EVOLUTION OF FUZZY LOGIC SYSTEM USING GENETIC ALGORITHMS FOR TIME-SERIES FORECASTING

Shivani Kakkar[1],Satvir Singh[2],Sarabjeet Singh[3],Jaspreet Kaur[4],Vijay Kumar Banga[5]
[1,2,3,4]SBS State Technical Campus, Ferozepur, (Punjab), India, [5]Amritsar College of Engg. & Tech., Amritsar, (Punjab), India
Email: [1]kakkarshivani47@yahoo.in, [2]Drsatvir.in@gmail.com, [3]sarabjeet_singh13@yahoo.com, [4]er.jaspreetkaur1@yahoo.com, [5]v_banga@rediffmail.com

Abstract— As the number of possible rules for a Fuzzy Logic System(FLS) increases, its complexity also increases. So, the work highlights the method to evolve the best possible rules using one of the evolutionary algorithm technique i.e. Genetic Algorithm(GA) for Mackey-Glass Time Series(MGTS) forecasting problem. As FLS deals with uncertainty so it can be easily employed for this application keeping in mind the chaotic nature of the MGTS.GA is known to provide powerful search mechanism that can be used in optimization application in order to evolve the best possible solution to a problem. So, integrating both FLS and GA would end up with a best possible output to the problem. In this work the methodology employed for the minimization of the fitness function i.e. Mean Square Error(MSE) has been discussed and at last the best rulebase corresponding to the least MSE has been fixed into the designed FLS in order to get the output closer to the desired output.

***Index Terms*—Fuzzy Logic System(FLS), Genetic Algorithm(GA), Type-1, Type-2, Mean Square Error (MSE).**

I. INTRODUCTION

This paper presents method of integrating Fuzzy Logic system and Genetic Algorithm (GA) to solve the problem of forecasting the Mackey-Glass chaotic time series. GA is a swarm based global search algorithm inspired by natural mechanism of genetical improvement in biological species [1],described by Darwinian Theory of "Survival of Fittest" [2]. So, GA is widely used as an optimization tool in several fields such as medical, engineering and finance etc [3]. Though the FLS has the ability to provide imprecision, uncertainty and human oriented knowledge yet the need for the hybridization of Fuzzy Logic approach with GA arose due to lack of possibility for Fuzzy Logic for self learning and generalization of rules [4]. The work highlights the framework to evolve Fuzzy rules automatically using GA and also the advantages of hybridization of Fuzzy Logic systems and Genetic Algorithm have been discussed. In the application of time-series forecasting for the given time-series x(t) if x(t-3), x(t-2), x(t- 1), x(t) are given then we need to predict x(t+1) [5]. So, we have been using 4 input variables and 1 output variable for a given FLS with each input variable characterized by 4 fuzzy sets [5]. From 256 possible rules we need to use only a portion of rules [6] which are required to be evolved using this method in order to obtain time-series with minimum possible error.

A. Mackey-Glass Time-Series

Mackey-Glass Time Series is basically required for time-series forecasting in order to predict the behavior of the various dynamic systems like weather and climate etc. The

Mackey- Glass Time Series is generated using the following delay differential equation:-

$$dx(t)/dt = ((0.2x(t-\tau)/1 + x^{10}(t-\tau)) - 0.1x(t) \quad (1)$$

In this paper we have designed a 4-input, 1-output FLS in order to evolve rules using 500-input/output pairs and then the system is tested for next 500 inputs.The graph of Mackey- Glass Time Series is shown in Fig.1.

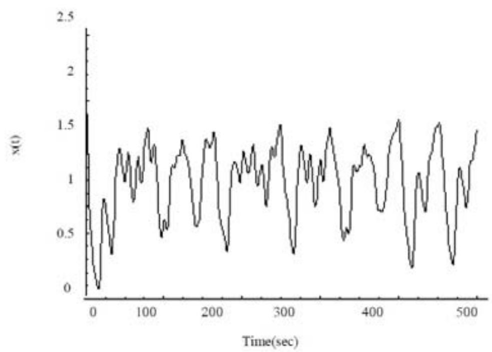


Fig. 1: Mackey-Glass Time Series [7]

II. RELATED WORKS AND OBJECTIVES

The chaotic nature of the Mackey-Glass Time Series makes its prediction uncertain because of the noise in the inputs available from the time series data. So, in 1999 Karnik et. al. and Mendel et. al. presents an approach to forecast time-series by incorporating information about noise strength into Type-2 FLS [5]. Using Type-2 FLS the bounds on the output was obtained within which the true output likely to lie [5]. The performance of the system was evaluated using a parameter i.e. Mean Square Error (MSE). The MSE between the crisp outputs obtained from the designed system and the true outputs was calculated for three different Signal to Noise Ratios (SNR) i.e. 1.34×10^{-2}, 4.38×10^{-3} and 1.59×10^{-2} for 0 db, 10 db and 20db SNR [5]. The minimum value of Mean Square Error (MSE) in case of 0db SNR was found to be 1.34×10^{-2} . In 1991 P.Thrift et. al. [8] and in 1994 Hwang et.al. and Thompsom et.al. [9]in their papers presented a methodology for combining genetic algorithms and Fuzzy Logic Systems for learning the optimal rules while fixing the shapes of the membership functions. So, the objective of this paper is to provide an alternative method of training Type- 1 FLS using Genetic Algorithms with 0 db SNR in order to forecast the time-series with the minimum possible MSE i.e. even less then 1.34×10^{-2}. Various applications using this methodology have been experiencing a recent acceleration and accuracy as the application becomes more and more complex [8]. After clarifying briefly our objectives in this paper the remaining paper is organized as follows: Section III gives the brief introduction to Type-1 and Type-2 FLS, section IV depicts the flowchart describing the working of the Genetic Algorithms followed by section V explaining the implementation of evolving Fuzzy rules using Genetic Algorithms along with its detailed flowchart. Results and the performance evaluation of the system are depicted in section VI and at last the conclusion and future work of the paper is discussed in section VII.

III. FUZZY LOGIC SYSTEM

Fuzzy Logic Systems introduced by Lotfi. A. Zadeh [10] [11] was discovered in order to provide us with a provision to deal with uncertainty or the knowledge which do not have well defined sharp boundaries. In Fuzzy Logic Systems, fuzzy sets are characterized by membership functions mapped between [0,1] [12]. For example temperature can be represented in the form of different fuzzy sets depending upon its range such as too cold, cold, warm, hot, too hot [13]. Different types of membership functions are Gaussian, triangular, trapezoidal, piecewise linear and singleton where the choice of membership function depend upon the type of application we are using. Every Fuzzy Logic System use a set of If-then rules called Rulebase where if part contain a condition and then part contain a conclusion. For example if the temperature is hot then the command is cool. Inference is performed by evaluating and combining various fuzzy rules using fuzzy set operations in order to get the output fuzzy set from which a single crisp output is obtained using defuzzification.

A. *Type-1 Fuzzy Logic System:*

The basic architecture of Type-1 Fuzzy Logic System is shown in Fig.2. We start with fuzzification where crisp values of input data are converted to fuzzy sets using membership functions. After performing inference on the set of if-then rules the resulting output fuzzy set is used to obtain single crisp output using defuzzification. So, the Type-1 Fuzzy Logic System does not have the ability to handle uncertainty over uncertainty. In order to deal withsuch kind of uncertainty another type FLS known as Type-2 FLS are used [11].

B. *Type-2 Fuzzy Logic System:*

Fuzzy sets models words that are being used in rulebase and inference engine. However, word mean different thing to different people and, therefore, are uncertain. Membership degree of a Type-1 fuzzy set cannot capture uncertainties about the words. Hence, another type of fuzzy set, i.e., Type-2 fuzzy Sets, came into existence which is capable of handling such uncertainties. For such a fuzzy set membership value corresponding to some crisp input is not a crisp value rather a Type-1 fuzzy set called secondary membership [14] [13] . This concept can be extended to Type-n fuzzy sets. Computations based on Type-2 fuzzy sets are very intensive, however, when secondary membership is assumed unity the computational burden reduces drastically. This is another variant to fuzzy set representation and is known as Interval Type 2 fuzzy sets [13] [15]. Type-2 membership function is as shown in fig.

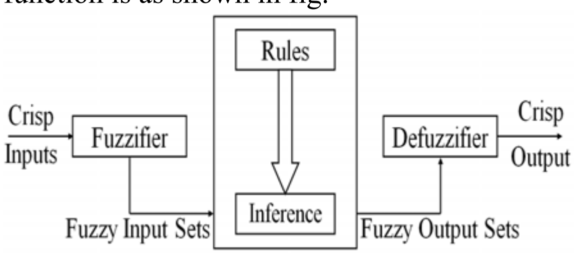


Fig. 2: Type-1 Fuzzy Logic System [10]

IV. GENETIC ALGORITHMS

The detailed flowchart depicting the Genetic Algorithm is shown in Fig.3.:
Genetic Algorithm developed by John Holland in 1970 [2] is
a swarm based global search algorithm inspired by natural mechanism of genetical improvement in biological species [1], described by Darwinian Theory of Survival of Fittest [16]. GA is widely used as an optimization tools in various fields such as medical, engineering and finance and can also be used for the purpose of automatic evolution using crossover, mutation and survival of the fittest [16]. GA starts with initialization of randomly generated population consisting of vectors of chromosomes [6]. After that the fitness for each chromosome in the population is calculated using some standard fitness function. A new population is generated by applying crossover and mutation over selected chromosomes. Selection is most commonly performed using a ”Roulette wheel” mechanism. The next step crossover involves the interchanging of some values of 2 parent chromosomes depending upon the crossover probability [6]. Finally mutation is performed which stands for changing the values of the elements of the population randomly using mutation probability. The all new population is generated now which is copied back to the initial population in order to calculate the new fitness values. The new population will yield the improved values of fitness. The whole algorithm repeats until some required condition is not met.

V. IMPLEMENTATION OF EVOLUTIONARY FUZZY SYSTEMSUSING GENETIC ALGORITHM

Genetic algorithms have demonstrated to be a robust and very powerful tool to perform tasks such as the generation of fuzzy rule base, optimization of fuzzy rule bases [17]. All these tasks can be considered as optimization or search processes within large solution spaces [18] [19]. The detailed flowchart explaining the implementation is shown in Fig.4.

A. *Initialization of population*

In the approach of evolving rules using GA the 20 different rulebases with each rulebase containing 10 rules are randomly
generated keeping in mind the range of antecedents and consequents of the rulebase of the FLS system to be designed
for time-series forecasting. The randomly generated rulebases
are then copied into the GA population with each rulebase representing each chromosome of the population. So, in the start of the algorithm each chromosome of GA population is encoded with the rule set of FLS [6]. At this step we cannot decide the number of rules to be included in the rulebase so we have just assumed the number of rules to be included in each rulebase.

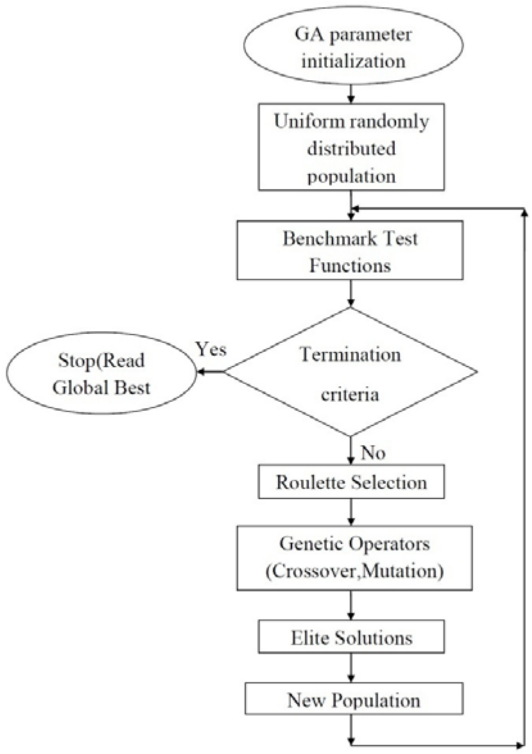


Fig. 3: Genetic Algorithm Flowchart [16]

B. Fitness Function

The fitness function here is evaluated using a type-1 FLS designed using 4 input and 1 output variables for time-series prediction. Each input variable is fuzzified using 4 Gaussian membership functions characterized by two parameters i.e. mean (m) and standard deviation (σ) [20]. Gaussian membership function is represented by:-

$$\mu(x) = \exp(-(x-m)^2 / 2\sigma^2) \qquad (2)$$

Where x stands for the input values. The rule sets randomly generated and encoded as GA population are fired using inputs available from the training data. Min and max operators are employed for implication and aggregation. Defuzzification is performed with the Height Defuzzification method [20]. Output of Height Defuzzification is given by:-

$$y = \sum_{i=1}^{M} C_i \mu(x_i) / \sum_{i=1}^{M} \mu(x_i) \qquad (3)$$

where M stands for number of rules, *C* represents the location of singleton consequent fuzzy sets and $\mu(x_i)$ stands for the clipping level after implication. Corresponding to 500 inputs available from the training data of time-series we will obtain 500 outputs from the designed FLS. From time-series prediction problem the commonly used fitness function is Mean Square Error (MSE) which is calculated as below:

$$MSE = \sum_{i=1}^{N} (y_i - d_i)^2 / N$$
(4)

where N is the number training data, y is the output obtained from the designed FLS and d is the true output from the system.

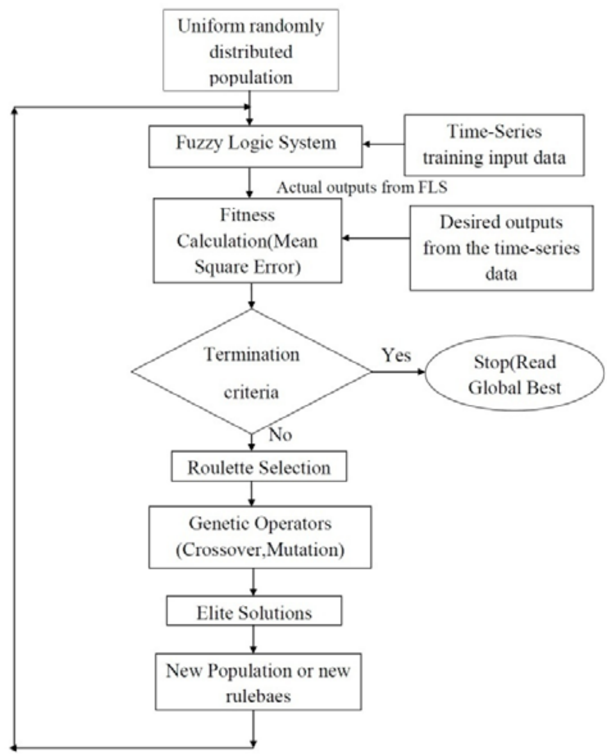


Fig. 4: Flowchart of Evolutionary Fuzzy Systems using Genetic Algorithm

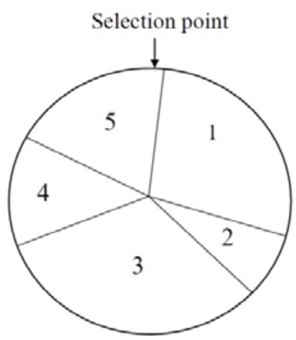
C. Selection.

Selection is basically carried out in order to prefer the fittest solutions for the next generation. Here the method used for selection is “Roulette Wheel” selection where on rotating the
Wheel once the probability of each parent chromosome being
chosen depends upon its fitness. The more will be the fitness
of the parent chromosome, the more area will it cover on the “roulette wheel” hence has the more probability of being selected [16] [6]. Hence selection favors the concept of the “Survival of

the Fittest". The mechanism of selection is shown in Fig.5.

D. Crossover

After selection the crossover between the parent chromosomes is carried out resulting into two new better off springs [6]. In order to carry out crossover between two individuals the crossover probability was first defined at 0.9 and a uniform random number (r) between 0 and 1 was generated. If the uniform random number, r generated is less than the crossover probability then the crossover between the two individuals takes place otherwise no crossover takes place and the original copies of parent chromosomes is reproduced in the next generation. Uniform crossover as shown in Fig.6. is carried out at gene level using a mixing ratio of 0.8.



Population	Fitness
1	25
2	5
3	40
4	10
5	20

Fig. 5: Roulette Wheel Selection

E. Mutation

After crossover the Mutation is carried out in order to reproduce better and better chromosomes with more improved solutions then that produced by crossover as with one point crossover their is a probability of two parents having same string at a given gene and the population will have the same string at a given gene and the population will have the same string forever if mutation is not carried out. Here the mutation is carried out by varying the elements of the parent chromosomes. The elements here are varied by 0.08 depending upon the uniform random number generated either greater then or smaller then the mutation probability. The validation check is applied on the newly generated population depending upon the range of the antecedents and the consequents of a FLS.

The newly generated population is fed to the rulebases of the FLS in order to calculate the new improved fitness values. The algorithm repeats until the condition for fitness value is not met.

At last the global best solution is used as a single rulebase and fed to the designed FLS in order to get time-series output closer to the desired output.

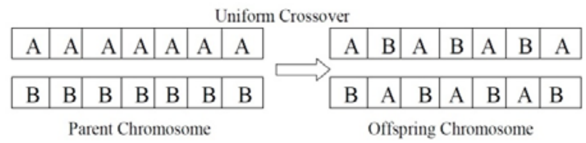


Fig. 6: Uniform Crossover

VI. RESULTS AND DISCUSSIONS

The system code was written in Visual Studio C++(2012 Release Mode) and was compiled on nvcc compiler. The results of the above experiments were evaluated using 1000 iterations. The dimension size of the experiment is kept fixed. On evolving the system using 1000 iterations the MSE which is the performance parameter of the system was minimized up to approximately 1.19 $\times 10^{-2}$ for 1000 iterations of GA and 1.17×10^{-2} for 10,000 iterations and the resulting time series obtained were very close to the desired time-series. The minimization of the MSE for 1000 and 10,000 iterations are as shown in Fig.7. and Fig.8.

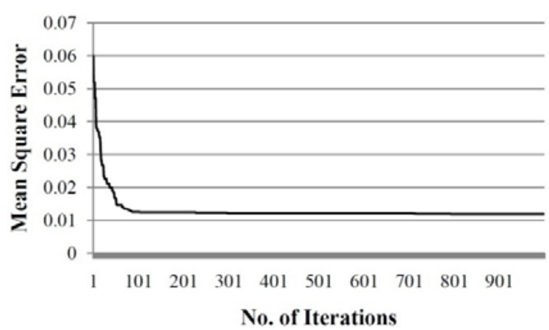


Fig. 7: Mean Square Error minimization for 1000 iterations.

The graph showing the comparison between the time-series output obtained from the designed system and the true time series is shown in Fig.9.

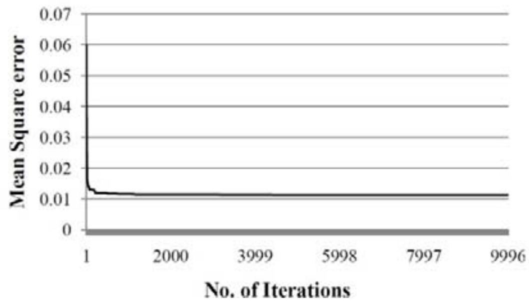


Fig. 8: Mean Square Error minimization for 10,000 iterations.

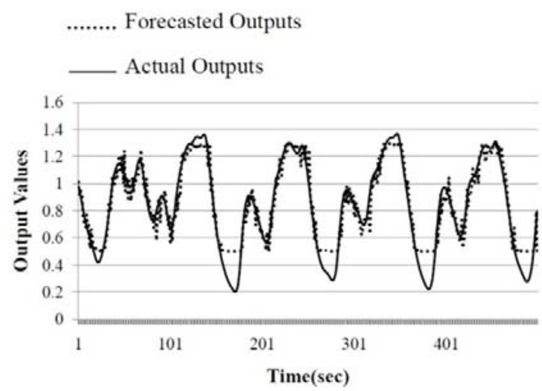


Fig. 9: Comparison between the Actual and the Forecasted time series

VII. CONCLUSION AND FUTURE WORK

In this paper we have proposed the way of evolving rules in Type-1 FLS using GA and used the designed system for time series prediction. The designed system with 0 db SNR has proved to be the best for forecasting time-series problem with least MSE. The system involves intense calculations and hence increases the execution time so implementing the same system parallely on GPU using CUDA can be the future work of this paper. Also in future the system can also be implemented for many other applications and problems involving wide range of classification.

REFERENCES

[1] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for thevdistributed and flexible job-shop scheduling problem," *European journal of operational research*, vol. 200, no. 2, pp. 395–408, 2010.

[2] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.

[3] A. Kumar, A. Khosla, J. S. Saini, and S. Singh, "Meta-heuristic range based node localization algorithm for wireless sensor networks," in *Localization and GNSS (ICL-GNSS), 2012 International Conference on*. IEEE, 2012, pp. 1–7.

[4] K. Mankad, P. S. Sajja, and R. Akerkar, "Evolving rules using genetic fuzzy approach: an educational case study," *International Journal onSoft Computing*, vol. 2, no. 1, pp. 35–46, 2011.

[5] N. N. Karnik and J. M. Mendel, "Applications of type-2 fuzzy logic systems to forecasting of time-series," *Information Sciences*, vol. 120,no. 1, pp. 89–111, 1999.

[6] Y. Shi, R. Eberhart, and Y. Chen, "Implementation of evolutionary fuzzy systems," *Fuzzy Systems, IEEE Transactions* on, vol. 7, no. 2, pp. 109– 119, 1999.

[7] A. J. Jones, A. Tsui, and A. De Oliveira, "Neural models of arbitrary chaotic systems: construction and the role of time delayed feedback incontrol and synchronization," *complexity international*, vol. 9, pp. 1–9, 2002.

[8] P. R. Thrift, "Fuzzy logic synthesis with genetic algorithms." in *ICGA*, 1991, pp. 509–513.

[9] W.-R. Hwang and W. E. Thompson, "Design of intelligent fuzzy logic controllers using genetic algorithms," in *Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence*., Proceedings of the Third IEEE Conference on. IEEE, 1994, pp. 1383–1388.

[10]J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[11]L. A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning*. Springer, 1974.

[12]D. J. Dubois, Fuzzy sets and systems: theory and applications. Academic press, 1980, vol. 144.

[13]S. Singh and S. Kakkar, "Fuzzy systems using gpgpu - a survey," in *International Conference on Communication, Computing and Systems*, Ferozepur, Punjab, India, August 2014, pp. 238–241.

[14]N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy sets and systems*, vol. 122, no. 2, pp. 327–348, 2001.

[15] O. Castillo and P. Melin, *3 Type-2 Fuzzy Logic*. Springer, 2008.

[16] R. S. Sinha and S. Singh, "Optimization techniques on gpu," in *International Multi Track Conference on Science, Engineering & Technical Innovations*, CT Group of Institutions, Jalandhar, June 2014, pp. 566–568.

[17] O. Cord´on, F. Herrera, F. Gomide, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 3. IEEE, 2001, pp. 1241–1246.

[18] A. Bastian and I. Hayashi, "An anticipating hybrid genetic algorithm for fuzzy modeling," *aNx,* vol. 7, no. 5, pp. 997–1006, 1995.

[19] Y. Yuan and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," *Fuzzy sets and systems*, vol. 84, no. 1, pp. 1–19,1996.

[20] D. Chauhan, S. Singh, S. Singh, and V. K. Banga, "Speedup of type- 1 fuzzy logic systems on graphics processing units using cuda," in *International Conference on Communication, Computing and Systems*, Ferozepur, Punjab, India, August 2014, pp. 267–271.