BRAIN. Broad Research in Artificial Intelligence and Neuroscience
ISSN: 2068-0473 | e-ISSN: 2067-3957
Covered in: Web of Science (WOS); PubMed.gov; IndexCopernicus; The Linguist List; Google Academic; Ulrichs; getCITED; Genamics JournalSeek; J-Gate; SHERPA/RoMEO; Dayang Journal System; Public Knowledge Project; BIUM; NewJour; ArticleReach Direct; Link+; CSB; CiteSeerX; Socolar; KVK; WorldCat; CrossRef; Ideas RePeC; Econpapers; Socionet.

2020, Volume 11, Issue 2, pages: 82-103 | https://doi.org/10.18662/brain/11.2/76

Biogeography-Based Optimization for Weight Optimization in Elman Neural Network Compared with Meta-Heuristics Methods

Habib DHAHRI[1,2]

[1] College of Applied Computer Sciences (CACS), King Saud University, Riyadh, Saudi Arabia;

[2] Faculty of Sciences and Techniques of Sidi Bouzid, University of Kairouan, Tunisia.

Abstract: *In this paper, we present a learning algorithm for the Elman Recurrent Neural Network (ERNN) based on Biogeography-Based Optimization (BBO). The proposed algorithm computes the weights, initials inputs of the context units and self-feedback coefficient of the Elman network. The method applied for four benchmark problems: Mackey Glass and Lorentz equations, which produce chaotic time series, and to real life classification; iris and Breast Cancer datasets. Numerical experimental results show improvement of the performance of the proposed algorithm in terms of accuracy and MSE eror over many heuristic algorithms.*

Keywords: *Biogeography-Based Optimization; Time series predictions; classification.*

How to cite: Dhahri, H. (2020). Biogeography-Based Optimization for Weight Optimization in Elman Neural Network Compared with Meta-Heuristics Methods. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 11*(2), 82-103. https://doi.org/10.18662/brain/11.2/76

1. Introduction

Among the varied types of Neural Nets, Recurrent Neural Network (RNN) is able to forecast the most accurate results (Senjyu et al., 2006). In case of the RNN, the fixed back-connections save a copy of the previous values of the hidden units in the context units. Actually, the RNN was applied in various applications, such as, pattern recognition (Hori et al., 2016), robotic control (Sharma et al., 2016), and genetic data prediction (Baldi & Pollastri, 2003). Therefore, RNN has been widely used as a tool in data classification (Nawi et al., 2015) and time series prediction (Chandra, 2015; Koskela et al., 1996).

In fact, there are two types of RNN; a fully RNN used by Kechriotis Zervas and Manolakos (1994), and a Partially RNN used by Robinson and Fallside (1991). Concerning the fully RNN, each unit of the NN is connected to every other unit. BAM (Bidirectional Associative Memory) (Kosko, 1988) and Hopfield (1982) are examples of FRNN. The recurrent networks are still complicated in dealing with complex problems. While the partially training is faster compared to globally recurrent NNs. Recent researches exhibit that PRNN can be a highly effective forecasting method in fields like Electricity Consumption and Wind Speed (Cao, Ewing, & Thompson, 2012; Marvuglia & Messineo, 2012). PRNN offer both features. This topology is considered for non-linear applications and also to modulate time series data (Müller-Navarra, Lessmann & Voß 2015). Elman Neural Network (ENN) (Elman, 1990) is the most widely used PRNN architecture. Its structure chosen over the Jordan network (Jordan, 1997) thanks to its hidden layer being wider than the output layer. This wider layer allows more values to be feedback to the input, consequently authorizing more information to be available to the network (Venayagamoorthy, Welch & Ruffing, 2009). Optimisation can be performed by metaheuristic methods (Yao & Kim, 2014). This class of network could be trained with heuristics algorithms because of the inconveniences gradient-based algorithms such as suffering from the local minima. Generally, it was three tasks for RNN optimization; weight and bias optimization, architecture optimisation and parameter gradient optimization. This work concerns the first optimization task in order to find the minimum training error.

A metaheuristic is officially known as an iterative generation process, which guides a subordinate heuristic by combining rationally different concepts for exploring and exploiting the search space, and learning the strategies which are used for organizing information to find efficiently nearoptimal solutions (Osman & Laporte, 1996). In general, the Nature Inspired Algorithms is mainly classified in three major’s groups: Evolutionary Algorithms, Ecology-Based Algorithms and Bio-Inspired Algorithm.

The Evolutionary computation algorithms are based on biological Darwinian evolution to design a solution and it include the Genetic Algorithms (Pham & Karaboga, 1999), the Differential Evolution (Storn & Price, 1997) and Evolutionary Strategies (Kawada, Yamamoto, & Mada, 2004).

The Ecology-Based Algorithms is based on the ecosystems to solve the problem. This group include Biogeography-Based Optimization (BBO) and Invasive Weed colony Optimization (IWO) algorithms (Mehrabian & Lucas, 2006).

Bio-Inspired Algorithm are inspired from the interactions between / with the species. Based on the behaviours of species, different algorithms have been invented. This category includes Particle Swarm Optimization (Kennedy & Eberhart, 1995), Artificial Bee Colony (Karaboga & Basturk, 2007), Fish Swarm Algorithm (Li, Shao & Qian, 2002), Firefly Algorithm (Yang, 2009), Bacterial Foraging Algorithm (Passino, 2002), Ant Colony Optimization (Zhipeng et al., 2012), Cuckoo Search Optimization (Yang, & Deb, 2009), Fruit fly Algorithm (Pan, 2012) and Bat Algorithm (Yang, 2010).

Despite the generation of an evolving solution is common for the most of approaches, they have their distinctive way to exploit and explore the search space of the problem.

BBO algorithm considered one of the powerful algorithms due to its exploration and exploitation strategies depend on the two BBO operators; migration and mutation.

The main objective of the mutation operator is to enhance the diversity of the population. Based on this operator, the solution with low HSI can improved as well as the solutions with high HSI. Consequently, the probabilistic operator can be applied for any candidate solution. Unlike the evolutionary algorithms, at each generation the solutions are the combination of the parents ‘solutions and their offspring.

The rate of emigrations is evolved from one generation to another. The habitat with a high emigration rate can share the information with the one with low emigration rate.

Several techniques have been used to optimize Elman RNN performance such as Genetic Algorithm (GA) (Pham & Karaboga, 1999), Particle Swarm Optimization PSO (Xiao, Venayagamoorthy, & Corzine, 2007), Ant Colony Optimization (ACO) (Zhipeng et al., 2012), Evolutionary

Strategies (ES) (Kawada, Yamamoto, & Mada, 2004) and Population Based Incremental Learning (PBIL) (Palafox & Iba, 2012).

Both BBO and GA are evolutionary algorithm, but each of them has a specific characteristic. In (Simon et al., 2011), the authors claim that BBO and GA have the same chances of finding the optimal solution, but BBO able to conserve this optimum once it found. Thanks to immigration rate, which help to retain good solutions in the population and reduces with fitness. In addition, the use of mutation for each individual in a population enhances the exploitation capability of BBO compared to GA, which applies a single mutation for the entire population. In fact, (Simon, Ergezer & Du, 2009) prove that the advantage of BBO compared to GA is more marked with larger and higher dimensional problems.

PSO is based on the behaviour of birds seeking their feeds while BBO uses the principle of migration to the islands, despite this difference, these two algorithms have similar characteristics as the sharing of information between populations but the strength of BBO is that it retains solutions from one iteration to another and ameliorate the solutions by the migration mechanism. BBO uses the mutation mechanism that represents a strong point compared to the Swarm Intelligence techniques (PSO, ACO).

In (Hordri, Yuhaniz, & Nasien, 2013), the author compares the performance of BBO, PSO and GA, treating fourteen benchmark functions, and finds that BBO makes a success in the convergence time and a great performance in avoiding local minima.

In this work, the used BBO algorithm is for optimizing the weight of the ENN. We also examine the advantages of this algorithm on the training ENN for the classification and prediction of benchmark problems. The performance of our algorithm will be compared also with other well-known heuristics algorithms.

The results indicate that BBO algorithm proves its effectiveness on training Elman Neural Network.

The used methodology in our analysis is as follows: Section II presents a broad description of Elman Neural Network (ENN); Section III explains the basic concept of the BBO algorithm and its use for the design ENN. The experimental results will be given in the fourth Section. Finally, the last Section gives the conclusions.

2. Elman Neural Network

Elman Neural Network (ENN) proposed in (Elman, 1990) designed with the input layer, the hidden layer, the recurrent link known as context layer and the output layer. It is based on the context layer that contains a copy of the hidden layer, which are subsequently used as input. The main advantage of this layer is to store the information in the hidden layer and to preserve the memory, which gives more information entered as input. As is well known, this simple Recurrent Network has many advantages, such as faster convergence, more accurate mapping and nonlinear prediction capability (Chandra, 2015).

Let assume x_i($i = 1 .. m$) the input vector, y_k the output of ENN and z_j($j = 1 .. n$) the output of hidden layer. b_j and b_k are the biases in the hidden layer and the output layer respectively. u_j denotes the context layer neurons. w_{ij} is the weight that connects between the input nodes (i) and the hidden nodes (j). c_j denotes the weight that connects between the hidden nodes and the context nodes. v_{jk} is the weight that connects the node j in the hidden layer to the output nodes.

$$net_j(t) = \sum_{i=1}^{m} w_{ij}\, x_i\, (t-1) + \sum_{j=1}^{n} c_j\, u_j\,(t) \qquad (1)$$

u_j is the context node value, calculated by (2)

$$u_j(t) = z_j(t-1) \qquad (2)$$

The activation function selected in hidden layer is the sigmoid function defined as follows:

$$z_j(t) = f(net_j(t)) \qquad (3)$$

$$f(x) = 1/(1+e^{-x})$$

The output of ENN is given as follows:

$$net_k(t) = \sum_{j=1}^{n} v_{jk}\, z_j\,(t) + b_k \qquad (4)$$

$$y_k(t) = f(net_k(t))$$

The architecture of ENN is presented in Figure 1.

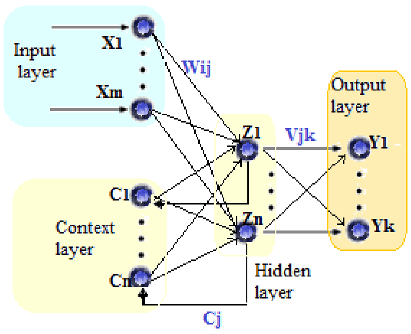

Fig. 1. Elman Neural Network architecture.

3. BBO Trained Elman RNN

Biogeography-Based Optimization (BBO) proposed in (Simon, 2008) is an Evolutionary Algorithm (EA), which is based on migration and immigration to the islands. Recently BBO algorithm has proved its efficiency and success to supply global optimal solution in different problems such as (Ma et al., 2015; Mirjalili, Mirjalili, & Lewis, 2014; Rodan, Faris & Alqatawna, 2016; Zhang et al., 2019).

The general idea of this algorithm is to get the relation between the species by emigration, immigration and mutation. Similarly, to GA, BBO employs the habitats as chromosome. Each habitat is assigned by a vector of habitants (genes in a GA), which are used as changeable variables to optimize the process problem. To achieve this objective, BBO offers Habitat Suitability Index (HSI) as performance index. High HSI represents a good solution, so a large number of habitants, which are more likely to immigrate to other islands with low HIS. Those poor solutions have a low HSI and a higher immigration rate. The BBO algorithm is characterised by emigration, immigration and mutation rates.

The time complexity analysis of BBO depends on numbers of used resources. Based on the O-notation the time complexity expressed as function describing the asymptotic upper bound. The big O notation is defined as follows: $O(g(n) = \{f(n), \exists\, c, n_0, 0 \leq f(n) \leq cg(n) \text{ for } n \geq n_0\}$

The computational complexity of the BBO algorithm depends on the number of species (habitats), the number of generations, the migration (selection of the solutions) and mutation operator and finding the best solution. Therefore, at each iteration, the computational complexity of BBO is as follows

O(BBO) = O(O(Initialization) + O(migration) + O(mutation) + O (best habitat))

The time complexity of Initialization is of O(nmd) where d is the dimension of habitats, m is the number of habitants and n is the number of habitats but in our implementation d is one dimension. In the migration operation, the roulette wheel selection is used to select of a candidate solution from which to immigrate so the computation complexity of migration is of O(mn^2). For each habitant, the mutation operation has been applied, thus the computational complexity of the mutation is of O(nm). The selection of the best solution is based on the fitness value of each habitat. Consequently, the computational complexity of best habitat is of O(n^2). Therefore, the final computational complexity of the proposed method is as follows: O(BBO)= O(g(mn^2 +mn+ n^2) where g is the number of generations. In the expression of time complexity, all variables with zero space dimension are ignored because to their constant complexity time.

Generally, the main steps of BBO in (Simon, Ergezer & Du, 2009) defined as follows:

1- *Initialize habitats values and BBO parameters.*
2- *Calculate HSI for each island.*
3- *Update the immigration, emigration and mutation rates.*
4- *Modify habitats according to the last rates.*
5- *Mutate some habitants of different habitats, which selected randomly.*
6- *Save the best habitats as elites.*
7- *Replace the worst habitats with the elites.*
8- *If satisfying condition, terminate, else, repeat from step 2.*

The BBO algorithm is used to an ENN to finding the best combination of biases and weights based on two phases:

i. Encoding strategy: represent weights and biases in the proper scheme (habitats) for BBO.

ii. Fitness evaluation: Calculate HSI as fitness function defined by the error of ENN to evaluate habitats performance.

Encoding scheme of ENN trained by BBO

The optimization algorithm evolves the parameter of ENNs. Thus, in BBO, a structure habitat is encoded based on vector scheme which is defined as follows ENN = [W_{12} W_{32} W_{24} b_1 b_2]

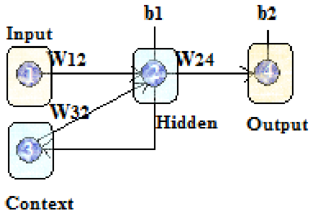

Fig. 2. ENN with the structure of 1-1-1.

In Figure 2 example, each layer (input, hidden and output) is composed by only one node. W12 denotes the weight between input node and hidden node. W32 denotes the weight between context node and hidden node. W24 denotes the weight between hidden node and output node. b1 and b2 are the biases value of hidden node and output node respectively. So the encoding vector scheme contains the list of weights between input and hidden layer, the list of weights between the context and hidden layer, the list of weights between hidden and output layer and biases values.

Fitness function

The fact remains that training RNNs is a challenging optimization problem. So each training set should be evaluated by a fitness measure. Thus, for each individual, the HSI function should be assigned depending to the desired optimization. In this work, the Mean Square Error (MSE) is used to compute the output error as HSI function:

$$\mathbf{HSI(ENN)} = \mathbf{MSE(ENN)} = \sum_{K=1}^{S} \frac{\sum_{K=1}^{m}\left(o_i^k - d_i^k\right)^z}{S} \qquad (5)$$

Where S is the number of training samples, m denotes the number of output, o_i^k is the obtained output of the *i*th input unit and d denotes the desired output. In this study, the proposed algorithm aims to minimize network performance. The computational complexity can have written as follows:

$$O(\text{BBO-ENN}) = O\,(\,i(\,x(z+y) + hH2 + Hh + H2\,)) \qquad (6)$$

Where i is the number of iterations, x is the number of input training sets, z and y are the number of nodes in the hidden layer and the output layer respectively, h is the number of habitants (weights and biases) and H is the number of habitats (ENNs). Denotes that H2 represent the elitism complexity, Hh is the mutation complexity, hH2 is the migration complexity and x(z+y) is our ENN complexity. The proposed model BBO-ENN given in Figure 3.

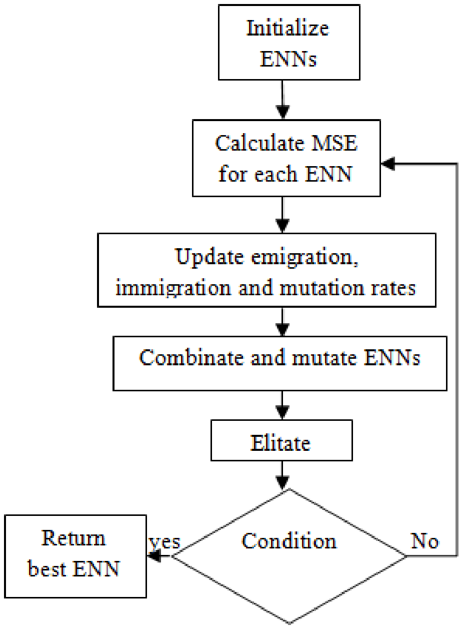
Initialize ENNs
Calculate MSE for each ENN
Update emigration, immigration and mutation rates
Combinate and mutate ENNs
Elitate
Condition
yes
No
Return best ENN

Fig. 3. BBO-ENN model.

In fact, the first step of the proposed model is to generate a random set of ENNs as habitats and initialize randomly weights and biases values as habitants. The second step is to calculate MSE for each ENN by Eq. (1) to distinguish between the best and the worst habitat. The third step is to update and modify the emigration, immigration and mutation rates. After having an idea about the good and poor solutions, we must make the combination between different islands then select some habitats to mutate various habitants. The last step is to select the good solution as elitism for future generations. These steps repeated until satisfaction of the stop condition, which can be the number of iterations or the error rate. Figure 4 presents a conceptual picture of the BBO-ENN.

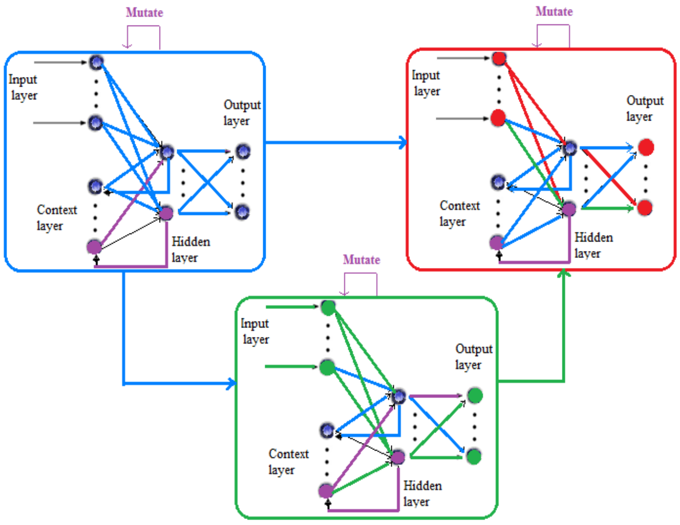

Fig. 4. Conceptual picture of BBO-ENN model.

As seen in this figure, there are three habitats (ENNs). Habitat 1 provides a lower HIS, highest emigration and lower immigration. It presents the good solution, so it is more likely to share weights and biases with Habitat 2 and Habitat 3. Whereas, Habitat 2 provides a highest HIS, llower emigration and a highest immigration, it presents the poor solution, then, it is more likely to accept shared features (weights and biases) from Habitat 1 and Habitat 3.

Theoretically, the proposed BBO-ENN model can improve the training phase according to the various advantages of emigration and immigration rates, which are evolutionary mechanisms for each habitat, which encourages exploration. Thus, BBO forced not to fall in local optima.

In addition, thanks to the migration of the better weights/biases towards the worse ENNs, the error rate MSE (HSI) of ENN (habitat) can be improved during the generation. Not to forget that the mutation mechanism helps each habitat to show the various exploitations mechanisms. Finally, elitism phase helps the proposed method to keep some of the best solutions, which are never lost.

After having an idea about the theoretical functionality of the proposed method, in the following section, we will see the results of the practical handling, followed by a comparative study between different algorithms.

4. Experiments

To verify the performance of BBO algorithm for training Elman NN, it's necessary to compare it with PSO, GA, ACO, ES and PBIL over four benchmark problems: Breast Cancer (Wolberg & Mangasarian, 1990), Iris (Fisher, 1936) for classification and Mackey and Glass (1977), and Lorenz Attractor (1963) for time series prediction.

The classification datasets based on two performance criteria: (a) MSE value and (b) classification accuracy.

In fact, the increase in population size and the number of iterations, could improve the performance of the algorithms, but in this work, we are interested on comparing the six algorithms during a fixed number of iterations. Thus, we are not forced to find the best parameters. Just use the same network parameters such as number of nodes, the value of weight initialization and size of population. In this architecture, the log-sigmoid is used as activation function.

For all algorithms, we initialise the habitat randomly in the range [-10, 10]. The population size is 200 for each dataset. For all the experiments, the performance was computed after 30 executed runs with 300 generations for all the used methods.

According to (Shamsuddin, 2004) there is no standard rule for determining the suitable number of hidden nodes. We fixed it based on this theorem "One hidden layer and 2N + 1 hidden neuron sufficient for N inputs". Table 1 show the different number of input, hidden and output node of each datasets.

Table 1. Structure of each datasets

Classification datasets	**Number of input nodes**	**Number of hidden nodes**	**Number of output nodes**
Iris	4	9	3
Breast Cancer	9	19	1
Mackey Glass	4	9	1
Lorenz	3	7	1

The initial parameters of meta-heuristics algorithms fixed in table 2; it shows various initialization settings for the optimization methods. All the parameters are chosen based on literature used value.

Table 2. Parameters settings of algorithms

Method	**Parameter**	**Value**
BBO	Max immigration/emigration Mutation immigration bounds per gene	1 0.005 [0,1]
ACO	Initial pheromone (τ0) Pheromone update constant (Q) Pheromone constant (q0) Global Pheromone decay rate (pg) Local Pheromone decay rate (pt) Pheromone sensitivity (α) Visibility sensitivity (β)	1e -06 20 1 0.9 0.5 1 5
GA	Selection mechanism Crossover probability Mutation probability	Roulette wheel 1 0.01
PSO	Cognitive constant (c1) Social constant (c2) Inertia weight (w)	1 1 0.3
ES	Λ Σ	10 1
PBIL	Learning rate Elitism parameter Mutation	0.05 1 0.1

A. Breast Cancer

This dataset was obtained from the UCI Machine Learning Repository. This dataset contains 699 instances and 9 attributes with 458 benign and 241 malignant instances. The first 599 patterns are used for training phase, and remaining for testing. The outputs convergences of different algorithms are presented in Figure 5. Table 3 presents the experimental results of different algorithms.

Table 3. Experimental results for Breast Cancer dataset

Algorithms	MSE error	Accuracy
BBO	**0.0024175**	**99.99**
GA	0.0025149	98.45
PSO	0.0043705	94.50
ACO	0.0073633	76.25
ES	0.0062843	73.00
PBIL	0.032001	03.99

From the table 3, it can be seen that the MSE value for our BBO-ENN is less than PSO, GA, ACO, ES and PBIL algorithms, which demonstrates the efficacy of BBO-ENN for data classification. The proposed algorithm achieves the small MSE (**0.0024175**) and the highest accuracy with 99.99. Meanwhile, the other methods (PSO, ACO, ES, and PBIL) converge with large MSE and lower accuracy. Whereas, the MSE value of GA is closer to MSE BBO. As shown in Figure 5, the BBO technique has a faster and lower convergence curves among all the methods for Breast Cancer. From the simulation results, the BBO algorithm proves its superiority in terms of MSE and accuracy.

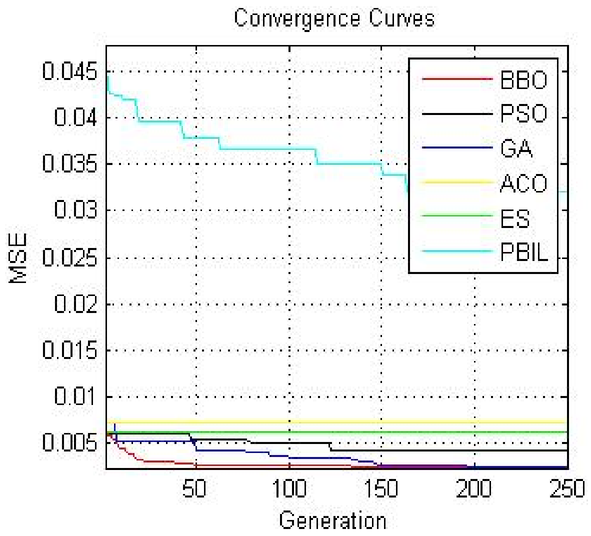

Fig. 5. Convergence Breast Cancer Elman RNN.

B. Iris dataset

The Iris Plants data set contains 150 samples and four attributes (sepal length, sepal width, petal length, petal width). It has actually, three major classes: Setosa, Versicolour and Virginica. In this experiment, we used four inputs, nine hidden nodes and three outputs . The first 150 patterns are selected for training phase, and remaining 150 for testing.

Table 4 presents the results of training algorithms. It illustrates the comparison between performances of BBO-ENN with GA, PSO, ACO, ES and PBIL algorithms. From table 4, it can be easier to show that the proposed algorithm achieves with a lower MSE (**0.017371)** and higher accuracy **(93.96)**

Table 4. Experimental results for Iris dataset

Algorithms	MSE error	Accuracy
BBO	**0.017371**	**93.96**
GA	0.029781	91.22
PSO	0.21166	66.53
ACO	0.40017	38.56
ES	0.30877	66.00
PBIL	0.1116	57.66

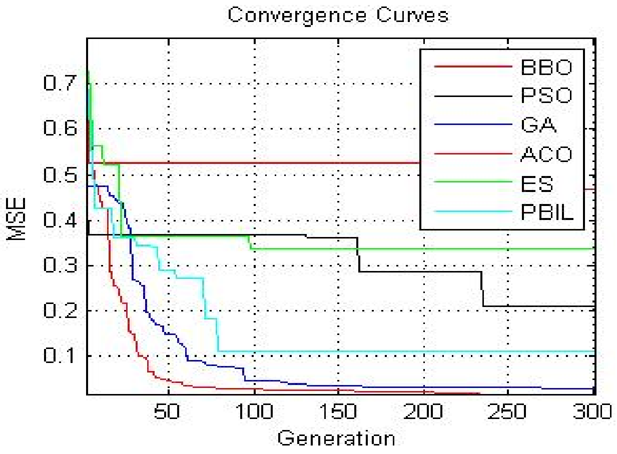

Fig. 6. Convergence Iris Elman RNN.

Figure 6 shows the convergence of each algorithm, and illustrates the success of BBO compared to the other methods. From these results, the BBO algorithm achieves with higher performance.

C. Mackey–Glass time series prediction

The Mackey-Glass time series prediction is defined using the following equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^c(t-\tau)} - bx(t) \qquad (7)$$

In our work, the input of ENN with four data points: x(t), x(t-6), x(t-12) and x(t-18). The output is defined in equation 8:

$$x(t+6) = f\left(x(t),\ x(t-6),\ x(t-12),\ x(t-18)\right) \qquad (8)$$

The first 500 samples are selected for training phase, and remaining 500 for testing. After 300 generations of the training process, the outputs convergences of different algorithms are presented in Figure 7. Table 5 shows the comparison of MSE error of the BBO-ENN to the other used meta-heuristics algorithms. In this experiment, the BBO and GA algorithm achieves with the smallest MSE error of **0.009702**. In some implementation, the MSE-GA equal to MSE-BBO. However, BBO-ENN is still promising in cases where the convergence to the best solution is faster than the other methods.

Table 5. Experimental results for Mackey-Glass dataset

Algorithms	**MSE training error**	**MSE Test error**
BBO	**0.00925**	**0.009702**
PSO	0.01043	0.071985
GA	0.01130	0.009851
ACO	0.04851	0.093598
ES	0.04748	0.082876
PBIL	0.02293	0.091885

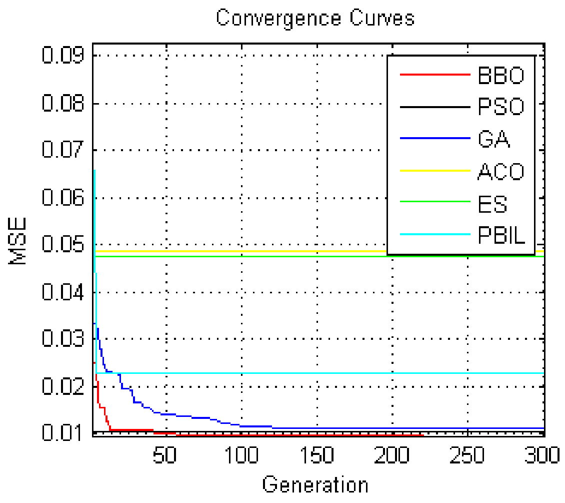

Fig. 7. Convergence Mackey-Glass Elman RNN.

D. Lorenz attractor

The Lorenz system was given by the following differential equations:

$$\begin{cases} \dfrac{dx(t)}{dt} = \alpha\left(y(t) - x(t)\right) \\ \dfrac{dy(t)}{dt} = \rho x(t) - y(t) - x(t)z(t) \\ \dfrac{dz(t)}{dt} = x(t)y(t) - \beta z(t) \end{cases} \tag{9}$$

Where σ, ρ and β are positive real parameters.

In these three equations, the component x denotes the used time series. In this work, the input of ENN is defined by x(t), x(t-1), x(t-2). The output is presented in equation 10:

$$x(t+1) = f\left(x(t), x(t-1), x(t-2)\right) \qquad (10)$$

The first 500 samples from 1000 simulation data points are chosen for training phase, and the remaining 500 for testing. The convergence curve of each algorithm summarized in Figure 8. Table 6 illustrates the MSE training error and MSE testing error of each algorithm. From this table, it can be seen that BBO achieves with a lower MSE error for both training and testing phase (0.14278, 0.241291). However, the other algorithms: PSO, GA, ACO, ES and PBIL have MSE of 0.21271, 0.48498, 1.27504, 0.29753, and 0.28753, quite larger than BBO. Similarly, Figure 8 represents the MSE convergence for Lorenz problem. This Figure demonstrates that the proposed BBO-ENN have better result than the other algorithms. The BBO-ENN shows again its efficiency for the prediction of Lorenz time series.

Table 6. Experimental results for Lorenz dataset

Algorithms	**MSE training error**	**MSE test error**
BBO	**0.14278**	**0.241291**
PSO	0.21271	0.253047
GA	0.48498	0.242105
ACO	1.27504	0.962869
ES	0.29753	0.293404
PBIL	0.28753	0.283424

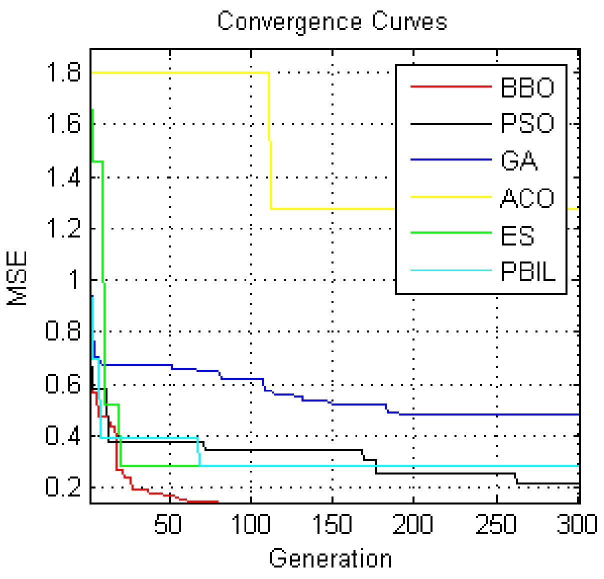

Fig. 8. Convergence Lorenz Elman RNN.

During the previous experiments, the BBO proves its good performances compared to the other applied algorithms. The obtained results can be explained based on the philosophy of the BBO technique over the other evolutionary algorithms. During the generation, the BBO solutions are maintained depending on their emigration rate. At each iteration, the BBO improves the habits by changing some features. The poor solutions can be improved from the good solutions by sharing theirs SIVs (attributes). However, in GA, ACO, PBIL techniques, the worse solutions are discarded from the populations and only the best candidate's solutions are maintained. Thus, the population evolves using the elite solutions. BBO also clearly similar to PSO and DE approach in maintaining solutions. The solution learns from theirs neighbours and evolves based on the movements of the around particles.

Conclusion

In this work, a Biogeography-Based Optimization (BBO) algorithm proposed to train Elman Neural Network (ENN) for four benchmark problems. The experiment results show that the BBO-ENN model can effectively classify the data such as Breast Cancer and Iris data sets. The method applied to Mackey Glass and Lorentz equations, which produce chaotic time series. Statistical results show that the proposed algorithm outperforms the GA, PSO, EA, ACO and PBIL algorithms. Performance and success of BBO-ENN is mainly due to the use of the Biogeography-Based Optimization (BBO) algorithm, which can successfully optimize the weight parameter of Elman Neural Network. BBO-ENN makes a success in the convergence time and a great performance in avoiding local minima. Although, BBO has shown a good performance when being applied to classification and time series prediction, BBO inherently lacks exploration ability to increase the diversity of habitats, which lead to slow down the convergence of the algorithm. The expansion of applying BBO algorithms in many types of problems open several research areas. One suggested research for the future work is to automating parameter tuning. Additional study is to apply the BBO algorithm for complicated problems.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research and RSSU at King Saud University for funding this work through research group no. RG-1438–071.

References

Baldi, P. & Pollastri, G. (2003). The Principled Design of Large-Scale Recursive Neural Network Architectures-DAG-RNNs and the Protein Structure Prediction Problem. *Journal of Machine Learning Research 4*(4), 575-602. https://doi.org/10.1162/153244304773936054

Cao, Q. Ewinga, B. T. & Thompson, M. A. (2012). Forecasting wind speed with recurrent neural networks. *European Journal of Operational Research, 221*(1), 148-154. https://doi.org/10.1016/j.ejor.2012.02.042

Chandra, R. (2015). Competition and Collaboration in Cooperative Coevolution of Elman Recurrent Neural Networks for Time-Series Prediction. *IEEE Trans Neural Netw Learn Syst, 26*(12), 3123-3136. https://doi.org/10.1109/TNNLS.2015.2404823

Elman, J. L. (1990). Finding structure in time. *Cognitive science, 4*(2), 179-211. https://doi.org/10.1016/0364-0213(90)90002-E

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of human genetics, 7*(2), 179-188. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America, 79*(8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554

Hordri, N. F., Yuhaniz, S. S. & Nasien, D. (2013). A Comparison Study of Biogeography based. Optimization for Optimization Problems. *International Journal of Advances in Soft Computing and its Applications, 5*(1), 1-16.

Hori, T., Hori, C., Watanabe, S. & Hershey, J. R. (2016). Minimum word error training of long short-term memory recurrent neural network language models for speech recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai*, 5990-5994. https://doi.org/10.1109/ICASSP.2016.7472827

Jordan. M. I. (1997). A parallel distributed processing approach. *Advances in Psychology, 121*, 471-495. https://doi.org/10.1016/S0166-4115(97)80111-2

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471. https://doi.org/10.1007/s10898-007-9149-x

Kawada, K., Yamamoto, T., & Mada, Y. (2004). A design of evolutionary recurrent neural-net based controllers for an inverted pendulum. *5th Asian Control Conference (IEEE Cat. No.04EX904), 3,* 1419-1422.

Kechriotis, G., Zervas, E. & Manolakos, E. S. (1994). Using recurrent neural networks for adaptive communication channel equalization. *IEEE Transactions on Neural Networks, 5*(2), 267-278. https://doi.org/10.1109/72.279190

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 4, 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Koskela, T., Lehtokangas, M., Saarinen, J. & Kaski, K. (1996). Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. *Proceedings of the World Congress on Neural Networks*, 491-496. https://pdfs.semanticscholar.org/82c8/e5d0cd4a7467f7f54ad823b2136b973eeb6e.pd

Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics, 18*(1), 49-60. https://doi.org/10.1109/21.87054.

Li, X. L., Shao, Z. J. & Qian, J. X. (2002). An Optimizing Method Based on Autonomous Animats: Fish-swarm Algorithm. *Systems Engineering - Theory & Practice, 22*(11), 32-38. https://doi.org/10.12011/1000-6788(2002)11-32

Lorenz, E. N. (1963). Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences, 20*, 130–141. https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

Ma, H., Fei, M., Simon, D., & Chen, Z. (2015). Biogeography-based optimization in noisy environments. *Transactions of the Institute of Measurement and Control, 37*(2), 190–204. https://doi.org/10.1177/0142331214537015

Mackey, M. C. & Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science, 197*(4300), 287-289. https://doi.org/10.1126/science.267326

Marvuglia, A. & Messineo, A. (2012). Using Recurrent Artificial Neural Networks to Forecast Household Electricity Consumption. *Energy Procedia, 14*, 45-55.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics, 1*(4), 355-366. https://doi.org/10.1016/j.ecoinf.2006.07.003

Mirjalili, S., Mirjalili, S. & Lewis, A. (2014). Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences, 269*, 188-209. https://doi.org/10.1016/j.ins.2014.01.038

Müller-Navarra, M., Lessmann, S. & Voß, S. (2015). Sales Forecasting with Partial Recurrent Neural Networks: Empirical Insights and Benchmarking Results. *48th Hawaii International Conference on System Sciences*, Kauai, HI, 1108-1116. https://doi.org/10.1109/HICSS.2015.135

Nawi, N. M. & Khan, A. & Rehman, G., Syed, M., Chiroma, H. & Herawan, T. (2014). Weight Optimization in Recurrent Neural Networks with Hybrid Metaheuristic Cuckoo Search Techniques for Data Classification. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2015/868375

Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations Research, 63*, 511–623. https://doi.org/10.1007/BF02125421

Palafox, L. & Iba, H. (2012). On the use of Population Based Incremental Learning to do Reverse Engineering on Gene Regulatory Networks. *IEEE Congress on Evolutionary Computation*, Brisbane, QLD, 1-8. https://doi.org/10.1109/CEC.2012.6256580

Pan, W-S. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, *26,* 69-74. https://doi.org/10.1016/j.knosys.2011.07.001

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine, 22*(3), 52-67. https://doi.org/10.1109/MCS.2002.1004010

Pham, D. T. & Karaboga, D. (1999). Training Elman and Jordan networks for system identification using genetic algorithms. *Artificial Intelligence in Engineering, 13*(2), 107-117. https://doi.org/10.1016/S0954-1810(98)00013-2

Robinson, T. & Fallside, F. (1991). A recurrent error propagation network speech recognition system. *Computer Speech & Language, 5*(3), 259-274. https://doi.org/10.1016/0885-2308(91)90010-N

Rodan, A., Faris, H. & Alqatawna, J. (2016). Optimizing Feedforward Neural Networks Using Biogeography Based Optimization for E-Mail Spam

Identification. *International Journal of Communications, Network and System Sciences, 9*, 19-28. https://doi.org/10.4236/ijcns.2016.91002

Senjyu, T., Yona, A., Urasaki, N. & Funabashi, T. (2006). Application of Recurrent Neural Network to Long-Term-Ahead Generating Power Forecasting for Wind Power Generator. *IEEE PES Power Systems Conference and Exposition*, Atlanta, GA, 1260-1265. https://doi.org/10.1109/PSCE.2006.296487

Shamsuddin, M. (2004). *Lecture note advanced artificial intelligence: Number of hidden neurons* [Unpublished Doctoral Thesis]. Universiti Teknologi Malaysia.

Sharmaa, R., Kumar, V., Gaur, P. & Mittal, A. P. (2016). An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload. *ISA Transactions, 62*, 258-267. https://doi.org/10.1016/j.isatra.2016.01.016

Simon, D. (2008). Biogeography-based optimization. *2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX* (702-713). https://doi.org/10.1109/TEVC.2008.919004

Simon, D., Ergezer, M. & Du, D. (2009). Population distributions in biogeography-based optimization algorithms with elitism. *2009 IEEE International Conference on Systems, Man and Cybernetics,* San Antonio, TX, 991-996. https://doi.org/10.1109/ICSMC.2009.5346058

Simon, D., Rarick, R., Ergezer, M. & Du, D. (2011). Analytical and numerical comparisons of biogeography-based optimization and genetic algorithms. *Information Sciences,181*(7), 1224-1248. https://doi.org/10.1016/j.ins.2010.12.006

Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization, 11*, 341–359. https://doi.org/10.1023/A:1008202821328

Venayagamoorthy, G. A., Welch, R. L. & Ruffing, S. M. (2009). Comparison of Feedforward and Feedback Neural Network. Architectures for Short Term Wind Speed Prediction. *IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks* pp. 3141–3146. https://doi.org/10.1109/IJCNN.2009.5179034

Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences of the United States of America, 87*(23), 9193–9196. https://doi.org/10.1073/pnas.87.23.9193

Xiao, P., Venayagamoorthy, G. K. & Corzine, K. A. (2007). Combined Training of Recurrent Neural Networks with Particle Swarm Optimization and Backpropagation Algorithms for Impedance Identification. *IEEE Swarm Intelligence Symposium,* Honolulu, HI, 9-15. https://doi.org/10.1109/SIS.2007.368020

Yang, X-S. (2009). Firefly algorithms for multimodal optimization. *International Symposium on Stochastic Algorithms*, 169-178. https://doi.org/10.1007/978-3-642-04944-6_14

Yang, X. S. (2010). A New Metaheuristic Bat-Inspired Algorithm. In J. R. González, D.A. Pelta, C. Cruz, G. Terrazas, & N. Krasnogor (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence*, vol 284. Springer.

Yang, X-S. & Deb, S. (2009). Cuckoo search via Levy flights. *World Congress on Nature & Biologically Inspired Computing (NaBIC),* Coimbatore, 210-214. https://doi.org/10.1109/NABIC.2009.5393690

Yoo, D. G., & Kim, J. H. (2014). Meta-heuristic algorithms as tools for hydrological science. *Geoscience Letters, 1*, 4. https://doi.org/10.1186/2196-4092-1-4

Zhang, X., Kang, Q., Tu, Q., Cheng, J. & Wang, X. (2019). Efficient and merged biogeography-based optimization algorithm for global optimization problems. *Soft Computing, 23*, 4483–4502. https://doi.org/10.1007/s00500-018-3113-1

Zhipeng, Y. Minfang, P., Hao, H. & Xianfeng, L. (2012). Fault Locating of Grounding Grids Based on Ant colony Optimizing Elman Neural Network. *2012 Third International Conference on Digital Manufacturing & Automation,* GuiLin, 406-409. https://doi.org/10.1109/ICDMA.2012.97