

Fuzzy Model Identification: A Firefly Optimization Approach

Shakti Kumar
Computational Intelligence
Laboratory, IST Klawad,
Haryana, INDIA

Parvinder Kaur
Department of Electronics &
Communications, SLIET,
Longowal, Punjab, INDIA

Amarpartap Singh
Department of Electronics &
Communications, SLIET,
Longowal, Punjab, INDIA

ABSTRACT

Nature-inspired methodologies are currently among the most powerful algorithms for optimization problems. This paper presents a recent nature-inspired algorithm named Firefly algorithm (FA) for automatically evolving a fuzzy model from numerical data. FA is a meta-heuristic inspired by the flashing behavior of fireflies. The rate and the rhythmic flash, and the amount of time form part of the signal system to attract other fireflies. The paper discusses fuzzy modeling for zero-order Takagi-Sugeno-Kang (TSK) type fuzzy systems. Simulations on two well known problems, one battery charger that is a fuzzy control problem and another Iris data classification problem are conducted to verify the performance of above approach. The results indicate that the FA is a very promising optimizing algorithm for evolving fuzzy logic based Systems as compared to some of the existing approaches.

General Terms

Soft Computing, Fuzzy Model Identification.

Keywords

Fuzzy logic, Firefly algorithm, Rule Base, Nature-inspired optimization, Fuzzy Modeling.

1. INTRODUCTION

Developing models of complex real-systems is an important topic in many fields of science and engineering. Models are generally used for simulation, identifying the system's behavior and design of controllers etc. The principles of fuzzy modeling were outlined by Zadeh when he gave the concept of grade of membership and published his seminal paper on fuzzy sets that lead to the birth of fuzzy logic technology [1]. Fuzzy systems provide a scheme to represent the knowledge in a way that resembles human communication and reasoning. Design of fuzzy model or fuzzy model identification is the task of finding the parameters of fuzzy model so as to get the desired behavior. Two different approaches are used for the design of fuzzy models: Knowledge driven and Data driven models. In the first approach, the design is constructed from the knowledge acquired from the expert, while in the second; the input-output data is used for building model.

In the first approach there were many problems and shortcomings; the interviews are generally long, inefficient and frustrating for both the domain experts and knowledge engineers, especially so in domains where experts make decisions based on incomplete or imprecise information. This knowledge acquisition phase is often the main bottleneck within the knowledge engineering process and therefore considerable effort has been expended in designing algorithms that automatically induce fuzzy rules from data [3]. Tagaki, Sugeno and Kang [4][5] developed the first approach for building and tuning fuzzy rules from the training data that laid the foundation for an important sub-area in fuzzy logic, referred to as fuzzy modeling or fuzzy model

identification. Many intelligent optimization techniques such as neural networks, genetic algorithms, swarm intelligence, ant colony optimization, biogeography based optimization etc. have been proposed to automatically generate fuzzy rules from numerical data [6]-[52].

This paper discusses a new approach to fuzzy model identification problem making use of Firefly algorithm. The paper is set up as follows. In Section 2 a brief introduction to fuzzy system modeling is presented. Section 3 provides a brief account of FA optimization algorithm and the framework for fuzzy model identification through FA is presented in Section 4. Section 5 represents simulation results considering two examples, one control problem and other Iris data classification problem. Finally, conclusions are drawn in Section 6.

2. FUZZY SYSTEMS MODELING

Fuzzy modeling is the task of identifying the parameters of fuzzy inference system so as to achieve a desired behaviour. The fuzzy model identification process involves the question of providing a methodology for development i.e. a set of techniques for obtaining the fuzzy model from information and knowledge about the system.

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

- 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.

These issues can be grouped into three subproblems: structure identification, parameter estimation and model validation as shown in Figure 1. If the performance of the model obtained is not satisfactory, the model structure is modified and the parameters are re-estimated till the performance is satisfactory.

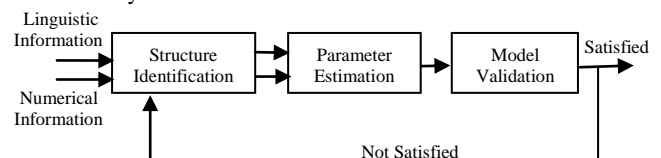


Figure 1: Fuzzy Model Identification Process

3. FIREFLY ALGORITHM

The Firefly Algorithm (FA) is a meta-heuristic, nature-inspired, optimization algorithm which is based on the social

(flashing) behavior of fireflies. The flashing light of fireflies is an amazing sight in the summer sky in the tropical and temperate regions. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. In addition, flashing may also serve as a protective warning mechanism.

For simplicity, the flashing characteristics of fireflies idealized in following three rules [53]-[55]:

- 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.
- 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 2.

In the firefly algorithm, there are two important issues: the variation of light intensity and formulation of the attractiveness. For simplicity, it is assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function.

3.1 Attractiveness

The form of attractiveness function of a firefly is the following monotonically decreasing function [53]:

$$\beta(r) = \beta_0 e^{-\gamma r^m} \quad (m \geq 1) \quad (1)$$

where r is the distance between any two fireflies, β_0 is the attractiveness at $r = 0$ and γ is a fixed light absorption coefficient.

3.2 Distance

The distance between any two fireflies i and j at X_i and X_j , respectively, is the Cartesian distance as follows:

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2)$$

where $x_{i,k}$ is the k th component of the spatial coordinate X_i of i th firefly and d is the number of dimensions.

3.3 Movement

The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by following equation:

$$X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2} (X_j - X_i) + \alpha(\text{rand} - 0.5) \quad (3)$$

where the second term is due to the attraction while the third term is randomization with α being the randomization parameter. *rand* is a random number generator uniformly distributed in $[0, 1]$. For most cases in the implementation, $\beta_0 = 1$ and $\alpha \in [0, 1]$.

Firefly Algorithm (FA)

Begin

 /* FA parameter initialization */

 Define Objective function $f(X)$, $X = (x_1, \dots, x_d)^T$;

 Generate initial population of fireflies X_i ($i = 1, 2, \dots, n$)

 Compute the light intensity I_i at X_i by $f(X_i)$;

 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 ($I_j > I_i$)

 Move firefly i towards j in d -dimension;

End if

 Attractiveness varies with distance r via $\exp[-\gamma r]$;

 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 2: Pseudo code of the firefly algorithm

4. IDENTIFICATION OF FUZZY MODELS USING FIREFLY ALGORITHM

The fuzzy model identification can be formulated as a search and optimization problem. It 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 represent a fuzzy system by a firefly (set of fireflies represent a population). Each firefly 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.

$$MSE = \frac{1}{N} \sum_{k=1}^N [y(k) - \tilde{y}(k)]^2 \quad (4)$$

where,

$y(k)$ = Actual output as given in data set

$\tilde{y}(k)$ = 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, \dots, x_n and the number of fuzzy sets for these inputs are m_1, m_2, \dots, m_n respectively. Our encoding is based on the following assumptions:

- 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.
- First and last membership functions of each input and output variable are represented with z-type and sigma-type membership functions respectively.
- Complete rule-base is considered, where all possible combinations of input membership functions of all the input variables are considered for rule formulation.
- Overlapping between the adjacent membership functions for all the variables is ensured through some predefined constraints.

4.1 Encoding Method (Membership functions)

Let's assume that a variable is represented by three fuzzy sets as in Figure 3. The vertices are indicated by E_i 's in this figure, where E_1 ($i=1$) represent vertex of first fuzzy set and so on. 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, pp. 418, [2]) can be represented as below:

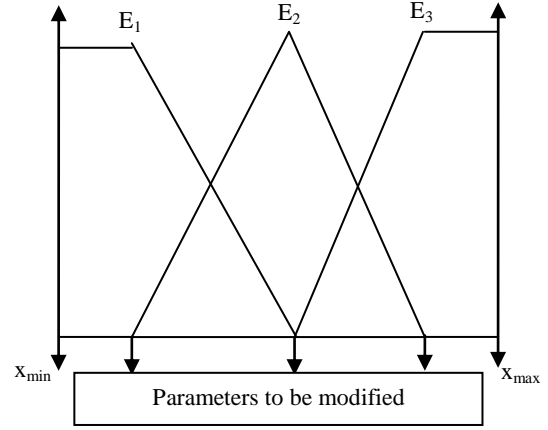


Figure 3: Representation of a variable with 3 membership functions with overlapping between the adjacent membership functions

$$X_{\min} < E_1 < E_2 < \dots < E_{m_n} < X_{\max} \quad (5)$$

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

As stated earlier each firefly consists of two parts; the first constituent part consists of membership functions of all the input variables. The number of membership functions (from Figure 3) while considering the assumptions made earlier, can be computed as follows:

Number of membership functions (size of first constituent part)

$$= \sum_{j=1}^n m_j \quad (6)$$

where n is the number of input variables, and m_j the number of fuzzy sets for j^{th} input variable.

4.2 Encoding Method (Rule-base)

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)

$$= \prod_{l=1}^n m_l \quad (7)$$

Hence, size of each firefly required to encode the zero-order TSK fuzzy model is the sum of equations (6) and (7).

$$\text{Firefly Size (Sugeno model)} = \sum_{j=1}^n m_j + \prod_{l=1}^n m_l \quad (8)$$

Each firefly represents one fuzzy model whose performance is evaluated in terms of MSE as defined in Eq. 4. This MSE is used as the fitness function for rating the fuzzy model.

Our optimization algorithm adjusts membership function parameters and consequents in such a way so as to minimize the objective function i.e. MSE. Now the whole problem of system identification boils down to a minimization problem as stated below:

Minimize objective function (MSE)

$$MSE = \frac{1}{N} \sum_{k=1}^N [AO - CO]^2$$

Subject to the constraint that

$$C_k \in \{\text{specified set of consequents}\}; \quad (9)$$

where AO is the actual output, CO is the computed output, N is number of data points taken for model validation and C_k is constant set of chosen consequents.

The process for the identification of fuzzy model using firefly algorithm is represented as pseudo-code in Figure 4.

Begin

Define operating parameters for firefly algorithm;

Iteration = 0;

Generate a random set of fireflies (initial population);

While Iteration \leq Maximum Iteration

Constraint Population;

Build fuzzy model corresponding to each firefly;

Evaluate each fuzzy model for its fitness (MSE) using Eq. (4);

Call FA algorithm to determine new position of each firefly by using Eq. (3) (as in Section III);

Iteration = Iteration + 1;

End

Display optimized fuzzy model;

End

Figure 4: Pseudo-code for Identification of Optimized Fuzzy Model

5. SIMULATION RESULTS

Problem 1: Battery Charger

The suggested approach has been applied for identification of fuzzy model for the rapid Nickel-Cadmium (Ni-Cd) battery charger [56]. The objective of this charger was to charge 2AA Ni-Cd batteries as quickly as possible but without causing 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 [57]. The input and output variables identified for rapid Ni-Cd battery charger along with their universes of discourse are given in Table 1.

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

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

Let us assume that the temperature with the universe of discourse ranging from 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 [58]. 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 $C_1 = \text{Trickle} = 0.1 \text{ Amp}$, $C_2 = \text{Low} = 1 \text{ Amp}$, $C_3 = \text{Med} = 2 \text{ Amp}$, $C_4 = \text{High} = 3 \text{ Amp}$ and, $C_5 = \text{Ultrafast} = 4 \text{ Amp}$. We have to choose parameters of antecedent and consequents in such a way so as to fulfill condition given by expression (9).

In this problem the size of a firefly to encode a Sugeno type fuzzy model for battery charger may be calculated from equation (8) as follows:

$$\begin{aligned} \text{Firefly Size (Sugeno model)} &= \sum_{j=1}^2 m_j + \prod_{l=1}^2 m_l \\ &= (3+2) + (3*2) = 5 + 6 = 11 \end{aligned}$$

Simulation Results and Discussions:

We have implemented the proposed approach of fuzzy model identification in Matlab on Intel Core i5-450M @ 2.4 GHz HP ENVY¹⁴ laptop with 4GB of RAM.

To obtain suitable parameters for algorithm, a large number of experiments were conducted with different parameters settings. The final parameters that were obtained through experimentation are as follows: $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 1$, number of fireflies = 50 and maximum iterations = 500. With the above parameters settings, the large numbers of sets of trials were conducted. A set of 10 trials is given in Table 2. The performance of all the sets is nearly the same as given in Table 2. We found that the proposed approach evolved optimized fuzzy model with average MSE of 0.008 in average time of 13 seconds. Further we observed from figure 5 that FA stabilizes the evolved fuzzy system to its optimum value in lesser number of iterations (in less than 100 iterations). This performance is for training data set (Appendix 1). For training purposes we selected 14 training examples out of 561 data points. The performance of obtained fuzzy system is then checked using the test data set. For testing purpose, we have chosen 20 points excluding those in the training data set.

Figure 6 presents the performance of evolved model on test data set. It is clear that the computed output (using FA) is approaching actual output which is given in the data set with MSE of 0.06.

The performance for the evolved fuzzy system was compared with that available in the literature and the same is given in Table 3. It can be observed that FA gives excellent model performance as compared to other approaches.

Table 2. Performance of FA algorithm in a set of 10 trials

Trials	Number of iterations = 500; Number of fireflies = 50	
	MSE	CPU Time (in seconds)
1	0.007	13
2	0.016	13
3	0.009	13
4	0.007	14
5	0.005	14
6	0.007	13
7	0.011	13
8	0.007	14
9	0.011	13
10	0.006	13
Average MSE = 0.008		

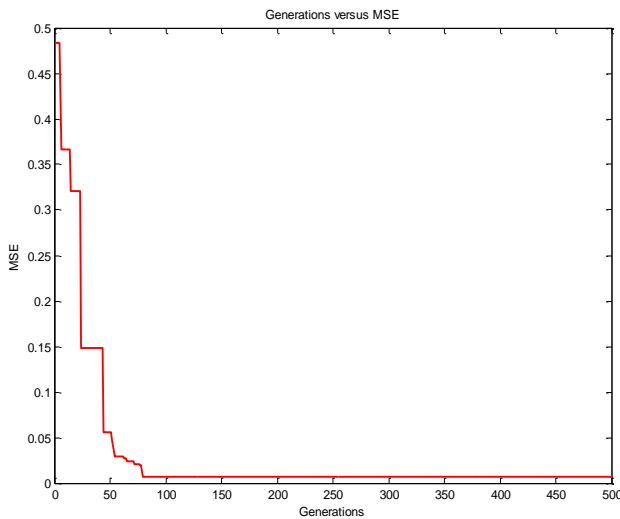


Figure 5: Graph of Generations vs MSE for FA

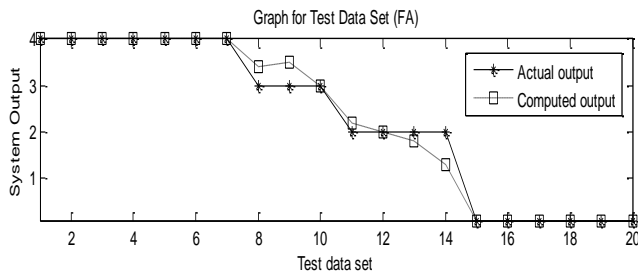


Figure 6: Performance of evolved fuzzy model on test data set

Table 3. Comparison of the Proposed Approach with Other Approaches

Algorithm	MSE
Hybrid Learning [59]	0.132
Genetic Algorithm [60]	0.130
Particle Swarm Optimization [61]	0.112
FA (Proposed approach)	0.008

Problem 2: Iris data set classification problem

The Iris data set [62] contains 150 patterns with four features that belong to three classes (Iris Setosa, Iris Versicolour and Iris Virginica). The four features are the sepal length, the sepal width, the petal length and the petal width. All the four features are specified in centimeters. The data set contains three classes, each of 50 patterns; each class refers to a type of Iris plant. 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. The parameters settings of FA algorithm for classification problem are same as that of battery charger problem.

Simulations were carried out to design a fuzzy classification system using FA algorithm for the Iris data set. Three membership functions were associated with each input variable. This algorithm successfully generates a fuzzy system, which yields only two misclassifications. The classification rates on training patterns and test patterns using proposed method are summarized in Table 4. Table 5 compares our results with the results in [63], [64] for the Iris classification problem. From the simulation results for the Iris classification problem, it is clear that the FA based fuzzy classification system has high generalization ability.

Table 4. Simulation results of the proposed method for the Iris data set

Classification rate on training patterns	Classification rate on test patterns
100%	98.66%

Table 5. Comparison of the proposed method with the existing methods

Method	Classification rate on test patterns
Pruning	93.30%
Multi-rule-table	94.30%
GA-based	90.67%
PSO-based	96.80%
FA (Proposed method)	98.66%

6. CONCLUSIONS

This paper presents a frame work to evolve a complete fuzzy model from available data using relatively new nature inspired algorithm based on flashing behavior of fireflies. The proposed algorithm successfully generated optimized fuzzy models from training data. The proposed approach was successfully validated on two problems. For control problem, FA appears to be more efficient in terms of computational time and MSE as compare to other approaches. Simulation results show that FA generated a fuzzy model in less than 100 iterations with average MSE of 0.008. For classification problem, the simulation results show that the proposed method evolved fuzzy classification system with high classification rate of 98.66 % (two misclassifications).

7. REFERENCES

- [1] Zadeh, L. A. 1965. Fuzzy Sets, Information and Control vol.8, 338-353.

- [2] Yen, J. and Langari, R. 1999. Fuzzy Logic Intelligence, Control and Information. Prentice Hall, New Jersey.
- [3] Wang, L. X. and Mendel, J. M. 1992. Generating Fuzzy rules from examples. IEEE transactions on Systems, Man and Cybernetics, vol. 22, no.6 (Nov/Dec 1992), 1414-1427.
- [4] Takagi, T. and Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics, 116-132.
- [5] Sugeno, M. and Kang, G.T. 1988. Structure identification of fuzzy model. Fuzzy Sets and Systems, 15-33.
- [6] Abe, S. and Lan, M. S. 1995. Fuzzy rule extraction directly from numerical data for function approximation. IEEE transactions on Systems, Man and Cybernetics, vol. 25, no.1 (Jan 1995), 119-129.
- [7] Juang, C. F. 2005. Combination of Online clustering and Q-value based GA for reinforcement fuzzy system design. IEEE transactions on Fuzzy Systems, vol. 13, no.3 (June 2005), 289-302.
- [8] He, Z., Wei, C., Yang, L., Gao, X., Yao S., Eberhart, R. C. and Shi, Y. 1998. Extracting rules from fuzzy neural network by particle swarm optimization. In Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA.
- [9] Homaifar, A. and McCormick, E. 1995. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Transactions on Fuzzy Systems, vol.3, no.2, 129-139.
- [10] Lee, M. A. and Takagi, H. 1993. Integrating design stages of fuzzy systems using genetic algorithms. In Proceedings of Second IEEE International Conference on Fuzzy Systems, 612-617.
- [11] Khosla, A., Kumar, S., Aggarwal, K.K. and Singh, J. 2005. Particle swarm optimizer for building fuzzy models. In Proceeding of one week workshop on applied soft computing SOCO-2005, Haryana Engg. College, Jagadhri, India (July 2005), 43-71.
- [12] Mahfouf, M., Jamei, M. and Linkens, D.A. 2001. Rule-Base generation via symbiotic evolution for a mamdani-type fuzzy control system. IEEE International Fuzzy Systems Conference, 396-399.
- [13] Denna, M., Mauri G. and Zanaboni A.M. 1999. Learning fuzzy rules with Tabu Search-An application to control. IEEE Transactions on Fuzzy Systems, vol. 7, no.2 (April 1999), 295-318.
- [14] Zeng, X. J. and Singh, M.G. 2003. Knowledge bounded least squares method for the identification of fuzzy systems. IEEE Transactions on Systems, Man and Cybernetics- Part C, vol. 33, no. 1 (Feb. 2003), 24-32.
- [15] Ishikawa, M. 1996. Structural learning with forgetting. Neural Networks, vol. 9, 509-521.
- [16] Duch, W., Adamczak, R., and Grabczewski, K. 1998. Extraction of logical rules from neural networks. Neural Process. Lett. vol. 7, 211-219.
- [17] Fu, L. 1991. Rule learning by searching on adapted nets. In Proceedings. of 9th National Conf. on Artificial Intelligence, 590-595.
- [18] Fu, L. 1994. Rule generation from neural networks. IEEE Transactions on Systems, Man and Cybernetics, vol. 24, no. 8 (Aug. 1994), 1114-1124.
- [19] Towell, G.G. and Shavlik, J. W. 1993. Extracting refined rules from knowledge-based neural networks. Machine learning, vol. 13, no.1, 71-101.
- [20] Setiono, R. 1997. Extracting rules from neural networks by pruning and hidden-unit splitting. Neural Computat., vol. 9, 205-225.
- [21] Setiono, R. 2000. Extracting M of N rules from trained neural networks. IEEE Trans. Neural Networks.
- [22] Setiono, R. and Liu, H. 1997. Neuro Linear: From neural networks to oblique decision rules. Neurocomputation. vol. 17, 1-24.
- [23] Ishibuchi, H. et al. 1993. Neural Networks that learn from Fuzzy if then rules. IEEE Trans. on Fuzzy Systems. vol. 1, 85-97.
- [24] Keller, J., Yager, R. and Tahani, H. 1992. Neural Network implementation of fuzzy logic. Fuzzy Sets and Systems. vol.45, 1-12.
- [25] Karr, C. L. 1991. Design of an adaptive fuzzy logic controller using a genetic algorithm. In Proceedings of 4th Int. Conf. Genetic Algorithms. 450-457.
- [26] Nomura, H., Hayashi, I. and Wakami, N. 1992. A self tuning method of fuzzy control by genetic algorithm. In Proceedings Int'l Fuzzy systems Intell. Contr. Conf. (IFSICC'92), 236-245.
- [27] Karr, C.L. and Gentry, E.J. 1993. Fuzzy control of pH using genetic algorithms. IEEE Transactions on Fuzzy Systems, vol. 1, no. 1, 46-53.
- [28] Nelles, O., Fischer, M. and Muller, B. 1996. Fuzzy rule extraction by a genetic algorithm and constrained nonlinear optimization of membership functions. In Proceedings of the 5th IEEE Int'l Conf. on Fuzzy Systems. vol. 1 (Sept. 1996), 213-219.
- [29] Shi, Y., Eberhart, R. and Chen, Y. 1999. Implementation of evolutionary fuzzy systems. IEEE Transactions on Fuzzy Systems, vol.7, no.2, 109-119.
- [30] Surmann, H. 2000. Learning a fuzzy rule based knowledge representation. In Proceeding of 2. ICSC Symp. on neural computation, NC'2000, Berlin. 349-355 (May 2000).
- [31] Ashwani, K., Agrawal, D.P., and Joshi, S.D. 2003. A GA-based method for constructing TSK fuzzy rules from numerical data. IEEE Int'l Conf. on Fuzzy Syztems, 131-136.
- [32] Hwang, H.S. 1999. Automatic design of fuzzy rule base for modeling and control using evolutionary programming. IEEE Proceedings-Control Theory Applications, vol. 146, no. 1, 9-16.
- [33] Kang, S.J., Woo, C.H., Hwang, H.S. and Woo, K.B. 2000. Evolutionary design of fuzzy rule base for nonlinear system modeling and control. IEEE Transactions on Fuzzy Systems. vol. 8. no.1, 37-45.

- [34] Pal, T., and Nikhil, R. P. 2003. SOGARG: A self organized genetic algorithm based rule generation scheme for fuzzy controllers. *IEEE Transactions on Evolutionary Computation*. vol. 7. no. 4 (Aug. 2003).
- [35] Mansoori, E. G., Zolghadri, M. J., and Katebi, S.D. 2008. SGERD: A steady-state genetic algorithm for extracting fuzzy classification rules from data. *IEEE Transactions on Fuzzy Systems*. vol.16. no.4 (Aug. 2008), 1061-1071.
- [36] Dorigo, M., Maniezzo, V., and Colomi, A. 1996. The Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*. vol.26. no.1, 1-13.
- [37] Colomi, A., Dorigo, M., and Maniezzo, V. 1991. Distributed optimization by ant colonies. In *Proceedings of the European Conference on Artificial Life, ECAL'91*, Elsevier Publishing, Amsterdam.
- [38] Dorigo, M. 1992. Optimization, learning and natural algorithms, Ph.D. Thesis, Politecnico di Milano, Milano.
- [39] Dorigo, M., and Gambardella L.M. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transaction on Evolutionary Computation* 1 (1997), 53-66.
- [40] Dorigo, M., and Stutzle, T. 2005. *Ant Colony Optimization*, Eastern Economy Edition, PHI.
- [41] Casillas, J., Cordon, O., and Herrera, F. 2000. Learning fuzzy rules using ant colony optimization algorithms. In *Proceedings of 2nd Int. Workshop Ant Algorithms*, 13-21.
- [42] Parpinelli, R.S., Lopes, H.S., and Freitas, A.A. 2002. An ant colony algorithm for classification rule discovery in Data Mining: A Heuristic Approach. *Idea Group Publishing*, 190-208.
- [43] Carmona, P., and Castro, J. L. 2005. Using ant colony optimization for learning maximal structure fuzzy rules. In *Proceedings of IEEE International Conference on Fuzzy Systems*, 999-999.
- [44] Kumar, S. 2005. Automatic Fuzzy rulebase generation. In *Proceedings of the One week workshop on Applied Soft Computing, CAT, Harayana Engg. College, Jagadhri, Haryana, India*, 26-42.
- [45] Chia-Feng, J., Huang, H.J., and Lu, C.M. 2007. Fuzzy controller design by ant colony optimization. *IEEE Proc. on Fuzzy Systems*.
- [46] Kumar, S., and Bhalla, P. 2007. Fuzzy rulebase generation from numerical data using ant colony optimization. *MAIMT- Journal of IT & Management*. vol.1, no.1, 33-47.
- [47] Kumar, S., Kaur P., and Singh, A.P. 2009. Soft computing approaches to fuzzy system identification: A survey. In *Proceedings of 3rd International Conference on Intelligent Systems and Networks (IISN-2009)*, ISTK. Jagadhri. Haryana. India, 402-411.
- [48] Simon, D. 2008. Biogeography-Based Optimization. *IEEE Trans. on Evolutionary Computation*, vol. 12, no. 6, 702-713.
- [49] Erol, O.K., and Eksin, I. 2006. A new optimization method: Big Bang-Big Crunch. *Advances in Engineering Software*, 37, 106-111.
- [50] 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*, 697-702.
- [51] Kripka, M., and Kripka, R.M.L. 2008. Big crunch optimization method. In *Proceedings of International Conference on Engineering Optimization, Brazil*.
- [52] Kumar, S., Kaur, P., and Singh A.P. 2010. Knowledge Extraction from Numerical Data for Mamdani Type Fuzzy System using Big Bang-Big Crunch Optimization. In *proceedings of International Conference on Future Challenges in Wireless Communication, Optical Communication and Networks (Nov. 2010)*, 45-53.
- [53] 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, 169-178.
- [54] Yang, X.S. 2010. Firefly algorithm, stochastic test functions and design optimisation, *Int. J. Bio-Inspired Computation*, vol. 2, no. 2, 78-84
- [55] 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, 209-218.
- [56] Khosla, A., Kumar, S. and Aggarwal, K. K. 2002. 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*, 9-14.
- [57] Linden, D. 1995. *Handbook of Batteries*, Mc.Graw Hill Inc.
- [58] 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. In *Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society, Chicago, Illinois, USA, (July 24-26, 2003)*, 536-539.
- [59] Khosla, A., Kumar, S., and Aggarwal, K. K. 2006. Fuzzy controller for rapid nickel-cadmium batteries charger through adaptive neuro-fuzzy inference system (ANFIS) architecture. In *Proceedings of 22nd International Conference of the North American Fuzzy Information Processing Society, Chicago, Illinois, USA (July 24-26, 2003)*, 540-544.
- [60] Kumar, S. 2010. Introduction to fuzzy logic based systems. In *Proceedings of Workshop on Intelligent System Engineering (WISE-2010)*.
- [61] Khosla, A., Kumar, S. and Aggarwal, K. K. 2005. A framework for identification of fuzzy models through particle swarm optimization algorithm. In *Proceedings of IEEE Indicon 2005 Conference, Chennai, India, 11-13 (Dec. 2005)*, 388-391.
- [62] Blake, C., Keogh, E., and Merz, C.J. 1998. *UCI Repository of Machine learning database*, Univ. California, Irvine. <http://www.ics.uci.edu/~mllearn/>
- [63] Nozaki, K., Ishibuchi, H., and Tanaka, H. 1996. Adaptive fuzzy rule-based classification systems. *IEEE*

Trans. on Fuzzy Systems, vol.4, no.3 (Aug. 1996), 238-250.

[64] Chen, C.C. 2006. Design of PSO-based fuzzy classification systems.. Tamkang journal of science and engineering, vol. 9, no.1, 63-70.

APPENDIX – 1

Training Data Set for Battery Charger

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

Test 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

Training Data Set for Iris Plants

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

Input 1 – Temperature

Input 2 – Temperature Gradient

Actual Output - Charging Current

Class 1 – Iris Setosa

Class 2 – Iris Versicolour

Class 3 – Iris Virginica