

ANN Model Identification: Parallel Big Bang Big Crunch Algorithm

Ashima Kalra, Shakti Kumar, Sukhbir Singh Walia

Abstract: This paper proposes a modification to existing big bang big crunch optimization algorithm that uses the concept of more than one population. In this the search begins with all the populations independently in parallel and as the algorithm proceeds the local best of the individual populations interact with global best to avoid local minima. In order to validate the proposed approach the authors have identified two models one from control field namely rapid battery charger and second a rating system for institutes of higher learning and compared its results with simple BB-BC based approach. The author further compared results of the proposed approach with the results of other recent soft computing based algorithms for ANN model identification. The proposed algorithm outperformed all of the other 7 algorithms in terms of MSE and convergence time.

Index Terms: model identification, ANN (artificial neural network), big bang big crunch (BB-BC) optimization, parallel big bang big crunch (PBB-BC) optimization Levenberg-Marquardt algorithm (LM), error back propagation (EBP), Resilient prop (RPROP), particle swarm optimization (PSO), ant colony optimization(ACO) and artificial bee colony(ABC) .

I. INTRODUCTION

An ANN is a massively parallel distributive processing system made up of simple processing elements which has ability to store experiential knowledge and later making it available for the use. ANNs have a property of learning by examples, thus ANNs can be trained with known set of examples for a given problem before they are tested for their inference capability on unknown instances of problem. These can therefore, identify the objects for which they are not previously trained. ANNs are characterized by properties such as mapping capabilities or pattern association, and generalization capability for tolerance and higher reliability. ANNs have been widely used in finance and banking, manufacturing, marketing, medicine, environment applications, pattern recognition, and control applications[1]-[4].

This paper proposes an iterative method of modeling layered ANN system. The method is general in nature and can be applied to any other problem where ANNs can be applied. The author proposed ANN modeling approach on two different problems namely Rapid Battery Charger (RBC) and Performance Evaluation of Institutions of higher Learning from the given training data set. Shakti et al.[5] presented an ANN based model for a Rapid Battery Charger (RBC). The method had high computational complexity and hence, was not suitable for larger systems. Khosla, A et al. [6],[7] also discussed fuzzy system modeling for Rapid Battery Charger.

Revised Manuscript Received on July 22, 2019.

Ashima Kalra, PhD research scholar, Punjab Technical University, Punjab, India.

Prof. (Dr.) Shakti Kumar, Director, Panipat Institute of Engineering Technology, Panipat, Haryana, India

Dr. Sukhbir Singh Walia, Registrar, Punjab Technical University, Punjab, India

Many methods for the ANN System training and design are available in literature. Shakti et al.[8] presented an ANN design automation tool which was a generalization of (Shakti et al. [5]. it also suffered from complexity issues. Gallant[9] introduces the layered connectionist model. Ishikawa, M. [10] demonstrated the training of a network using structural learning with forgetting and Duch et al. [11] modified this algorithm by constraining the weights to 1, 1, or 0. Various other authors in their papers [11]-[15] presented extraction of rules from neural networks using different techniques. In contrast to ANN model identification, literature is rich for fuzzy model identification from the given data set. Fuzzy model identification from given data set, based upon GA [16]-[19], based upon bio-geography based optimization (BBO)[20]-[23], based upon ant colony optimization (ACO)[24]-[27], based upon particle swarm optimization (PSO)[28]-[30] and based upon big bang big crunch (BB-BC)[31][32] and parallel BB-BC [33][34]are available in literature. Similar techniques are applicable for ANN model identification as well. A rich survey on available classical as well as soft computing based approaches for the same has been done by A Kalra.et al [35]. A Kalra et al. [36] proposed an optimized ANN model identification approach using two soft computing based approaches i.e. PSO and ABC for two different problems. A Kalra et al. [37] further presented an optimized ANN model identification approach using BBBC. In this the author presented an ANN model identification approach based on Parallel BB-BC optimization. The system identification problem was formulated as minimization problem. These approaches are used to find out the optimal values of synaptic weights, number of neurons and number of layers of the ANN based model.

The author first discussed ANN system modeling process with brief introduction of parallel big bang big crunch algorithm. It is then followed by the proposed ANN model based on PBBBC, simulation results, performance comparison of PBBBC with other algorithms and conclusion in the end.

II. ANN SYSTEM MODELING PROCESS

In this paper the author discussed ANN system design for two different problems, Rapid Battery Charger (RBC) and Performance Evaluation of Institutions of Higher Learning from the given training data set. The complete modeling of an ANN system consists of two processes: first the selection of ANN architecture in which number of hidden layers and the number of neurons in each hidden layer is to be decided. Second is the training of this ANN system by the given training data. The problem here is formulated as search and minimization problem. The optimization algorithms are applied in a way to automatically adjust the

number of hidden layers, neurons in each of the hidden layers and identified values of synaptic weights in such a way so as to minimize the objective function i.e. MSE:

$$MSE = \frac{1}{N} \sum_{k=1}^N [OA - OC]^2 \quad (1)$$

where *OA* is the actual output or desired output, *OC* is the computed output, *N* is number of training examples used for model identification.

Modeling an ANN System is a complex process involving number of steps. This complexity further increases with the increase in the number of input parameters and number of hidden layers. The main steps to be followed for designing a complete ANN System are given as below:

1. Begin with number of hidden layers $NH = 0$
2. Fix the number of neurons in each hidden layer.(values of $k_1, k_2 \dots$ and so on)
3. Randomly initialize the weights of ANN
4. For each training pattern, evaluate output and error between the computed and desired output.
5. Compute mean square error for the model (MSE).
6. Minimize the objective function (MSE) by adjusting the weights using proposed approach. (We have used simple Big Bang Big Crunch (BB-BC) and Parallel BB-BC algorithms based optimization approaches.)
7. If MSE is acceptable or termination criterion is met, then go to step 9, else if number of hidden layers are non zero then increase the number of neurons in the hidden layers. After an upper limit of the number of neurons in the hidden layers has reached and if the performance is still not acceptable we increase the number of hidden layers.
8. Go to step 4.
9. Stop

The ANN model for these problems has been identified using parallel BB-BC algorithms based Optimization Approaches. The figure 1 represents the flow diagram of ANN System identification process.

A. Parallel BB-BC Optimization based Algorithms for ANN System Identification

Big Bang-Big Crunch is one of the simplest optimization algorithms used in the soft computing. This approach is based on the theory of the evolution of the universe [38]. In the Big Bang phase, energy dissipation produces disorder and in the Big Crunch phase, randomly distributed particles are drawn into an order. An optimization algorithm based on Big Bang-Big Crunch optimization algorithm was first proposed by Shakti et al[33].The pseudo code of BB-BC is shown below:-

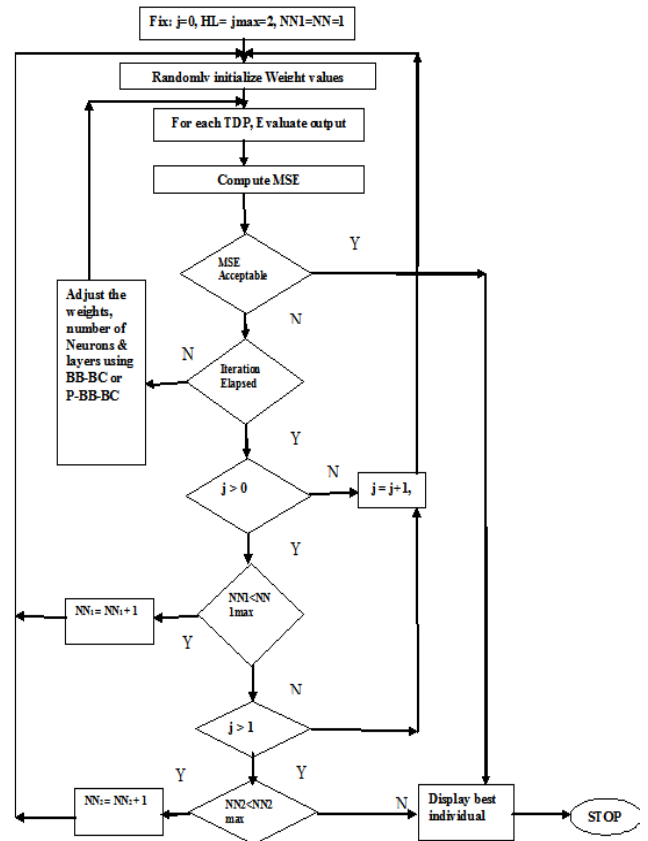


Figure 1. Flow Diagram ANN System Design process

Begin

/ Big Bang Phase */*

Generate a random set of *NC* candidates (population);

/ End of Big Bang Phase */*

While not *TC* */* TC 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 (x^c) using equation 2;

$$x^c = \frac{\sum_{i=1}^{np} \frac{1}{f^i} x^i}{\sum_{i=1}^{np} \frac{1}{f^i}} \quad (2)$$

Where x^c = position of the CoM;

x^i = position of

candidate *i*;

f^i = fitness function value of candidate *i*;

np = The population size in Big Bang phase.

Best fit individual can be chosen as the center of mass instead of using equation 2;

/ 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 equation 3;

$$x^{new} = x^c + l(rand)/k \quad (3)$$

End while

End



B. Parallel BB-BC Optimization Algorithm

This optimization algorithm was proposed by [33][34] and is a modification of the existing BB-BC optimization algorithm called simple BB-BC algorithm. This is a multi-population approach in which more than one populations of individuals work through the search space in parallel to find out the optimal solution. The pseudo code of PBB-BC is shown below:-

```

Begin
  /* Big Bang Phase */
  Generate N populations each of size NC candidates randomly;
  /* End of Big Bang Phase */
  While not TC /* TC is a termination criterion */
  /* Big Crunch Phase */
  For i = 1: N
    Compute the fitness value (centre of mass using Equation 1) of all the candidate solutions of ith population;
    Best fit individual can be chosen as the centre of mass instead of using Eq. 1;
    Sort the population from best to worst based on fitness (cost) value;
    Select local best candidates  $l_{best}$  (i) for ith population;
  End
  From amongst "N"  $l_{best}$  candidates select the globally best  $g_{best}$  candidate;
  For i = 1: N
    With a given probability replace a gene of  $l_{best}$  (i) with the corresponding gene of global best  $g_{best}$  candidate
  End
  /* End of Big Crunch Phase */
  /* Big Bang Phase */
  Calculate new candidates around the centre of mass by adding or subtracting a normal random number whose value decreases as the iterations elapse using Equation 2;
  /* End of Big Bang Phase */
End while
End

```

C. ANN System Identification based on Parallel BB-BC Optimization

The author now discussed below the algorithm for design and training ANN models for two above said problems using this approach.

Nomenclature

Size of the individual = S
 Number of individuals in each population = N
 Number of Inputs = NI
 Number of Hidden Layers = NH
 Max number of Hidden Layers = $NH_{max} = j$
 Number of data points = NP
 Number of Neurons = $NN_1 = k_j$ Where $I = 1$ to j
 Number of iterations = NT
 Number of populations = NP

1. Begin % initialize number of hidden layers%
2. for NH = 0: j (number of hidden layers: $j <= 2$)

```

   $NN_{1max} = NDP/3$ 
  3. for  $NN_1 = 2: NN_{1max}$ 
    (number of neuron in the 1s thidden layer)

     $NN_{2max} = \text{floor}(NN_{1max} * 0.6)$ 
  4. for  $NN_2 = 2 : NN_{2max}$ 
    (number of neuron in the 2nd hidden layer)

```

Compute the size of individual as per the equations/criteria given below:

```

  5. If NH = 0
    S = Number of inputs + 1 = NI+1
  6. elseif NH = 1
    S =  $NN_1 * (NI+2) + 1$ 
  7. elseif NH = 2
    S =  $NN_1 * (NI+1) + NN_2 * (NN_1+2) + 1$ 
  8. end

```

% Initialize the Populations %
 Randomly generate NP populations (Pop_number = 1:NP)
 Each population consists of N individuals each of size S genes.

```

  9. for jj = 1: NT ( number of iterations)
  10. for population kk = 1: NP (number of populations) .
  11. for ii = 1: N (population size)
  12. for mm = 1: NDP (No. of Data points)

```

for each data point evaluate the output of ANN and calculate the error between the desired and the computed output.

```

  13. end (end of mm loop)

```

Compute the MSE for each individual.

```

  14. end (end of ii loop)

```

Compute local best individual (on the basis of MSE) for each population

```

  15. end (end of kk loop)

```

Evaluate the global best individual from the local best individual.

```

  16. if global best <= acceptable value then exit .
  17. if jj > NT/2 (Number of iterations/2)
  18. for pp = 1: NP

```

With a given probability replace the gene of local best individual with the value = (value of corresponding gene of global best + value of gene of local best)/2 .

```

  19. end (end of pp loop)
  20. for rr = 1: NP

```

With a given probability, generate the next population around the modified value of the local best individual.

```

  21. end (end of rr loop)
  22. end (end of jj loop)
  23. end ( $NN_2 = 2 : NN_{2max}$  loop)
  24. end ( $NN_1 = 2: NN_{1max}$  loop)

```



25. end (NH =1: j loop, number of hidden layers)
26. end (global best <= acceptable value then exit loop)
27. Display the global best individual and the Global best fitness.

The global best candidate gives the optimum values of the weights for each layer such that network MSE is minimum for whole set of training data.

III. PROPOSED MODEL AND METHODOLOGY

An ANN System is a network of neurons arranged in different layers. Designing an ANN model mainly involves two steps: i) selecting of optimal number of layers, number of neurons in each layer and type of activation function and ii) finding optimal values for the synaptic weights also called training of ANN. In this paper the author applied simple BB-BC and Parallel BB-BC algorithm based optimization approaches to design ANNs from the given training data set. The system parameters/variables such as number of neurons in each layer, number of hidden layers and weights for every connection are identified using these approaches, so as to achieve goal of minimum MSE. Figure 2 below shows the a multi-layer ANN system . $x_1, x_2, x_3, \dots, x_n$ are the “n” input variables for the system, $N_1, N_2, N_3, \dots, N_n$, are the input layer neurons.

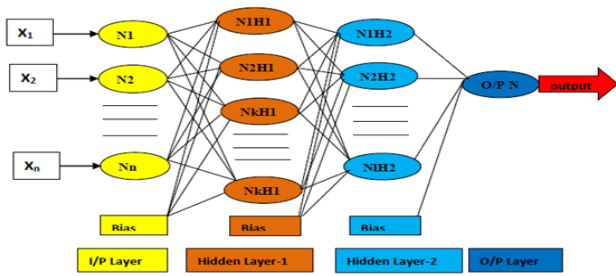


Fig. 2: Multilayer ANN system

The number of input layer neurons equals number of inputs to the system. In our case number of O/P is one. The net input to the system can be represented as follows,

$$net_{in} = N_0 + \sum_{i=0}^N W_i X_i \quad (4)$$

The output “y” of the system is taken at the output neuron and can be represented as

$$Y = 1 / (1 + e^{-(net_{in} - (t) - \theta) / \rho}) \quad (5)$$

The $W_0, W_1, W_2, \dots, W_n, W_{0N1k1}, W_{1N1k1}, W_{2N1k1}, \dots, W_{nN1k1}$ are the synaptic weights of the connections between different layers of neurons. The number of hidden layers and number of neuron in these hidden layers of this ANN system may vary as per requirements of design.

For identifying the ANN model using above mentioned optimization algorithms, a population of individuals is randomly generated. Each individual in the population represents a complete artificial neural-network system. The size of an individual i.e. set of synaptic weights, is variable and depends upon following factors.

- Number of inputs (n)
- Number of hidden layers (HL)(j)
- Number of neurons in each HLs (k_1, k_2, \dots, k_j)

With the increase in number of layers or the number of neurons in these layers the number of weights i.e. the size of individual increases. The author has taken both the data sets from computational lab (www.cilab.in). The author discussed it for two examples namely rapid battery charger and institute rating system[36],[37]. In case of rapid battery charger system, the number of neurons in the input layer is 2 which is equal to that of input variables. And another system, institute rating system is one with 14 input variables and one output variable named “overall rating”. Thus input layer will have 14 neurons in it. The number of layers of ANN is fixed between 0 to j and number of neuron in each layer is assumed to be k_1, k_2, \dots, k_j respectively. Both these values are also modified to find out the optimal solution. For a single layer ANN system each individual will consist of following members as shown below in the figure 3.

W_1	W_2	...	W_n	W_o
1	2		Nl	n+1

Fig.3: Individual for a single (zero hidden layer) layer ANN

The size of individual can be computed as given below:

$$S = Nl + 1 = n + 1 \text{ if } NH = 0 \quad (6)$$

For multi-layer system having one hidden layer individual can be represented as shown in the figure 4. The size of individual is the total number of elements in all the rows of the figure 4 and is given by Equation:

$$S = k_1 (n + 2) + 1 \text{ if } NH = 1 \quad (7)$$

W_{iN1j} = Weights between i th neuron of input layer and j th neuron of first hidden layer				
W_{0N1j} = Weights between bias and j th neuron of first hidden layer				
W_{N1p} = Weights between p th neuron of hidden layer 1 and output node				
W_{i0} = Weight between bias point and output node				
W_{1N11}	W_{2N11}	---	W_{nN11}	W_{0N11}
W_{1N12}	W_{2N12}	---	W_{nN12}	W_{0N12}
----	----	---	----	----
W_{1N1k1}	W_{2N1k1}	---	W_{nN1k1}	W_{0N1k1}
+				
W_{N11}	W_{N12}	-----	W_{N1k1}	W_{i0}

Fig.4: Individual for one hidden layer ANN

W_{iN1j} = Weights between i th neuron of input layer and j th neuron of first hidden layer				
W_{0N1j} = Weights between bias and j th neuron of first hidden layer				
W_{1N11}	W_{2N11}	---	W_{nN11}	W_{0N11}
W_{1N12}	W_{2N12}	---	W_{nN12}	W_{0N12}
----	----	---	----	----
W_{1N1k1}	W_{2N1k1}	---	W_{nN1k1}	W_{0N1k1}
+				
W_{N1pN2m} = Weights between p th neuron of first hidden layer and m th neuron of second hidden layer				
W_{0N2m} = Weights between bias for second layer and m th neuron of second hidden layer				
W_{N2m} = Weights between m th neuron of second layer and output node				
W_{20} = Weight between bias and output node				

WN11N21	WN12N21	---	WN1k1N2 1	W10N21
WN11N22	WN12N22	---	WN1k1N2 2	W10 N22
---	----	---	---	---
WN11N2k 2	WN12N2k 2	---	WN _{1k1} N _{2k2}	W10N2k 2
+				
WN21	WN22	---	WN2k2	W20

Fig.5 : Individual for Two Hidden Layer ANN

For multi layer ANN system having two hidden layers of neuron the individual can be represented as in figure 5. The size of individual for two hidden layer ANN system is given by Equation 8:

$$S = k_1 (n+1) + k_2 (k_1+2) + 1 \text{ if } NH=2 \quad (8)$$

Similarly the size of individual can be calculated for an ANN system with any number of hidden layers and any number of inputs.

IV. SIMULATION RESULTS AND DISCUSSIONS

In order to validate the proposed approach the author implemented these algorithms in MATLAB on a DEL Laptop with Intel core i3 processor, running on Windows 7 platform. The author used rapid battery charger (RBC) data and identification for institute rating system (IRS) data to evolve the architecture as well as for training purpose. The data has taken from computational laboratory at www.cilab.in. The author used 20% data for training purpose. For each of the implemented approach, 20 trials were taken with 500, 1000, 2000, 5000 and 10,000 iterations and the MSE of each evolved model is then recorded. The author has taken 50 as population size, 2 maximum number of hidden layers, 5 maximum no. of neurons in first hidden layer and 3 maximum no. of neurons in second hidden neurons for ANN model identification.

Table 2 and Table 3 compare the performance of proposed algorithm with other approaches namely Levenberg-Marquardt algorithm (LM), error back propagation (EBP), Resilient prop (RPROP), particle swarm optimization (PSO), ant colony optimization (ACO) and artificial bee colony (ABC) and big bang big crunch (BBBC). based model identification approaches for Rapid Battery Charger (RBC) and institute rating system (IRS). The value of MSE with this proposed approach has been found to be far superior than the other 7 algorithms.

V. PERFORMANCE COMPARISON

For each of the implemented approach, the network is trained with 500, 1000, 2000, 5000 and 10,000 iterations and recorded the MSE of each evolved model. For all the trials the author observed the minimum, mean and maximum MSE and mean execution time. Out of the 8 approaches used, PBBBC based approach produced the minimum MSE model with MSE value of 10^{-7} followed by BBBC based approaches with MSE value of 10^{-5} . The author observed that PBBBC identified RBC model with mean MSE of 0.000275 in 167 sec while BBBC identified the same with mean MSE of 0.0059 in 47.44 sec for 5000 iterations. It is also observed that PBBBC

identified IRS model with mean MSE of 1.4×10^{-6} in 43.5 sec while BBBC identified the same with mean MSE of 0.000114 in 30.2 sec for 5000 iterations.

As the numbers of iterations are increased from 5000 to 10000, PBBBC identified RBC model with minimum MSE of 2.95×10^{-7} while BBBC achieved minimum MSE of 0.0018. Also in the case for IRS model identification, PBBBC identified IRS model with minimum MSE of 4×10^{-10} in 56.06 sec while BBBC achieved minimum MSE of 2.49×10^{-6} for 10000 iterations. This became apparent from the observations that as the number of iterations was increased to 10000 for training, PBBBC based approach outperforms BBBC based approach for both the examples. Table 2 and 3 clearly highlights the performance improvement provided by the proposed algorithm and compared it with other 7 approaches. But as far as accuracy and computing time is concerned PBB-BC completely outperforms the other soft computing based and conventional approaches based ANN system modeling approach.

VI. CONCLUSION

The paper presented parallel BBBC based model identification approach. We applied this approach to identify two systems i) a 14 input, single output ANN system for evaluating the over-all rating of universities and institutes of higher learning and ii) a 2 input, single output ANN system for rapid battery charger. The proposed parallel BB-BC algorithm is a multi-population algorithm. The parallel search has improved the search and optimization capability of the simple BB-BC. In the beginning search begins by all the populations independently in parallel. As the algorithm proceeds the local best i.e. best of the each population interacts with global best with a given probability and thus avoids the local minima if any. We used a battery data set consisting of total of 561 patterns and IRS data set of 135 patterns. This algorithm is used to identify an ANN model for a battery charging system and institute rating system from a given training data set. We compared its results with simple BB-BC based approach and other 7 approaches. The results were found to be much more superior than the simple BB-BC based approach. It produced better minimum, better average and better Maximum MSE observed over training data set. We further compared the proposed algorithm with 7 other algorithms found in literature. These 7 algorithms include recently used soft computing based algorithms namely ACO, ABC, PSO and simple BB-BC and two classical approaches namely LM, RPROP and EBP for ANN model identification. The proposed algorithm outperformed all of the other 7 algorithms.

It is apparent that LM based approach is quickest to converge with fair amount of accuracy. However, if the accuracy requirement is high then PBBBC and BBBC based approaches perform best for both the examples in that order but with a requirement of higher training times followed by PSO, ABC, LM, Rprop and EBP approaches. However, if adequate training time is available then out of the seven algorithms that we have considered, PBBBC was found to offer best performance with minimum MSE of all the approaches.

REFERENCES

- [1] Bishop Chris M., (1994) Neural Networks and their applications, Review of Scientific Instruments, Vol. 65, No. 6, pp. 1803-1832.
- [2] Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Neural Network Design, ISBN:0-9717321-0-8.
- [3] Soroush A.R., Kamal-Abadi, NakhaiBahreininejad A, 2009 "Review on Applications of Artificial neural networks in supply chain management", World Applied Sciences Journal 6 (supplement 1), pp. 12-18.
- [4] Yaochu Jin, Jingping Jiang, Jing Zhu, June 1995, "Neural Network Based Fuzzy Identification and Its Applications to Control of Complex systems", IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 6, pp. 990-997.
- [5] Shakti Kumar, Nitika Ohri, Savita Wadhvan 2004 a, "ANN based design of rapid battery charger", Trends Of Computational Techniques In Engineering Oct 15-16, SLIET, Longowal Punjab pp 129-132.
- [6] 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.
- [7] Khosla, A., Kumar, S. and Aggarwal, K. K. (2003 b). 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.
- [8] Shakti Kumar, Savita Wadhawan and Neetika Ohri 2004 b, "ANDEng: An Artificial Neural Network Design Automation Tool". National Conference on Intelligent Systems and Networks, ISN-2004, Feb27-28, Haryana Engineering College Jagadhri, Haryana, pp 26-29
- [9] Gallant, S. (1988). Connectionist expert systems. Comm. ACM. 31(2) : 152-169 .
- [10] Ishikawa, M., 1996, "Structural learning with forgetting," Neural Networks, vol. 9, pp. 509-521.
- [11] Duch, W., R. Adamczak, and K. Grabczewski, (1998), "Extraction of logical rules from neural networks," Neural Process. Lett., vol. 7, pp. 211-219.
- [12] Ishibuchi, H. et al., (1993), "Neural Networks that learn from Fuzzy if then rules," IEEE Trans. on Fuzzy Systems, Vol.1, pp.85-97.
- [13] Towell, G.G., J. W. Shavlik, 1993, "Extracting refined rules from knowledge-based neural networks," Machine learning, vol. 13, no.1, pp. 71-101.
- [14] Setiono, R., 1997, "Extracting rules from neural networks by pruning and hidden-unit splitting," Neural Computat., vol. 9, pp. 205-225.
- [15] Setiono, R., 2000, "Extracting M of N rules from trained neural networks," IEEE Trans. Neural Networks.
- [16] Ashwani K., D.P. Agrawal and S.D. Joshi (2003). A GA-based method for constructing TSK fuzzy rules from numerical data *IEEE Int'l Conf. on Fuzzy Systems*, pp.131-136.
- [17] Eghbal G. Mansoori, M.J. Zolghadri and S.D. Katebi, (2008) "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.
- [18] 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.
- [19] Eghbal G. Mansoori, M.J. Zolghadri and S.D. Katebi, Aug. 2008 "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.
- [20] Shakti Kumar, Parvinder Bhalla, AP Singh, July 2009, "Fuzzy rule generation from numerical data using biography based optimization. Journal of Institutions of Engineers (India), IE(I) Journal-ET, Volume 90, pp. 8-13. .
- [21] Simon D. December 2008, "Biogeography-Based Optimization," IEEE Transactions on Evolutionary Computation, vol. 12, no. 6, pp. 702-713.
- [22] 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.
- [23] 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.
- [24] Carmona, P. and J. L. Castro, 2005, "Using ant colony optimization for learning maximal structure fuzzy rules," Proc. IEEE Int. Conf. Fuzzy Systems, pp.999-999.
- [25] Chia-Feng J., H.J. Huang and C.M. Lu, 2007, "Fuzzy controller design by ant colony optimization," IEEE Proc. on Fuzzy Systems.
- [26] Dorigo M and L.M. Gambardella (1997), Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transaction on Evolutionary Computation 1, pp. 53-66.
- [27] 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. 26(1) : 1-13.
- [28] Chen, C.C. 2006. Design of PSO-based fuzzy classification systems. Tamkang journal of science and engineering, vol. 9, no.1, 63-70. 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.
- [29] He Z., Wei C., Yang L., Gao X., Yao S., Eberhart R. C., Shi Y., 1998, "Extracting Rules from Fuzzy Neural Network by Particle Swarm Optimization", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA.
- [30] Khosla, A., Kumar, S. and Aggarwal, K. K. 2005. A framework for identification of fuzzy models through particle swarm optimization algorithm. IEEE Indicon 2005 Conference, Chennai, India, Dec. 11-13. pp. 388-391.
- [31] Shakti Kumar et al, January 2011 "Fuzzy Rulebase Generation from Numerical Data using Big Bang-Big Crunch Optimization", IE(I)Journal -ET, Volume 91, pp 1-8.
- [32] Kumbasar, T, E Yesil, I Eksin and M Guzelkaya., 2008, "Inverse Fuzzy Model Control with Online Adaptation via Big Bang-Big Crunch Optimization" ISCCSP2008, Malta, March 12-14, pp. 697.
- [33] Shakti Kumar, Sukhbir Singh Walia, A Kalanidhi , Nov 2013 "Fuzzy Model Identification: A New Parallel BB-BC Optimization Based Approach" International Journal of Electronics and Communication Engineering. Vol 2, Issue 5, pp 167-178.
- [34] Shakti Kumar, Sukhbir Singh Walia, Amar Singh Sept 2013 a, "Parallel Big Bang-Big Crunch Algorithm", International Journal of Advanced Computing, Vol.46, Issue.3 1, pp 1330-1335.
- [35] A Kalra, S. Kumar, S.S Walia. (2016). ANN Training: A Survey of classical and Soft Computing Approaches, *International Journal of Control Theory and Applications*, Vol. 9, pp.715-736.
- [36] A Kalra, S. Kumar, S.S Walia (2017). ANN Model identification: Two Soft Computing Based Approaches, *International Journal of Research and Analytical Reviews*, Vol. 4 , issue 2, pp 79-86.
- [37] A Kalra, S. Kumar, S.S Walia (2018). ANN Model Identification: A BB-BC Optimization Algorithm Based Approach, *International Journal of Computer Sciences and Engineering*, vol 6 issue 12, pp 264-271.
- [38] Erol O. K., Eksin I., 2006, A new optimization method: Big Bang-Big Crunch, *Advances in Engineering Software*, vol 37, 106-111.

Table 2: Performance comparison of MSE with different approaches for Rapid Battery Charger (RBC)

Performance Measures	Model identification approach							
	Iterations=500							
	PSO	BBBC	ACO	EBP	RPROP	LM	ABC	PBBBC
maximum MSE	0.0401	0.0154	0.0193	0.1178	0.1171	1.0415	0.1191	0.0015
Mean MSE	0.018933	0.005422	0.014427	0.11756	0.1164	0.41166	0.113933	0.000324
Minimum MSE	0.0046	0.000727	0.0118	0.117	0.1105	0.0212	0.1098	2.01E-09
Elapsed Time (sec)	1.733587	4.571813	5.940047	9.48534	8.492327	0.070707	16.14322	16.97283
Iterations=1000								
maximum MSE	0.0356	0.0224	0.0157	0.1177	0.117	0.2383	0.1182	0.0011
Mean MSE	0.017413	0.009638	0.01326	0.11742	0.11546	0.08066	0.1127	0.000264
Minimum MSE	0.0059	0.000666	0.0114	0.1171	0.1043	0.0091	0.1077	7.32E-06
Elapsed Time (sec)	4.24708	9.248847	10.54223	16.37903	14.74454	0.023947	37.54106	33.91405
Iterations=2000								
maximum MSE	0.0418	0.0207	0.0208	0.1176	0.117	1.1595	0.1179	0.0017
Mean MSE	0.01432	0.006124	0.014293	0.117293	0.116327	0.231733	0.113293	0.000211
Minimum MSE	0.0029	1.99E-11	0.0126	0.117	0.1113	0.0093	0.1086	1.16E-06
Elapsed Time (sec)	7.27198	17.55212	23.21797	31.49277	26.62006	0.026447	44.56536	64.61417
Iterations=5000								
maximum MSE	0.041	0.017	0.0202	0.1177	0.117	0.4743	0.1184	0.0022
Mean MSE	0.017727	0.005959	0.014527	0.117273	0.115827	0.101053	0.11392	0.000275
Minimum MSE	0.0066	1.25E-13	0.0117	0.1161	0.111	0.002	0.1103	4.34E-06
Elapsed Time (sec)	36.71967	47.44284	51.81607	77.50712	66.96082	0.071493	119.4229	167.9235
Iterations=10000								
maximum MSE	0.0453	0.0165	0.0214	0.1177	0.1169	0.6716	0.1222	0.000415
Mean MSE	0.02098	0.008527	0.01472	0.117407	0.113667	0.138187	0.114267	9.45E-05
Minimum MSE	0.0023	0.0018	0.0118	0.1171	0.0967	0.0061	0.1088	2.95E-07
Elapsed Time (sec)	109.5225	90.37113	98.74841	2423.294	173.4092	0.063533	270.8297	333.301

Table 3: Performance comparison of MSE with different approaches for IRS (institute rating system)

Performance Measures	Model identification approach							
	Iterations=500							
	PSO	BBBC	ACO	EBP	RPROP	LM	ABC	PBBBC
maximum MSE	0.0351	0.0012	0.0091	0.0297	0.0294	0.9778	0.0288	1.04E-05
Mean MSE	0.024273	0.000113	0.005773	0.02836	0.02762	0.246773	0.025487	1.43E-06
Minimum MSE	0.0158	7.68E-07	0.0039	0.0268	0.0253	0.0059	0.0226	7.55E-10
Elapsed Time (sec)	1.074487	3.38884	8.066253	1.03858	1.40842	0.0265	10.8648	12.35939
Iterations=1000								
maximum MSE	0.0349	0.00022	0.0064	0.0289	0.0296	0.2956	0.0311	1.51E-05
Mean MSE	0.02206	5.22E-05	0.00474	0.027547	0.027747	0.095513	0.025453	2.31E-06
Minimum MSE	0.0129	3.41E-07	0.0032	0.0248	0.0258	0.0026	0.0212	1.14E-09
Elapsed Time (sec)	3.181193	6.235267	14.64955	2.082787	2.661413	0.024447	18.67363	25.13445
Iterations=2000								
maximum MSE	0.0336	0.000432	0.0082	0.0284	0.0286	0.6279	0.0328	1.88E-06
Mean MSE	0.022127	0.000125	0.00462	0.027173	0.027467	0.160533	0.02754	5.00E-07
Minimum MSE	0.0117	1.76E-11	0.003	0.026	0.0261	0.0035	0.0224	2.85E-09
Elapsed Time (sec)	7.302653	12.20574	31.57399	4.687187	3.586947	0.07166	29.31675	43.26845
Iterations=5000								
maximum MSE	0.0372	0.000364	0.0096	0.03	0.0295	1.0154	0.0328	1.21E-05
Mean MSE	0.02316	0.000114	0.005	0.027473	0.027453	0.196747	0.026173	1.40E-06
Minimum MSE	0.0103	9.54E-12	0.0026	0.0256	0.0238	0.0073	0.0236	2.73E-10
Elapsed Time (sec)	42.92591	30.21604	81.19537	11.85265	11.65764	0.065373	65.10712	122.7725
Iterations=10000								



maximum MSE	0.0322	0.000363	0.0072	0.0294	0.0285	0.3067	0.028	5.61E-06
Mean MSE	0.02026	6.94E-05	0.004693	0.02706	0.027087	0.083533	0.024633	9.14E-07
Minimum MSE	0.0095	2.49E-06	0.0034	0.024	0.0255	0.0013	0.0198	4.00E-10

AUTHORS PROFILE



Ashima Kalra , Gold Medalist in B Tech in Electronics from Kurukshetra University, Kurukshetra in 2003 .Received M.Tech degree from Punjab Technical University, Kapurthla (Punjab)in 2008 and pursuing PhD from Punjab Technical University , Kapurthla (Punjab)in the field of soft Computing. She has published more than 25 papers in reputed journals and 3 book chapters in Springer series. Her research activities include designing model identification using neural networks, fuzzy systems, supervised learning .machine learning. Lifetime IEEE member and has been serving as a fellow member of IEEE Delhi section, India. Organized 3 IEEE international conferences as a chairperson.



Prof. (Dr.) Shakti Kumar did his MS (Electronics & Control) from BITS, Pilani, India and Ph.D. (Electronics & Computer Engg.) from National Institute of Technology, Kurukshetra, India. Presently, he is Director, Panipat Institute of engineering Technology, Panipat, Haryana , India. His research areas include Communication Networks, Soft Computing and Intelligent Systems. He has published more than 200 research articles in various Journals and Conferences and 7 book chapters. He has organized 7 international and 6 national conferences in the areas of intelligent systems and networks.



Dr. Sukhbir Singh Walia has done his B.Tech. from Guru Nanak Dev University Amritsar, M.Tech & Ph.D. from Punjab Technical University (PTU), Kapurthla (Punjab) India. His research areas include Soft Computing, Education Management, Mobile Communication, and Wireless Sensor Networks. He has published around 28 research articles in various Journals and Conferences. Currently he is Director, International Resource Centre in Human Values and Professional Ethics at PTU