J. of Mult.-Valued Logic & Soft Computing, Vol. 25, pp. 555–587
Reprints available directly from the publisher
Photocopying permitted by license only

Nature Inspired Approaches for Identification of Optimized Fuzzy Model: A Comparative Study

PARVINDER KAUR[1], SHAKTI KUMAR[2] AND AMAR PARTAP SINGH[3]

[1]*Maharishi Markandeshwar University, Mullana, Ambala, Haryana, INDIA*
[2]*Surya World Institutions of Academic Excellence, Rajpura, Punjab, INDIA*
[3]*Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, INDIA*
parvinderbhalla@gmail.com, shaktik@gmail.com, amarpartapsingh@yahoo.com

Received: June 11, 2013. Accepted: March 27, 2014.

The identification of an optimized model is one of the key issues in the field of fuzzy system modeling. This has gained significant importance since; most of the real life systems are highly complex and nonlinear. Fuzzy model identification involves two stages i.e. identification of input and output membership functions as well as generation of rule base for the system being modeled. The fuzzy modeling or fuzzy model identification can be formulated as a search and optimization problem where the goal is to find the parameters of fuzzy model based on some evaluation criteria such that model gives optimal performance. Therefore, search and optimization techniques can be applied to the problem of model identification. Owing to their ability to deal with highly complex problems nature/biologically inspired approaches are currently amongst the most powerful algorithms suitable for fuzzy model identification problems. The research work reported in this paper is focused on five new approaches to model identification based on nature inspired optimization algorithms namely Biogeography Based Optimization Approach (BBO), Big Bang - Big Crunch Optimization Based Approach (BB-BC), Artificial Bee Colony Optimization Based Approach (ABC), Ant Colony Optimization Based Approach (ACO) and Firefly Algorithm Based Approach (FA). These approaches have been implemented and validated successfully on two standard benchmark data sets to suggest robust, tractable and low cost solutions.

Keywords: Soft Computing, Nature Inspired Approaches, Membership function, Rule base, Fuzzy Modeling, Fuzzy Model Identification.

1. INTRODUCTION

The principles of fuzzy modeling were outlined by Zadeh in [1] where he proposed a new approach that "provides an approximate and yet effective means of describing the behavior of systems which are too complex or too ill-defined to admit use of precise mathematical analysis." The models proposed by Zadeh present three distinguishable features: (1) the use of linguistic variables in place or in addition to numerical variables, (2) the description of simple relations between variables by conditional fuzzy statements, and (3) the characterization of complex relations by fuzzy algorithms. Current fuzzy modeling techniques still follow these principles. An important issue in designing fuzzy models involves the question of providing a methodology for their development. i.e., a method for obtaining the fuzzy model from numerical information and knowledge about the system [2].

Fuzzy modeling is the task of identifying the parameters of a fuzzy inference system so that a desired behavior is attained [3]. Due to linguistic and numeric requirements, the fuzzy-modeling process has generally to deal with an important trade-off between the *accuracy* and the *interpretability* of the model. In other words, the model is expected to provide high numeric precision while incurring as little a loss of linguistic descriptive power as possible. With the *direct* approach a fuzzy model is constructed using knowledge from a human expert (Knowledge-driven modeling) [3], [4]. This task becomes difficult when the available knowledge is incomplete or when the problem space is very large, thus motivating the use of *automatic* approaches to fuzzy modeling (Data-driven modeling) [5]. This modeling approach makes use of numeric information obtained from input-output measurements. It is also possible to integrate both the modeling approaches [6]. One of the major problems in fuzzy modeling is the *curse of dimensionality*, meaning that the computation requirements grow exponentially with the number of variables.

The problem of fuzzy model identification includes the following issues [2], [7]:

- Selecting the type of fuzzy model.
- Selecting input and output variables for the model.
- Choosing the structure of membership functions.
- Determining the number of fuzzy rules.
- Identifying the parameters of antecedent and consequent membership functions.
- Identifying the consequent parameters of rules.
- Defining some performance criteria for evaluating fuzzy models.

The survey of the field reveals that literature is rich with a large number of classical/hard computing (Exact reasoning) based approaches such as Clustering, Singular value decomposition, Decision trees etc. We also found that few soft computing (Approximate reasoning) based approaches like Evolutionary Computation, Neural Networks and Swarm Intelligence have also been developed and applied for fuzzy model identification [8]-[47].

In classical/hard computing based approaches we require a precisely stated analytical model with associated computational complexity. The real word problems exist in a non-ideal environment i.e. the real world problems are pervasively imprecise and uncertain. Precision and certainty carry a cost. This cost grows exponentially with the system complexity. This is where approximate reasoning or soft computing approaches come into play. The guiding principle of soft computing is to exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractable, robust and low cost solutions [48]. Soft computing plays an important role in situations where we could replace "the best for sure" solution with "good enough with high probability" solutions. Hence, soft computing approaches seem to be very effective solutions for the model identification problems.

In real life we have to design engineering systems which are very complex and highly nonlinear. In order to reduce design cycle time there will always is a need for a fast and better modeling approach which provides tractable, robust and low cost solution to engineering system modeling. Hence, a search of more and more of such techniques for fuzzy modeling will always be a key question. This research work is also an attempt to answer such an important question partially. This paper proposes five new methodologies and techniques for optimized fuzzy model identification.

This paper is organized as follows. Section 2 provides a brief account of five nature/biologically inspired optimization algorithms. In Section 3, the framework for fuzzy model identification based upon nature inspired algorithms namely BBO, BB-BC, ABC, ACO and FA is presented. Section 4 represents experimental results considering two real world problems one control problem and other Iris data classification problem. Finally, conclusions are drawn in Section 5.

2. NATURE INSPIRED APPROACHES

2.1 BBO Algorithm

Biogeography-based optimization is a population-based evolutionary algorithm that is based on the mathematics of biogeography. Biogeography is the study of the geographical distribution of biological organisms. In BBO, problem solutions are represented as islands, and the sharing of features between solutions is represented as emigration and immigration. An island is any habitat that is geographically isolated from other habitats [49]-[58].

Biogeography-based optimization was first presented in [49] and is an example of how a natural process can be modeled to solve general optimization problems. This is similar to what has occurred in the past few decades with genetic algorithms, neural networks, ant colony optimization, particle swarm optimization, and other areas of computer intelligence. Biogeography is nature's way of distributing species, and is analogous to general problem solving. Suppose that we have some problem, and that we also have a certain number of candidate solutions. A good solution is analogous to an island with a high HSI (Habitat suitability index), and a poor solution is like an island with a low HSI. Features that correlate with HSI include factors such as rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature. The variables that characterize habitability are called suitability index variables (SIVs). High HSI solutions are more likely to share their features with other solutions, and low HSI solutions are more likely to accept shared features from other solutions. As with every other evolutionary algorithm, each solution might also have some probability of mutation, although mutation is not an essential feature of BBO. The pseudo-code for BBO Algorithm is shown in Figure 1.

```
Biogeography–Based Optimization
  Begin
    /* BBO parameter initialization */
          Generate a random set of habitats (integer population) H1,H2,...,Hn ;
          Compute corresponding HSI value;     /* HSI represents the fitness value */
          Sort the population from best to worst based on HSI value;
    /* End of BBO parameter initialization */
    While not T    /* T is a termination criterion */
        Keep the best individuals as elites;
        Compute immigration rate λ and emigration rate μ for each habitat based on HSI;
        /* Migration */
        Select Hi with probability based on λi;          /* see probability Eq. (A.1)*/
        If Hi is selected
            Select Hj with probability based on μj;
            If Hj is selected
                Randomly select an SIV from Hj;
                Replace a random SIV in Hi with one from Hj;
            End if
        End if
        /* End of migration */
        /* Mutation*/
        Select an SIV in Hi with probability based on the mutation rate Pmutate;
        If Hi(SIV) is selected
            Replace Hi(SIV) with a randomly generated SIV;
        End if
        /* End of mutation */
          Recompute the HSI value of the modified habitats;
          Sort the population from best to worst based on HSI value (cost);
          Replace the worst with the previous generation's elites;
          Sort the population from best to worst based on HSI value;
    End while
  End
```

$$\dot{P}_s = \begin{cases} -(\lambda_s + \mu_s)P_s + \mu_{s+1}P_{s+1}, & S = 0 \\ -(\lambda_s + \mu_s)P_s + \lambda_{s-1}P_{s-1} + \mu_{s+1}P_{s+1}, & 1 \le S \le S_{\max} - 1 \\ -(\lambda_s + \mu_s)P_s + \lambda_{s-1}P_{s-1}, & S = S_{\max} \end{cases} \quad \text{.......(A.1)}$$

where λ_s and μ_s are the immigration and emigration rates when there are S species in the habitat.

FIGURE 1
Pseudo-code for BBO Algorithm.

2.2 BB-BC Algorithm

The Big Bang theory is a broadly accepted theory for the origin and evolution of our universe [59]. Inspired by this theory, an optimization algorithm is constructed, which is referred to as the Big Bang-Big Crunch optimization algorithm [59]-[61].

Randomness can be seen as equivalent to the energy dissipation in nature while convergence to a local or global optimum point can be viewed as gravitational attraction [60]. Since energy dissipation creates disorder from ordered particles, we will use randomness as a transformation from a converged solution (order) to the birth of totally new solution candidates (disorder). The creation of the initial population randomly is called the Big Bang phase. In this phase, the candidate solutions are spread all over the search space in an uniform manner. The Big Bang phase is followed by the Big Crunch phase. The Big Crunch is a convergence operator that has many inputs but only one output, which can be named as the centre of mass. An optimization algorithm must converge to an optimal point; but, at the same time, in order to be classified as a global algorithm, it must contain certain different points within its search population with a decreasing probability. This ratio of solution points around the optimum value to points away from optimum value must decrease as the number of iterations increases. This convergence can be accomplished by spreading new off-springs around the centre of mass using a normal distribution operation in every direction where the standard deviation of this normal distribution decreases as the number of iterations of the algorithm increases. These successive explosion and contraction steps are carried repeatedly until a stopping criterion has been met. The pseudo-code for BB-BC Algorithm is shown in Figure 2.

2.3 ACO Algorithm

Ant colony optimization [21] is a metaheuristic that belongs to the group of swarm intelligence based techniques. In group of insects, which live in colonies, such as ants and bees, an individual can only do simple tasks of its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows. ACO is an agent-based meta-heuristic for combinatorial optimization problems motivated by the ability of real ants to find the shortest path between their nest and a food source. This is attributed to the fact that ants lay a chemical substance, called a pheromone, along the paths they take, and when presented with a choice between alternative paths, they tend to choose the one with the greatest amount of pheromone. Pheromone, however, evaporates so that over time the shortest path accrues more pheromone as it is traversed more quickly. In a number of experiments presented in [21]-[23], Dorigo et al. illustrated the complex behaviour of ant colonies towards path length optimization. The Simple ACO (S-ACO) is an algorithmic implementation that adapts the behavior of real ants to solutions of mini-

Big Bang Big Crunch Algorithm

Begin

/* **Big Bang Phase** */

Generate a random set of candidates (population);

/* **End of Big Bang Phase** */

While not T /* T is a termination criterion */

Compute the fitness value of all the candidate solutions;

Sort the population from best to worst based on fitness (cost) value;

/* **Big Crunch Phase** */

For guiding the new search compute the center of mass using Eq. (B.1). Best fit individual can be chosen as the centre of mass instead of using Eq. (B.1);

/* **End of Big Crunch Phase** */

Calculate new candidates around the center of mass by adding or subtracting a normal random number whose value decreases as the iterations elapse using Eq. (B.2); /* Big Bang Phase */

End while

End

$$x_c = \frac{\sum_{i=1}^{N} \frac{1}{f^i} x_i}{\sum_{i=1}^{N} \frac{1}{f^i}} \quad \text{....(B.1)}; \qquad x^{new} = x_C + lr/k \quad \text{....(B.2)}$$

where x_c = position of the centre of mass; x_i = position of candidate; f^i = fitness function value of candidate i; N = population size; l = upper limit of the parameter; r = normal random number; k = iteration step and x^{new} = new candidates around the center of mass.

FIGURE 2
Pseudo-code for BB-BC Algorithm.

mum cost path problems on graphs [21]. The pseudo-code for S-ACO Algorithm is shown in Figure 3.

2.4 ABC Algorithm

ABC algorithm proposed by Karaboga in 2005 for real parameter optimization, simulates the foraging behaviour of a bee colony [62]-[65]. In the ABC algorithm, the colony of artificial bees contains three groups of bees:

Ant Colony Optimization Algorithm

Nomenclature:

L^k = Length of ant k's path

ρ = evaporation constant, $\rho \in (0,1]$

$\Delta\tau^k$ = increment in pheromone quantity = $1/L^k$

N_i^k = neighborhood of ant k when at node i.

α = a constant = 2

1. Create N ants. Place them all on source node. Deposit a constant amount of pheromone $\tau_{ij} = 1$ on all the edges.
2. for time step = 1 to S do,
   - 2.1. for each ant do,
     - 2.1.1. Select an out-going edge from the current node where ant is, with probability

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha}} & , if \ \ j \in N_i^k; \\ 0, if, j \notin N_i^k \end{cases}$$

       If more than one edge has same amount of pheromone, choose one randomly.
     - 2.1.2. Record this ant took this edge.
     - 2.1.3. If end of edge is terminal node then
       - 2.1.3.1. remove the loops from the path between source and terminal node just traced.
       - 2.1.3.2. update the pheromone trail of the path traversed by ant k to $\tau_{ij} \leftarrow \tau_{ij} + (1/L^k)$ on each edge traversed by this ant. Record the path taken by the ant in path_table.
       - 2.1.3.3. for each edge in the graph, evaporate pheromone trail using $\tau_{ij} \leftarrow (1-\rho)\tau_{ij}$ expression.
       - 2.1.3.4. Reset the ant back to the initial state, clear its edge history and let it move as before. (end of "for each ant" loop)
3. End of "for time step 1 to S do" loop.(time step $\leftarrow$ time step +1), if not end of loop then go to step 2).
4. If the number of time steps are reasonable then all the ants start moving on the shortest path.

FIGURE 3
Pseudo-code for ACO Algorithm.

employed bees, onlookers and scouts. A bee waiting on the dance area for making decision to choose a food source is called an *onlooker* and a bee going to the food source visited by itself previously is named an *employed* bee. A bee carrying out random search is called a *scout*. In the ABC algorithm, first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout. The pseudo-code for ABC Algorithm is shown in Figure 4.

2.5 FA Algorithm

The Firefly Algorithm is a nature-inspired, optimization meta-heuristic algorithm which is based on the social (flashing) behavior of fireflies. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. For simplicity, the flashing characteristics of fireflies are idealized in following three rules [66]-[68]:

- All fireflies are unisex, so that one firefly is attracted to other fireflies regardless of their sex.
- Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If no one is brighter than a particular firefly, it moves randomly.

Artificial Bee Colony Algorithm
Begin
/* ABC parameter initialization*/
Initialize the random population of solutions X_{ij}, $\{i = 1,2,\dots, SN, j = 1,2,\dots, D\}$ using Eq. (D.1)
Evaluate fitness value of the population;
/* End of ABC parameter initialization */
While not T /* T is a termination criterion */
Produce new solutions V_{ij} for the employed bees by using Eq. (D.2) and evaluate them;
Apply the greedy selection process for the employed bees;
Calculate the probability values P_{ij} for the solutions X_{ij} by Eq. (D.3);
Produce the new solutions V_{ij} for the onlookers from the solutions X_{ij} selected depending on P_{ij} and evaluate their fitness value;
Apply the greedy selection process for the onlookers;
Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution X_{ij} by Eq. (D.1);
Memorize the best solution achieved so far;
End while
End

$$X_{ij} = l_i + \varepsilon(u_i - l_i) \dots \text{(D.1)};\ V_{ij} = X_{ij} + \varphi_{ij}(X_{ij} - X_{kj}), k \in \{1,2,\dots,SN\} \dots \text{(D.2)};\ P_i = F(X_i) \Big/ \sum_{k=1}^{SN} F(X_k) \dots \text{(D.3)}$$

where
SN = Number of food source position; D = Number of optimization parameters;
$F(X_i)$ = Nectar amount of the food source located at X_i; φ_{ij} = random number between [-1,1];
ε = random number in the range [0,1); l_i and u_i are the lower and upper bound of the parameter X_i

FIGURE 4
Pseudo-code for ABC Algorithm.

```
Firefly Algorithm (FA)
 Begin
      /* FA parameter initialization */
            Define Objective function f(X), X= (x1,..., xd)^T;
            Generate initial population of fireflies Xi (i = 1, 2,... n)
            Compute the light intensity Ii at Xi by f(Xi);
            Define light absorption coefficient γ;
      /* End of FA parameter initialization */
      While not T                              /* T is a termination criterion /*
         For i = 1 : n                          /* all n fireflies /*
            For j = 1 : i
               If (Ij > Ii)
                  Move firefly i towards j in d-dimension;
               End if
               Attractiveness varies with distance r via exp[− γr];
               Evaluate new solutions and update light intensity;
            End for
         End for
         Rank the fireflies and find the current best;
      End while
         Postprocess results and visualization;
 End
```

FIGURE 5
Pseudo code for FA algorithm.

- The brightness of a firefly is affected or determined by the landscape of the objective function to be optimized.

Based on these three rules, the basic steps of the firefly algorithm can be summarized as the pseudo code shown in Figure 5.

3. IDENTIFICATION OF OPTIMIZED FUZZY MODEL BASED UPON NATURE INSPIRED APPROACHES

The fuzzy model identification involves finding the optimal values of the parameters of the fuzzy model based on some evaluation criteria. The application of optimization algorithm for fuzzy model identification involves a number of important considerations. The first step in applying such an algorithm is to define solution space (ranges of variables to be optimized), a set of constraints and the fitness function. Another important consideration is the solution encoding i.e. to completely encode a fuzzy system in terms of population (set of individuals). Each individual represents a fuzzy system which consists of two parts: one represents membership functions of antecedents and consequents and second part represents rule-base. It is also suggested to modify the membership functions and rule-base simultaneously, since these are codependent in a fuzzy system. In this paper, MSE is used as fitness function to evaluate the quality of fuzzy model. The ideal value of MSE would be zero.

$$MSE = \frac{1}{N}\sum_{K=1}^{N}\left[O_A - O_C\right]^2 \tag{1}$$

where,
O_A = Actual output as given in data set
O_C = Computed output of model
N = Number of data points taken for model validation
For the purpose of encoding, we consider a multi-input single-output system with n number of inputs with labels $x_1, x_{2,\ldots\ldots\ldots\ldots,} x_n$ and the number of fuzzy sets for these inputs are $m_{1,}\ m_{2,\ldots\ldots\ldots\ldots,}\ m_n$ respectively. Our encoding is based on the following assumptions:

i. Fixed number of triangular membership functions are used for both input and output variables and placed symmetrically over corresponding universes of discourse. The universe of discourse or simply universe is the working range of variable.
ii. First and last membership functions of each input and output variable are represented with z-type and sigma-type membership functions respectively.
iii. Complete rule-base is considered, where all possible combinations of input membership functions of all the input variables are considered for rule formulation.
iv. Overlapping between the adjacent membership functions for all the variables is ensured through some predefined constraints.

A) Encoding Mechanism for Tuning of the Fuzzy Membership Functions
Let us assume that first input variable is represented by three fuzzy sets (MFs) as given in Figure 6. The vertex of fuzzy sets is represented by *E's* in this

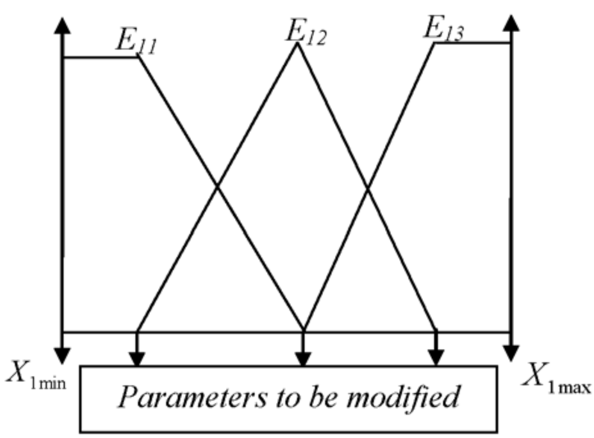


FIGURE 6
Representation of a variable with 3 membership functions with overlapping between the adjacent membership functions.

figure, where E_{11} represent vertex of first fuzzy set of first input variable. In general, E_{ni} represents vertex of i^{th} fuzzy set of n^{th} input variable.

Then the constraints to ensure the overlapping between the adjacent membership functions for all the input variables for the zero-order TSK fuzzy model (Sugeno Model) can be represented as below:

$$X_{n\,\min} < E_{n1} < E_{n2} < \ldots < E_{nm_n} < X_{n\,\max} \tag{2}$$

where m_n represents number of fuzzy sets for n^{th} input variable and $X_{n\,\min}$ and $X_{n\,\max}$ are the minimum and maximum value of the n^{th} input variable respectively.

For the adjustment of membership functions the following equations are defined:

Input Variable #n

For ensuring a movement of membership functions to right, we use the following equations:

$$E_{\mathrm{ni}} = E_{ni} + (E_{n(i+1)} - E_{ni}) * P_{\mathrm{k}} \tag{3}$$

where i =1, 2, . . . , $m_{\mathrm{n},}$ k =1,2………etc.

If ($i = m_n$), then

$$E_{ni} = E_{ni} + (X_{n\,\max} - E_{ni}) * P_k \tag{4}$$

Here P_{k} decides the percentage of movement.

For the movement of membership functions to left, we use the following equations:

$$E_{ni} = E_{ni} - (E_{ni} - E_{n(i-1)}) * P_k \tag{5}$$

If (i = 1), then

$$E_{ni} = E_{ni} - (E_{ni} - X_{n\,\min}) * P_{\mathrm{k}} \tag{6}$$

As stated earlier each individual consists of two parts; the first constituent part consists of membership functions of all the input variables. The size of membership functions (from Figure 6) while considering the assumptions made earlier, can be computed as follows:

Size of membership functions (First constituent part)

$$= \sum_{j=1}^{n} m_j \tag{7}$$

where "n" is the number of input variables, and m_j the number of membership functions for j^{th} input variable.

B) Encoding Mechanism for Rule-base Generation

The second constituent part consists of rule-base represented by a set of consequents selected from a given set. The size of rule-base can be computed as follows:

Size of rule-base (Second constituent part)

$$= \prod_{l=1}^{n} m_l \tag{8}$$

Hence, size of each individual required to encode the zero-order TSK fuzzy model is the sum of equations (7) and (8).

$$\text{Size of one individual (Sugeno model)} = \sum_{j=1}^{n} m_j + \prod_{l=1}^{n} m_l \tag{9}$$

Thus, an individual representing the parameters of the membership functions for input variables and rule base corresponding to a Sugeno model can be represented as shown in Figure 7. In Sugeno model the output variable is represented by fuzzy singleton.

C) Computing Output of each Individual

Each individual represent one fuzzy model whose performance is evaluated in terms of MSE as defined in equation 1.

For a given input training data set, output of each individual can be computed as follows (Eq. 14.5 in [7]):

$$\text{Computed output (training example)} \; \frac{\sum_{k=1}^{R} w_k \left(R_K C\right)}{\sum_{K=1}^{R} W_k} \tag{10}$$

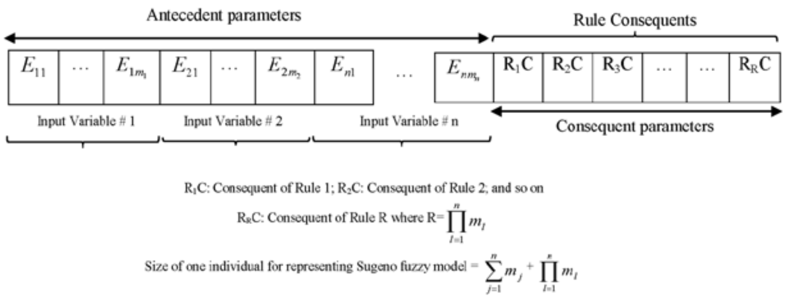


FIGURE 7
Representation of a Sugeno fuzzy model by one individual.

where w_k is the matching degree (i.e., the firing strength) of the k^{th} rule defined by Eq. 14.6 in [7] and R_kC is the consequent of k^{th} rule.

For each training example we compute output using equation (10) and compared it with actual output as given in training example and find the error as follows:

$$\text{Error} = O_A \text{ (as given in training data set)} - O_C \text{ (using Equation 10)} \qquad (11)$$

For complete data set MSE is computed. This will give the MSE of each individual which is used as the fitness function for rating the fuzzy model.

D) Problem Formulation

For a given training data set we compute output of each individual (one fuzzy model) in population. Every training example produces one output which is compared with output as given in training example. For all sets of input MSE is computed.

Our optimization algorithm simultaneously adjusts membership function parameters and consequents in such a way so as to minimize the objective function i.e. MSE. This problem of system identification can now be stated as minimization problem as given below:
Minimize objective function (MSE)

$$MSE = \frac{1}{N}\sum_{k=1}^{N}\left[O_A - O_c\right]^2$$

Subject to the constraint that

```
Pseudo Code for Identification of Optimized Fuzzy Model
  Begin
       Define operating parameters for optimization algorithm;
       Iteration = 0;
       Generate a random set of candidates (initial population);
    While Iteration ≤ Maximum Iteration
       Constraint Population;
       Build fuzzy model corresponding to each candidate solution;
       Evaluate each fuzzy model for its fitness (MSE) using Equation (1);
       Call optimization algorithm (BBO/BB-BC/ACO/ABC/FA) to determine new optimal set of candidates
       (as in Section II);
       Iteration = Iteration + 1;
    End
  Display optimized fuzzy model;
  End
```

FIGURE 8
Pseudo-code for Identification of Optimized Fuzzy Model.

1. $R_kC \in \{\text{specified set of consequents}\};$

2. $$X_{n\,\min} < E_{n1} < E_{n2} < \ldots < E_{nm_n} < X_{n\,\max} \quad (12)$$

where O_A is the actual output, O_C is the computed output, N is number of data points taken for model validation and R_kC represent consequent of k^{th} rule.

The methodology for the identification of optimized fuzzy model through various optimization algorithms (BBO, BB-BC, ACO, ABC and FA) is represented as pseudo–code in Figure 8.

4. APPLICATION EXAMPLES

4.1 Problem 1: Battery Charger

The suggested approaches have been applied for identification of fuzzy model for the rapid Nickel-Cadmium (Ni-Cd) battery charger [69]. The objective of this charger was to charge 2AA Ni-Cd batteries as quickly as possible but without doing any damage to them. Input-output data consists of 561 points is available at *http://www.research.4t.com.* 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). Charging rates are expressed as multiple of rated capacity of the battery, e.g. C/10 charging rate for a battery of C=500 mAh is 50 mA [70]. The input and output variables identified for rapid Ni-Cd battery charger along with their universes of discourse are given in Table 1.

The block diagram for the system to be identified is given in Figure 9. Let us assume that the temperature with the universe of discourse ranging from

Input Variables	Minimum Value	Maximum Value
Temperature (T)[^{0}C]	0	50
Temperature Gradient (dT/dt)[^{0}C/sec]	0	1
OUTPUT VARIABLE Charging Rate (Ct)[A]	0	8C

TABLE 1
Input and Output variables for rapid Ni-Cd battery charger alongwith their universes of discourse.

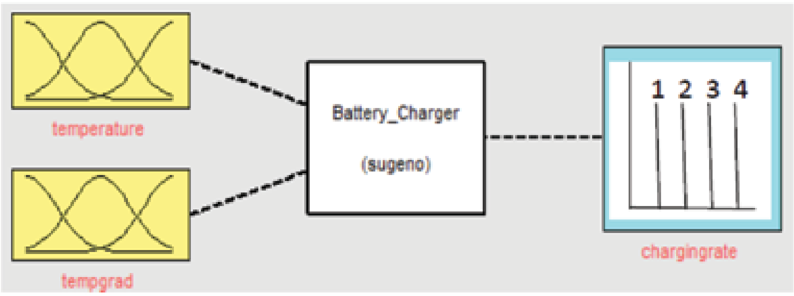


FIGURE 9
Sugeno type Fuzzy Model for Battery Charger.

0-50 degree centigrade has been partitioned into 3 fuzzy sets namely temperature low, med (medium), and temperature high.

The temperature gradient is partitioned into 2 fuzzy sets (membership functions) namely low and high. Initially we set the parameters of membership functions of input variables using modified FCM clustering technique [44]. Once fuzzification of the inputs is carried out, we get the 6 combinations of input membership functions (3*2 = 6) representing 6 antecedents of rules. These 6 rules form the rule base for the system under identification. The rule base is yet incomplete as for each rule the consequent need to be found out. From the datasets of Battery Charger (Appendix 1), we find that there are only 5 consequents that form the set of consequents from where we have to choose one particular element as the consequent for a particular rule. The specified set of consequents in this case are C_1= Trickle = 0.1 Amp, C_2=Low = 1 Amp, C_3= Med = 2 Amp, C_4= High= 3 Amp and, C_5= Ultrafast = 4 Amp. We have to choose parameters of antecedent and consequents in such a way so as to fulfill condition given by expression (12).

In this problem the individual size to encode a Sugeno model may be calculated from equation (9) as follows:

$$\text{Individual Size (Sugeno model)} = \sum_{j=1}^{2} m_j + \prod_{l=1}^{2} m$$
$$= (3+2) + (3*2) = 5 + 6 = 11$$

The 11-dimension individual representing a Sugeno fuzzy model for the Ni-Cd rapid battery charger is shown in Figure 10.

A Parameter Settings for Algorithms

To obtain suitable parameters of algorithms which affect the performance in terms of solution quality and processing time, a large number of experiments were conducted with different parameter settings. We found that a population size of 50 and maximum number of generations of 5000 was adequate for all algorithms except ACO. For ACO the number of ants was kept to 30 and maximum number of iterations was 10,000. The other specific parameters of algorithms which came through experiments are listed in Tables 2-5. All observations are made on Intel Core i5-450M @ 2.4 GHz HP ENVY[14] laptop with 4GB of RAM.

B Simulation Results and Discussions

With the optimal parameters obtained, the large numbers of sets of trials were conducted with all approaches as given in Section II. Each set consisted of 10 trials. In Table 6 we list the performance of one best set of each approach in terms of MSE and computing time. This performance is for the training data

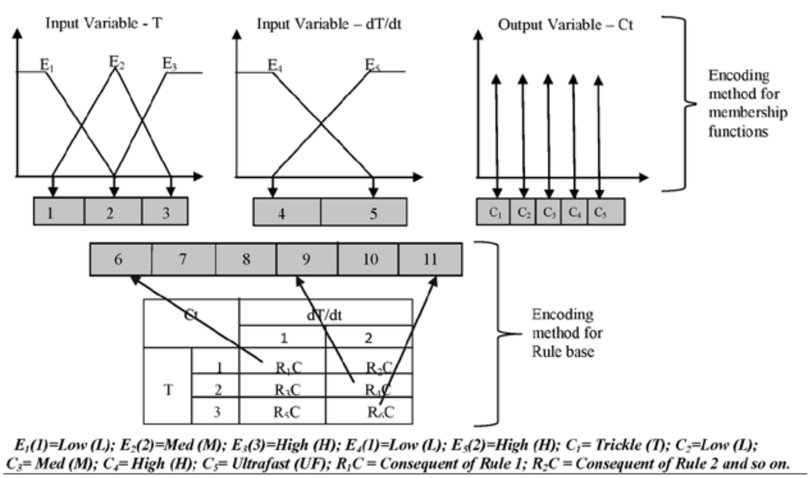


FIGURE 10
Encoding of Sugeno Model for rapid Ni-Cd Battery Charger.

Parameter	VALUE
Mutation probability	0.1
Habitat modification probability (P_{modify})	1
Maximum immigration (I) rates for each island	1
Maximum emigration (E) rates for each island	1
Lower bound for immigration probability per SIV (λ_{Lower})	0
Upper bound for immigration probability per SIV (λ_{Upper})	1
Step size (dt) for numerical integration of probabilities	1

TABLE 2
BBO algorithm parameters for fuzzy model identification of Battery Charger.

Parameter	Value
limit (A trail number after which solution cannot be improved)	10

TABLE 3
ABC algorithm parameters for fuzzy model identification of Battery Charger.

Parameter	Value
α (a constant)	2
ρ (evaporation constant)	0.018
$\Delta\tau^k$(Pheromone deposit factor)	0.0072

TABLE 4
ACO algorithm parameters for fuzzy model identification of Battery Charger.

Parameter	Value
α	0.5
β	0.2
γ	1

TABLE 5
FA algorithm parameters for fuzzy model identification of Battery Charger.

set. For training purposes we selected 14 training examples out of 561 data points. This training data set was chosen as peaks of membership functions. Each approach was executed for population size of 50 and maximum iterations of 5000 except ACO based approach which was stable for 10000 iterations and the minimum, average and maximum MSE's were recorded as given in Table 7.

	Model Identification Approach Number of iterations = 5000; Population size = 50 Number of iterations for ACO = 10000; Number of ants = 30									
	BBO		BB-BC		ACO		ABC		FA	
Trials	MSE	CPU Time (in seconds)	MSE	CPU Time (in seconds)	MSE	CPU Time (in seconds)	MSE	CPU Time (in seconds)	MSE	CPU Time (in seconds)
1	0.004	204	0.004	114	0.005	241	0.005	266	0.005	132
2	0.010	205	0.008	114	0.006	257	0.005	266	0.006	142
3	0.007	207	0.006	114	0.007	254	0.005	266	0.005	138
4	0.009	211	0.006	114	0.009	254	0.005	266	0.011	132
5	0.007	206	0.004	114	0.020	262	0.005	266	0.005	135
6	0.005	203	0.007	113	0.009	259	0.005	266	0.005	137
7	0.009	206	0.009	114	0.006	266	0.005	266	0.005	141
8	0.002	205	0.005	113	0.005	253	0.005	266	0.006	140
9	0.003	203	0.006	114	0.006	252	0.005	266	0.005	139
10	0.005	207	0.006	112	0.007	249	0.005	266	0.005	131

TABLE 6
Performance of five new model identification approaches for battery charger example in a set of 10 trials

Performance Measures	BBO	BB-BC	ACO	ABC	FA
Minimum MSE	0.002	0.005	0.005	0.005	0.005
Average MSE	0.006	0.006	0.008	0.005	0.006
Maximum MSE	0.010	0.009	0.020	0.005	0.011

TABLE 7
Comparison of five new model identification approaches for battery charger example in terms of MSE.

The best of the average MSE was observed to be 0.005 by ABC algorithm based approach. The best of the minimum MSE given by BBO approach was observed to be 0.002. For 5000 iterations, BB-BC took the minimum time of 112 sec. whereas maximum time recorded by ABC was 266 sec. We further observe from Figure 11 that ABC based approach stabilizes the evolved model to MSE goal in lesser number of iterations as compared to other approaches.

Further for training data as observed from Figure 11 one can observe that ABC approach was fastest to stabilize followed by FA approach,

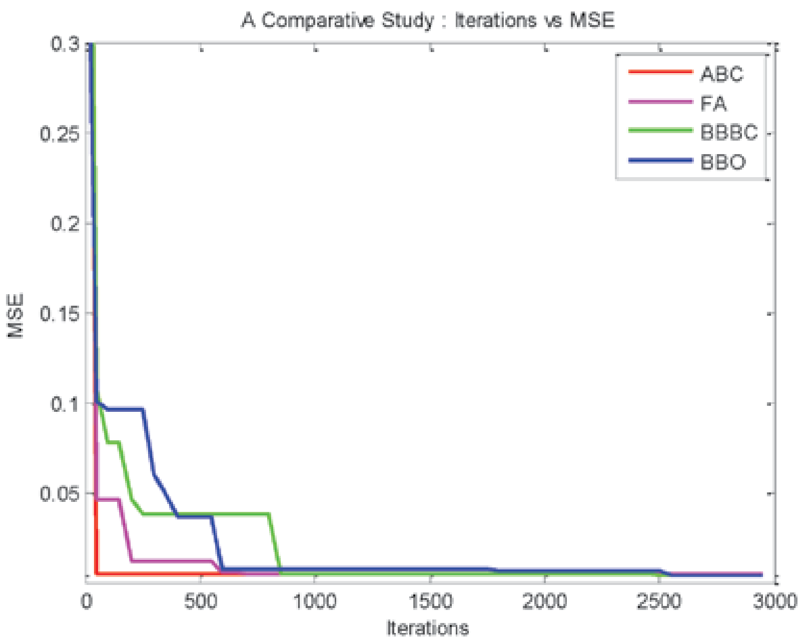


FIGURE 11
Iteration vs MSE : Comparison of proposed model identification approaches for control problem.

BB-BC approach and last to stabilize was BBO approach. In general ABC approach stabilize the evolved model in less than 100 iterations, FA approach in around 500 iterations, BB-BC approach in less than 700 iterations and BBO approach in and around 3000 iterations whereas ACO approach (as seen in Figure 12) stabilize the evolved model in around 10000 iterations.

We further observed from Table 8 that for MSE goal of 0.006, ABC algorithm based approach identified the desired model in average time of 3 seconds, FA approach in 13 seconds, BB-BC approach in 29 seconds, BBO approach in 68 seconds and ACO approach generated the results in 255 seconds. Thus clearly establishing the superiority of ABC algorithm based approach over other approaches for the control problem under identification (Figure 13). The performance for the evolved system was compared with that available in the literature and the same is given in Table 9. It can be observed that ABC approach followed BBO approach, BB-BC approach, FA approach and ACO approach in that order give excellent model performance as compared to other approaches. The evolved models with all the proposed approaches are given below in Table 10. Further, with the use of rule reduction algorithm [12], [13] we can have a compact rule-base of only 3 rules as shown in Table 10.

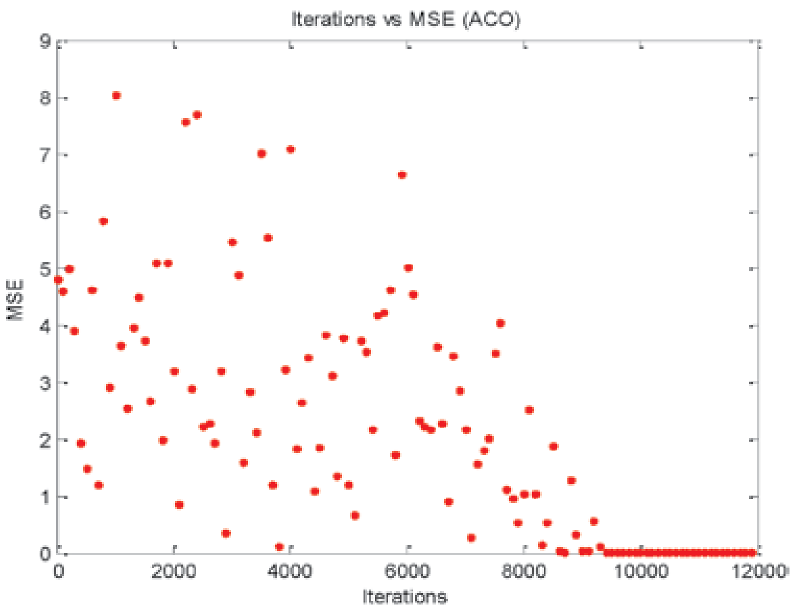


FIGURE 12
Graph of Iterations vs MSE for ACO Based Approach.

	Model Identification Approach Time taken (in sec.); Fixed MSE Goal = 0.006				
Trials	ABC	FA	BB-BC	BBO	ACO
1	4	11	24	18	241
2	3	10	17	11	257
3	5	15	58	164	266
4	2	12	17	154	253
5	3	14	6	21	252
6	2	11	15	42	263
7	2	13	21	38	250
8	3	14	67	17	254
9	3	15	21	29	262
10	3	14	49	188	258
	Avg. Time = 3	Avg.Time = 13	Avg. Time= 29	Avg. Time = 68	Avg. Time = 255

TABLE 8
Comparison of proposed approaches in terms of processing time.

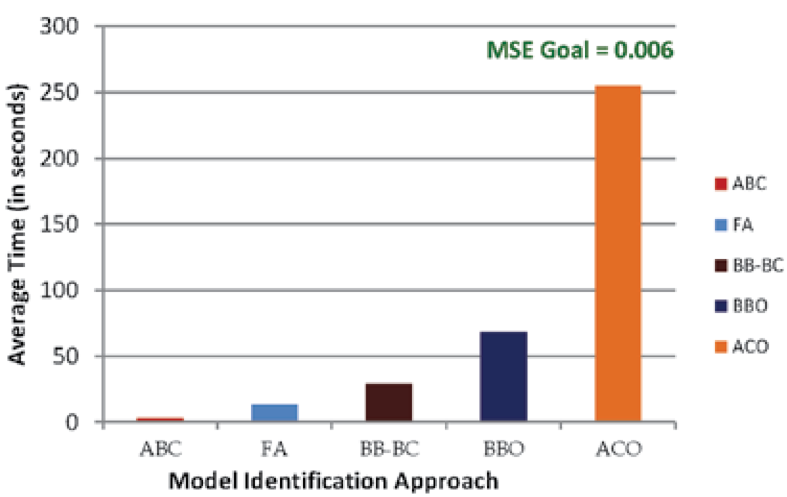


FIGURE 13
Comparison of five model identification approaches in terms of average time of 10 trials for fixed MSE goal in control problem.

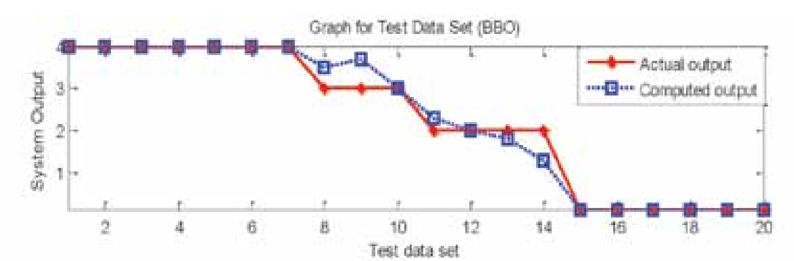
Model Identification Approach	MSE
Hybrid Learning [71]	0.132
Genetic Algorithm [72]	0.130
Particle Swarm Optimization [20]	0.112
ACO (Proposed)	0.008
BBO (Proposed)	0.006
BB-BC (Proposed)	0.006
FA (Proposed)	0.006
ABC (Proposed)	0.005

TABLE 9
Comparison of proposed model identification approaches with existing approaches for battery charger problem in terms of MSE.

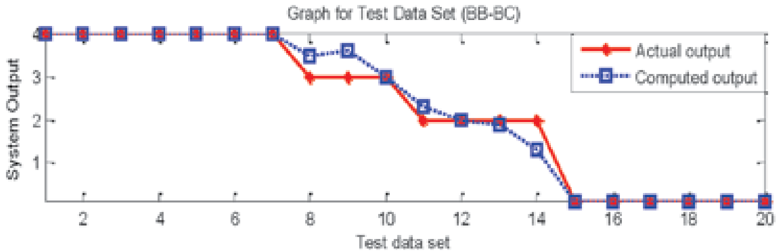
4.2 Evolved Model performance for Test Data set

For control problem under discussion out of 561 points, a set of 14 data points were chosen as training dataset. For test data set we selected 20 points excluding those in the training data set (Appendix-1). Figure 13 presents the performance of evolved fuzzy model on test data set. The actual output is the desired output i.e. given in the data set and the computed output is the output computed by the proposed approaches i.e. BBO, BB-BC, ACO, FA and ABC respectively. It was observed that for test data set MSE stayed within 0.06 to 0.08 for all the observations.

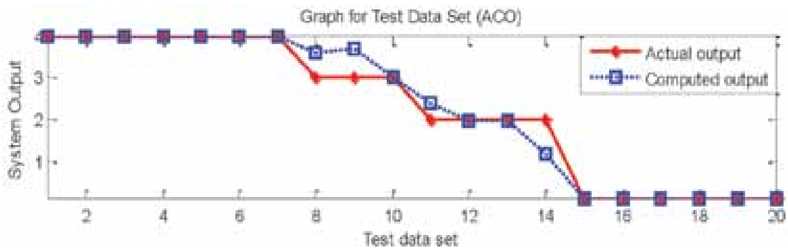
Problem 2: Iris data set classification problem
The Iris data set [73] contains 150 patterns with four features that belong to three classes (Iris Setosa, Iris Versicolour and Iris Virginica) and each class contains 50 patterns. The four features are the sepal length, the sepal width,



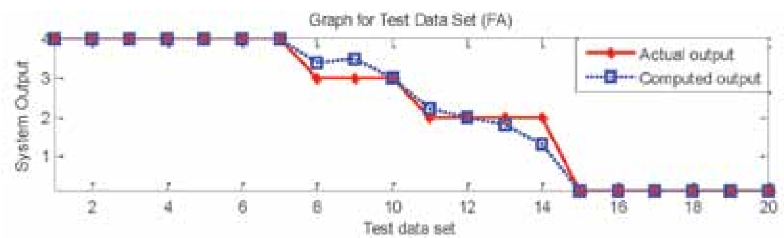
(a)



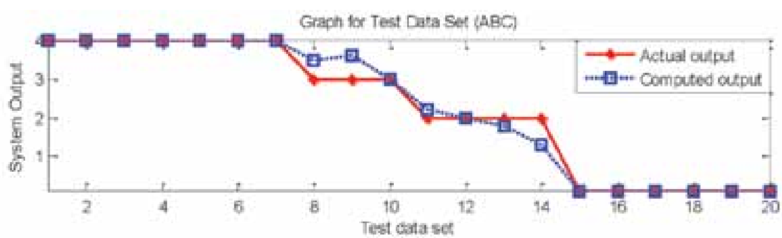
(b)



(c)



(d)



(e)

FIGURE 13
Performance of evolved system through proposed approaches on test data set of Ni-Cd battery charger.
a) BBO b) BB-BC c) ACO d) FA e) ABC

Model Identification	Parameters of Membership functions					Parameters of Consequent part						Reduced Rule-Base [12], [13]		
Approach	E_1	E_2	E_3	E_4	E_5	R_1C	R_2C	R_3C	R_4C	R_5C	R_6C	T	dT/dt	Ct
BBO	37	40	44	0.3	0.7	UF	UF	H	H	T	T	L	X	UF
BB-BC	37	40	44	0.2	0.8	UF	UF	H	H	T	T	M	X	H
ACO	37	40	44	0.3	0.8	UF	UF	H	H	T	T	H	X	T
ABC	37	40	44	0.3	0.7	UF	UF	H	H	T	T			
FA	36.3	40.3	43.8	0.35	0.71	UF	UF	H	H	T	T			

UF=Ultrafast; H=High; T=Trickle; X =Don't care

TABLE 10
Evolved fuzzy model through proposed model identification approaches for control problem.

Algorithm	Population size	Maximum Iterations
BBO	500	500
BB-BC	30	10000
ABC	500	1000
FA	30	5000

TABLE 11
Parameter settings of algorithms for Iris data classification problem.

the petal length and the petal width. All the four features are specified in centimeters. We choose only 20 percent data (30 out of 150 patterns) as training examples to form the training data set as given in Appendix 1. The system was trained using training data set and the system performance was evaluated using entire data set of 150 patterns. After experimentation, the best parameters obtained for proposed model identification approaches are given in Table 11.

The system was evolved 10 times for each approach and each system being evolved was tested for the entire data set. The classification rate and number of misclassifications for each model being evolved are given in Table 12. Three membership functions were associated with each input variable. It was observed that without expert's opinion ABC approach evolved system with maximum classification rate of 99.3% in 5 trials which gave 1 misclassification out of 150 patterns, BBO approach gave maximum rate of 98.7% with 2 misclassifications, FA approach gave maximum rate of 98.7% with 2

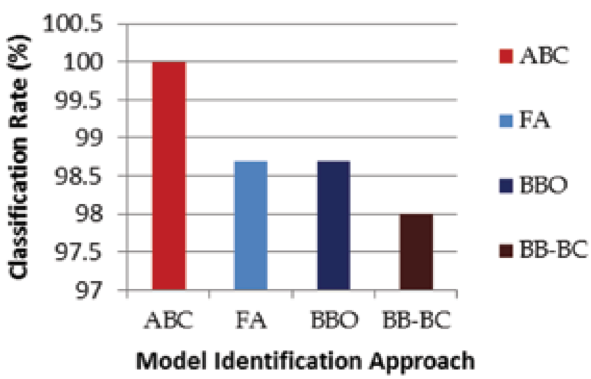


FIGURE 14
Comparison of proposed model identification approaches in terms of classification rates

Model Identification Approach	Classification rate on training patterns (%)	Classification rate on test patterns (%) / No. of Misclassifications									
		Identified Model No.									
		Without Expert's intervention					With Expert's intervention				
		1	2	3	4	5	6	7	8	9	10
ABC	100	98.7/2	99.3/1	99.3/1	98.7/2	98.7/2	100/0	100/0	100/0	100/0	100/0
BBO	100	98.7/2	97.3/4	98.7/2	98/3	98.7/2	98.7/2	98.7/2	98.7/2	98.7/2	98.7/2
BB-BC	100	97.3/4	97.3/4	96.7/5	96.7/5	97.3/4	98/3	98/3	98/3	98/3	98/3
FA	100	97.3/4	98/3	98/3	97.3/4	98.7/2	98.7/2	98.7/2	98.7/2	98.7/2	98.7/2

TABLE 12
Comparison of proposed model identification approaches in terms of classification rate for training and test data set of Iris classification problem.

Method	Classification rate on total data set (%)
Nozaki K. et al. [74]	93.3
C. Chong Chen [75]	96.8
Wang et al. [76]	97.5
Wu et al. [77]	96.2
Shi et al. [17]	98.0
Abe et al. [78]	98.7
Setnes et al. [79]	98.9
Ruso [80]	100
Ishibuchi et al. [81]	98.0
X. Chang et al. [82]	99.3
Abonyi et al. [83]	98.0
K. Nozaki et al. [84]	90.7
Tanaka et al. [85]	94.3
ABC (Proposed)	100
BBO (Proposed)	98.7
FA (Proposed)	98.7
BB-BC (Proposed)	98.0

TABLE 13
Comparison of the proposed model identification approaches for Iris data classification problem with the existing methods.

misclassifications and BB-BC approach gave its best with 97.3% classification rate which gave 4 errors.

Next, we adjusted the membership functions shapes and rule base manually taking expert's opinion into account (after observing misclassifications). By little effort, the best system we evolved gave classification rates of 100%, 98.7%, 98.7% and 98% for ABC, BBO, FA and BB-BC based approaches respectively as shown in Figure 14. ACO based approach was not considered for comparison due to complexity involved in graphical representation in computer presentation of problem. The results are tabulated and compared with results obtained by most other classification methods found in the literature as shown in Table 13.

5. CONCLUSIONS

As the complexity of problem grows soft computing approaches assume significance. This paper presents a frame work to evolve a complete fuzzy model from available data using five recent nature inspired approaches namely BBO, BB-BC, ACO, ABC and FA. The proposed approaches successfully generated optimized fuzzy models from training data. The training data set was a small percentage of total available data set. The proposed approaches were successfully validated on one control and one classification problem. For control problem, ABC based approach appears to be more efficient in terms of computational time and MSE as compare to other approaches. Simulation results show that for an MSE goal of 0.006, ABC approach generated a fuzzy model within an average time of 3 seconds, FA in 13 sec., BB-BC in 29 sec., BBO in 68 sec. and ACO in 255 sec. For control problem, the results clearly indicate that as for as training data is concerned, ABC approach outperforms the other four approaches on computational time and MSE basis. The performance of proposed algorithms was validated using test data set. For test data set all of these algorithms performed within an MSE of 6 to 8 percent.

For classification problem, we used 20% of the available data for training purpose. For validation purpose entire data set was used as test data set. The proposed approaches generated fuzzy classification systems with high classification rates. ABC based approach obtained fuzzy classifier with average classification rate of 98.94%, BBO approach with 98.28%, FA approach with 97.86% and BB-BC approach with 97.06%. With expert's opinion incorporated, this classification rate was improved to 100% (0 misclassification) for ABC approach, 98.7% (2 misclassifications) for BBO approach, 98.7% (2 misclassifications) for FA approach and 98% (3 misclassifications) for BB-BC approach. Thus we conclude that the proposed approaches perform better than or equivalent to most of the existing approaches available in the literature.

REFERENCES

[1] Zadeh, L. A. 1973. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man and Cybernetics. 3(1) : 28–44.

[2] Berkan, R. C. and Trubatch, S. L. 2000. Fuzzy Systems Design Principles. 1st Edition, Wiley-IEEE Press, New York, USA, p 496.

[3] Yager, R. R. and Filev, D. P. 1994. Essentials of Fuzzy Modeling and Control. John Wiley & Sons, New York, p 388.

[4] Rouse, W. B., Hammer, J. M. and Lewis, C. M. 1989. On capturing human skills and knowledge: Algorithmic approaches to model identification. IEEE Transactions on Systems, Man and Cybernetics. 19(3) : 558-573.

[5] Takagi, T. and Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics. 15(1) : 116-132.

[6] Plamen, A. et al. 2002. Identification of evolving fuzzy rule-based models. IEEE Transactions on Fuzzy Systems. 10(5) : 667-677.

[7] Yen, J. and Langari, R. 1999. Fuzzy Logic Intelligence, Control and Information. Prentice Hall, New Jersey, p 548.

[8] J.Yen and L.Wang, "An SVD-based fuzzy model reduction strategy," Proceedings of the Fifth IEEE International conference on Fuzzy Systems, New Orleans, LA, pp. 835-841, 1996.

[9] J.Yen and L.Wang, "Application of statistical information criteria for optimal fuzzy model construction," IEEE Transactions on Fuzzy Systems, Vol. 6, No.3, pp. 362-372, 1998.

[10] J.Yen and L.Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," IEEE Transactions on Systems, Man and Cybernetics, Vol.29, 1999.

[11] Y.Yam, P.Baranyi and C.T. Yang, "Reduction of Fuzzy Rule Base via Singular Value Decomposition," IEEE Transactions on Fuzzy Systems, Vol.7, No.2, pp.120-132, 1999.

[12] Arun Khosla, Shakti Kumar, K.K. Aggarwal, "Hardware Reduction for Fuzzy based systems via Rule Reduction Through Exhaustive Search Technique", National Seminar on emerging convergent technologies and systems (SECTAS-2002), Dayalbag Educational Institute, Agra, India, March 1-2, 2002, pp 381-385.

[13] Arun Khosla, Shakti Kumar, K.K. Aggarwal, "Optimizing Fuzzy Rule Base Through State Reduction", National Seminar on emerging convergent technologies and systems (SECTAS-2002), Dayalbag Educational Institute, Agra, India, March 1-2, 2002, pp. 415-419.

[14] Ken Nozaki, Hisao Ishibuchi and H.Tanaka, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," Fuzzy Sets and Systems, Vol.86, pp. 251-270, 1997.

[15] Li-Xin Wang and Jerry M. Mendel, "Generating fuzzy rules by Learning from Examples," IEEE Transactions on Systems, Man and Cybernetics, Vol.22, No.6, pp. 1414-1427, 1992.

[16] A.Homaifar and E.Mc.Cormick, "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," IEEE Transactions on Fuzzy Systems, Vol.3, No.2, pp. 129-139, 1995.

[17] Y.Shi, R. Eberhart and Y.Chen, "Implementation of Evolutionary Fuzzy Systems," IEEE Transactions on Fuzzy Systems, Vol.7, No.2, pp. 109-119, 1999.

[18] H.S. Hwang, "Automatic design of fuzzy rule base for modeling and control using evolutionary programming," IEE Proceedings- Control Theory Applications, Vol. 146, No. 1, pp. 9-16, 1999.

[19] S.J. Kang, C.H. Woo, H.S. Hwang and K.B. Woo, "Evolutionary Design of Fuzzy Rule Base for Nonlinear System Modeling and Control," IEEE Transactions on Fuzzy Systems, Vol. 8, No.1, pp. 37-45, 2000.

[20] Arun Khosla, Shakti Kumar, K.K.Aggarwal, Jagatpreet Singh, "Particle Swarms for fuzzy models identification," Studies in Computational Intelligence (SCI) 26, Springer-Verlag Berlin Heidelberg 2006, pp. 149-173.

[21] Marco Dorigo and Thomas Stutzle, Ant Colony Optimization, Eastern Economy Edition, PHI, 2005.

[22] Marco Dorigo, Vittorio Maniezzo and Alberto Colorni, "The Ant System: Optimization by a colony of cooperating agents" IEEE Transactions on Systems, Man, and Cybernetics–Part B, Vol.26, No.1, pp.1-13, 1996.

[23] M. Dorigo and L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transaction on Evolutionary Computation, 1(1) (1997), pp. 53-66, 1997.

[24] J. Casillas, O. Cordon and F. Herrera, "Learning fuzzy rules using ant colony optimization algorithms," Proc. 2nd Int. Workshop Ant Algorithms, 2000, pp. 13-21.

[25] R.S. Parpinelli, H.S. Lopes and A.A. Freitas, "An ant colony algorithm for classification rule discovery," in Data Mining: A Heuristic Approach, pp. 190-208, H.A. Abbass, R.A. Sarkar. Idea Group Publishing, 2002.

[26] Bo Liu, H.A. Abbass and R.I McKay,"Classification Rule Discovery with Ant Colony Optimization," IEEE Intelligent Informatics Bulletin (2004) pp. 31-35.

[27] M. Galea and Q. Shen, "Fuzzy rules from ant-inspired computation," Proc. IEEE Int'l Conf. Fuzzy Systems, pp. 1691-1696, 2004.

[28] P. Carmona and J. L. Castro, "Using ant colony optimization for learning maximal structure fuzzy rules," Proc. IEEE Int. Conf. Fuzzy Systems, pp. 999-999, 2005.

[29] H.Nobahari and Seid H. Pourtakdoust, "Optimization of fuzzy rule bases using continuous Ant Colony System," Proceeding of the first International Conference on Modeling, Simulation and Applied Optimization, Sharjah, U.A.E., Feb. 2005.

[30] R.Martinez, O. Castillo and J.Soria, "Parameter tuning of membership functions of a Type-1 and Type-2 fuzzy logic controller for an autonomous wheeled mobile robot using Ant Colony Optimization," Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, Oct. 2009.

[31] C. Juang and Po-Han Chang, "Designing fuzzy-rule-based systems using continuous Ant-Colony Optimization," IEEE Transactions on Fuzzy Systems, Vol. 18, No.1, Feb. 2010.

[32] S. Kumar, P. Kaur and A.P. Singh, "Soft Computing Approaches to Fuzzy System Identification: A Survey," 3rd International Conference on Intelligent Systems and Networks (IISN-2009), Feb 14-16, 2009, ISTK, Jagadhri, Haryana, India, pp.402-411.

[33] S. Kumar, P. Kaur and A.P. Singh, "Fuzzy Rulebase Generation from numerical data using Biogeography Based Optimization Approach," Journal of Institution of Engineers IE (I), Vol. 90, pp.8-13, July 2009.

[34] P. Kaur, S. Kumar and A.P. Singh, "Optimization of Membership Functions Based on Ant Colony Algorithm," International Journal of Computer Science and Information Security (IJCSIS), Vol. 10, No. 4, pp. 38-45, April 2012.

[35] S. Kumar, P. Kaur and A.P. Singh, "Fuzzy Rulebase Generation from numerical data using Big Bang-Big Crunch Optimization," Journal of Institution of Engineers IE (I), Vol. 91, pp. 18-25, January 2011.

[36] S. Kumar, P. Kaur and A.P. Singh, "Fuzzy Rule Base Generation: An Artificial Bee Colony (ABC) Approach," 5th International Multi Conference on Intelligent Systems, Sustainable, New and Renewable Energy Technology and Nanotechnology (IISN-2011), Feb 18-20, 2011, ISTK, Jagadhri, Haryana, India, pp. 262-270.

[37] A.A.A. Esmin, A.R. Aoki, G. Lambert-Torres, "Particle swarm optimization for fuzzy membership functions optimization," IEEE Int'l Conf. on Syst., Man and Cybern., pp. 6, vol. 3, Oct. 2002.

[38] Seema Chopra, Ranjit Mitra and Vijay Kumar, "Reduction of Fuzzy Rules and Membership Functions and its application to Fuzzy PI and PD type controllers," Int'l journal of Control, Automation, and Systems, vol.4, no.4, pp. 438-447, Aug. 2006.

[39] Hyong-Euk Lee, Kwang-Hyun Park and Z.Z.Bien, "Iterative Fuzzy Clustering Algorithm with Supervision to construct probabilistic Fuzzy Rule Base from numerical data," IEEE Transactions on Fuzzy Systems, Vol. 16, No.1, pp.263-277, Feb. 2008.

[40] P. Carmona, J.L. Castro and J. M. Zurita, "FRIwE: Fuzzy rule identification with exceptions," IEEE Transactions on Fuzzy Systems, Vol. 12, No.1, pp.140-151, Feb. 2004.

[41] B. Apolloni, A. Brega, D.Malchiodi, G. Palmas and A. M. Zanaboni, "Learning rule representations from data," IEEE Transactions on Systems, Man and Cybernetics- Part A, Vol. 36, No. 5, pp. 1010-1028, Sep. 2006.

[42] Xiao-Jun Zeng and M.G. Singh, "Knowledge bounded least squares method for the identification of fuzzy systems," IEEE Transactions on Systems, Man and Cybernetics- Part C, Vol. 33, No. 1, pp. 24-32, Feb. 2003.

[43] S. B. Morphet, L.B. Morphet, "Combining single input/single output fuzzy decision trees," IEEE Int'l Conf. on Fuzzy Syatems, Vancouver, Canada, pp. 1792-1798, July 2006.

[44] Khosla, A., Kumar, S. and Aggarwal, K. K. 2003. Identification of fuzzy controller for rapid nickel cadmium batteries charger through fuzzy c–means clustering algorithm. Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society, Chicago, Illinois, USA, July 24–26, pp. 536–539.

[45] T. Pal and Nikhil R. Pal, "SOGARG: A self organized genetic algorithm based rule generation scheme for fuzzy controllers," IEEE Transactions on Evolutionary Computation, vol. 7, no. 4, pp. 397-415, Aug. 2003.

[46] Eghbal G. Mansoori, M.J. Zolghadri and S.D. Katebi, "SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data," IEEE Transactions on Fuzzy Systems, Vol.16, No.4, pp. 1061-1071, Aug. 2008.

[47] Z. Ning, Y S. Ong, K.W. Wong and K.T. Seow, "Parameter identification using Memetic algorithms for fuzzy systems," Proc. of the fourth Int'l conf. on intelligent technologies (Intech'03), pp 833-839, 2003.

[48] L.A. Zadeh, “Fuzzy logic, neural networks, and soft computing,” Commun. ACM, vol. 37, pp. 77-84, 1994.
[49] Simon, D. 2008. Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation. 12(6) : 702–713.
[50] Simon, D., Ergezer, M., Du, D. and Rarick, R. 2011. Markov models for biogeography–based optimization. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics. 41(1) : 299–306.
[51] MacArthur, R. and Wilson, E. 1967. The theory of island biogeography. Princeton, NJ: Princeton University Press, p 203.
[52] Simon, D., Ergezer, M. and Du, D. 2009. Population distributions in biogeography-based optimization algorithms with elitism. Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA, pp. 991–996.
[53] Ma, H. 2010. An analysis of the equilibrium of migration models for biogeography–based optimization. Information Sciences. 180(18) : 3444–3464.
[54] Bhattacharya, A. and Chattopadhyay, P. K. 2010. Hybrid differential evolution with biogeography–based optimization for solution of economic load dispatch. IEEE Transactions on Power Systems. 25(4) : 1955–1964.
[55] Ma, H. and Simon, D. 2011. Blended biogeography–based optimization for constrained optimization. Engineering Applications of Artificial Intelligence. 24(3) : 517–525.
[56] Ma, H., Ni, S. and Sun, M. 2009. Equilibrium species counts and migration model tradeoffs for biogeography based optimization. IEEE Conference on Decision and Control and 28th Chinese Control Conference Shanghai, P. R. China, pp. 3306–3310.
[57] Roy, P. K., Ghoshal, S. P. and Thakur, S. S. 2009. Biogeography based optimization to solve economic load dispatch considering valve point effects. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, 21-22 Dec., pp. 1213–1218.
[58] Simon, D., Rarick, R., Ergezer, M. and Du, D. 2011. Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. Information Sciences. 181(7) : 1224–1248.
[59] Erol, O. K. and Eksin, I. 2006. A new optimization method: Big Bang-Big Crunch. Advances in Engineering Software. 37(2) : 106-111.
[60] Kumbasar, T., Yesil, E., Eksin, I. and Guzelkaya, M. 2008. Inverse fuzzy model control with online adaptation via Big Bang-Big Crunch optimization. ISCCSP 2008, Malta, 12-14 March, pp. 697-702.
[61] Kripka, M. and Kripka, R. M. L. 2008. Big Crunch Optimization method. International Conference on Engineering Optimization, Brazil, 1-5 June.
[62] Karaboga, D. and Akay, B. 2009. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 214 : 108-132.
[63] Karaboga, D. 2005. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
[64] Karaboga, D. and Basturk, B. 2007. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39(3) : 459–471.
[65] Karaboga, D. and Basturk, B. 2008. On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing. 8(1) : 687–697.
[66] Yang, X. S. 2009. Firefly algorithms for multimodal optimization in: Stochastic Algorithms: Foundations and Applications (Eds O. Watanabe and T. eugmann), SAGA 2009, Lecture Notes in Computer Science, 5792, Springer-Verlag, Berlin, pp. 169-178.
[67] Yang, X. S. 2010. Firefly algorithm, stochastic test functions and design optimization. Int. J. Bio-Inspired Computation. 2(2) : 78-84.
[68] Yang, X. S. 2010. Firefly algorithm, levy flights and global optimization. Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, Petridis), Springer London, pp. 209-218.

[69] Khosla, A., Kumar, S. and Aggarwal, K. K. 2002. Design and development of RFC-10: A fuzzy logic based rapid battery charger for Ni-Cd batteries. HiPC (High Performance Computing). Workshop on Soft Computing, Bangalore, pp. 9-14.

[70] Linden D. 1995. Handbook of Batteries. Mc. Graw Hill Inc, p 1216.

[71] Khosla, A., Kumar, S. and Aggarwal, K. K. 2003. Fuzzy controller for rapid Nickel-Cadmium batteries charger through adaptive neuro-fuzzy inference system (ANFIS) architecture. Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society, Chicago, Illinois, USA, July 24–26, pp. 540–544.

[72] Kumar, S. 2005. Introduction to fuzzy logic based systems. Proceeding of one week workshop on applied soft computing SOCO-2005, Haryana Engg. College, Jagadhri, India, July 25-30, pp. 43-71.

[73] Blake, C., Keogh, E. and Merz, C. J. 1998. UCI Repository of Machine learning database, Univ. California, Irvine. http://www.ics.uci.edu/~mlearn/.

[74] Nozaki K., Ishibuchi H. and Tanaka H. 1996. Adaptive fuzzy rule based classification systems. IEEE Trans. on Fuzzy Systems. 4(3) : 238-250.

[75] Chen, C. C. 2006. Design of PSO-based fuzzy classification systems. Tamkang Journal of Science and Engineering. 9(1) : 63-70.

[76] Wang, J. S. and Lee, G. C. S. 2002. Self-adaptive neuro-fuzzy inference system for classification application. IEEE Trans. Fuzzy System, 10(6) : 790-802.

[77] Wu, T. P. and Chen, S. M. 1999. A new method for constructing membership functions and fuzzy rules from training examples. IEEE Trans., Syst., Man, Cybern. B. 29(1) : 25-40.

[78] Abe, S. and Thawonmas, R. 1997. A fuzzy classifier with ellipsoidal regions. IEEE Trans. Fuzzy System. 5(3) : 358-368.

[79] Setnes, M. and Roubos, H. 2000. GA-fuzzy modeling and classification: complexity and performance. IEEE Trans. Fuzzy System. 8(5) : 509-522.

[80] Russo, M. 2000. Genetic fuzzy learning. IEEE Transactions on Evolutionary Computation, 4(3) : 259-273.

[81] Ishibuchi, H. and Nakashima, T. and Murata, T. 2001. Three-objective genetic-based machine learning for linguistic rule extraction. Inf. Sciences. 136 : 109-133.

[82] Chang, X. and Lilly, J. H. 2004. Evolutionary design of a fuzzy classifier from data. IEEE Trans., on Systems, Man, and Cybernetics –Part B: Cybernetics. 34(4) : 1894-1906.

[83] Abonyi, J., Roubos, J. A. and Szeifert, F. 2003. Data-driven generation of compact, accurate, and linguistically sound fuzzy classifiers based on a decision-tree initialization. Int. J. Approx. Reasoning. 32(1) : 1-21.

[84] Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H. 1995. Selecting fuzzy if-then rules for classification problems using genetic algorithms. IEEE Trans. on Fuzzy Systems. 3(3) : 260-270.

[85] Ishibuchi, H., Nozaki, K. and Tanaka, H. 1992. Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets and Systems. 52(1) : 21–32.

APPENDIX – 1

Data Point	Input 1	Input 2	Actual Output
1	0	0.0	4.0
2	30	1.0	4.0
3	37	0.2	4.0
4	40	0.0	3.0
5	40	1.0	2.0
6	41	0.5	2.0
7	42	1.0	1.0
8	43	0.5	1.0
9	43	1.0	0.5
10	44	0.0	0.1
11	44	0.4	0.1
12	45	0.1	0.1
13	45	0.5	0.1
14	50	1.0	0.1

Training Data Set for Battery Charger

Data Point	Input 1	Input 2	Actual Output
1	0	0.1	4.0
2	5	1.0	4.0
3	10	0.5	4.0
4	20	0.2	4.0
5	30	0.9	4.0
6	37	0.8	4.0
7	37	1.0	4.0
8	38	0.4	3.0
9	38	1.0	3.0
10	40	0.8	3.0
11	41	0.7	2.0
12	42	0.5	2.0
13	42	0.6	2.0
14	43	0.4	2.0
15	44	0.2	0.1
16	44	1.0	0.1
17	45	0.8	0.1
18	48	0.1	0.1
19	50	0.1	0.1
20	50	1.0	0.1

Test Data Set for Battery Charger

Data Point	Sepal length (in cm.)	Sepal width (in cm.)	Petal length (in cm.)	Petal width (in cm.)	Class
1	49	30	14	2	1
2	50	36	14	2	1
3	49	31	15	1	1
4	54	37	15	2	1
5	54	34	15	4	1
6	44	30	13	2	1
7	50	35	16	6	1
8	50	33	14	2	1
9	64	32	45	15	2
10	55	23	40	13	2
11	59	30	42	15	2
12	59	32	48	18	2
13	63	25	49	15	2
14	64	29	43	13	2
15	67	30	50	17	2
16	54	30	45	15	2
17	62	29	43	13	2
18	51	25	30	11	2
19	71	30	59	21	3
20	65	30	58	22	3
21	49	25	45	17	3
22	67	25	58	18	3
23	64	27	53	19	3
24	57	25	50	20	3
25	65	30	55	18	3
26	69	32	57	23	3
27	67	33	57	21	3
28	64	28	56	21	3
29	64	28	56	22	3
30	59	30	51	18	3

Training Data Set for Iris Plants

Input 1 – Temperature
Input 2 – Temperature Gradient
Actual Output - Charging Current
Class 1 – Iris Setosa
Class 2 – Iris Versicolour
Class 3 – Iris Virginica