Enhancing Competitive Island Cooperative Neuro-evolution through Backpropagation for Pattern Classification

Gary Wong and Rohitash Chandra

School of Computing Information and Mathematical Sciences
University of South Pacific, Suva, Fiji.
gary.wong.fiji@gmail.com, c.rohitash@gmail.com

Abstract. Cooperative coevolution is a promising method for training neural networks which is also known as cooperative neuro-evolution. Cooperative neuro-evolution has been used for pattern classification, time series prediction and global optimisation problems. In the past, competitive island based cooperative coevolution has been proposed that employed different instances of problem decomposition methods for competition. Neuro-evolution has limitations in terms of training time although they are known as global search methods. Backpropagation algorithm employs gradient descent which helps in faster convergence which is needed for neuro-evolution. Backpropagation suffers from premature convergence and its combination with neuro-evolution can help eliminate the weakness of both the approaches. In this paper, we propose a competitive island cooperative neuro-evolutionary method that takes advantage of the strengths of gradient descent and neuro-evolution. We use feedforward neural networks on benchmark pattern classification problems to evaluate the performance of the proposed algorithm. The results show improved performance when compared to related methods.

1 Introduction

Cooperative coevolution (CC) decomposes a problem into subcomponents that are implemented as sub-populations which cooperatively evolves while mating is restricted within sub-populations [1]. The process of breaking a problem down into subcomponents is called problem decomposition. In the case of neuro-evolution, efficient problem decomposition depends on the network architecture and nature of the application problem in terms of separability [2]. Cooperative coevolution has been mostly used for large scale optimisation [3] and evolution of feedforward and recurrent neural networks in pattern classification and time series prediction [4, 5, 6, 7]. The use of cooperative coevolution for neuro-evolution is referred to as cooperative neuro-evolution.

In cooperative neuro-evolution, much attention has been given to problem decomposition, i.e. how to break the neural network into sub-problems through the interconnected weights that contain inter-dependencies [2]. The major problem decomposition methods involve those that fully or partially decompose the network, i.e. in full decomposition, the neural network is decomposed into the lowest level where a single subcomponent represents a weight connection, this is also called synapse level decomposition [8]. In partial decomposition, the network is decomposed with reference to weight connections linked to each hidden and output neurons that is also called neuron level decomposition [9, 2]. The performance of a decomposition method varies on different types of problems, for instance, synapse level decomposition showed very good results in pole balancing [8] but have been unsuccessful in pattern classification [9]. Both synapse and neuron level decomposition have shown competitive performance for time series problems [5]. There has been much focus on adaptation of the problem decomposition method during the learning process in order to take advantage global - local search and inter-dependencies [10, 11, 12]. In competitive island-based cooperative neuro-evolution (CICN), two or more problem decomposition methods are implemented as islands that compete and collaborate at different phases of evolution [4]. The competitive feature gives subcomponents the ability to compete for resources. There is altruism feature in the algorithm where the winner island shares its solution with the losing islands so that they can catch up in the next phase of evolution. The competitive and collaborative features enables strong solutions to be retained and has been very promising for training neural networks for time series and pattern classification problems [13, 14]

Neuro-evolution has limitations in terms of training time although they are known as global search methods. Backpropagation algorithm employs gradient descent which helps in faster convergence. Backpropagation suffers from premature convergence and its combination with neuro-evolution can help eliminate the weaknesses of both the approaches. In this paper, we propose a competitive island cooperative neuro-evolutionary method that takes advantage of the strengths of gradient descent and neuro-evolution. Integrating backpropagation in competitive island cooperative coevolution can help in achieving faster convergence to a near global optimum solution. We implement backpropogation as an island in competitive island-based cooperative neuro-evolution (CICN) and use it for training feedforward networks for selected pattern classification problems.

The remaining sections of the paper are structured as follows. Section 2 provides the details of the proposed method that features backpropagation in CICN. Section 3 presents the results with discussion and Section 4 concludes the paper with discussion on future work.

2 Proposed Method

2.1 Backpropagation in CICN

In Competitive Island Cooperative Neuro-evolution (CICN), two or more decomposition methods are implemented as islands that compete and provide altruism where the winning islands share solutions with the losing islands over a period until termination. In an environment with multiple species, the competitive feature relates to the ability of the species to outperform each other for possession of resources [15]. In the proposed method, two standalone methods are used that include backpropogation and cooperative neuro-evolution that employs neuron level problem decomposition. The details of each island are given below.

1. **Backpropagation algorithm (BP)**: Standard backpropagation algorithm where the entire network is used 'as-is' without decomposition.
2. **Cooperative coevolution with neuron level problem decomposition (CC-NL)**: The number of neurons in the hidden and output layer determine the number of subcomponents [5, 2].

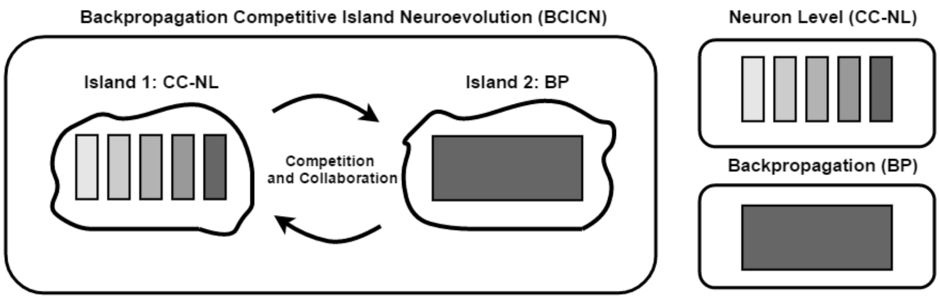

Fig. 1. The three algorithms employed in this study. These are standalone Backpropagation, CC-NL and the 2-Island BCICN algorithm.

The proposed backpropagation competitive island cooperative neuro-evolution (BCICN) method is given in Algorithm 2. In Stage 1, the sub-populations are randomly initialised and cooperatively evaluated using neuron level problem decomposition. After evaluation, the current best individual in the cooperative neuro-evolution island is copied to the Backpropagation Island. This is to ensure that both islands start from the same set of initial solution(s). In Stage 2, the islands are evolved per the island evolution time and in Stage 3 the best solutions from both islands are compared and the winner is selected to be transferred to the losing island in Stage 4 of the algorithm and then the process is repeated for the next phase of evolution. As presented in previous work [4], the respective islands need to be given the same number of function evaluations for each phase in evolution and this is due to the requirement that each island be evaluated for complete cycles.

2.2 Backpropagation Island

The conventional backpropagation procedure consists of forward pass where information is propagated forward through neurons using their activation function that computes weighted sum of incoming weight-connections to the respective neurons. Once the information is propagated from input, hidden to output later, the network error is computed and used to calculate gradients for each weight connected that are then updated. The process is repeated until the overall error reaches a desired level or when maximum training time in terms of epochs has been reached [16].

Algorithm 1: Backpropagation Algorithm (BP)

```
Initialisation:
foreach Epoch until Max-Epoch do
    foreach Training-Sample until Total-Training-Samples do
        Forward propagation through network
        Backward propagation through network
    end
    Increment Epoch
end
```

If the backpropagation island wins a phase of competition, it transfers the solution to the island that features cooperative neuro-evolution taking into account that the solution needs to be decomposed as defined by neuron problem decomposition in order to maintain solution validity. In the case where the Backpropagation Island loses the competition, the solution from the winner island will be concatenated by combining the best solutions from all its respective subpopulations. This individual is then refined using backpropogation and then the competition continues.

3 Experiments and Results

Table 1. Data set information and neural network configuration

Problem	Input	Output	Min. Train (%)	Max. Time	Samples
Wisconsin Breast Cancer	9	1	95	15000	699
4-Bit	4	1	1E-3	30000	16
Wine	13	3	95	15000	178
Iris	4	3	95	15000	150
Cleveland Heart Disease	13	1	88	50000	303

In this section, we apply the proposed BCICN to pattern classification problems. In our previous work, we applied competitive neuro-evolution to pattern classification and time series prediction [4]. We use the same classification problems from the UCI Machine Learning Repository [17]. The problems are Cleveland Heart Disease, Wisconsin Breast Cancer, Iris and the 4-Bit parity problem.

Algorithm 2: BCICN for Pattern classification

```
Stage 1: Initialisation:
i. Generate and cooperatively evaluate NL Island
ii. Copy Best Individual from NL Island to BP Island
Stage 2: Evolution:
while FE ≤ Global-Evolution-Time do
    while FE ≤ Island-Evolution-Time do
        foreach Sub-population at NL Island do
            foreach Depth of n Generations do
                Create new individuals using genetic operators.
                Cooperative evaluation.
            end
        end
    end
    while FE ≤ Island-Evolution-Time do
        Execute BP (Algorithm 1) .
    end
    Stage 3: Competition: Compare NL Island fitness with BP Island fitness.
    Stage 4: Collaboration: Inject the best individual from the island with
    better fitness into the other island.
    if NL Island fitness ≤ BP Island fitness then
        Copy NL Island best individual into the BP Island.
    end
    else
        Copy BP Island Individual to NL Island
    end
end
```

They have been used in other studies to evaluate performances of new methods [14, 2]. The details of problems tested are provided in Table 1.

The termination condition for an unsuccessful run is provided in Table 1 as maximum time (*Max. Time*). Each problem is set to have 50 independent runs where the evaluation time, generalisation performance and success rate is given. We evaluate the performance on different number of hidden neurons (H) in order to test robustness and scalability of BCICN. For all the 3 methods employed, the maximum time or island evolution time remained the same regardless of the number of islands used (in this case, we used 2 islands).

3.1 Results and Discussion

The results of the experiment are presented in Tables 2 - 3. A comparison is made between standalone cooperative coevolution with neuron level decomposition (CC-NL) and the BCICN.

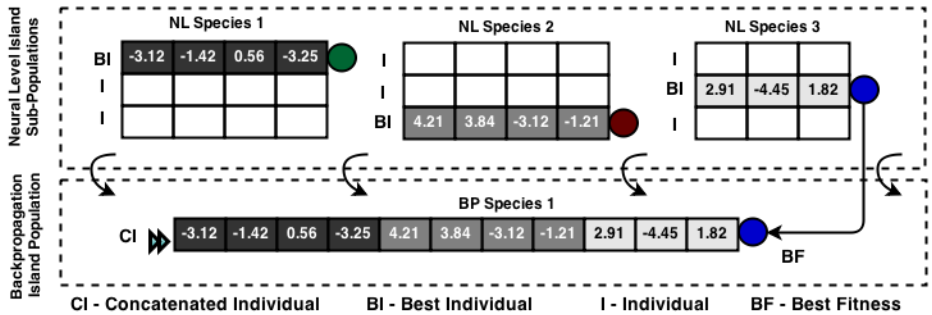

Fig. 2. Concatenation of the best individuals from the neuron level island and injection into the backpropagation island. Note the fitness of the concatenated individual is acquired from the fitness of the last best individual from the neural level island. When transferring, backpropagation's solution is decomposed as defined by neuron level problem decomposition.

The results show that the method that performed best in terms of convergence time for the Iris, Cancer, Wine and Heart problems was the standard backpropogation algorithm while the worst performance was that of CC-NL. BCICN obtained faster convergence and outperformed CC-NL as shown in Figure 3 where results for 10 hidden neurons are compared. The success rate of BCICN improved in some problems but was the same when the standalone methods had an average success rate of 100%. BCICN performed best in the 4-Bit problem where once again the worst performance was that of CC-NL. This is the only problem BCICN method outperformed backpropogation. The success rate of BCICN improved over backpropogation in all cases given by number of hidden neurons (H). The focus of this study was to reduce the convergence time of CC-NL and this was achieved in all the problems tested. The performance measure was in terms of minimising the function evaluations and improving the success rates.

This improved performance is due to the collaborative feature employed here where the two islands shared best solutions throughout the island evolutionary phases. In the backpropagation island, gradient information is used for weight update whereas in the cooperative neuron-evolution island, genetic operators are used. Gradient information features local search and ensures faster convergence when compared to neuro-evolution that features global search which is slower in convergence. Backpropagation island does not require network decomposition and hence does not face the problems of grouping interacting variables. BCICN provides the balance between global and local search and also features network

Keys for Table 2 and Table 3

$\bar{\mathbf{x}}$**ev** = Mean Fitness Evaluations, $\bar{\mathbf{x}}$**er** = Mean Generalisation Performance, (**H**) = No. Hidden Neurons, and (**sr**) = Success Rate

Table 2. Performance for the Iris, Cancer and Heart classification problems

		Iris				Cancer				Heart		
Method	**H**	$\bar{x}$**ev**	$\bar{x}$**er**	**sr**	**H**	$\bar{x}$**ev**	$\bar{x}$**er**	**sr**	**H**	$\bar{x}$**ev**	$\bar{x}$**er**	**sr**
BP	4	687	87.50	100	**4**	**246**	**97.99**	**100**	**6**	**4034**	**81.29**	**96**
	6	684	87.50	100	6	165	98.16	100	8	3390	81.17	100
	8	**680**	**87.50**	**100**	8	145	98.38	100	**10**	**1440**	**81.28**	**50**
	10	692	87.50	100	10	129	98.39	100	12	1569	81.53	100
	12	**700**	**87.50**	**100**	**12**	**124**	**98.42**	**100**	14	1451	81.33	100
CC-NL	**4**	**4356**	**95.50**	**100**	4	**5562**	**96.98**	**94**	6	19097	79.50	90
	6	5184	94.88	100	**6**	**4519**	**97.70**	**100**	**8**	**15719**	**79.88**	**100**
	8	5430	96.75	100	8	5227	97.96	100	**10**	**35760**	**80.00**	**50**
	10	5860	96.00	100	10	5174	98.08	100	12	24445	80.55	100
	12	**6636**	**96.20**	**100**	12	5475	98.31	98	14	21051	79.11	100
BCICN	**4**	**2204**	**95.50**	**100**	**4**	**2034**	**96.71**	**94**	6	8990	80.94	100
	6	3618	95.00	100	6	1454	97.33	100	**8**	**12580**	**79.55**	**100**
	8	4117	94.00	100	8	1485	97.34	100	10	9675	79.95	100
	10	3632	94.75	100	**10**	**1140**	**97.45**	**100**	12	7321	81.53	100
	12	**4369**	**95.25**	100	12	1248	97.62	100	**14**	**6340**	**81.02**	**100**

Table 3. Performance for the Wine and 4-Bit classification problems

		Wine				4-Bit		
Method	**H**	$\bar{x}$**ev**	$\bar{x}$**er**	**sr**	**H**	$\bar{x}$**ev**	$\bar{x}$**er**	**sr**
BP	**4**	**262**	**98.12**	**100**	**4**	**30010**	**-**	**0**
	6	279	98.50	100	6	13995	100.00	70
	8	282	98.75	100	8	9442	100.00	95
	10	300	99.62	100	10	5228	100.00	95
	12	**323**	**99.75**	**100**	**12**	**4795**	**100.00**	**95**
CC-NL	**4**	**6573**	**94.73**	**95**	**4**	**11151**	**100.00**	**100**
	6	7371	92.75	100	**6**	**6001**	**100.00**	**100**
	8	7293	94.25	100	8	5772	100.00	100
	10	8268	94.00	100	10	7012	100.00	100
	12	**8730**	**94.12**	**100**	12	6318	100.00	100
BCICN	**4**	**1959**	**95.25**	**100**	**4**	**9324**	**100.00**	**100**
	6	1994	94.87	100	6	4944	100.00	100
	8	1728	95.50	100	8	3298	100.00	100
	10	**921**	**95.12**	**100**	**10**	**3067**	**100.00**	**100**
	12	1023	94.37	100	12	3967	100.00	100

decomposition. It approaches the problem as partially separable through neuron level problem decomposition and non-separable through backpropagation.

In terms of scalability, we look at the mean evaluations at each total number of hidden neurons used. It is observed that increasing the number of hidden neurons in the Cancer, 4-Bit and Heart problems decreased the mean evaluations needed. On the other hand, mean evaluation performance improved when more hidden neurons were used in the Wine and Iris problems. The BCICN method showed good scalability in the Wine and Iris problems, but poor scalability in Cancer, 4-Bit and Heart problems. It can be generalised that that scalability features depend greatly on the problem nature, which is in terms of the size of the problem, noise, number of attributes and level of inter-dependencies amongst them.

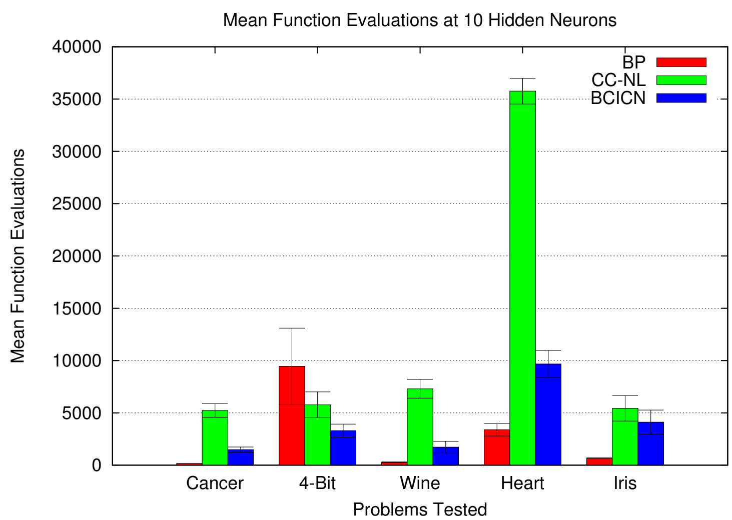

Fig. 3. Visualisation of the performance of BCICN with the standalone methods for 10 hidden neurons taken from Table 2 and Table 3.

4 Conclusions and Future Work

This paper proposed an algorithm that incorporates backpropagation in competitive island cooperative neuro-evolution for pattern classification. The results show that the proposed method outperformed the standalone methods through faster convergence. The backpropagation algorithm provided neuro-evolution gradient information that led to faster convergence. This can be very beneficial in the use of neuro-evolution for big data related problems that require faster learning. Evolutionary computation methods have limitations in the field of big data due to time required for convergence. The proposed method can motivate the development of other hybrid algorithms that speed up evolutionary learning methods for big data problems.

In future work, the proposed method can be used for training recurrent neural networks for time series prediction problems such as renewable energy load forecasting. It can also be used in selected big data problems.

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, M. Frean, and M. Zhang, "On the issue of separability for problem decomposition in cooperative neuro-evolution," *Neurocomputing*, vol. 87, pp. 33–40, 2012.

[3] M. Omidvar, X. Li, and X. Yao, "Cooperative co-evolution with delta grouping for large scale non-separable function optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, 2010, pp. 1762 –1779.

[4] R. Chandra, "Competitive two-island cooperative coevolution for training Elman recurrent networks for time series prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014, pp. 565–572.

[5] R. Chandra and M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction," *Neurocomputing*, vol. 186, pp. 116 – 123, 2012.

[6] N. Garcia-Pedrajas, C. Hervas-Martinez, and J. Munoz-Perez, "COVNET: a cooperative coevolutionary model for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 575–596, 2003.

[7] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, pp. 1–29, 2000.

[8] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Accelerated neural evolution through cooperatively coevolved synapses," *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, 2008.

[9] R. Chandra, M. Frean, and M. Zhang, "An encoding scheme for cooperative coevolutionary feedforward neural networks," in *AI 2010: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, J. Li, Ed. Springer Berlin / Heidelberg, 2010, vol. 6464, pp. 253–262.

[10] R. Chandra, "Adaptive problem decomposition in cooperative coevolution of recurrent networks for time series prediction," in *International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, USA, August 2013, pp. 1–8.

[11] R. Chandra, M. Frean, and M. Zhang, "Adapting modularity during learning in cooperative co-evolutionary recurrent neural networks," *Soft Com-*

puting - A Fusion of Foundations, Methodologies and Applications, vol. 16, no. 6, pp. 1009–1020, 2012.

[12] R. Chandra, M. R. Frean, and M. Zhang, "Crossover-based local search in cooperative co-evolutionary feedforward neural networks," *Appl. Soft Comput.*, vol. 12, no. 9, pp. 2924–2932, 2012.

[13] 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.

[14] R. Chandra and G. Wong, "Competitive two-island cooperative coevolution for pattern classification problems," in *International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, July 2015, pp. 393–400.

[15] G. Bella, *A Bug's Life: Competition Among Species Towards the Environment*, ser. Fondazione Eni Enrico Mattei Working Papers. Fondazione Eni Enrico Mattei, 2007.

[16] S. Russell and P. Norvig, "Backpropagation in neural networks," *Artificial Intelligence A Modern Approach*, 1969.

[17] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html