not the demand for housing), Average Weekly Earnings (is the average amount of wages that a full time worker takes home before any taxes), Current Account (is the sum of the balances on merchandise trade, services trade, income and unrequited transfers), Trade Weighted Index (measures changes in our currency relative to the currencies of our main trading partners), RBA Commodity Price Index (provides an early indication of trends in Australia's export Prices), All Industrial Index (provides an indication of price movements on the Australian Stock Market), Company Profits (are defined as net operating profits or losses before income tax) and New Motor Vehicles is the number of new vehicles registered in Australia.

A fuzzy logic system that use all above indicator and had five fuzzy sets associated with every indicator would result in a large FKB consisting of over six billion rules! This would require large computing power to not only train the fuzzy logic system with a evolutionary algorithms, but also large storage and run-time costs when the system is operational. To overcome this problem a hierarchical fuzzy logic structure for the fuzzy logic system can be constructed to reduce the number of fuzzy rules of the system. The input variable described above are grouped together to build a fuzzy knowledge base for each group. The first step is to divide the indicators into smaller-related groups [11, 12] such as: Employment Group (Job Vacancies, Unemployment Rate), Country Group (Gross Domestic Product, Consumer Price Index), Savings Group (Household Saving Ratio, Home Loans, Average Weekly Earnings), Foreign Group (Current Account, RBA Index, Trade Weighted Index), Company Group (All Industrial Index, Company Profit, New Motor Vehicles)

The five fuzzy knowledge bases created form the top layer of the hierarchy are shown in Figure 2. The authors designed and connected together the fuzzy knowledge bases to form a final fuzzy knowledge base system. The final fuzzy knowledge base is shown in Figure 1; it uses the predicted interest rate from the five above groups to produce a final interest rate prediction. The number of fuzzy rules for each group is shown in Figure 1. The final hierarchical FKB in this system contains 3125 rules giving the total number of rules to be learnt as 5250. This is a significant reduction from the 6 billion rules that would have been obtained using a traditional fuzzy knowledge base. This allows quicker training time without the need for huge computer resources [10, 11]

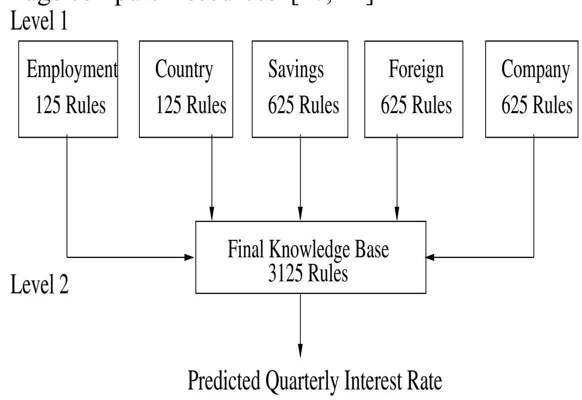



Figure1. Hierarchical Knowledge Base Flow

The interest rate is included as input to each of the groups above. To learn the fuzzy knowledge base for each group, a evolutionary algorithm was implemented as described above. The evolutionary algorithm had a population size of 500 with a crossover rate of 0.6, a mutation rate of 0.01, and it was run for 10000 generations over 10 years (a period of 40 quarters) of data. The fitness of each string was calculated as the sum of the absolute differences from the predicted quarter and the actual quarters interest rate.

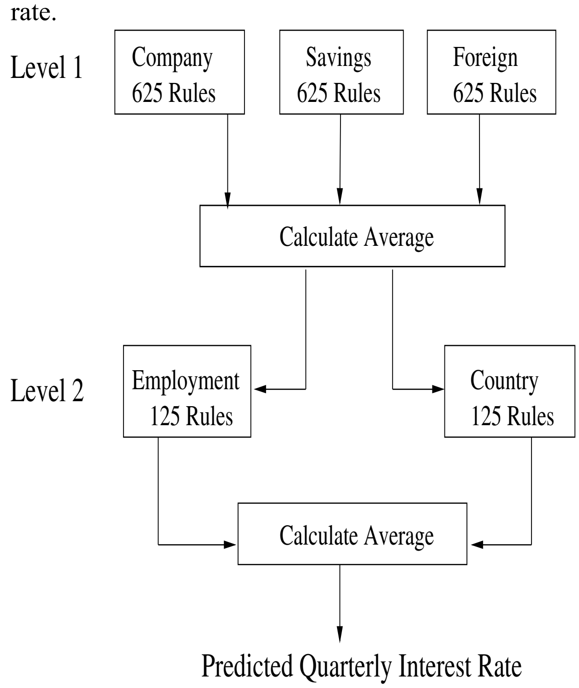



Figure 2. A three-layer hierarchical fuzzy logic system with 3125 fuzzy rules

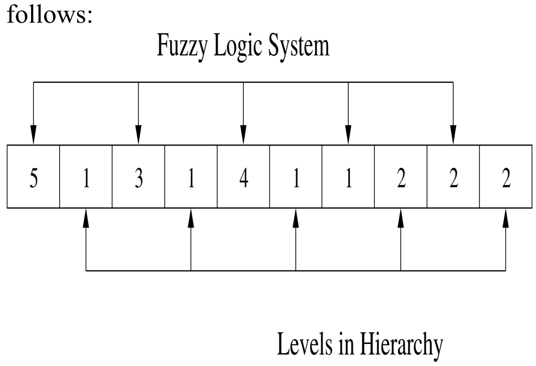
Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

The fitness was subtracted from an 'optimal' fitness amount, which was decided to be 30 as it was unlikely the error amount would be higher than this over 10 years [11, 12]. The fitness is thus calculated by the following formula:

$$fitness = 30 - \sum_{i=0}^{30} abs(PI_i - I_{i+1}).$$

Good prediction of Australian quarterly interest rate was obtained using the above system, see [11, 12]. The number of fuzzy rules used has been also reduced dramatically. However, there is still a question: Does a two layer hierarchical architecture provide the best solution?

We choose to develop hierarchical structures that have the form shown in Figure 2. Using the economic indicators five fuzzy logic systems were developed using the genetic algorithm approach described, one for each of the five groups described above. Each system produces a predicted interest rate for the next quarter. For encoding and decoding of the hierarchical fuzzy logic system, first a number is allocated to each fuzzy logic system developed from the group of indicators. For this simulation the number allocated to each group see below: that is: 1 = Employment, 2 = Country, 3 = Savings, 4 = Foreign, 5 = Company. There are five layers possible so we can encode a string or individual in the population as follows:




Here Company (5) is in layer 1 of the hierarchy, Savings (3) is in layer 1 of the hierarchy and so on, finally Country (2) is in layer 2. We then used evolutionary algorithms to learn the number of layers as well as position of FKB of each group in each layer of hierarchical fuzzy logic system for interest rate prediction. For this string we evaluate its fitness based on the decoded hierarchical structure. It is the average error of the system for the training set and tests sets using the following formula [11, 12].

$$E = \frac{\sum_{i=1}^{n} abs(P_i - A_i)}{n}$$

where $E$ is the average error, $P_i$ is the predicted interest rate at time period $i$, $A_i$ is the actual interest rate for the quarter and $n$ is the number of quarters predicted. The initial population of the evolutionary algorithm is created with strings whose integer elements are randomly chosen between 1 and 5. The evolutionary algorithm is allowed to evolve as previously described until a satisfactory hierarchical fuzzy logic system defined by a small acceptable error $E$ is obtained. The results of the top performing five hierarchical fuzzy logic systems designed by genetic algorithms are given in Table 1.

| | Training Error | Testing Error |
|---|---|---|
| HFL #1 | 0.356 | 0.659 |
| HFL #2 | 0.343 | 0.663 |
| HFL #3 | 0.289 | 0.494 |
| HFL #4 | 0.274 | 0.441 |
| HFL #5 | 0.291 | 0.398 |

Table 1. Comparison of Average Errors of Hierarchical Fuzzy Logic Systems (HFL)

The best performing structure is HFL #5, in terms of training and test errors. This is a much improved result when compared with the hierarchical model in Figure 1 which gave a training error of 0.402 and test error of 0.465.

## 6. Unsupervised concept learning using Genetic Algorithms

In this section, the concept of unsupervised learning, using evolutionary Algorithm, is illustrated by generating the fuzzy rules for coordinating two adjacent urban traffic signals. The traffic flow approaching a set of two intersections *A*, and *B*, is regulated by adjusting the signal timing parameters, green phase split and offset, at each intersection. Phase split is the division of the cycle time into periods of green phase for competing approaches and offset is the time difference in the starting times of the green phases of adjacent intersections. A mean vehicle arrival rate is assigned to each approach and at each successive time unit (one sec), a random number is generated and compared with the vehicle arrival rate to decide the arrival of a vehicle. At each signalised intersection sensors to count the number of vehicles are used instead of proximity sensors which only indicate the presence of a vehicle. Vehicle densities are taken from two sensors placed on the road, one at the intersection and the other at 100 metres from the intersection. The rear sensor increments a counter every time a vehicle passes over it, while the forward sensor decrements the same counter. This gives a count of the number of

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

vehicles waiting 100 metres before the light and a count of the number of vehicles that passed through the intersection when the light was green. A Fuzzy Logic Traffic Controller (FLTC) comprising twenty five fuzzy rules, developed manually, is used to adjust the green phase splits of the three traffic signals. The number of vehicles waiting (queue length) at the end of a red phase and the number of vehicles that passed through the intersection in the previous green phase, are the deciding factors in adjusting the green phase split of the signal. The FLTC regulates the traffic by making on-line adjustments to the green phase duration of the traffic light. The input variables are:

(**i**) The ratio of queue length to number of vehicles that passed through, in the East-West approaches.

(**ii**) The ratio of queue length to number of vehicles that passed through, in the North-South approaches.

The output variables are :

(**i**) The amount of adjustment to the current green phase of the North-South approach.

(**ii**) The amount of adjustment to the current green phase of the East-West approach

A supervisory fuzzy logic controller comprising a set of 25 fuzzy rules is developed to coordinate the two intersections *A* and *B*. Evolutionary algorithms as described earlier was employed to acquire the fuzzy rule base for adjusting the offset. Evolutionary algorithm is employed to acquire the fuzzy rule base for adjusting the offset at the two intersections called intersection *A* and intersection *B*. The input variables for the supervisory fuzzy logic controller are *Vol_diff 1* and *Vol_diff2* where

$$\text{Vol\_diff 1} = V_{SB} - (V_{EA} + V_{WA}) / 2$$
$$\text{Vol\_diff 2} = V_{NA} - (V_{EB} + V_{WB}) / 2$$

$V_{SB}$ is the queue length at the south approach of intersection *B*, $V_{NA}$ is the queue length at the north approach of intersection *A*, $V_{EA}$ and $V_{WA}$ are the queue lengths at the east and west approaches of intersection *A*, and $V_{EB}$ and $V_{WB}$ are the queue lengths at the east and west approaches of intersection *B*. The output variables are the offset adjustments to the green phase of the two traffic signals, *Ext1* and *Ext2*. The fuzzy rule base is evaluated by the supervisory fuzzy logic controller based on a fitness function. The fitness function is the sum of all the vehicles waiting at the north and south approaches of the two intersections during the simulation. It is desired to generate a fuzzy rule base that minimizes the fitness function. The evolutionary algorithm operators - reproduction, crossover, and mutation are then applied to the individual strings of the population based on the fitness value. This process is repeated for a number of generations till a suitable fuzzy rule base is obtained. A suitable fuzzy rule base is one which minimizes the queue length at the north and south approaches of both intersections. The fuzzy control rules for adjusting the offset of the two traffic signals is developed by running the evolutionary algorithm for 300 generations using a population size of 30 and a crossover and mutation probability of 0.6 and 0.015 respectively. The total number of vehicles waiting at intersection *A* is reduced by 39% and the total number of vehicles waiting at intersection *B* is reduced by 20%. There is a significant reduction in the queue length when the fuzzy rules generated by evolutionary algorithm is used to adjust the offset of the two traffic signals, *A* and *B*. The fuzzy rule base obtained after 300 generations for a population size of 30 is shown in Table2.

| | VL | LO | MD | HI | VH |
|---|---|---|---|---|---|
| VL | **SM**<br>**VS** | **VH**<br>**VS** | **SM**<br>**SM** | **VH**<br>**HI** | **SM**<br>**HI** |
| LO | **VS**<br>**VH** | **SM**<br>**VH** | **SM**<br>**VH** | **VS**<br>**SM** | **VH**<br>**HI** |
| MD | **VS**<br>**SM** | **VS**<br>**VH** | **MD**<br>**SM** | **VH**<br>**HI** | **SM**<br>**SM** |
| HI | **MD**<br>**HI** | **MD**<br>**MD** | **HI**<br>**MD** | **VS**<br>**VH** | **MD**<br>**MD** |
| VH | **MD**<br>**MD** | **SM**<br>**HI** | **VH**<br>**VH** | **VS**<br>**VH** | **HI**<br>**HI** |

Table 2 Fuzzy rule base for the supervisory FLC, adjusting offset

It can thus be seen from the simulation results that the integrated FKB for fuzzy logic system to control two adjacent intersections optimizes the performance of the entire system as a whole. It tries to minimize the total queue length at all the intersections. Table 3 shows the effectiveness of the fuzzy rules generated by evolutionary algorithm over the fuzzy rule base constructed by hand. Evolutionary algorithm attempts to optimize the overall performance of the system. It minimizes the number of vehicles waiting at all the approaches to intersections *A* and *B* rather than just reducing the queue length at the north approach of intersection *B* and the queue length at the south approach of intersection *A*.

| | Fuzzy rules generated by hand | Learnt fuzzy rules |
|---|---|---|
| QL at the north approach of intersection A | 2323 | 802 |
| QL at the south approach of intersection A | 562 | 436 |
| QL at the north approach of intersection B | 441 | 543 |
| QL at the south approach of intersection B | 1529 | 778 |
| QL at all four approaches of intersection A | 4253 | 2595 |
| QL all four approaches of intersection B | 2974 | 2375 |

Table 3 Comparison of the fuzzy rules constructed by hand and rules generated by EA

When the fuzzy rules are generated by evolutionary algorithm, the queue length at the north approach of intersection *A* is reduced by 62%, the queue length at the south approach of intersection *A* is reduced by 22%, the queue length at the north approach of intersection *B* is increased by 18% and the queue length at the south approach of intersection *B* is reduced by 49%. The total number of vehicles waiting at intersection *A* is reduced by 39% and the total number of vehicles waiting at intersection *B* is reduced by 20%. There is a significant reduction in the queue length when the fuzzy rules generated are used to adjust the offset of the two traffic signals, *A* and *B*.

## 7. CONCLUSION

A method is used to design and develop hierarchical fuzzy logic systems. Evolutionary algorithm is used as an adaptive learning method to design a hierarchical fuzzy logic system to predict the fluctuations of the 10-year Australian Treasury bond using Australian economic data. Hierarchical fuzzy logic systems is used to reduce the number of fuzzy rules in the system. Simulation results are promising and the resulting hierarchical fuzzy logic systems obtained are capable of making accurate predictions of the following quarter's interest rate. The research work performed in this paper is unique in the way the hierarchical fuzzy logic systems are developed. However, it should be noted that the learning of layers in hierarchical fuzzy logic systems may not be possible for all hierarchical fuzzy logic systems. Further research needs to be undertaken on the full input structure with all fourteen input variables, maximum number of layers possible in the hierarchy being fourteen. Research has recently been presented which has fully investigated the possible hierarchical structures for the control of the inverted pendulum [18]. However it was noted that in this case due to the dynamics of this system it may not be possible to learn the layers of hierarchical fuzzy system. Research work is under progress to investigate and develop a universal method using evolutionary algorithms that can overcome the current limitations. The application of this method to several industrial problems such as robotic control and collision avoidance of multi-robot systems is also currently under consideration.

The author acknowledges the initial work performed by Mark Kingham, Irshad Nianar and involvement of Russel J. Stonier in this research project.

## 8. REFERENCES

[1] M. M. Anver, R. J. Stonier, “Evolutionary learning of a fuzzy edge detection algorithm based on multiple masks”, Proceedings of the7th Asia-Pacific Complex Systems Conference - Complex 2004. Cairns, (pp. 183-195). Cairns, Australia.

[2] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, “Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases”, A*dvances in Fuzzy Systems: Applications and Theory,* World Scientific Publishing Vol. 19, 2001.

[3] E. Cox, “Adaptive fuzzy systems” *IEEE Spectrum*, 27-31, 1993.

[4] D. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Reading, Massachusetts: Addison Wesley, USA, 1989.

[5] S. Goonatilake, J. A. Campbell, N. Ahmad, “Genetic-Fuzzy Systems for financial decision making. In *Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms, IEEE/Nagoya-University World Wisepersons Workshop*. Lecture Notes in Artificial Intelligence, Springer, USA, 1995.

[6] S. Goonatilake, P. Treleaven, *Intelligent Systems for Finance and Business*, Wiley and Sons, USA, 1995.

[7] B. Kosko, *Neural networks and fuzzy systems, a dynamic system*. Englewood Cliff: Prentice-Hall. USA, 1992.

[8] B. Kosko, S. Isaka, “Fuzzy Logic” *Scientific American*, 76-81. 1993.

[9] C. C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Controllers - part I, part II.” *IEEE Transactions on Systems, Man and Cybernetics*, 2092, 404-435, 1990.

[10] m. Mohammadian, R. J. Stonier, “Adaptive two layer control of a Mobile Robot Systems”, IEEE International Conference on Evolutionary Computing, Perth, Australia, Vol.1. (pp. 204-209), 1995.

[11] m. Mohammadian, M. Kingham, “Hierarchical Fuzzy Logic for Financial Modelling and Prediction”, Tenth Australian Joint Conference on Artificial Intelligence, Perth, Australia, 1997.

[12] M. Mohammadian, M. Kingham, B. Bignall, “Hierarchical Fuzzy Logic for Financial Modelling and Prediction”, *Journal of Computational Intelligence in Finance,* UK, 1998.

[13] G. V. S. Raju, J. Zhou, “Adaptive Hierarchical Fuzzy Controller” *IEEE Transactions on Systems, Man & Cybernetics*. 23(4), 973-980, 1993.

[14] d. Ruelle, Chaotic Evolution and Strange Attractors: The Statistical Analysis of Time Series for Deterministic Nonlinear Systems, Cambridge University Press, 1989.

[15] P. T. Thomas, R. J. Stonier, "*Hierarchical Fuzzy Control in Robot Soccer using Evolving Algorithms*", Proceedings of the International Congress on Evolutionary Computation (CEC2003), Vol. 4. Canberra, Australia, pp. 2434-2440, 2003.

[16] T. Welstead, *Neural networks and fuzzy logic applications in C/C++*, Wiley and Sons. New York, 1994.

[17] L. Zadeh, Fuzzy Sets. *Journal of Information and Control*, 8, 338-353. 1965.

[18] J. Zajaczkowski, R. J. Stonier, "Analysis of hierarchical control for the inverted pendulum", Proceedings of the 7th Asia-Pacific Complex Systems Conference (Complex 2004), Cairns, Australia, (pp. 350-374). 2004.

[19] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, GermanY, 1998.