I.J. Information Technology and Computer Science, **2013, 08, 24-37**
Published Online July 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2013.08.03

Data Driven Fuzzy Modeling for Sugeno and Mamdani Type Fuzzy Model using Memetic Algorithm

Savita Wadhawan
M.M. University, Mullana ,Distt. Ambala, India
E-mail: savitawadhawan@gmail.com

Gunjan Goel
Jaypee University of Information Technology, Solan, HP, India
E-mail: gunjanmicro@gmail.com

Srikant Kaushik
M.M. University, Mullana
E-mail: kaushik_srikant@gmail.com

Abstract**—** The process of fuzzy modeling or fuzzy model identification is an arduous task. This paper presents the application of Memetic algorithms (MAs) for the identification of complete fuzzy model that includes membership function design for input and output variables and rulebase generation from the numerical data set. We have applied the algorithms on four bench mark data: A rapid Ni-Cd battery charger, the Box & Jenkins's gas-furnace data, the Iris data classification problem and the wine data classification problem. The comparison of obtained results from MAs with Genetic algorithms (GAs) brings out the remarkable efficiency of MAs. The result suggests that for these problems the proposed approach is better than those suggested in the literature.

Index Terms**—** Memetic Algorithms (MAs), Genetic Algorithms (GAs), Fuzzy Modeling, Fuzzy Systems

I. Introduction

Fuzzy logic provides an effective means to capture the approximate and inexact nature of the real world. As the system complexity grows, it becomes more difficult to describe them by precise mathematical models. Fuzzy logic can describe such complex systems with linguistic rules [1-2]. The most important applications of fuzzy logic are control systems and decision support systems.

Successful design of a rule based fuzzy system depends on several factors such as choice of the rulebase, membership functions, inference mechanism, and the defuzzification strategy. Of these factors, selection of an appropriate rule base is more difficult because it is a computationally expensive combinatorial optimization problem. Sometimes for fuzzy systems, rules are derived from human experts who have acquired their knowledge through experience. This approach is known as knowledge driven modeling. This modeling approach becomes difficult, when the available knowledge is incomplete or when the problem space is very large. Even though this design methodology has led to a large number of successful applications, it is time consuming and subjected to criticism for its lack of principles and systematic methodologies. Thus extraction of an appropriate set of rules from the observed data is an important and essential step towards the design of any successful fuzzy logic based systems. For the data driven modeling approach, no prior knowledge of the system under consideration is assumed to be available.

Evolutionary algorithms (EAs) are stochastic search methods that mimic the metaphor of natural biological evolution and/or the social behavior of species. Genetic algorithms (GAs) [3] were the first evolutionary based technique introduced in literature. GAs are developed based on the Darwinian principle of the 'survival of the fittest' and the natural process of evolution through reproduction. Based on its demonstrated ability to reach near-optimum solutions to large problems, the GAs techniques have been used in many applications in science and engineering [4-5]. Despite their benefits, GAs may require long processing time for a near optimum solution to evolve. Also, not all problems lend themselves well to a solution with GAs [6].

In an attempt to reduce processing time and improve the quality of solutions, particularly to avoid being trapped in local optima, other EAs have been introduced during the past decade. In addition to various GAs improvements, recent developments in

EAs include other techniques inspired by different natural processes: memetic algorithms (MAs), particle swarm optimization (PSO), ant colony systems (ACO) etc. The general idea behind memetic algorithms is to combine the advantages of evolutionary operators that determine interesting regions in the search space with local neighborhood searches that quickly finds good solutions in a small region of the search space.

In this paper we introduce a modeling approach to identifying the complete fuzzy model i.e. membership function design for input and output variables and rulebase generation from data using MAs and compare its results with the results obtained from genetic algorithms (GAs) and the results found in literature. This paper is organized as follows. Section II is a brief introduction to fuzzy systems. This is followed by a brief introduction to Memetic algorithms (MAs) in section III. Section IV presents MAs based procedure to identify fuzzy models. Section V considers four bench mark data: A rapid Ni-Cd battery charger, the gas-furnace data, the iris data classification problem and the wine data classification problem. Section VI concludes the paper.

II. Fuzzy System

This section presents a brief overview of a fuzzy logic based system. Fuzzy inference is the actual process of mapping from a given input space to an output space through fuzzy logic. The term fuzzy inference system applies to any system whose operations are based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning. The fuzzy inference systems are also known by several other names such as fuzzy rule-based systems, fuzzy expert systems, fuzzy associative memories and fuzzy logic controllers.

The basic structure of a fuzzy inference system can be represented as in fig. 1, which consist of following main modules: (1) the fuzzifier that converts the crisp inputs into a fuzzy inputs, (2) a knowledge base which contains fuzzy rules along with a data base or dictionary defining the membership functions (3) an inference mechanism that applies a fuzzy reasoning mechanism to derive a fuzzy output and (4) a defuzzifier, that translates the fuzzy output into a crisp value.

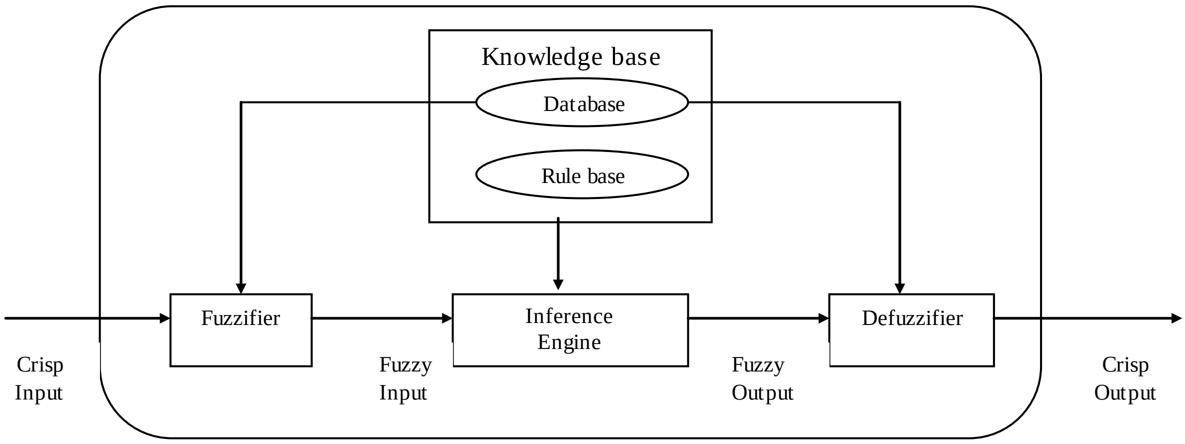

Fig. 1: Basic structure of a fuzzy inference system

Types of fuzzy models:

There are three main types of fuzzy models that differ in the way the rule consequents and the implication process. These are Mamdani, Takagi-Sugeno-Kang (TSK) and Sugeno/TSK type 0 fuzzy models[1-2].

In Mamdani models, each fuzzy rule is of the form:

R_i: If x_1 is A_{i1} and ……and x_n is A_{in} then y is B

Where, x_1, …….., x_n are the input variables and y is the output variable. A_{i1}, ………., A_{in} and B are the linguistic values of the input and output variables in the ith fuzzy rule.

The Mamdani fuzzy models have fuzzy sets as rule consequents. The main advantage of such model is their high interpretability since the output variables are defined linguistically. However, these models lack accuracy and have high computational cost. Lack of accuracy is due to the rigidity of linguistic values, whereas high computational cost can be attributed to computational intensive defuzzification process.

In TSK models, each fuzzy rule is of the form:

R_i: If x_1 is A_{i1} and ……and x_n is A_{in} then y is

$$\sum_{i=1}^{n} a_i x_i + c$$

where, a_i and c are constants.

whereas, for Singleton models, each fuzzy rule is of the form:

R_i: If x_1 is A_{i1} and ……and x_n is A_{in} then y is C

where, C is a fuzzy singleton.

In Singleton models, the consequents are represented by a fuzzy singleton. Singleton models can be considered as a special case of either Mamdani or TSK fuzzy models, as a constant value is equivalent to both a singleton fuzzy set i.e. a fuzzy set that has its membership value in a single point of the universe of discourse and a linear function as defined by TSK rule consequent, when $a_i = 0$. Due to the discrete representation of the output variable, the defuzzification process requires less computational efforts than for Mamdani and TSK models.

III. Memetic Algorithms

Evolutionary algorithms is a general term for *evolutionary programming*, *evolution strategies*, *genetic algorithms*, and *genetic programming* have been applied successfully in various domains of search, optimization, and artificial intelligence. In the field of combinatorial optimization, it has been shown that augmenting evolutionary algorithms with problem-specific heuristics can lead to highly effective approaches. These hybrid evolutionary algorithms combine the advantages of efficient heuristics incorporating domain knowledge and population-based search approaches. One form of hybridization is the use of local search in evolutionary algorithms [7-8]. These algorithms, sometimes called genetic local search algorithms, belong to the class of Memetic algorithms.

3.1 Memetic Algorithms

MAs are inspired by Dawkins notion of a meme [9]. MAs are similar to GAs but the elements that form a chromosome are called memes and not the genes. The unique aspect of the MAs is that all chromosomes and offsprings are allowed gaining some experience, through a local search, before being involved in the evolutionary process [10]. A local search algorithm starts from a configuration generated at random or constructed by some other algorithm. Subsequently, it iterates using at each step a transition based on neighborhood of the current configuration. Transition leading to preferable configuration are accepted i.e. the newly generated configuration becomes the current configuration in the next step. Otherwise the current configuration is kept. The process is repeated until a certain termination criterion is met. A pseudo code for a MA procedure is given in Figure 2(a).

```
Procedure Local_Search(current);
Begin;
    while TerminationCriterion() not met;
        new = GenerateNeighbor(Current)
        if fitness(new) < fitness(current)
            return;
        else
            current = new;
        endif
        return current;
    end while
End;
```

Fig. 2(a): Pseudo code for local procedure

As such, the term MAs is used to describe GAs that heavily use local search [11-12]. A pseudo code for a MA procedure is given in Figure 2(b). Similar to the GAs, an initial population is created at random. Afterwards, a local search is performed on each population member to improve its experience and thus obtain a population of local optimum solutions. Then, crossover and mutation operators are applied, similar to GAs, to produce offsprings. These offsprings are then subjected to the local search so that local optimality is always maintained.

```
Procedure MA;
Begin;
    Initialize population P;
    For each individual i ∈ P: calculate fitness (i);
      For j ∈ 1 to #generations;
        For each individual i ∈ P: do Local-Search (i);
        Perform crossover;
          Select two parents i_a , i_b ∈ P randomly;
          Generate offspring i_c = Crossover (i_a , i_b);
          i_c = Local-Search (i_c);
        Perform mutation;
          Select an individual i ∈ P randomly;
          Generate offspring i_c = Mutate (i);
          i_c = Local-Search (i_c);
        Calculate the fitness of the offspring i_c;
        Add individual i_c to P;
        P = select (P);
    End for
End:
```

Fig. 2(b): Pseudo code for a MA procedure

IV. Fuzzy Model Identifications

The fuzzy model identification can be formulated as a search and optimization problem in high dimensional space, where each point correspond to a fuzzy system i.e. represents membership functions, rule-base and hence the corresponding system behaviour. Evolutionary algorithms have the capability to find an optimal or near optimal solution in a given complex search space and can be used to modify/learn the parameters of fuzzy model. Evolutionary algorithms offer a number of advantages over other search methods as they integrate elements of directed and stochastic search. These algorithms do not require any knowledge about the characterstics of the search space. Moreover, due to the parallel nature of the evolutionary algorithms, the possibility to reach a global minima (or maxima) is high.

The application of EAs for fuzzy model identification involves a number of important considerations.

1.Completely represent the fuzzy system within the chromosome through some encoding mechanism.

2.Define an appropriate fitness function for evaluating the chromosomes representing fuzzy models. Here Mean Square Error (MSE) defined in (1) is used for rating the quality of fuzzy model. Lower the value of MSE, better is the quality of fuzzy model.

$$MSE = \frac{1}{N}\sum_{k=1}^{N}[y(k) - y(k)']^2 \tag{1}$$

where, y(k) is the desired output and y(k)' is the actual output of the fuzzy model. N is number of data points taken for calculating MSE with training data set as well as test data set.

4.1 Encoding Mechanism

We have considered only multi-input single-output (MISO) fuzzy model with n number of inputs as shown in figure 3. The number of fuzzy sets for the inputs are m_1, m_2, m_3,..., m_n respectively.

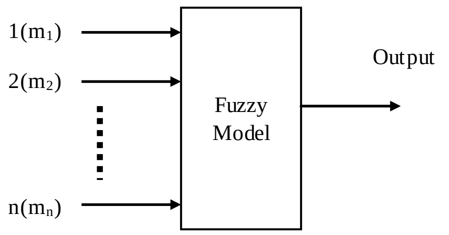

Fig. 3: A multi-input, single-output fuzzy model

Some of the assumptions used for model formulation are listed below.

1 Only triangular type membership functions are used for both input and output variables.

2 Number of membership functions for each input and output variables are kept fixed.

3 First and last membership functions of each input and output variables are represented by zed and sigma type respectively.

4 Complete rule base is considered.

5 Overlapping between the adjacent membership functions for all the variables are ensured through some defined constraints.

4.1.1. Encoding method for Membership functions

Consider a triangular membership function and let x_k^l, x_k^c, and x_k^r represent the coordinates of left anchor, cortex and right anchor of the k^{th} linguistic variables as shown in figure 4(a). Zed type membership function is shown in figure 4(b) with x_1^l, x_1^c, and x_1^r as the coordinates of left anchor, cortex and right anchor of the 1^{st} linguistic variables. Here $x_1^l = x_{min}$ and x_{min} is the lower limit of the universe of discourse of that variable. Sigma type membership function is shown in figure 4(c) with x_n^l, x_n^c, and x_n^r as the coordinates of left anchor, cortex and right anchor of the n^{th} linguistic variables. Here $x_n^r = x_{max}$ and x_{max} is the upper limit of the universe of discourse of that variable.

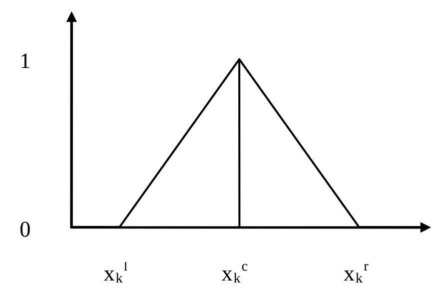

Fig. 4(a): Characteristics of a triangular membership function

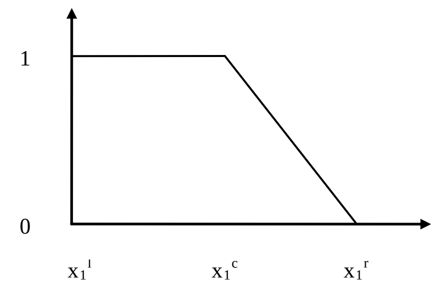

Fig. 4(b): Characteristics of a triangular membership function

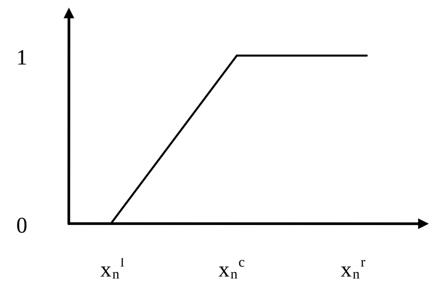

Fig. 4(c): Characteristics of a triangular membership function

The following constraints are imposed for every membership function of input and output variables.

$$x_k^l < x_k^c < x_k^r$$

Thus the parameters of the membership functions for the input and output variables are represented by the chromosome as follows:

$(x_1^l, x_1^c, x_1^r, x_2^l, x_2^c, x_2^r, \ldots\ldots, x_n^l, x_n^c, x_n^r, x_{n+1}^l, x_{n+1}^c, x_{n+1}^r)$

The index n+1 corresponds to the membership functions of the output variable.

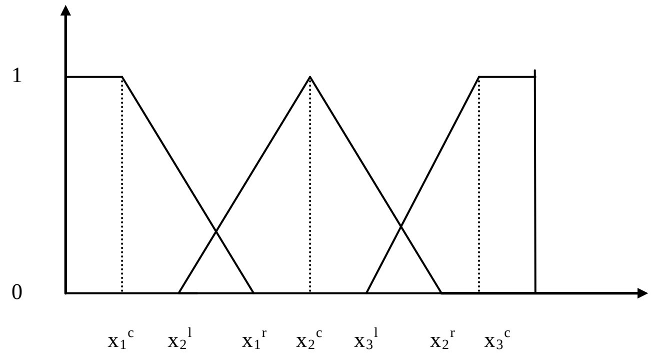

Fig. 5: Representation of overlapping through constraints for a variable with 3 membership functions

Also imposing additional constraints ensures the overlapping between the adjacent membership functions. Consider that a variable is represented by three fuzzy sets as in fig. 5, and then those additional constraints to ensure overlapping can be represented below.

$$x_{min} < x_1^c < x_2^l < x_1^r < x_2^c < x_3^l < x_2^r < x_3^c < x_{max}$$

where, x_{min} and x_{max} are the universe of discourse for that particular variable.

Also $x_{min} = x_1^l$ and $x_{max} = x_3^r$

The additional constraints represented above can be generalized for any number of membership functions and are represented as:

$$x_{min} < x_1^c < x_2^l < x_1^r < x_2^c < x_3^l < x_2^r < x_3^c, \ldots\ldots, < x_{n-2}^c < x_{n-1}^l < x_{n-2}^r < x_{n-1}^c < x_n^l < x_{n-1}^r < x_n^c < x_{max} \quad (2)$$

where $x_{min} = x_1^l$ and $x_{max} = x_n^r$

The chromosome size required to encode the membership functions for each variable is given as

$$\text{Chromosome size} = 3m_i - 2 \quad (3)$$

For fig 5, $m_i = 3$ and hence the chromosome size for encoding the variable consisting of 3 membership functions shall be 3*3 – 2 = 7.

The equation(3) can be generalized for the fuzzy system with multi input single output (MISO). The size of the chromosome to encode only membership functions for input and output variables for a Mamdani fuzzy model can be represented as in (3).

$$\text{Chromsome size (for membership functions)} = \sum_{i=1}^{n+1}(3m_i - 2) \quad (4)$$

Where, n is number of input variables, m_i is number of fuzzy sets for i^{th} input and the index n+1 corresponds to the membership functions of the output variables.

4.1.2 Encoding method for fuzzy rules

Because the complete rule base is to be considered, the size required for representing the entire rule base is given by (5).

$$\text{Chromosome size (for rule base)} = \prod_{i=1}^{n} m_i \quad (5)$$

Here each element is representing the index of the membership functions of the output variable.

Thus, the chromosome size required for encoding the Mamdani fuzzy model can be obtained in (6) by simply adding the (4) and (5).

Chromosome size (Mamdani model) =

$$\sum_{i=1}^{n+1}(3m_i - 2) + \prod_{i=1}^{n} m_i \tag{6}$$

Thus, a chromosome representing the parameters of the membership functions for input variable, output variable, rule consequents and rule base corresponding to a mamdani fuzzy model can be represented as shown in fig. 6 that also carries the details about the chromosome size. The parameters that are modified through MA have been put in shaded blocks and un-shaded blocks represent fixed parameters.

For the purpose of clarity in representation another subscript has been attached with the parameters of triangular membership functions so as to associate them with the input and output variables. For example, the three parameters for second fuzzy set of the first variable are represented as

$$x^l_{12}, x^c_{12}, x^r_{12}$$

Here first subscript represents input variable and 2nd subscript represents corresponding fuzzy set of that input.

												Chromosome Size
Input Variable #1	x^l_{11}	x^c_{11}	x^r_{11}	x^l_{12}	x^c_{12}	x^r_{12}	…	…	x^l_{1m1}	x^c_{1m1}	x^r_{1m1}	$3m_1$-2
Input Variable #2	x^l_{21}	x^c_{21}	x^r_{21}	x^l_{22}	x^c_{22}	x^r_{22}	…	…	x^l_{2m2}	x^c_{2m2}	x^r_{2m2}	$3m_2$-2
…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…
Input Variable #n	x^l_{n1}	x^c_{n1}	x^r_{n1}	x^l_{n2}	x^c_{n2}	x^r_{n2}	…	…	x^l_{nmn}	x^c_{nmn}	x^r_{nmn}	$3m_n$-2
Output Variable	y^l_1	y^c_1	y^r_1	y^l_2	y^c_2	y^r_2	…	…	y^l_m	y^c_m	y^r_m	3m-2
Rule Base (rule number)	R(1)	R(2)	R(3)	R(4)	…	…	…	…	R(m_1*m_2*…*m_n)			$\prod_{i=1}^{n} m_i$

Chromosome Size for representing Mamdani fuzzy model = $\sum_{i=1}^{n+1}(3m_i - 2) + \prod_{i=1}^{n} m_i$

Fig. 6: Representation of a Mamdani type fuzzy model by a Chromosome

In case of singleton fuzzy model with t being the number of singleton output values, (7) represent the required size of the chromosome to encode the Singleton fuzzy model.

Chromosome size (Sugeno model) =

$$\sum_{i=1}^{n}(3m_i - 2) + t + \prod_{i=1}^{n} m_i \tag{7}$$

In fig 7 a chromosome is shown which represents the parameters for input variable, output variable, rule consequent and rule base corresponding to sugeno fuzzy model. It also tells about the detail of the chromosome size.

The framework for the identification of fuzzy model through the Memetic algorithm is represented in fig. 8 and pseudo code for local search is given in fig. 9, where d is an incremental value used to find the neighbor of the variable[13].

Begin;
Define initial parameters for MA algorithm;*
Iteration = 0;
Create initial population of chromosomes;
***While** iteration ≤ Maximum iteration*
Constraint chromosomes acc. to (2);
Build fuzzy model for each chromosome@;
Evaluate each fuzzy model and calculate MSE using (1);
Apply standard procedure of MA (figure 2(b)) to get the new population;
Iteration = iteration + 1;
End
End

Fig. 8: A framework for fuzzy model identification through Memetic algorithm

*Initial parameters like population size crossover probability, mutation probability, maximum number of iterations. Local search probability.

@One chromosome represents the parameters of the membership function for each input variable, output variable, rule consequents and rule base corresponding to required fuzzy model.

```
Begin;
    Select an incremental value d=a*Rand(), where a is a
        constant that suits the variable values;
    For a given chromosome i ∈P: calculate fitness(i);
    For j ∈1 to number of variables in chromosome i;
        Value (j)=value (j)+d;
          If chromosome fitness not improved then
             value(j)=value (j)-d;
          If chromosome fitness not improved then
             retain the original value(j);
    Next j;
End
```

Fig. 9: Pseudocode for the memetic local search

												Chromosome Size
Input Variable #1	x^l_{11}	x^c_{11}	x^r_{11}	x^l_{12}	x^c_{12}	x^r_{12}	…	…	x^l_{1m1}	x^c_{1m1}	x^r_{1m1}	$3m_1-2$
Input Variable #2	x^l_{21}	x^c_{21}	x^r_{21}	x^l_{22}	x^c_{22}	x^r_{22}	…	…	x^l_{2m2}	x^c_{2m2}	x^r_{2m2}	$3m_2-2$
…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…
Input Variable #n	x^l_{n1}	x^c_{n1}	x^r_{n1}	x^l_{n2}	x^c_{n2}	x^r_{n2}	…	…	x^l_{nmn}	x^c_{nmn}	x^r_{nmn}	$3m_n-2$
Output Variable	c_1	c_2	c_3	c_4	…	…	…	…	…	…	c_t	t(say)
Rule Base (rule number)	R(1)	R(2)	R(3)	R(4)	…	…	…	…	$R(m_1{*}m_2{*}\ldots{*}m_n)$			$\prod_{i=1}^{n} m_i$

Chromosome Size for representing Sugeno fuzzy model = $\sum_{i=1}^{n}(3m_i-2)+t+\prod_{i=1}^{n} m_i$

* The shaded block represents the parameters to be modified through MAs and an un-shaded block represents fixed parameters.

x^l_{i1} and y^l_1 are equal to the minimum values of the input and output variables and x^r_{i1} and y^r_1 are equal to the maximum value of the input and output variables.

Fig. 7: Representation of a Sugeno type fuzzy model by a Chromosome

V. Illustrative Examples

In the following four subsections, we consider four different modeling problems. The first example involves the modeling of a rapid Ni-Cd battery charger and second is the modeling of gas-furnace based upon given data. This is followed by the application of this approach to two benchmark datasets for classification problem i.e. Iris data classification problem and the wine data classification problem. Some of datasets from the complete data is chosen for training and validation dataset has been taken from rest of the data sets. We compare the obtained results with the results found in literature. Table 1 shows the preferred parameters for the modeling of these examples.

Here the initial population size 20 has been chosen in order to reduce the computational time and the complexity of the problem. After the selection of initial population size the other parameters were set on the basis of the study from literature and the personal expertise of the author.

Table 1: Parameters used in MAs

Parameter	Value
Population Size	20
Crossover rate (XOVR)	0.8
Mutation probability (p_m)	0.09
Generation Gap (GGAP)	0.7
Local Search Probability (p_{ls})	0.9
Interval (T)	10

A Memetic algorithm raises a number of important issues and the foremost issues may be stated as[7]:

"What is the best trade-off between local search and the global search provided by evolution?"

This leads naturally to questions such as:

- How often should local search be applied?
- On which solutions should local search be used?
- How long local search should be used?
- How efficient does local search need to be?

Parameters local search probability (p_{ls}) and interval (T) is used to overcome these issues. The local search phase is not activated at every generation, but at an interval of every T generation with T ≥ 1 and with local search probability (p_{ls}) selected by the user. This is done to avoid a high computational cost of building and optimizing approximations at every generation. Performing local learning and search at every generation may not leave time for global evolution and significant sampling of the search space. Here we choose a big probability after a selected interval 'T'(time) in order to achieve good results and to reduce complexity and computational cost.

5.1 Rapid Ni-Cd Battery Charger

The suggested approach has been applied for identification of fuzzy model for the rapid Nickel-Cadmium (Ni-Cd) battery charger, developed in [14]. The main objective of development of this charger was to charge the batteries as quickly as possible but without any damage. This data set consists of 561 input-output points, available at [15].

For this charger, the two input variables used to control the charging rate (Ct) are absolute temperature of the batteries (T) and its temperature gradient (dT/dt). Maximum charging current can be 8C where C is capacity of battery [16]. In case of 2 AA battery with a capacity of 500 mAh, charging current is 500 * 8 = 4A. The input and output variables identified for rapid Ni-Cd battery charger along with their universe of discourse are listed in table 2.

Table 2: Input and output variables for rapid Ni-Cd battery charger along with their universe of discourse

INPUT VARIABLES	MINIMUM VALUE	MAXIMUM VALUE
Temperature (T)[^{0}C]	0	50
Temperature Gradient (dT/dt)[^{0}C/10sec]	0	1
OUTPUT VARIABLE		
Charging Rate (Ct)[A]	0	8C

From the complete data set approximately 10 percent data sets have been chosen for training and another 10 percent from rest of the data is used for validation. Table 3 shows the Results obtained from the present approach for mamdani type and singleton type fuzzy model using GAs and MAs. These are compared with the Results found in literature (Table 4).

Table 3: Results for Rapid Ni-Cd Battery Charger from Present Approach.

Model	No. of Iterations	Performance (MSE)	
		GAs	MAs
Mamdani	2000	0.0135	0.0032
Singleton	2000	0.0367	0.0235

Table 4: Results for Rapid Ni-Cd Battery Charger from Literature

Author	Model	No. of Iterations	Performance (MSE)
Khosla et. al [17]	Mamdani	2500	0.1455
Khosla et. al [17]	Singleton	2500	0.1123

Fig 10 shows the graph between actual output and output obtained from GAs for training dataset and validating dataset respectively and fig 11 shows the graph between actual output and output obtained from MAs for training dataset and validating dataset respectively.

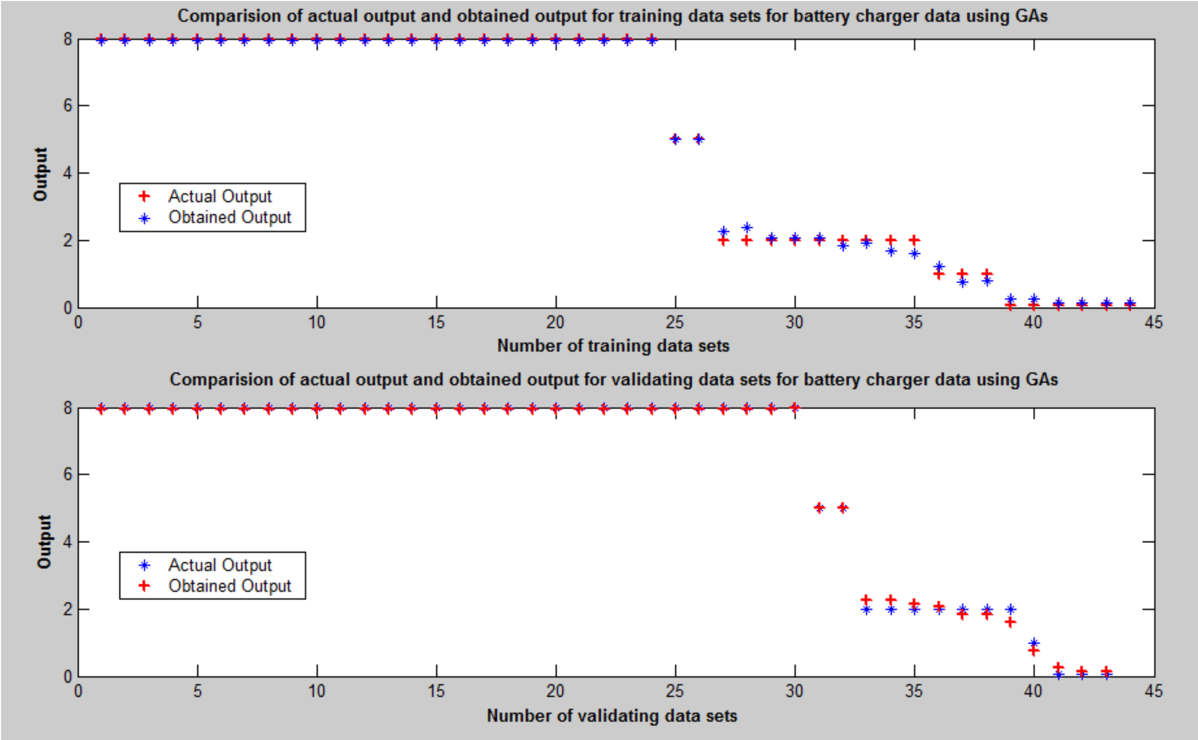

Fig. 10: Performance of training and validating data sets obtained from GAs

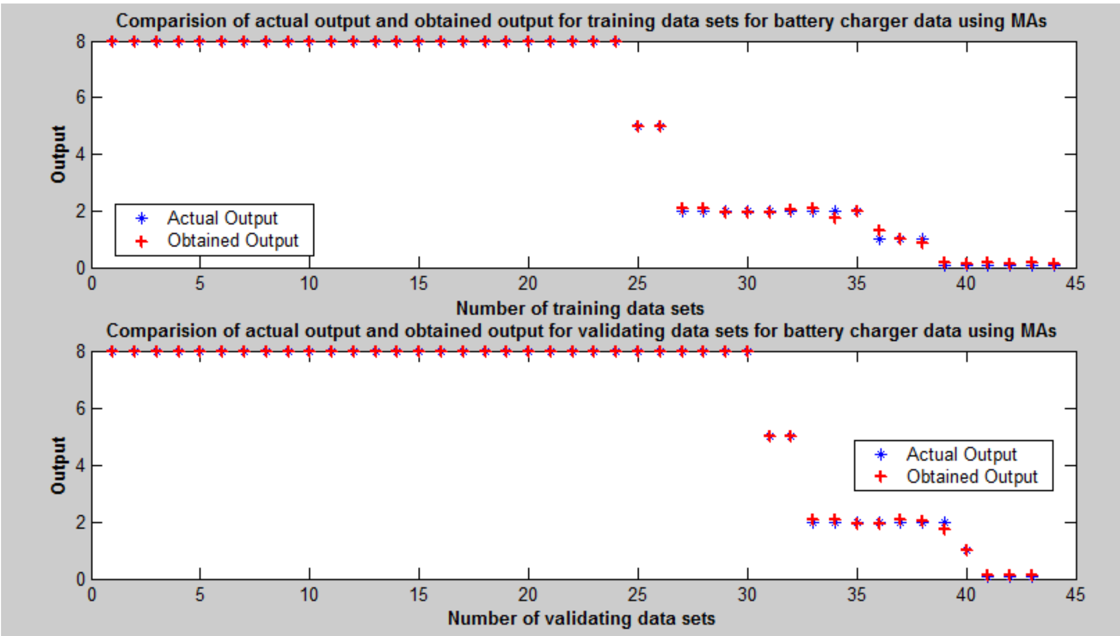

Fig. 11: Performance of training and validating data sets obtained from MAs

5.2 Box & Jenkin's Gas Furnace Data

Box & Jenkin's gas furnace data is a single input single output time series data for a gas furnace process with gas flow rate u(t) as the input and y(t), the CO_2 concentration as the output. Sugeno and Yasukava [17] consider 10 input variables which are y(t-1),…,y(t-4),u(t-1),…,u(t-6) as candidates to effect the output y(t). The original data set contains 296 data pairs and with these settings only 290 of them can be used. Results obtained from the present approach with GAs and MAs are shown in Table 6 and the results obtained from literature are shown in Table 5. Here we have chosen approximately 10 percent data sets from the complete data sets for training and 20 percent data sets from the rest data sets for validation. Performance is measured in terms of MSE[18].

Table 5: Results for the Gas Furnace Data from Literature

Author	Inputs	No. of Rules	Performance (MSE)
Position Gradient Model[17]	y(t-1) u(t-4) u(t-3)	6	0.190
Tong [19]	y(t-1) u(t-4)	19	0.469
Pedryez[20]	y(t-1) u(t-4)	81	0.320
Xu-Lu[21]	y(t-1) u(t-4)	25	0.328
Sugeno et. al[22]	y(t-1) u(t-4)	6	0.355
Linear model	y(t-1) y(t-2) u(t-3) u(t-4) u(t-6)	---	0.193

Table 6: Results for the Gas Furnace Data from Present Approach

Input	No. of Iterations	No. of Rules	Performance (MSE)	
			GAs	MAs
y(t-1) u(t-4)	5000	8	0.0769	0.0076
y(t-1) u(t-4) u(t-3)	5000	16	0.1585	0.1546

Fig 12 shows the graph between actual output and output obtained from GAs for training and validating datasets respectively and fig 13 shows the graph between actual output and output obtained from MAs for training and validating datasets respectively.

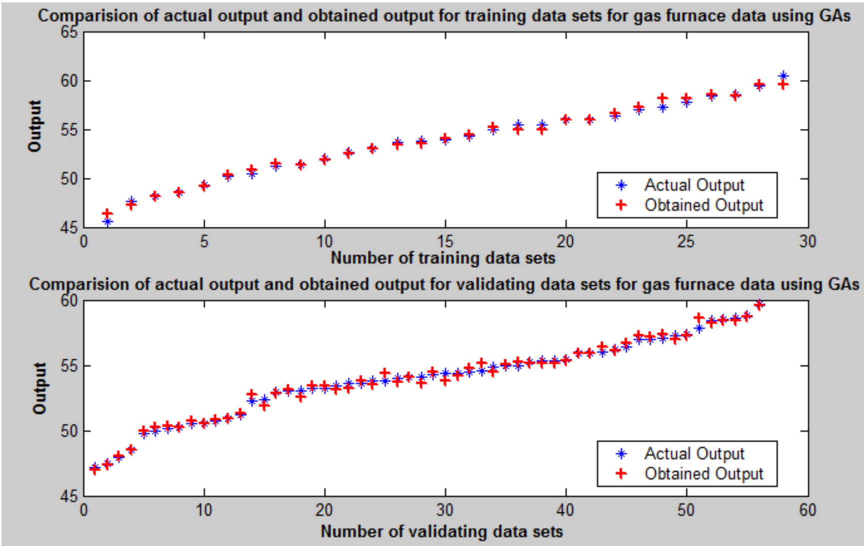

Fig. 12: Performance of training and validating data sets of Box & Jenkins Gas Furnace Data obtained from GAs

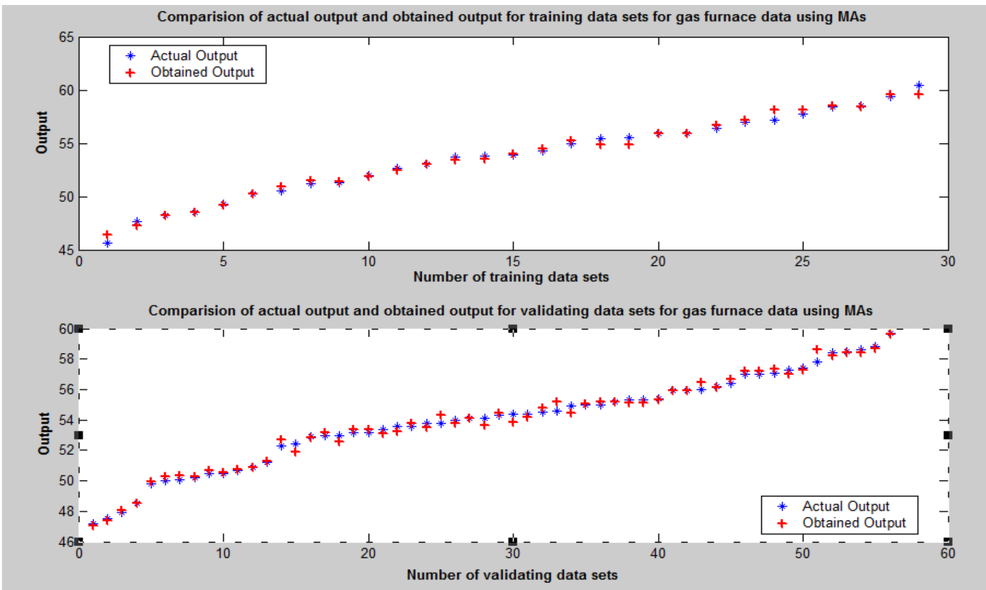

Fig. 13: Performance of training and validating data sets of Box & Jenkins Gas Furnace Data obtained from MAs

The results suggest that the performance of our approach is much better than the others found in literature.

5.3 Iris Data classification

The iris data is a common benchmark in classification and pattern recognitions [23-2]. It contains 50 measurement of three species *Iris Setosa*, *Iris Versicolor*,and *Iris Virginica*. We label the species 1,2 and 3 respectively, which gives a 150 * 5 pattern matrix Z of observational vectors.

$$Z_k = [x_{k1}, x_{k2}, x_{k3}, x_{k4}, c_k]$$

$$c_k = \begin{cases} 1 & \text{if } y_k < 1.5 \\ 2 & \text{if } 1.5 \le y_k < 2.5 \\ 3 & \text{if } 2.5 \le y_k \end{cases}$$

$c_k \in [1,2,3]$, $k = 1,2,\ldots150.$

Where x_{k1}, x_{k2}, x_{k3}, and x_{k4} are the sepal length, sepal width, petal length and petal width respectively. In order to perform classification the output y_k was used with the following classification rule:

Solutions obtained in the literature are shown in Table 7.

Table 7: Results for Iris Data from Literature.

Authors	Classification Rate	No. of Misclassifications
Shi et. al[23]	98%	3
Ishibuchi et. al[25]	84%	24
Bezdek et. al[26]	98%	3
FRBL[27]	98%	3
Magni Setnes et. Al[28]	99.3%	1

For training we choose 10 percent of the data from complete data set. The performance of the proposed approach with sugeno type system is shown in Fig. 14 and Table 8 shows the classification rate for Iris data classification using GAs and MAs.

Table 8: Results for fhe Iris Data from Present Approach

Algorithm	Classification Rate	No. of Misclassifications
MAs	100%	Nil
GAs	100%	Nil

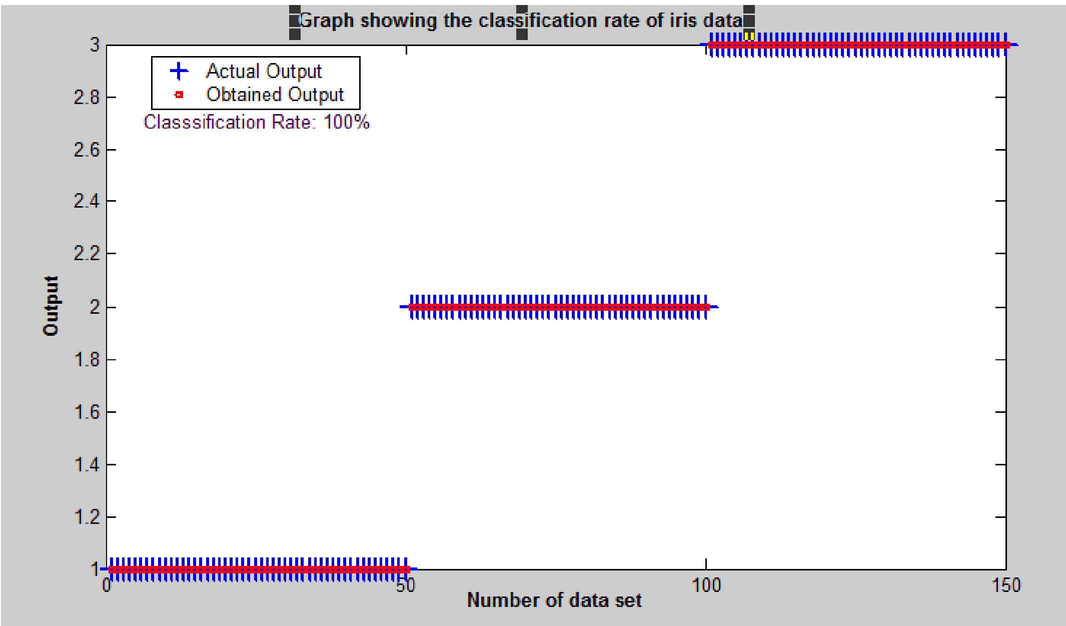

Fig. 14: Performance of Iris data classification

5.4 Wine Data Classification

This data is the result of a chemical analysis of wines from the same region but with different types of groups, using 13 continuous variables. The 13 continuous variables are: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non flavanoids, phenols, proanthocyaninsm color intensity, hue, OD280/OD315 of diluted wines and proline. The obtained result from the literature is shown in Table 9.

Table 9: Results for the Wine Data from Literature

Authors	Classification Rate	No. Of Misclassifications
Ishibuchi et. al[24]	97.8%	4
FRBL[27]	100%	Nil
Magni Setnes et. al[28]	98.3%	3
Corcoran and sen[29]	98.3%	3

From the observation we observe that the features ash, magnesium, total phenols, nonflavanoids phenols, proanthocyaninsm color intensity and hue have little variation over the three classes. Hence we ignore these features and retained the rest of features for working out with sugeno type model. We have taken 10 percent data sets from complete data sets to train the system. The performance of the proposed approach is shown in Fig. 15 and Table 10 gives the classification rate of MAs and GAs, which is better than the others found in literature.

Table 10: Results for the Wine Data from Present Approach

Algorithm	Classification Rate	No. Of Misclassifications
MAs	98.88%	2
GAs	97.19%	5

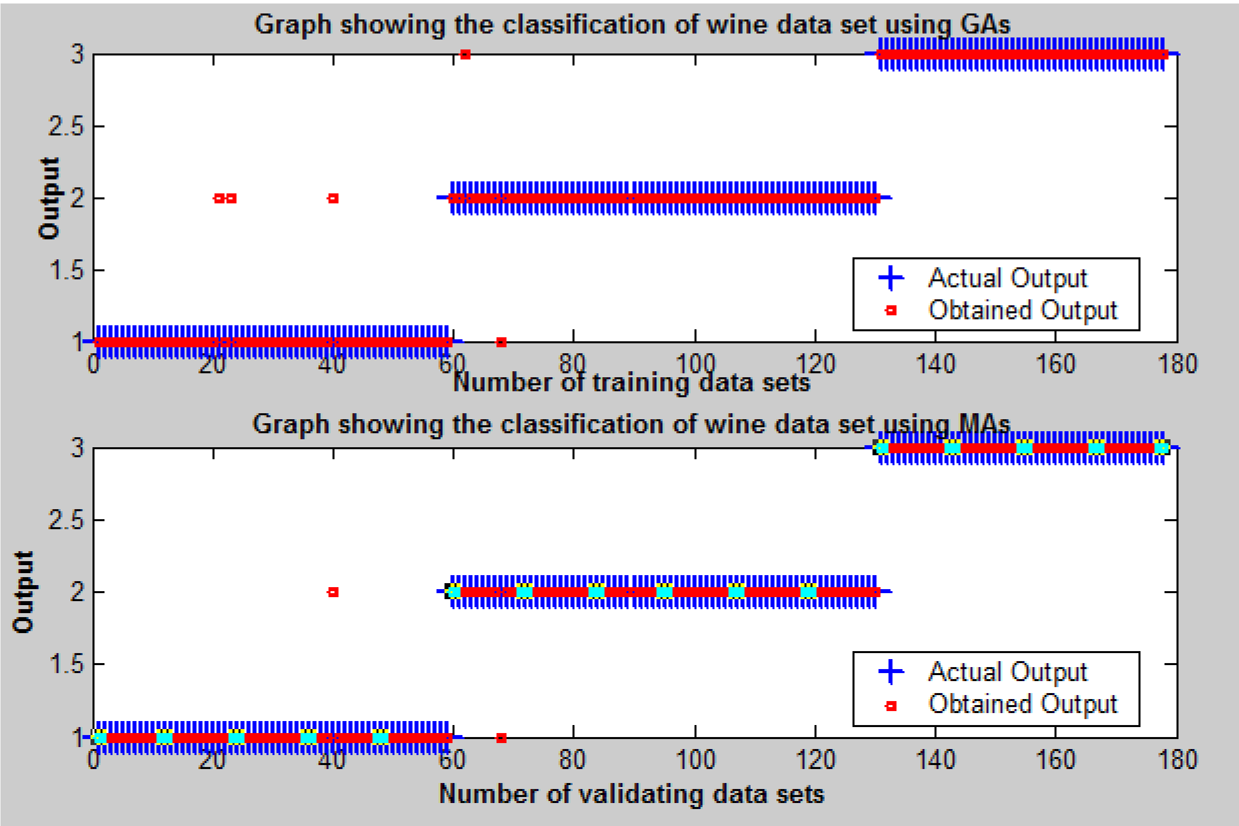

Fig. 15: Performance of Wine data classification obtained from MAs and GAs

VI. Conclusions & Future Scope

In this paper, we have described the use of memetic algorithms (MAs) for the identification of fuzzy models from the available data. The proposed modeling approach was successfully applied to four well-known problems from literature: A rapid Ni-Cd battery charger, the Box & Jenkins's gas-furnace data, the Iris data classification problem and the wine data classification problem. The proposed approach is better than the results reported in literature. This paper also compares the results obtained from MAs and GAs, when used for identifying fuzzy models of same complexity that were generated from the same data. The results bring out the tremendous efficiency of MAs. The suggested framework can be extended to increase the flexibility of the search by incorporating additional parameters so that the search for the optimal solution could be executed in terms of number of membership functions for each variable, the type of membership function and the number of rules.

References

[1] G. J. Klir and B. Yuan, "*Fuzzy Sets and Fuzzy Logic—Theory and Applications",* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] L. A. Zadeh, "Fuzzy logic and approximate reasoning,", *Syntheses,* vol. 30, pp. 407–428, 1975.

[3] Goldberg DE, "Genetic algorithms in search, optimization and machine learning. Reading", MA: Addison-Wesley Publishing Co; 1989.

[4] Hegazy T., "Optimization of construction time-cost trade-off analysis using genetic algorithms", Can J Civil Eng 26: 685–97, 1999.

[5] Al-Tabtabai H, Alex PA. "Using genetic algorithms to solve optimization problems in construction", Eng Constr Archit Manage;6(2):121–32,1999.

[6] Joglekar A, Tungare M., "Genetic algorithms and their use in the design of evolvable hardware", http://www.manastungare.com/articles/genetic/geneticalgorithms.pdf; 2003, accessed on,15 pp.,May 20,2004.

[7] P. Merz and B. Freisleben, "Fitness landscapes and memetic algorithm design", in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, 1999.

[8] A. Kolen and E. Pesch, "Genetic local search in combinatorial optimization", *Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, vol. 48, pp. 273–284, 1994.

[9] Dawkins R. The selfish gene. Oxford: Oxford University Press; 1976.

[10] Merz P, Freisleben B. "A genetic local search approach to the quadratic assignment problem", In: Bäck CT, editor. Proceedings of the 7th international conference on genetic algorithms. San Diego, CA: Morgan Kaufmann; pp. 465–72, 1997.

[11] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms", Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, Tech. Rep. 790, 1989.

[12] P. Moscato and M. G. Norman, "A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems", in *Parallel Computing and Transputer Applications*, M. Valero, E. Onate, M. Jane, J. L. Larriba, and B. Suarez, Eds. Amsterdam, The Netherlands: IOS Press, pp. 177–186,1992.

[13] Emad Elbeltagi, Tarek Hegazy, Donald Grierson, "Comparison Among Five Evolutionary-Based Optimization Algorithms", Advance Engineering Informatics ELSEVIER, vol. 19, pp. 43-53, 2005.

[14] Khosla, A., Kumar, S., & Aggarwal, K.K. "Design and development of RFC-10: A Fuzzy Logic Based Rapid Battery Charger for Nickel-Cadmium Batteries." HiPC(High Performance Computing) workshop on soft computing, Bangalore, pp 9-14, 2002.

[15] http://www.research.4t.com

[16] Linden, D. (Editor-in-Chief) Handbook of Batteries. McGraw Hill Inc., 1995.

[17] Arun Khosla, Shakti Kumar, K.K.Aggarwal, Jagatpreet Singh, " Particle Swarm Optimizer For Building Fuzzy Models", Proceeding of one week workshop on applied soft computing SOCO-2005, Haryana Engg. College,Jagadhri,India , July 25-30, pp 43-71,2005.

[18]]M. Sugeno, T.A. Yasukawa, " A fuzzy logic based approach to qualitative modeling", IEEE Trans. Fuzzy syst. 31, pp. 7-31, 1993.

[19] R.M. tong, "The evaluation of fuzzy models derived from experimental data", Fuzzy sets and systems, vol. 4, pp. 1-12,1980.

[20] Witold Pedrycz, "An Identification Algorithm In Fuzzy Relational Systems", Fuzzy Sets And Systems, Vol 13, pp. 153-167, 1984.

[21] Chen-Wei Xu and yong-Zai Lu, "Fuzzy Model Identification And Self Learning For Dynamic Systems", IEEE Trans. On Systems, Man And Cybernatics, Vol. Smc-17, pp. 683-689,1987.

[22] Michio Sugeno and Takahiro Yasukawa, " Linguistic Modeling Based On Numerical Data", Proc. Of IFSA'91, Brussel,1991.

[23] Y. Shi, R. Eberhart, and Y. Chen, "Implementation Of Evolutionary Fuzzy Systems", IEEE Trans. Fuzzy Syt., Vol 7, pp. 109-119, Apr. 1999.

[24] J.C. Bezdek, J.M. Keller, R. Krishnapuram, L. I. Kuncheva, and N.R. Pal, "Will The Real Iris Data Please Stand Up?", IEEE Trans. Fuzzy Syst., Vol. 7, pp. 368-369, June 1999.

[25] H. Ishibuchi and T. nakashima, "Voting In Fuzzy Rule-Based Systems For Pattern Classification Problems", Fuzzy Sets And Syst., Vol. 103, pp. 223-238, 1999.

[26] J.C. Bezdek, T. R. Reichherzer, G.S. Lim, and Y. Attokiouzel, "Multiple-Prototype Classifier Design," IEEE Trans. Syst., Man, Cybern., Vol. 28, pp. 67-79, Feb, 1998.

[27] Hartmut Surmann and Alexander Selenschtschikow, "Automatic Generation of Fuzzy Logic Rule Bases: Examples I", Proc. Of The NF2002: First International ICSC Conference

on Neuro-Fuzzy Technologies, pp 75, CUBA 16-19 Jan.2002.

[28] Magne Setnes and Hans Roubos, "GA – Fuzzy Modeling and Classification: Complexity and Performance", IEEE Trans. On Fuzzy Syst., Vol. 8, No. 5, October 2000.

Authors' Profiles

Savita Wadhawan is working as Assistant Professor in M.M. University Mullana. She received his M.Sc. in Information Technology and M.Tech in Computer Science and Engineering from Guru Jambheshwar University, Hisar. Her area of interest is fuzzy logic and Soft Computing.

Gunjan Goel is working as Assistant Professor in Jaypee University of Information Technology; He earned his PhD from National Dairy Research Institute followed by postdoctoral training in European universities namely University of Hohenheim, Germany and University of Ghent, Belgium. His area of interest is Microbiology and Genetic Engineering.

Srikant Kaushik is working as Assistant Professor in M.M. University Mullana. He received his BTech degree in Biotechnology from Kurukshetra University and MTech degree in Biotechnology from Maharishi Markandeshwar University, Ambala. His area of interest is Genetic Engineering.

How to cite this paper: Savita Wadhawan, Gunjan Goel, Srikant Kaushik,"Data Driven Fuzzy Modeling for Sugeno and Mamdani Type Fuzzy Model using Memetic Algorithm", International Journal of Information Technology and Computer Science(IJITCS), vol.5, no.8, pp.24-37, 2013. DOI: 10.5815/ijitcs.2013.08.03