Competitive Island Cooperative Neuro-Evolution of Feedforward Networks for Time Series Prediction

Ravneil Nand and Rohitash Chandra

School of Computing Information and Mathematical Sciences
University of South Pacific, Suva, Fiji.
ravneiln@yahoo.com, c.rohitash@gmail.com

Abstract. Problem decomposition, is vital in employing cooperative coevolution for neuro-evolution. Different problem decomposition methods have features that can be exploited through competition and collaboration. Competitive island cooperative coevolution (CICC) implements decomposition methods as islands that compete and collaborate at different phases of evolution. They have been used for training recurrent neural networks for time series problems. In this paper, we apply CICC for training feedforward networks for time series problems and compare their performance. The results show that the proposed approach has improved the results when compared to standalone cooperative coevolution and shows competitive results when compared to related methods from the literature.

Key words: Cooperative coevolution, feedforward network, problem decomposition, Neuron level, Synapse level.

1 Introduction

Cooperative coevolution (CC) is an evolutionary algorithm that divides a large problem into subcomponents that are implemented as sub-populations [1]. CC applied to neuro-evolution is referred to as cooperative neuro-evolution that has been used for training feedforward and recurrent neural networks [2] for time series prediction [2]. Problem decomposition is vital for cooperative neuro-evolution. The two major decomposition methods are synapse level (SL) and neuron level (NL) methods. In synapse level problem decomposition, the subcomponents are defined by the weight connection which is known as synapse [3,4,2]. In neuron level problem decomposition, the neural network gets decomposed by the number of hidden and output neurons as reference neurons in the network [5].

Competition and collaboration are vital components in the evolution of species in nature and motivated recent trend in design of evolutionary algorithms. The competitive coevolution technique was proposed for genetic algorithm where populations called host and parasite contested with each other with different mechanisms that enable fitness sharing and selection [6]. The competition characteristic in cooperative coevolution has also been used for multi-objective optimization that exploited connection and inter-dependencies between the components of the problem [7].

Cooperative island cooperative coevolution (CICC) employs problem decomposition strategies as islands that compete and collaborate during different phases of evolution [8,9]. It was initially proposed for training recurrent neural networks for time series problems [8] and later extended for global optimization [9] where it showed very promising results. The islands evolve in phases that is defined by evolution time. At the end of each phase of evolution, the best solutions from participating islands are compared and the best solution is copied to the rest of the islands.

In this paper, we employ CICC for training feedforward networks on chaotic time series problems. The proposed approach takes advantage of the different problem decomposition methods during evolution of feedforward neural network for time series prediction problems. The performance of the proposed approach is compared with recurrent neural networks used for chaotic time series problems [8].

The rest of the paper is organized as follows. In Section 2, the proposed method is discussed in detail while Section 3 reports on experimental setup and results. Section 4 reports on discussion and Section 5 concludes the paper with discussion of future work.

2 CICC for Feedforward Neural Networks

In this section, the details of Competitive Island Cooperative Coevolution (CICC) for training feedforward neural network is given. The proposed method employs the strengths of different problem decomposition methods that reflects on the different degrees of separability and diversity [5].

The proposed method is given in Algorithm 1 where a problem decomposition method is defined as an *island* that has distinct features in terms of how the problem is decomposed and encoded. There are two islands which have distinct problem decomposition methods as given below:

1. SL Island: Decomposes the neural network in its lowest level where each synapse becomes a subcomponent. The number of subcomponents depends on the number of synapse [3].
2. NL Island: Neuron level problem decomposition employs hidden and output neurons as reference point for each subcomponent. The number of subcomponents depends on the number of hidden and output neurons [5].

The subcomponents are implemented as sub-populations which are implemented using an evolutionary algorithm. Initially, all the sub-populations of each island is initialized and then evaluated.

Algorithm 1: Competitive Two-Island Cooperative Coevolution for training Feedforward Neural Networks

```
Step 1: Initialisation:
i. Cooperatively evaluate Neuron level Island
ii. Cooperatively Evaluate Synapse level Island
Step 2: Evolution:
while FuncEval ≤ GlobalEvolutionTime do
    while FuncEval ≤ Island-Evolution-Time do
        foreach Sub-population at Synapse level do
            foreach Depth of n Generations do
                Create new individuals using genetic operators
                Cooperative Evaluation
            end
        end
    end
    while FuncEval ≤ Island-Evolution-Time do
        foreach Sub-population at Neuron level do
            foreach Depth of n Generations do
                Create new individuals using genetic operators
                Cooperative Evaluation
            end
        end
    end
    Step 3: Competition: Compare and mark the island with best fitness.
    Step 4: Collaboration: Exchange the best fitness individual from the island
    into the other island.
end
```

In Step 2, the evolution of the islands take place where each island is evolved for a predefined time given by the number of fitness evaluations in a round-robin fashion. This is called *island evolution time* that is given by the number of function evaluations in the respective islands. Once both islands have been evolved for the island evolution time, the algorithm proceeds and checks if the best solution of the particular island is better. If the solution is better, then the *collaboration*, procedure takes place where the solution is copied to the other island.

Each island is evolved using genetic operators until the local evolution time has been reached as shown in Fig. 1. In the collaboration procedure, the method takes into account how the best solution from one island is copied into the other since both have different number of sub-populations.

The best individuals from each of the subcomponents needs to be carefully concatenated into an individual and transferred without losing any genotype (subcomponents in cooperative coevolution) to phenotype (feedforward neural network) mapping. When one island wins, the best solution is transferred to the other island as shown in Fig 2.

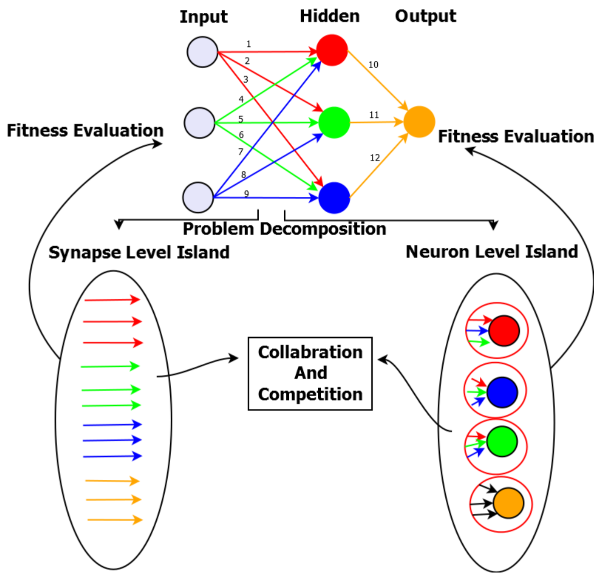


Fig. 1: The two islands compete with each other and the best solution is transferred to the losing Island.

As shown in Step 4 of Algorithm 1, the island that contains an individual with better solution needs to be shared with the other island. Synapse level island employs the highest number of sub-populations that are defined by number of weights in the network, whereas, neuron level island depends on the number of neurons in the hidden and output layer. Both islands have varied number of sub-

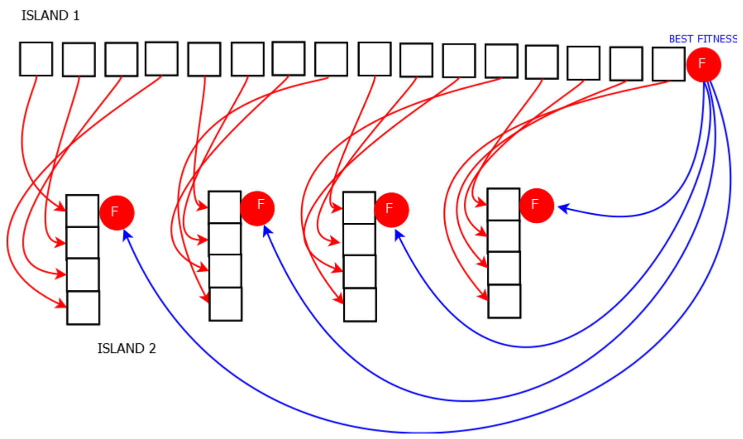


Fig. 2: The transfer of the best solution mapped between the islands (SL to NL).

populations with varied size, therefore, the transfer of the best solution needs to be mapped correctly between the islands which is shown in Fig. 2.

3 Experiments and Results

This section presents the experiments that feature neuron synapse level islands in proposed competitive island cooperative coevolution for feedforward networks for time series problems. The performance and results of the method were evaluated by using different numbers of hidden neurons.

We use four different benchmark chaotic time series to evaluate the proposed method. Mackey-Glass time series [10] and Lorenz time series [11] are the two simulated problems while Sunspot [12] and ACI Worldwide Inc.are the real-world problems. The results are compared with related work from the literature.

Taken's embedding theorem [13] allows for chaotic time series data to be reconstructed into a state space vector with the two conditions of *time delay (T)* and *embedding dimension (D)* [13]. The values of D and T are used as in the literature in order to provide a fair comparison [2,8].

All the data set used is reconstructed with the embedding dimensions as used in [2]. The Mackey-Glass and ACI time series datasets are scaled in the range of [0,1] whereas the Sunspot and Lorenz are scaled in the range of [-1,1].

The four time series are scaled in the range of [0,1] and [-1,1] as in the literature in order to provide a fair comparison [2,8].

The Generalized Generation Gap with Parent Centric Crossover (G3-PCX) evolutionary algorithm was used to evolve all the sub-populations with a pool size of 2 parents and 2 offspring as done in literature [2].

The number of generations for each sub-population known as *depth of search* and is kept as 1 since this number has achieved good results in previous work [2].

The algorithm terminates once the maximum number of function evaluations has been reached by the respective cooperative neuro-evolutionary methods. CICC employs a total of 100 000 function evaluations where each island employs 50 000. Standalone methods use 50 000 function evaluations.

The feedforward neural network employs sigmoid units in the hidden layer and in the output layer a sigmoid unit is used for the Mackey-Glass and ACI Worldwide Inc., while the hyperbolic tangent unit is used for Lorenz and Sunspot time series. Root mean squared error (RMSE) is used to evaluate the performance of the proposed method [2].

3.1 Results

This section reports on the performance of CICC for training the feedforward network on four different benchmark problems of the chaotic time series.

In Tables 1 - 4, the results are shown for different number of hidden neurons using the CICC method which is compared with the results of standalone cooperative coevolution based on feedforward network.

The results in the Tables 1 - 4 report the RMSE for training and generalisation performance. They are given by the mean and 95 percent confidence interval for 50 experimental runs for different number of hidden neurons (H). The best run is also given.

Table 1 shows experimental results of the Mackey-Glass time series problem where CICC has outperformed the standalone methods (SL and NL). All the methods produced better generalization performance with seven hidden neurons. The overall performance increased for all the methods as the number of hidden neurons increased.

In Table 2, the CICC method has performed better than SL method and closer to the NL method in terms of generalization for the Lorenz time series problem. The generalization performance of the CICC and the other two methods deteriorates as the number of the hidden neuron increases. The best result was seen for three hidden neurons for CICC.

Table 3 shows results for the Sunspot time series problem where three hidden neurons have given the best performance for CICC and also for the other two methods. The generalization performance of all the methods deteriorates as the number of the hidden neuron increases.

The ACI Worldwide Inc. time series problem is evaluated in Table 4. It shows that the CICC has performed much better than the other two standalone methods (SL and NL). The generalization performance of the CICC and the other two methods gets better as the number of the hidden neuron increases. The best result was seen for seven hidden neurons for all the methods.

Figures 3 - 4 show that a typical experimental run from the CICC method was able to cope with the noise from the real-world datasets. The error graph is also given which indicates the challenges for the chaotic nature of these time series problems at certain time intervals.

Table 5, compares the best results from Tables 1 - 4 with some of the related methods from the literature. The RMSE of the best run together with NMSE (Normalized Mean Squared Error) are used for the comparison. The proposed method has given better performance when compared to some of the methods in the literature.

Table 5, Mackey-Glass time series shows that CICC has outperformed all of the methods except for Auto Regressive Moving Average with Neural Network (ARMA-NN), Radial Basis Network with Orthogonal Least Squares (RBF-OLS) and Locally Linear Neuro-Fuzzy Model (LLNF).

In Lorenz problem given in the Table 5, the proposed method outperformed all the methods. In Table 5, for the Sunspot time series problem, the proposed method was able to outperform the rest of the methods from the literature. As for ACI Worldwide Inc. problem in Table 5, the proposed method outperforms majority of the methods.

Table 1: The prediction training and generalisation performance (RMSE) of NL and SL Mackey-Glass time series

Prob.	H	Training	Generalisation	Best
NL	3	0.0107 ± 0.00131	0.0107 ± 0.00131	0.0050
	5	0.0089 ± 0.00097	0.0088 ± 0.00097	0.0038
	7	0.0078 ± 0.00079	0.0078 ± 0.00079	0.0040
SL	3	0.0237 ± 0.0023	0.0237 ± 0.0023	0.0125
	5	0.0195 ± 0.0012	0.0195 ± 0.0012	0.0124
	7	0.0177 ± 0.0009	0.0178 ± 0.0009	0.0121
CICC	3	0.00950 ± 0.0013	0.0947 ± 0.0013	0.0043
	5	0.00690 ± 0.0005	0.0068 ± 0.0005	0.0035
	7	0.0063 ± 0.0005	0.0063 ± 0.0005	0.0026

Table 2: The prediction training and generalisation performance (RMSE) of NL and SL for the Lorenz time series

Prob.	H	Training	Generalisation	Best
NL	3	0.0170 ± 0.0031	0.0176 ± 0.0031	0.0043
	5	0.0249 ± 0.0062	0.0271 ± 0.0067	0.0021
	7	0.0379 ± 0.0093	0.0416 ± 0.0092	0.0024
SL	3	0.0680 ± 0.0325	0.0452 ± 0.0229	0.0153
	5	0.0526 ± 0.0084	0.0546 ± 0.0084	0.0082
	7	0.0574 ± 0.0075	0.0605 ± 0.0074	0.0079
CICC	3	0.0191 ± 0.00328	0.0198 ± 0.00359	0.0022
	5	0.0212 ± 0.00569	0.0225 ± 0.00611	0.0012
	7	0.0254 ± 0.00701	0.0281 ± 0.00748	0.0023

Table 3: The prediction training and generalisation performance (RMSE) of NL and SL Sunspot time series

Prob.	H	Training	Generalisation	Best
NL	3	0.0207 ± 0.0035	0.0538 ± 0.0091	0.015
	5	0.0289 ± 0.0039	0.0645 ± 0.0093	0.017
	7	0.0353 ± 0.0048	0.0676 ± 0.0086	0.021
SL	3	0.0539 ± 0.0261	0.04998 ± 0.0238	0.210
	5	0.0560 ± 0.0208	0.05210 ± 0.0177	0.302
	7	0.0568 ± 0.0178	0.05250 ± 0.0132	0.344
CICC	3	0.0193 ± 0.00351	0.0480 ± 0.00722	0.017
	5	0.0216 ± 0.00321	0.0549 ± 0.00993	0.014
	7	0.0316 ± 0.00488	0.0719 ± 0.00924	0.018

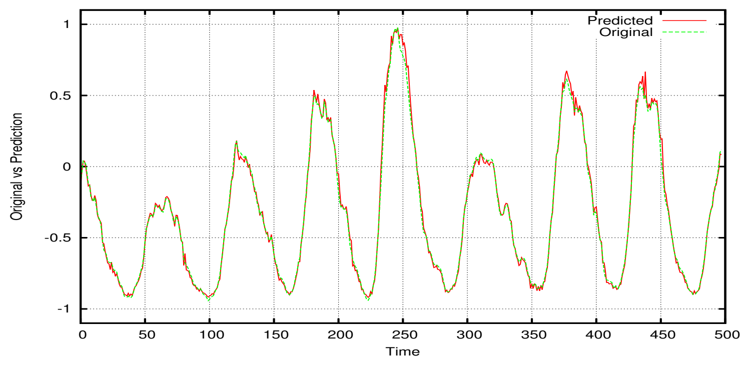
Table 4: The prediction training and generalisation performance (RMSE) of NL and SL for ACI time series

Prob.	H	Training	Generalisation	Best
NL	3	0.0214 ± 0.00039	0.0215 ± 0.00039	0.020
	5	0.0203 ± 0.00047	0.0212 ± 0.00041	0.019
	7	0.0201 ± 0.00038	0.0208 ± 0.00033	0.019
SL	3	0.0466 ± 0.0039	0.0411 ± 0.0036	0.080
	5	0.0413 ± 0.0038	0.0390 ± 0.0038	0.042
	7	0.0449 ± 0.0028	0.0424 ± 0.0027	0.134
CICC	3	0.0301 ± 0.0115	0.0332 ± 0.0197	0.0150
	5	0.0240 ± 0.0036	0.0227 ± 0.00566	0.0148
	7	0.0202 ± 0.00020	0.0178 ± 0.00088	0.0150

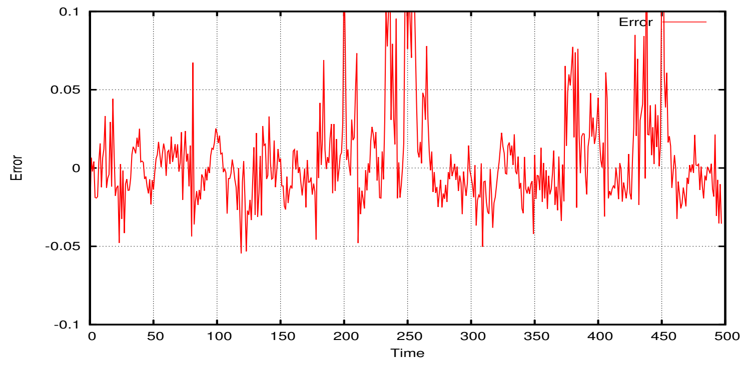
4 Discussion

Competitive island cooperative coevolution enables competition of diversity enforced through the number of sub-populations. The collaboration feature allows to improve diversity through injection of the best materials from the winning island after a predefined time. These features seem to improve the results of the standalone methods. Synapse level problem decomposition method seems to enable better global search features through higher diversity that is useful during the early stages of evolution. Collaboration and competition take advantages of these features during evolution. The proposed method takes advantage of solutions produced by two problem decomposition methods after each phase of competition.

The comparison of results with literature has shown that the proposed method performs better than cooperative co-evolutionary recurrent neural networks (CCRNN-SL and CCRNN-NL). We note that the performance also depends on the neural

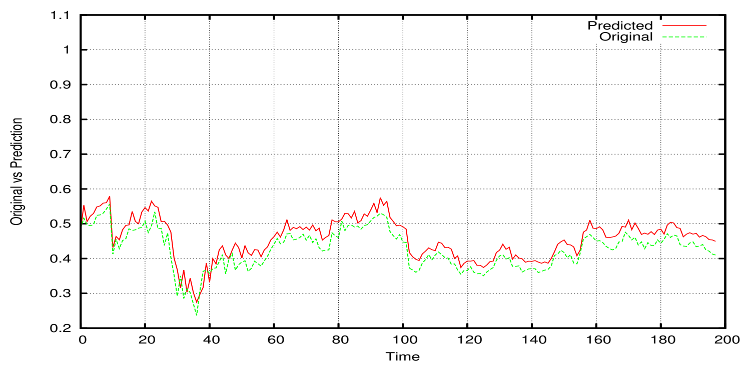


(a) Performance given by CICC on the testing set for Sunspot Worldwide.

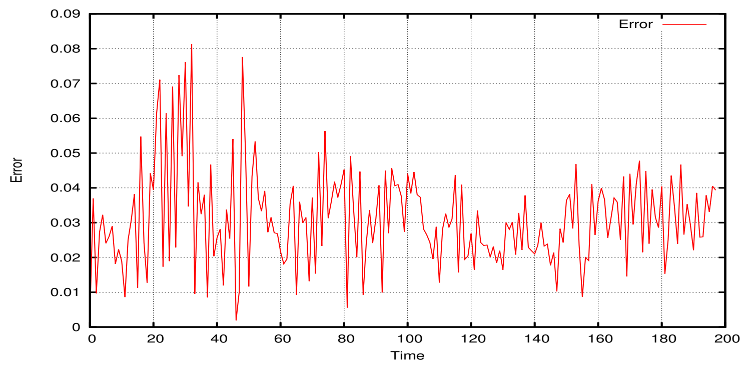


(b) Error on the test data set given by CICC for the Sunspot time series.

Fig. 3: Typical mean prediction given by CICC for Sunspot time series.



(a) Performance given by CICC on the testing set for ACI Worldwide Inc.



(b) Error on the test data set given by CICC for the ACI Worldwide Inc time series.

Fig. 4: Typical mean prediction given by CICC for ACI Worldwide Inc time series.

network architecture, hence, we cannot make strict comparison and only use the results from the literature of the related methods as a baseline to evaluate the proposed method. This is also the case when comparing the performance of proposed CICC-FNN with its counterpart CICC-RNN for the same problems. We also note that CICC was unable to outperform some of the related methods since they have additional enhancements such as the optimization of the embedding dimensions and strength of architectural properties [15].

5 Conclusion

In this paper, we applied competitive island-based cooperative coevolution of feedforward neural networks for chaotic time series prediction. The proposed approach employed two island competitive method where the islands were defined by neuron level and synapse level problem decomposition methods. The

Table 5: A comparison with the results from literature for all time series

Problem	Prediction Method	RMSE	NMSE
Mackey	Auto regressive moving average (ARMA-NN)(2008) [14]	2.5E-03	
	Radial basis network (RBF-OLS)(2006) [15]	1.02E-03	
	Locally linear neuro-fuzzy model (LLNF) (2006) [15]	9.61E-04	
	Neural fuzzy network (PS0) (2009) [4]	2.10E-02	
	Neural fuzzy network (CPS0) (2009) [4]	1.76E-02	
	Synapse Level-CCRNN (SL-CCRNN) (2012) [2]	6.33E-03	2.79E-04
	Neuron Level-CCRNN (NL-CCRNN) (2012) [2]	8.28E-03	4.77E-04
	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [8]	3.99E-03	1.11E-04
	Proposed CICC-FNN	2.61E-03	1.29E-05
Lorenz	Radial basis network (RBF-OLS)(2006) [15]		1.41E-09
	Locally linear neuro-fuzzy model (LLNF) (2006) [15]		9.80E-10
	Auto regressive moving average (ARMA-ANN)(2008) [14]	8.76E-02	
	Backpropagation neural network and genetic algorithms (2011) [16]	2.96E-02	
	Synapse Level-CCRNN (SL-CCRNN) (2012) [2]	6.36E-03	7.72E-04
	Neuron Level-CCRNN (NL-CCRNN) (2012) [2]	8.20E-03	1.28E-03
	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [8]	3.55E-03	2.41E-04
	Proposed CICC-FNN	1.16E-03	2.80E-05
Sunspot	Radial basis network (RBF-OLS)(2006) [15]		4.60E-02
	Locally linear neuro-fuzzy model (LLNF) (2006) [15]		3.20E-02
	Synapse Level-CCRNN (SL-CCRNN) (2012) [2]	1.66E-02	1.47E-03
	Neuron Level-CCRNN (NL-CCRNN) (2012) [2]	2.60E-02	3.62E-03
	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [8]	1.57E-02	1.31E-03
	Proposed CICC-FNN	1.44E-02	5.84E-04
ACI	Competitive Island Cooperative Coevolution (CICC-RNN) (2014) [8]	1.92E-02	
	Synapse Level (FNN-SL) (2014) [17]	1.92E-02	
	Neuron Level (FNN-NL) (2014) [17]	1.91E-02	
	MOCCFNN with 2-objectives (T=2)(MO-CCFNN-T=2) (2014)[18]	1.94E-02	
	MOCCFNN with 2-objectives (T=3)(MO-CCFNN-T=3) (2014)[18]	1.47E-02	
	Proposed CICC-FNN	1.48E-02	1.19E-03

results have shown good results on the different benchmark problems and has given competitive performance with the majority of the methods in the literature. It can be concluded that sharing of resources between the different islands that have different features helps in achieving better solutions.

In future work, the proposed method can be improved by exploring other problem decomposition methods that can provide more competition. A multi-threaded version of the algorithm can be developed to reduce the computation time. The method can be used to evolve other neural network architectures for similar problems and those that involve pattern classification and control.

References

1. M. Potter and K. De Jong, “A cooperative coevolutionary approach to function optimization,” in *Parallel Problem Solving from Nature PPSN III*, ser. Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Mnner, Eds. Springer Berlin Heidelberg, 1994, vol. 866, pp. 249–257.
2. R. Chandra and M. Zhang, “Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction,” *Neurocomputing*, vol. 186, pp. 116 – 123, 2012.
3. F. Gomez, J. Schmidhuber, and R. Miikkulainen, “Accelerated neural evolution through cooperatively coevolved synapses,” *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, 2008.

4. C.-J. Lin, C.-H. Chen, and C.-T. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 55–68, Jan. 2009.
5. R. Chandra, M. Frean, and M. Zhang, "On the issue of separability for problem decomposition in cooperative neuro-evolution," *Neurocomputing*, vol. 87, pp. 33–40, 2012.
6. C. D. Rosin and R. K. Belew, "New methods for competitive coevolution," *Evol. Comput.*, vol. 5, no. 1, pp. 1–29, Mar. 1997.
7. C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *European Journal of Operational Research*, vol. 202, pp. 42–54, 2010.
8. R. Chandra, "Competition and collaboration in cooperative coevolution of Elman recurrent neural networks for time-series prediction," *Neural Networks and Learning Systems, IEEE Transactions on*, p. In Press, 2015.
9. R. Chandra and K. Bali, "Competitive two island cooperative coevolution for real parameter global optimization," in *IEEE Congress on Evolutionary Computation*, Sendai, Japan, May 2015, p. In Press.
10. M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
11. E. Lorenz, "Deterministic non-periodic flows," *Journal of Atmospheric Science*, vol. 20, pp. 267 – 285, 1963.
12. SILSO World Data Center, "The International Sunspot Number (1834-2001), International Sunspot Number Monthly Bulletin and Online Catalogue," Royal Observatory of Belgium, Avenue Circulaire 3, 1180 Brussels, Belgium, accessed: 02-02-2015. [Online]. Available: http://www.sidc.be/silso/
13. F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, ser. Lecture Notes in Mathematics, 1981, pp. 366–381.
14. I. Rojas, O. Valenzuela, F. Rojas, A. Guillen, L. Herrera, H. Pomares, L. Marquez, and M. Pasadas, "Soft-computing techniques and arma model for time series prediction," *Neurocomputing*, vol. 71, no. 4-6, pp. 519 – 537, 2008.
15. A. Gholipour, B. N. Araabi, and C. Lucas, "Predicting chaotic time series using neural and neurofuzzy models: A comparative study," *Neural Process. Lett.*, vol. 24, pp. 217–239, 2006.
16. M. Ardalani-Farsa and S. Zolfaghari, "Residual analysis and combination of embedding theorem and artificial intelligence in chaotic time series forecasting," *Appl. Artif. Intell.*, vol. 25, pp. 45–73, 2011.
17. S. Chand and R. Chandra, "Cooperative coevolution of feed forward neural networks for financial time series problem," in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 202–209.
18. ——, "Multi-objective cooperative coevolution of neural networks for time series prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 190–197.