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 distributed 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.

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:

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

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 \( k_1, k_2 \ldots \) 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.
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:-

```
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;
      \[ x_c = \frac{\sum_{i=1}^{np} \frac{1}{f_i} x_i}{\sum_{i=1}^{np} \frac{1}{f_i}} \]  
      Where \( x_i \) = position of the CoM;
      \( f_i \) = 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;
      \[ x_{new} = x_c + l(rand)/k \]  
  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 */
/* Big Crunch 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 gbest 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

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_{i} = k_{i} 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)
   \[ NN_{1\text{max}} = \text{NDP}/3 \]
3. forNN_{1} = 2: NN_{1\text{max}}
   (number of neuron in the 1s hidden layer)
4. for NN_{2} = 2 : NN_{2\text{max}}
   (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 M SE) 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 (NN2 = 2 : NN2max loop)
24. end (NN1 = 2: NN1max loop)
ANN Model Identification: Parallel Big Bang Big Crunch Algorithm

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 multi-layer ANN system. x₁, x₂, x₃, … xn are the “n” input variables for the system, N₁, N₂, N₃, … Nₙ are the input layer neurons.

The net input to the system can be represented as follows,

\[ \text{net}_{in} = N_0 + \sum_{i=0}^{N} W_i X_i \]  \hspace{1cm} (4)

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

\[ Y = 1/1 + e^{-(\text{net}_{in})} \]  \hspace{1cm} (5)

The W₀,W₁,W₂, …, Wₙ, W₀N₁k₁, W₁N₁k₁, W₂N₁k₁, …, WₙN₁k₁ 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 (k₁, k₂, …, kᵢ)

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 k₁, k₂, …, kᵢ 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.

<table>
<thead>
<tr>
<th>W₁</th>
<th>W₂</th>
<th>…</th>
<th>Wₙ</th>
<th>W₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td>n+1</td>
</tr>
</tbody>
</table>

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

The size of individual can be computed as given below:

\[ S = N_I + 1 = n+1 \text{ if } NH = 0 \]  \hspace{1cm} (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:

\[ S = k_1 (n+2) + 1 \text{ if } NH = 1 \]  \hspace{1cm} (7)

Fig.4: Individual for one hidden layer ANN

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.

| W₀N₁ = Weights between input neuron and bias |
| W₀N₁k₁ = Weights between bias and first hidden neuron |
| W₀N₁ = Weights between p₀ neuron of hidden layer 1 and output node |
| W₁N₁1 = Weights between j₀ neuron of input layer and j₁ neuron of first hidden layer |
| W₁N₁1 = Weights between j₀ neuron of first hidden layer and k₁ neuron of second hidden layer |
| W₁N₁1 = Weights between k₁ neuron of second hidden layer and output node |
| W₁N₁ = Weights between j₁ neuron of second hidden layer and output node |
| W₁N₁1 = Weights between k₁ neuron of second hidden layer and output node |

W₉ = Weight between bias point and output node

W₀N₁ = Weights between 0 neuron of input layer and 1 neuron of first hidden layer
W₀N₁ = Weights between 1 neuron of first hidden layer and 2 neuron of second hidden layer
W₀N₁ = Weights between 2 neuron of second hidden layer and output node
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 been 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 layers 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, PBBC 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 $\times$ 0.6 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 2.95e-0.7 while BBBC achieved minimum MSE of 0.0018. Also in the case for IRS model identification, PBBBC identified IRS model with minimum MSE of 4e-10 in 56.06 sec while BBBC achieved minimum MSE of 2.49e-0.6 for 10000 iterations. This became apparent from the observations that as the number of iterations was increased to 10000 for training, PBBC 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 the out of the seven algorithms that we have considered, PBBBC was found to offer best performance with minimum MSE of all the approaches.

<table>
<thead>
<tr>
<th>WN11N21</th>
<th>WN12N21</th>
<th>---</th>
<th>WN1k1N2</th>
<th>W10N21</th>
</tr>
</thead>
<tbody>
<tr>
<td>WN11N22</td>
<td>WN12N22</td>
<td>---</td>
<td>WN1k1N2</td>
<td>W10 N22</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>WN1k2</td>
<td>WN12N2k</td>
<td>---</td>
<td>WNk1N2k</td>
<td>W10N2k</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**Fig.5** : Individual for Two Hidden Layer ANN

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

$$S = k_1 (n+1) + k_2 (k_1+2)+ 1$$ if $NH=2$

Equation 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

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.
REFERENCES


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

<table>
<thead>
<tr>
<th>Performance Measures</th>
<th>PSO</th>
<th>BBBC</th>
<th>ACO</th>
<th>EBP</th>
<th>RPROP</th>
<th>LM</th>
<th>ABC</th>
<th>PBBBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations=500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0401</td>
<td>0.0154</td>
<td>0.0193</td>
<td>0.1178</td>
<td>0.1171</td>
<td>1.0415</td>
<td>0.1191</td>
<td>0.0015</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.018933</td>
<td>0.005422</td>
<td>0.014427</td>
<td>0.11756</td>
<td>0.1164</td>
<td>0.41166</td>
<td>0.113933</td>
<td>0.000324</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0046</td>
<td>0.000727</td>
<td>0.0118</td>
<td>0.117</td>
<td>0.1105</td>
<td>0.0212</td>
<td>0.1098</td>
<td>2.01E-09</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>1.735387</td>
<td>4.571813</td>
<td>5.940047</td>
<td>9.48534</td>
<td>8.492327</td>
<td>0.070707</td>
<td>16.14322</td>
<td>16.97283</td>
</tr>
<tr>
<td>Iterations=1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0356</td>
<td>0.0224</td>
<td>0.0157</td>
<td>0.1177</td>
<td>0.117</td>
<td>0.2383</td>
<td>0.1182</td>
<td>0.0011</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.017413</td>
<td>0.0009638</td>
<td>0.01326</td>
<td>0.11742</td>
<td>0.11546</td>
<td>0.08066</td>
<td>0.1127</td>
<td>0.000264</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0059</td>
<td>0.000666</td>
<td>0.0114</td>
<td>0.1117</td>
<td>0.1043</td>
<td>0.0091</td>
<td>0.1077</td>
<td>7.32E-06</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>4.24708</td>
<td>9.248847</td>
<td>10.54223</td>
<td>16.37903</td>
<td>14.74454</td>
<td>0.023947</td>
<td>37.54106</td>
<td>33.91405</td>
</tr>
<tr>
<td>Iterations=2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0418</td>
<td>0.0207</td>
<td>0.0208</td>
<td>0.1176</td>
<td>0.117</td>
<td>1.1595</td>
<td>0.1179</td>
<td>0.0017</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.01432</td>
<td>0.006124</td>
<td>0.014293</td>
<td>0.117293</td>
<td>0.116327</td>
<td>0.231733</td>
<td>0.113293</td>
<td>0.000211</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0029</td>
<td>1.99E-11</td>
<td>0.0126</td>
<td>0.117</td>
<td>0.1113</td>
<td>0.0093</td>
<td>0.1086</td>
<td>1.16E-06</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>7.27198</td>
<td>17.55212</td>
<td>23.21797</td>
<td>31.49277</td>
<td>26.62006</td>
<td>0.026447</td>
<td>44.56536</td>
<td>64.61417</td>
</tr>
<tr>
<td>Iterations=5000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0453</td>
<td>0.0165</td>
<td>0.0214</td>
<td>0.1177</td>
<td>0.1169</td>
<td>0.6716</td>
<td>0.1222</td>
<td>0.000415</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.02098</td>
<td>0.008527</td>
<td>0.01472</td>
<td>0.117407</td>
<td>0.113667</td>
<td>0.138187</td>
<td>0.114267</td>
<td>9.45E-05</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0023</td>
<td>0.0018</td>
<td>0.0118</td>
<td>0.1171</td>
<td>0.0967</td>
<td>0.0061</td>
<td>0.1088</td>
<td>2.95E-07</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>109.5225</td>
<td>90.37113</td>
<td>98.74841</td>
<td>243.2942</td>
<td>173.4092</td>
<td>0.063533</td>
<td>270.8297</td>
<td>333.301</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Performance Measures</th>
<th>PSO</th>
<th>BBBC</th>
<th>ACO</th>
<th>EBP</th>
<th>RPROP</th>
<th>LM</th>
<th>ABC</th>
<th>PBBBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations=500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0351</td>
<td>0.0012</td>
<td>0.0091</td>
<td>0.0297</td>
<td>0.0294</td>
<td>0.9778</td>
<td>0.0288</td>
<td>1.04E-05</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.024273</td>
<td>0.000113</td>
<td>0.005773</td>
<td>0.02836</td>
<td>0.02762</td>
<td>0.246773</td>
<td>0.025457</td>
<td>1.43E-06</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0158</td>
<td>7.68E-07</td>
<td>0.0039</td>
<td>0.0268</td>
<td>0.0253</td>
<td>0.0059</td>
<td>0.0226</td>
<td>7.5E-10</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>1.074487</td>
<td>3.38884</td>
<td>8.066253</td>
<td>1.03858</td>
<td>1.40842</td>
<td>0.0265</td>
<td>10.8648</td>
<td>12.35939</td>
</tr>
<tr>
<td>Iterations=1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0349</td>
<td>0.00022</td>
<td>0.0064</td>
<td>0.0289</td>
<td>0.0296</td>
<td>0.2956</td>
<td>0.0311</td>
<td>1.51E-05</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.02206</td>
<td>5.22E-05</td>
<td>0.00474</td>
<td>0.027547</td>
<td>0.027747</td>
<td>0.095513</td>
<td>0.025453</td>
<td>2.31E-06</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0129</td>
<td>3.41E-07</td>
<td>0.0032</td>
<td>0.0248</td>
<td>0.0258</td>
<td>0.0026</td>
<td>0.0212</td>
<td>1.14E-09</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>3.181193</td>
<td>6.235267</td>
<td>14.64955</td>
<td>2.082787</td>
<td>2.661413</td>
<td>0.024447</td>
<td>18.67363</td>
<td>25.13445</td>
</tr>
<tr>
<td>Iterations=2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0336</td>
<td>0.000432</td>
<td>0.0082</td>
<td>0.0284</td>
<td>0.0286</td>
<td>0.6279</td>
<td>0.0328</td>
<td>1.88E-06</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.022127</td>
<td>0.000125</td>
<td>0.00462</td>
<td>0.027173</td>
<td>0.027467</td>
<td>0.160533</td>
<td>0.02754</td>
<td>5.00E-07</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0117</td>
<td>1.76E-11</td>
<td>0.003</td>
<td>0.026</td>
<td>0.0261</td>
<td>0.0035</td>
<td>0.0224</td>
<td>2.85E-09</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>7.302653</td>
<td>12.20574</td>
<td>31.57399</td>
<td>4.687187</td>
<td>3.586947</td>
<td>0.07166</td>
<td>29.31675</td>
<td>43.26845</td>
</tr>
<tr>
<td>Iterations=5000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximum MSE</td>
<td>0.0372</td>
<td>0.003064</td>
<td>0.0096</td>
<td>0.0295</td>
<td>0.03</td>
<td>1.0154</td>
<td>0.0328</td>
<td>1.21E-05</td>
</tr>
<tr>
<td>Mean MSE</td>
<td>0.02316</td>
<td>0.000114</td>
<td>0.005</td>
<td>0.027473</td>
<td>0.027453</td>
<td>0.196747</td>
<td>0.026173</td>
<td>1.40E-06</td>
</tr>
<tr>
<td>Minimum MSE</td>
<td>0.0103</td>
<td>9.54E-12</td>
<td>0.0026</td>
<td>0.0256</td>
<td>0.0238</td>
<td>0.0073</td>
<td>0.0236</td>
<td>2.73E-10</td>
</tr>
<tr>
<td>Elapsed Time (sec)</td>
<td>42.92591</td>
<td>30.21604</td>
<td>81.19537</td>
<td>11.85265</td>
<td>11.65764</td>
<td>0.065373</td>
<td>65.10712</td>
<td>122.7725</td>
</tr>
</tbody>
</table>

Retrieval Number: 110520789S19/19E0BEIESP
DOI: 10.35940/ijitee.I1052.0789S19

Published By:
Blue Eyes Intelligence Engineering & Sciences Publication
<table>
<thead>
<tr>
<th></th>
<th>Maximum MSE</th>
<th>Mean MSE</th>
<th>Minimum MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0322</td>
<td>0.000363</td>
<td>0.0095</td>
</tr>
<tr>
<td></td>
<td>0.0072</td>
<td>6.94E-05</td>
<td>2.49E-06</td>
</tr>
<tr>
<td></td>
<td>0.0294</td>
<td>0.004693</td>
<td>0.0034</td>
</tr>
<tr>
<td></td>
<td>0.0285</td>
<td>0.02706</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>0.3067</td>
<td>0.027087</td>
<td>0.0255</td>
</tr>
<tr>
<td></td>
<td>0.028</td>
<td>0.083533</td>
<td>0.0013</td>
</tr>
<tr>
<td></td>
<td>5.61E-06</td>
<td>0.024633</td>
<td>0.0198</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.14E-07</td>
<td>4.00E-10</td>
</tr>
</tbody>
</table>

**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, Kapurthala (Punjab) in 2008 and pursuing PhD from Punjab Technical University, Kapurthala (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), Kapurthala (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.