The Scrutiny of Variation in the Number of Fuzzy Rules and Membership Functions in a New Genetic-Fuzzy System in Approximation and Prediction Problems

Vahideh Keikha, *Member, IACSIT*, Hayat Khoobipour, Mahdi Aliyari Shoorehdeli, and Hassan Rezaei

***Abstract*—This paper shows a new fuzzy system was improved using genetic algorithm to handle fuzzy inference system as a function approximator and time series predictor. The system was developed generality that trained with genetic algorithms (GAs) corresponding to special problem and would be evaluated with different number of rules and membership functions. Then, compare the efficacy of variation of these two parameters in behavior of the system and show the method that achieves an efficient structure in both of them. Also, the proposed GA-Fuzzy inference system successfully predicts a benchmark problem and approximates an introduced function and results have been shown.**

***Index Terms*—Genetic algorithm, fuzzy inference system, fuzzy membership functions, fuzzy rules.**

I. Introduction

The idea of fuzzy sets first was introduced by Zadeh [1] and, Holland [2] proposed the basic principles of the genetic algorithms. Thereafter, a large number of literatures in fuzzy theory and genetic algorithm became available. Also, many papers have been published regarding the combination of fuzzy logic (FL) and genetic algorithms [3]-[7], and also fuzzy logic and time series prediction and applications [8]-[14].

In general, due to their efficiency and interpretability, fuzzy systems are very powerful tools for function approximation and time series prediction problems. Prediction and approximation is becoming one of important research and major goal in application areas, so several methodologies have been applied to these problems. In [13] Kang used Triangular membership functions and Evolutionary Algorithm for time series prediction and showed his results by comparison between Evolutionary Strategy and Genetic Algorithm with bit representation of chromosomes, in benchmark problem Mackey-Glass. Traditional methods used for prediction are based on technical analysis of time series, such as looking for trends, stationarity, seasonality, random noise variation, moving average. These are exponential smoothing method, well known Box-Jenkins method which have shortcomings [15], [16]. Some of these are linear and some are nonlinear solutions. Likewise, soft computing methods as nonlinear solutions were used for these problem that in [17] have been shown that these methods have more ascendency regards to precedency approaches. Likewise, other soft computing methods such Anfis and Neural Networks have been used for these problems [9], [14], [15], [18]. With inspiration of [13] we propose a new GA-Fuzzy system that can predict and also approximate the functions successfully.

The purpose of the present study is to propose a Mamdani fuzzy inference system as a general nonlinear model that were trained with genetic algorithm and can predict the chaotic time series and approximate the functions. Moreover, we show the method that achieves an efficient structure. This system against the previous in [13] use the floating point representation of chromosomes in the form of fuzzy rules to achieve an optimized structure with comparison of different number of rules and two most usage membership functions (MFs) that are Triangular and Gaussian. We use the mutation operator in GA trainer structure as a primary operator and in another way that causes to speed the convergence and running away from suboptimal solutions. Since, the aim is scrutiny of efficiency of changing the parameters of the fuzzy system and achieve an efficient structure and the way to obtain that, compare the systems achieved by controlled changing the parameters of the fuzzy system to each other and no other systems proposed in previous.

This paper is structured as follows: Section 2 provides a brief introduction to fuzzy concepts and genetic algorithms. Fuzzy module design and GA trainer structure are given in Section 3. Section 4 introduces used applications and section 5 presents the results of prediction and approximation and discussions on the experimental validation. Finally, some concluding remarks are drawn in Section 6.

II. Preliminaries

A. Fuzzy Concepts

Fuzzy logic is another area of artificial intelligence that has been implemented successfully in different basis and applications. The main idea of fuzzy logic is that something has its main character that causes belonging to a specific class or set with a degree of membership. For example, something in the class of true objects can be partially false. In fuzzy logic, variables are "fuzzified" through the use of membership function, the curve that defines the membership degree to fuzzy sets. These variables are called linguistic variables. A fuzzy subset Q of a universe of discourse U is

Manuscript received August 5, 2012; revised March 10, 2013.

V. Keikha and H. Rezaei are with the Computer Science Department, Sistan and Baluchestan University, Zahedan, Iran (e-mail: va.keikha@yahoo.com, hrezaei@cs.usb.ac.ir).

H. Khoobipour is with the Computer Engineering Department, Azad Islamic University, Branch of Dehdasht, Dehdasht, Iran (e-mail: hayat_1378@ yahoo.com).

M. Aliyari Shoorehdeli is with the Department of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran (e-mail: m_aliyari@eetd.kntu.ac.ir).

DOI: 10.7763/IJIEE.2013.V3.360

characterized by a membership function $\mu_Q(x)$ in the interval [0, 1] and represents the grade of membership in Q [19], [20]. The first application of fuzzy control was first developed by the work of Mamdani and Assilian [21] in 1975, with designing a fuzzy controller for a steam engine. Fuzzy control systems are rule-based systems which have a set of fuzzy IF-THEN rules represents a control decision mechanism to adjust the effects of certain causes coming from the system [5]. A fuzzy reasoning system consists of three other components fuzzification, inferencing, and defuzzification. The fuzzification process is concerned with finding a fuzzy representation of non-fuzzy input values. This is achieved through application of the membership functions associated with each fuzzy set in the rule input space. The task of the inferencing process is to map the fuzzified inputs (as received from the fuzzification process) to the rule base, and to produce a fuzzified output for each rule, the task of the defuzzification process is to convert the output of the fuzzy rules into a scalar, or non-fuzzy value[19]. Commonly, two different types of fuzzy inference system (FIS) can be implemented. These are the Mamdani and the Takagi-Sugeno algorithm. Among them, Mamdani fuzzy inference system is the most commonly used fuzzy methodology. Several types of membership function can be used to describe the fuzzy sets such as Triangular MFs, Gaussian MFs and Trapezoidal MFs. Also, there are other MFs such that Γ-membership function, S-membership function, logistic and exponential type [19], [22].

1) Triangular MFs

A "triangular MF" is specified by three parameters $\{a, b, c\}$ as follows:

$$y = \text{triangle}(x, a, b, c) \begin{cases} 0, & x \le a \\ (x-a)/(b-a), & a \le x \le b \\ (c-x)/(c-b), & b \le x \le c \\ 0, & c \le x \end{cases} \tag{1}$$

The parameters $\{a, b, c\}$ (with $a < b < c$) determine the x coordinates of the three corners of the underlying Triangular MF [22].

2) Gaussian MFS

A "Gaussian MF" is specified by two parameters $\{c, \sigma\}$:

$$y = \text{gaussian}(x; c, \sigma) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{2}$$

A "Gaussian MF" is determined completely by c and σ, c represents the MFs center of set and σ determines the MFs width(spread)[22].

B. Genetic Algorithms

GA's are stochastic global search methods that operate on a population and using the biological evolution and the survival of the fittest analogous to Darwinian models. In classical GAs, for constructing new population, using operators such as crossover and mutation and then selection has been accomplished. That is individuals with higher fitness values are more likely to survive and to participate in mating crossover operations. GA because of both of the genes can be manipulated and controlled to achieve desirable attributes and, containing members with better fitness values with passing some of the generations, is a good method for optimization problems. The chromosomes as the individuals can be used as the potential solution of the problem that should be optimized.

III. Fuzzy Module Design and GA Trainer Structure

First, we have to make the generation that should be evaluated, therefore it is important to precise the structure of chromosomes exactly. Make up of each chromosome differs in Gaussian and Triangular MFs. Length of each chromosome in Gaussian MFs with m rules is $3\times m$ that, m elements is for center of sets, m elements for spreads and m elements is for the functional values. In Triangular MFs with the same number of rules, length of each chromosome is $4\times m$, *that* $3\times m$ elements for m triangles and m elements is for the functional values. Spread in the Gaussian MFs has a direct effect on the convergence speed and the accuracy of the estimated values, if the spread has been selected very big, the estimated values have no difference for inputs with pelting different. Also, if it has been chosen very small, the system has been specialized for the train data. It can be found by trial and error method and then, the minimum number of rules that they can be continuous, can be achieved. E.g., for approximating the function $f(\gamma)$, γ is defined in the range$[\alpha_1, \alpha_2]$. First, we have to find the width of triangles and gaussians, use the spread of gaussians for the width of triangles. Then we can find minimum number of rules that can be continuous. By increasing them and using trial and error method, the best number has been achieved. If m is the minimum number of rules then:

$$m = (\alpha_2 - \alpha_1)/\text{spread} \tag{3}$$

To evaluate the system in approximation problem by a two-dimensional function (9), length of each chromosome becomes twice. Generally, in a D-dimensional optimization problem, L the length of chromosomes becomes:

$L=D\times(4\times m)$ in Triangular MFs

$L=D\times(3\times m)$ in Gaussian MFs,

D= dimensions of the function.

$$P=(p_{1,G}, p_{2,G}, \ldots, p_{N,G})^T = \begin{pmatrix} p_{1,1,G} & p_{1,2,G} & \cdots & p_{1L,G} \\ p_{2,1,G} & p_{2,2,G} & \cdots & p_{2,L,G} \\ \vdots & \vdots & \vdots & \vdots \\ p_{N,1,G} & p_{N,2,G} & \cdots & p_{N,L,} \end{pmatrix} \tag{4}$$

N=population size, m=number of fuzzy rules,
G= generation that population belongs

Spread value has been selected 0.5. As the individuals are the rules of Mamdani fuzzy inference systems, initializing is very important. First, start evaluation of population with random generation in specified range. Population size has been chosen 60 and keeps it fixed throughput the run. For producing the individuals in Gaussian MFs, centers have been selected randomly. In Triangular MFs, characters of triangles have been chosen regularly with a random epsilon different. The functional values of each rule let to be randomly in the beginning. In this stage, the main subject is that the rules be continuous, that in above styles the aim is achieved. Another way is choosing these specifications regularly that appropriately overlapped to each other, but because of similarity of individuals in some attributes, the speed of convergence becomes very low and selected method seems to be better solution.

The chromosomes prepare the rules of Mamdani inference system that have been evaluated throughput the run; evaluation result is tuning the rules. To stand on this, make a vector with N random entries (775 for (8) and 700 for (9)) as the train vector in specified interval with universe of discourse and keep it fixed. For each chromosome, compute the sum square error (SSE) of real functional value of each N its entries and estimated value, and give it as the chromosome's fitness value (6).

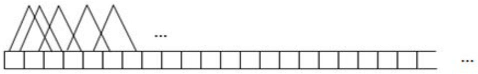

Fig.1. a

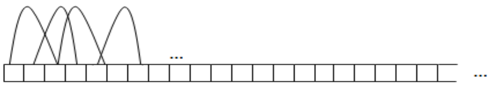

Fig.1. b

Fig. 1.a. structure of chromosomes in Triangular MFs that character of rules are regularly with random epsilon different in each chromosome, b. structure of chromosomes in Gaussian MFs that centers have been selected randomly.

A. Mutation

In this system, mutation is the main genetic operator that operates on one chromosome at a time and generates one offspring. A simple way to do this would be to choose a random point and mutate it, but this is not approval here, suggestion is to spot D random real numbers (D is number of the dimension of the problem) in specified range and after finding most closely genes in chromosome, replace them. The mutation rate (denoted by p_m) is defined as a ratio of the number of offspring produced in each generation to the population size with mutation operator, has been chosen fourfold crossover rate. High mutation rate at the begging of execution of algorithm, causes to run away from local minimums and throughput the run, provides from premature convergence and handle diversity of population.

B. Crossover

This operator producing new individuals from two parent individuals, that works by exchanging substrings between two individuals to obtain new offspring. This substring includes one rule either Gaussian or Triangular MF. Number of crossover points have been restricted to one per pair, corresponding to the type of membership function, 2 or 3 genes have been substituted. Also, one cut point regards to be better in both time and diversity in our method.

C. Evaluating Fitness Value

Each individual is tested empirically in an environment, receiving a numerical evaluation of its merit, assigned by the fitness function. Previously, mentioned that, fitness value is the sum square error of real functional value of specified number of train data and estimated value with proposed system. This train data keep fixed across the algorithm execution and involves N random data in universe of discourse. For example in (9), for each chromosome, the sum square error of estimated values by mentioned system and 700 real $f(x, y)$ values, must be computed ($N=700$).

$$X = \{x_i\},\ y = \{f(x_i)\},\ i = 1\,........,N \tag{5}$$

$$\text{fitness}(p_{i,G}) = \sum_{K=1}^{N} |y_k| - \text{estimated}(p_{i,G}, x_k)|,\ i = 1,........,60 \tag{6}$$

$$m_{SSE} = min(\text{fitness}(p_{i,G}))$$

$P_{i,G}$ is i-th individual in G-th generation

D. Selection

Selection is stage that next generation has been selected from old and new generation, that implemented by eliminating low fitness individuals from whole of old and new population with tournament method, therefore probably making multiple copies of high fitness individuals. Also, two more elite individuals directly inserted in new generation. For the reason of velocity and avoiding get stuck in suboptimal solutions, in each 10 generation some random individuals have been replaced with new offspring.

E. Fuzzy Rules

The rule base of our system follows the Mamdani method and center of area defuzzifier [19], [23] with the below formula for computing the output:

$$Y = \frac{\sum_{L=1}^{M} \prod_{i=1}^{n} \mu A_i^l(x_i)\overline{y}}{\sum_{L=1}^{M} \prod_{i=1}^{n} \mu A_i^l(x_i)} \tag{7}$$

where x is input vector, $\mu A_i^l(x_i)$ is the membership grade of to membership function A_i in rule l ($l=1,...,m$) and is computed with one of the Triangular or Gaussian MFs, m is number of rules, n is size of input vector and Y is the computed defuzzified output.

F. The Algorithm of How to Evaluating the System Is Classified in the Following

Step 1:

[Stage 1-1] Initialize a population with random generated individuals as the Mamdani fuzzy inference systems with specified number of rules and membership functions (population size is 60 and keep it fixed).

[Stage 1-2] Make the training vector with N random entries to evaluate chromosome's fitness values.

[Stage 1-3] Evaluate the fitness value of each individual according to (6).

Step 2:

[stage 2-1] Select two members from current population with probabilities proportional to their fitness value.

[Stage 2-2] Apply crossover operator with a probability equal to the crossover rate ($p_c=0.2$), that one rule in two selected individuals has been substituted.

[Stage 2-3] Apply mutation operator with a probability equal to the mutation rate ($p_m=0.8$), select D random real number with universe of discourse and then, nearest genes have been replaced.

[Stage 2-4] Compute the fitness value of each new offspring.

[Stage 2-5] Repeat (2-1) to (2-4) until enough members are generated to form the new generation.

[Stage 2-6] Select next generation from old and new population and insert two most elite individuals into the new population.

Step 3:

Repeat steps 2 to 3 until a stop condition is satisfied.

IV. Applications to Time Series Prediction and Function Approximation

Prediction of chaotic Mackey-Glass time series can be defined in following differential equation:

$$x(t)=\frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)}-0.1x(t) \tag{8}$$

Assume $\tau=17$, $x(0)=1.2$, $x(t)=0$ $(t<0)$

From 1200 exist samples, x (19) to x (794) is for tuning the rules(as the train vector) and x (795) to x(1200) hold for validation. Predicted and their real values have been compared in the next section (See Fig.4 and Fig.5).

The system was evaluated with the following function [24]:

$$F(x, y)=\sin c(x)\times \sin c(y), \text{ where } x, y \ [-3,\ 3] \tag{9}$$

That x and y in both train and validation data have been produced randomly. In order to construct the train data, 700 random x and y has been produced to make 700 $f(x, y)$ as the train vector. This vector used to compute the chromosome's fitness values. In order to evaluate the system efficiency, after tuning the rules, 100 random test data have been generated and then estimated values and their real values have been compared (See Fig. 2 and Fig. 3).

V. Experimental Result

The proposed system has been tested with different number of rules and two different membership functions for two applications that have been said before. For straightforward comparing, both of the estimated and the real values are plotted in the figures. The goal has been selected the value of sum square error, that as we reached to value less than 0.5 (m_{SSE}<0.5), stopped the evolutionary process. Also, in each execution initial situations are the same.

A. Approximation with Gaussian MFs

In the following, the result of estimating (9) by Gaussian MFs with different number of rules has been shown. With reference to Table I, approximation with 60 rules has more accuracy and promptitude regards to estimation with 70 rules. So, stop increasing number of rules and accept 60 rules as the best number. Fig.2 shows the approximated and real values of 100 random test data using Gaussian MFs with 60 rules.

TABLE I: Experimental Result of the Efficiency of Changing Number of the Rules Until the Goal Is Satisfied

Generation	MF	Num of Rules	m_{SSE}	Elapsed Time (s)
932	Gaussian	20	0.4997	73611.808
547	Gaussian	30	0.4987	29989.500
287	Gaussian	40	0.4971	18403.953
220	Gaussian	50	0.4874	16131.406
181	Gaussian	60	0.4606	15201.266
269	Gaussian	70	0.4991	25381.594

In Table I, saw that increasing number of rules to a specified number caused to more better solution, but by increasing more, because of belonging input to more rules that result the decrease in speed of training all the related rules and wastage the flexibility and role of each rule in estimating due to finding real position of each random rule, did not lead to more better answer. E.g., assume that in a sample fuzzy system, input x as the train data, belongs to w rules at a same time, that is w rules should be trained to find the best attributes, but by increasing q more rules, x belongs to $w+q$ rules simultaneously, i.e., $w+q$ rules should be tuned for x. In this way, one rule is continuous with more rules and should be tuned with all of them concurrently.

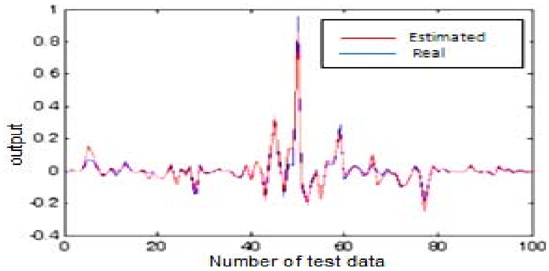

Fig. 2. Comparison of 100 random test data and their estimated values in sinc(x)×sinc(y) with Gaussian MFs and 60 rules

B. Approximation with Triangular MFs

With reference to Table II, by using Triangular MFs approximation with 30 rules have more accuracy and promptitude regards to estimation with 35 rules, thus because increasing the rules does not cause to better solution, stop the process and matriculate this as the best. So, the best answer in both time and accuracy in this problem is Gaussian MFs with 60 rules. Fig. 3 shows the approximation of 100 random test data and their real values using Triangular MFs with 30 rules. As see, in Fig. 2 toward Fig. 3 estimated data have more coincidence with their real values.

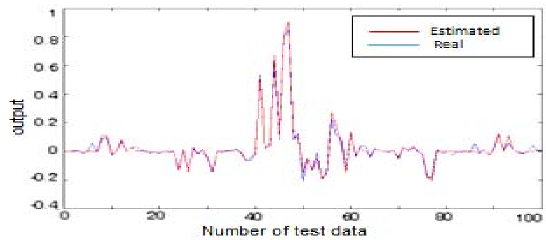

Fig. 3. Comparison of 100 random test data and their estimated values in sinc(x)×sinc(y) with Triangular MFs and 30 rules

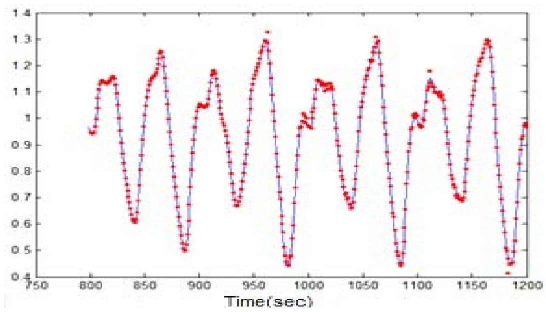

Fig. 4. Comparison of real and predicted values of mackey-glass with gaussian MFs and 330 rules

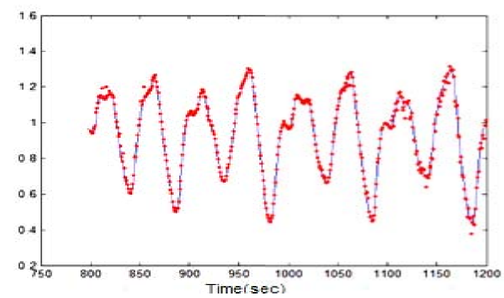

Fig. 5. Comparison of real and predicted values of mackey-glass with triangular MFs and 290 rules

TABLE II: EXPERIMENTAL RESULT OF THE EFFICIENCY OF CHANGING NUMBER OF THE RULES UNTIL THE GOAL IS SATISFIED

Generation	MF	Num of Rules	m_{SSE}	Elapsed Time (s)
1623	Triangular	15	0.4996	54908.066
907	Triangular	25	0.4947	47908.031
889	Triangular	30	0.4717	45445.390
1145	Triangular	35	0.4991	50896.165

C. Prediction with Gaussian MFs

In the following, the result of predicting (8) by Gaussian MFs with different number of rules has been become.

TABLE III: EXPERIMENTAL RESULT OF THE EFFICIENCY OF CHANGING NUMBER OF THE RULES UNTIL THE GOAL IS SATISFIED

Generation	MF	Num of Rules	m_{SSE}	Elapsed Time (s)
6349	Gaussian	290	0.4789	62654.051
5832	Gaussian	310	0.4652	58552.075
4998	Gaussian	330	0.3258	51304.105
5142	Gaussian	350	0.4147	57895.327

With reference to above result, predicting (8) by Gaussian MFs with 330 rules is better solution regards to another number of rules.

D. Prediction with Triangular MFs

As see in Table IV, in predicting (8) by fuzzy system with Triangular MF, 290 rules is better from other number of rules. Also, in Fig. 4 toward Fig. 5 estimated data have more coincidence with their real values. By comparison Table III and Table IV, see that training the Triangular MFs are more time spending and reach the goal posterior. In other words, in Triangular MFs, 3 parameters in each rule should be tuned regards to 2 parameters in Gaussian MFs.

TABLE IV: EXPERIMENTAL RESULT OF THE EFFICIENCY OF CHANGING NUMBER OF THE RULES UNTIL THE GOAL IS SATISFIED

Generation	MF	Num of Rules	m_{SSE}	Elapsed Time (s)
6738	Triangular	280	0.4733	65908.066
5216	Triangular	290	0.4286	58104.720
5779	Triangular	300	0.4407	61445.390

VI. CONCLUSION

The work reported here has concentrated on scrutiny of variation the behavior of a fuzzy inference system with differing MFs and number of rules and show the method that achieves efficient parameters in structure of the problems. The system trained with genetic algorithm for function approximation and time series prediction. Also, the system was successfully applied to a benchmark problem Mackey-Glass chaotic time series and approximation of proposed function. By proposed method, optimized number of rules and MFs has been achieved. As saw, when found a good solution regard to next and before, stopped the process of increasing number of rules, because of mentioned reasons this did not cause better solution. The significance of this work is that a new efficient system and the method of achieving the best structure have been developed with comparative and experimental methods.

REFERENCES

[1] L. A. Zadeh, "Fuzzy sets information and control," vol. 8, no. 3, pp. 338-353, 1965.

[2] J. H. Holland, *Adaptation in natural and artificial systems*, MIT Press, Cambridge, 1992.

[3] J. Kroeske, A. Ghandar, Z. Michalewicz, and F. Neumann, "Learning fuzzy rules with evolutionary algorithms - an analytic approach," *PPSN X, LNCS*, pp. 1051-1060, 2008.

[4] S. K. Oh, K. J. Park, and W. Pedrycz, "Optimization of fuzzy systems based on fuzzy set using genetic optimization and information granulation, "*MDAI 2005, LNAI 3558*, pp. 316-327, 2005.

[5] M. S. Kim, *Performance Improvement of the Attitude Estimation System Using Fuzzy Inference and Genetic Algorithms*, in Springer-Verlag, pp. 445–454, 2007.

[6] T. Arredondo *et al.*, *Dynamic penalty based GA for inducing fuzzy inference systems*, pp. 957–966, 2007.

[7] O. Cardon, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic Fuzzy Systems Evolutionary Tuning and Learning OF Fuzzy Knowledge Bases*, Singapore: world scientific publishing, 2001.

[8] Z. Li, Fuzzy Chaotic Systems: Springer, Stud Fuzz 199, pp. 73-89, 2006.

[9] C. Vairappan, H. Tamura, S. Gao, and Z. Tang, "Batch type local search-based adaptive neuro-fuzzy inference system (ANFIS) with self-feedbacks for time-series prediction," *Neurocomputing*, vol. 72, pp. 1870-1877, 2009.

[10] M. Firat, M. E. Turan, and M. A. Yurdusev, "Comparative analysis of fuzzy inference systems for water consumption time series prediction," *Journal of Hydrology*, vol. 374, pp. 235-241, 2009.

[11] H. Gu and H. Wang, "Fuzzy prediction of chaotic time series based on singular value decomposition," *Applied Mathematics and Computation*, vol. 185, pp. 1171-1185, 2007.

[12] N. Takagi and S. Tokinaga, "Prediction of chaotic time-series by using the multi-stage fuzzy inference systems and its applications to the analysis of operational flexibility," *Journal of the Operations Research Society of Japan*, vol. 45, no. 3, 2002.

[13] H. Kang, *A fuzzy time series prediction method using the evolutionary algorithm*, pp. 530- 537, 2005.

[14] L.J. Herrera *et al.*, "Multigrid-based fuzzy systems for time series prediction," *CATS competition, Neurocomputing*, vol. 70, pp. 2410-2425, 2007.

[15] R. H. Abiyev, "Time series prediction using fuzzy wavelet neural network model," pp. 191- 200, 2006.

[16] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time-series Analysis, Forecasting and Control*, 1 st ed: Prentice-Hall, Inc, Englewood Cliffs, NJ 07632, 1994.

[17] G. S. Maddala, *Introduction to econometrics*, Prentice-Hall, Englewood Cliffs, 1996.

[18] Y. Oysal, "Time delay dynamic fuzzy networks for time series prediction," pp. 775 -782, 2005.

[19] A. P. Engelbrecht, *Computational Intelligence*, John Wiley and sons, 2007.

[20] H. J. Zimmermann, *Fuzzy set theory - and its application*, 4 st ed: Kluwer Academic, Dordrecht, The Netherlands, 1991.

[21] E. Mamdani and H. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.

[22] O. Castillo and P. Melin, "Type-2 Fuzzy Log, Theory and Appl.," *Stud Fuzz*, vol. 223, pp. 5-28, 2008.

[23] K. H. Lee, *First Course on Fuzzy Theory and Applications*, Springer-Verlag, 2005.

[24] R. Mehran, A. Fatehi, C. Lucas, and B. N. Arabi, "Particle swarm extension to LOLIMOT," in *Proc. ISDA 06, IEEE*, 2006.

Vahideh Keikha received her B.Sc. degree in computer engineering from Ferdowsi University of Mashhad, Mashhad, Iran, in 2008. She received her M.Sc degree in Computer Science from Sistan and Baluchestan University, Zahedan, Iran, in 2010. She is Faculty member in department of Computer Science, Sistan and Baluchestan University, Zahedan, Iran. Her research interests include Fuzzy and Neuro-Fuzzy Systems, Neural Networks, Evolutionary Algorithms Approximation and Prediction, Image processing and Computational Geometry.

Hayat Khoobipour received his B.Sc. degree in computer engineering from Shiraz University, Shiraz, Iran, in 2004. He received his M.Sc degree in Computer Science from Sistan and Baluchestan University, Zahedan, Iran, in 2010. He is Faculty member in Azad Islamic University, Branch of Dehdasht, Dehdasht, Iran. His research interests include Fuzzy and Neuro-Fuzzy Systems, Neural Networks, Evolutionary Algorithms Approximation and Prediction, Natural Language Processing and Computational Geometry.

Mahdi Aliyari Shoorehdeli received his B.Sc. degree in Electrical Engineering (Electronics) from K. N. Toosi University of Technology, Tehran, Iran, in 2001. He received his M.Sc degree in Electrical Engineering (Control) from K. N. Toosi University of Technology, Tehran, Iran, in 2003. He received his Ph.D degree in Electrical Engineering (Control, Intelligent Systems) from K. N. Toosi University of Technology, Tehran, Iran, in 2008. He is Faculty member in department of Mechatronics, K. N. Toosi University of Technology, Tehran, Iran. His research interest includes Mechatronics Systems, Fault Diagnosis, Neural Networks, Fuzzy systems, Nero-Fuzzy control, Identification and Predication, Pattern Recognition and Swarm Intelligence.

Hassan Rezaei received his B.Sc. degree in Computer Science, from Sharif University of Technology, Tehran, Iran, in 1992. He received his M.Sc degree in Computer Science from, Shahid Bahonar University, Kerman, Iran, in 1995. He received his Ph.D degree in Computer Science from Meiji University, Japan, in 2007. He is Faculty member in department of Computer Science, Sistan and Baluchestan University, Zahedan, Iran. His research interest includes Soft Computing.