Information Collection Strategies in Memetic Cooperative Neuroevolution for Time Series Prediction

Gary Wong *, Anuraganand Sharma * and Rohitash Chandra †¶
* School of Computing Information and Mathematical Sciences
University of the South Pacific, Suva, Fiji.
† Centre for Translational Data Science, The University of Sydney, NSW 2006, Australia
¶ School of Geosciences, The University of Sydney, NSW 2006, Australia

***Abstract*—Memetic algorithms have been a promising strategy to enhance neuroevolution in the past. Cooperative coevolution has been combined as memetic cooperative neuroevolution with application to chaotic time series prediction. Although the method has shown promising performance, there are limitations in the balance between global and local search. The previous study used a specific local search strategy for intensification that affected the diversity of solutions. In this study, we address this limitation by information (meme) collection strategies that maintains and refines a pool of memes during global search. We present two strategies where one is sequential and the other is concurrent meme collection implemented at different stages of evolution. In the majority of the given problems, the proposed strategies showed improvement in prediction accuracy over the related methods.**

***Index Terms*—Cooperative Coevolution, Memetic Algorithms, Time Series Prediction Global Search, Local Search, Neuroevolution.**

I. Introduction

Memetic algorithms (MAs) are meta-heuristics that feature techniques to balance exploration and exploitation during optimization or evolution [1]–[5]. MAs are based on evolutionary biology where the term 'meme' refers to cultural information as opposed to genes [6]. The meme in memetic algorithms typically refers to local search techniques that guides the search during evolution [4], [7], [8]. Global search refers to the algorithms that explore the entire fitness landscape for global optima while local search algorithms focus around a particular area [1]. MAs are capable of tackling large-scale real-world problems where motivation stems from the inefficiency of canonical evolutionary algorithms for large-scale problems that require a balance between diversification and intensification [1]. It is well known that global search aids local search in escaping local minima while local search provides refinement [4], [8], [9].

Cooperative Coevolution (CC) decomposes a problem into subcomponents that are implemented as subpopulations. These are typically evolved in isolation where cooperation takes place for fitness evaluation. Evolution in isolation implies that the evolutionary operators such as crossover and mutation from a particular subpopulation do not affect the other subpopulations [10]. Cooperative neuroevolution (CNE) which refers to use of CC for neuroevolution has been successfully applied to several domains that include optimisation [11], [12], pattern classification [13], [14] and time series prediction [15]. CC appeals to problems that are separable which feature little interactions amongst subcomponents [16]. Synapse level decomposition creates a subpopulation for every synapse or weight in the neural network [17]. Synapse level decomposition has shown to be effective for time series prediction [15] and control problems [17], however, it gave poorer performance when applied to pattern classification [18]. Neuron level decomposition creates sub-populations with reference to each neuron in the hidden and output layers [15], [18]. It is preferred due to the efficiency and accuracy of solutions produced and has been successfully applied to pattern classification [18] and time series prediction [15]. More recently, a synergy of cooperative neuro-evolution with multi-task learning has been used for multi-step time series prediction [19].

The synergy between cooperative neuroevolution and memetic algorithms have been explored in the past [20]–[22]. Chandra *et. al.* [20] implemented the meme using crossover based local search in cooperative coevolution for pattern classification problems. This featured the decomposition of the network for global search via the subpopulations. The meme was not decomposed by the network. The authors highlighted the challenges in incorporating the memes back into the subpopulations. Later, the method was also used for recurrent neural networks for grammatical inference problems [21]. Wong and Chandra used gradient descent via backpropagation for the memes [22]. They gathered that the exchange of memes with the subpopulations obtained better accuracy in limited computation time than solely using global or local search. However, the method did not outperform some of the methods from the literature. Since local search was used at a uniform rate throughout the evolution, it could have affected the diversity of the solutions. This was a major limitation of the method. In a related framework, competition and collaboration between islands of cooperative neuroevoluton and backpropagation were implemented during evolution [23].

In this study, we improvise the memetic strategy in [22] by minimizing local search influence at a uniform rate while exploring the solution space. In order to achieve this, memes during the initial search phase (global search) are not incorporated into the subpopulations. We propose two strategies for improving interaction between the memes and the subpopulations, where one is *sequential* and the other is *concurrent* in incorporating memes during evolution. The results are compared to our previous implementation of memetic cooperative neuro-evolution for time series prediction [22].

The rest of the paper is organized as follows, Section 2 presents the proposed method that gives details of the strategies for improving memetic cooperative neuroevolution. Section 3 presents the experiment design, results, and discussion. Section 4 concludes the study with notes on future work.

II. Enhanced Memetic Cooperative Neuroevolution

As mentioned earlier, the meme in the earliest memetic cooperative neuroevolution framework represented the entire network weights and biases [20]. Global search was implemented by the subpopulations in cooperative neuroevolution by the neuron level decomposition method [24]. Note that global search becomes more useful in the initial phase of evolution. A single meme that was implemented by gradient descent was incorporated into the subpopulations which slightly improved the prediction performance [22]. We found that giving more time to evolution of the subpopulations increases computational costs (over-exploration), while giving more refinement time to the memes (over-exploitation) can get the search stuck in a local minima [22]. The balance between global search by the subpopulations and local refinement by the meme was the focus of the study. The local search frequency (lsf) determined how often the memes were implemented the while the local search intensity (lsi) defined the duration of the meme for refinement. When passing back to the subpopulations, the meme is disintegrated according to neuron level decomposition in order to replace the worst fit individual. The method performed better on the real world time series problems when compared to the other methods from the literature. In order to improve it further for other types of problems, we extend the memetic cooperative neuroevolution (MCNE) framework [22] using two strategies presented in Figures 1 and 2. We investigate different ways of efficiently utilising memes for keeping a balance between global and local search during evolution.

A. Sequential Meme Collection

In this case, the meme collection strategy extracts and concatenates the fittest individuals from all the subpopulations at a uniform rate in *sequential* order *during* the entire phase of evolution as shown in Figure 1. The memes in our previous implementation were refined at a fix rate, whereas here, we adapt the lsi during evolution.

In the *initialization* stage (Algorithm 1 - Line 1), the number of subcomponents is determined by the number of hidden (h) and output (o) neurons in the network. The function

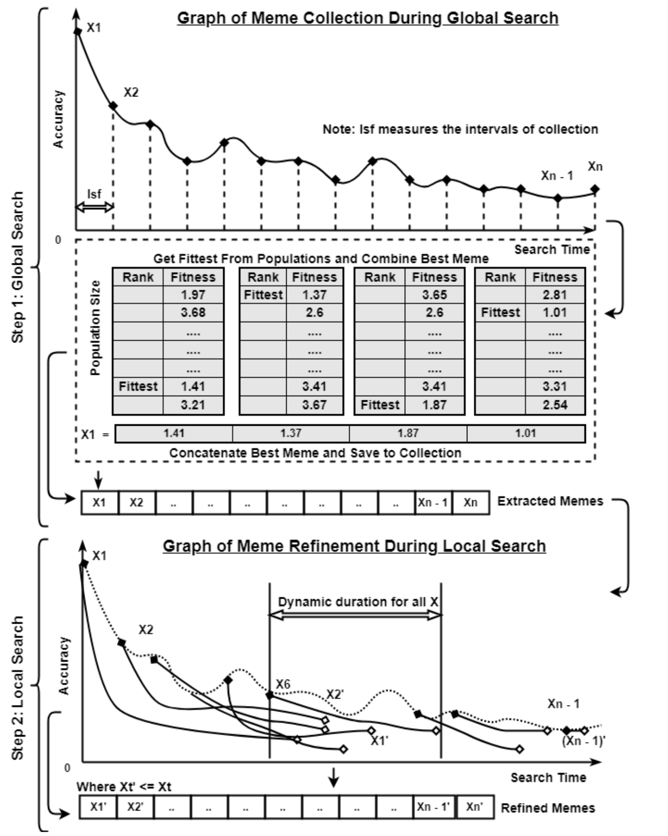


Fig. 1. The *sequential* memetic strategy applies the meme at a uniform rate during the entire evolution.

rand(μ, w_{max}, w_{min}) initializes the subpopulations of CC (L) (with a population size of μ) that represent neural network weights with real numbers in a range [w_{min} and w_{max}]. The fitness of each individual of the respective subpopulations is then computed and assigned in eval($L(y)$). The initialization cost is then added to the total number of fitness evaluations taken by the algorithm $\Gamma_{elapsed}$.

In the *meme collection* step (Algorithm 1 - Line 7), we begin by performing global search where the subpopulations are independently evolved and evaluated with a duration of Γ_{max} fitness evaluations. Here, each subcomponent is passed to the evolution function evolve($L(y)$) which returns a training error (*RMSE*) and updates the number of fitness evaluations used for global search $\Gamma_{elapsed}$. The subpopulations are evolved for the lsf, then the fittest individuals from all the subpopulations are concatenated and presented for refinement by gradient descent, which is also known as meme collection, mc. The elapsed global search time updated by the $e_{counter}$.

In the *refinement* step (Algorithm 1 - Line 16), each meme is refined with varying lsi as given by Equation 1. Finally, the best meme is selected according to fitness evalMemes(mc).

The sequential strategy employs an adaptive local search intensity similar to the work in Chandra *et al.* [20]. This was done to ensure that the intensity of local search increases with the number of fitness evaluations. For this study, a

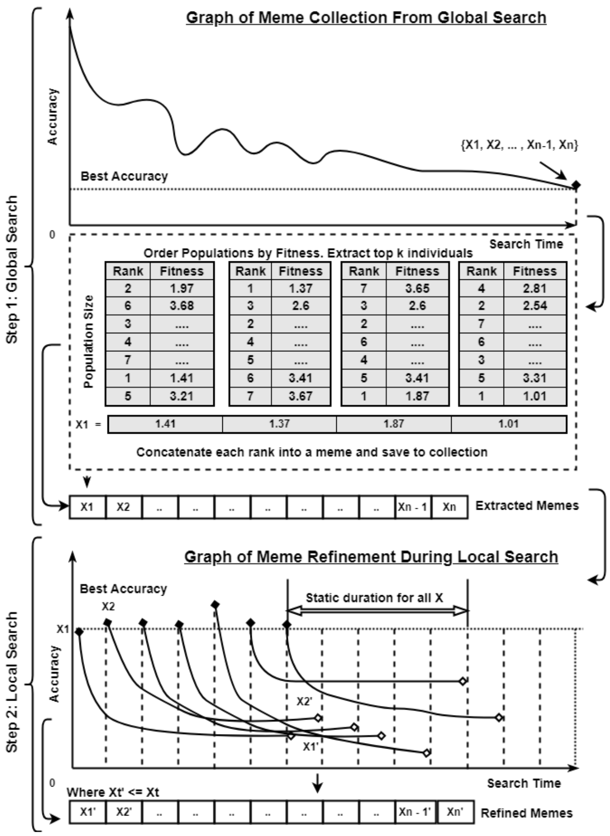


Fig. 2. The memetic *concurrent* strategy applies the meme at the end of the exploration stage.

Algorithm 1: *Sequential* Strategy

Step 1: Population Initialization
s = h + o; $\Gamma_{elapsed} = 0$;
for $y \in \{1,..., s\}$ **do**
  $L(y)$ = `rand`(μ, w_{max}, w_{min});
  $F(y)$ = `eval`($L(y)$);
  $\Gamma_{elapsed} = \Gamma_{elapsed} + |L(y)|$;

Step 2: Collection (Global Search)
while $\Gamma_{elapsed} < \Gamma_{max}$ **do**
  while $(\Gamma_{elapsed} - e_{counter}) < (lsf + 1)$ **do**
    for $y \in \{1,..,s\}$ **do**
      for $j \in \{1,..,\mu\}$ **do**
        $L(y)$ =`evolve`($L(y)$); $\Gamma_{elapsed}$ += μ x (γ + 1);
  δ^* = `getBestSolution`(L);
  $mc = mc \cup \delta^*$;
  $ec = ec \cup \Gamma_{elapsed}$;

Step 3: Refinement (Local Search)
for $u \in \{1,..,|mc|\}$ **do**
  $\Gamma_{elapsed} = ec(u)$;
  $lsi = lsf - \frac{(lsf \times \Gamma_{elapsed})}{\Gamma_{max}}$
  ε_t = `bpnn`($\delta^*, \lambda, mc(u), lsi$);
`evalMemes`(mc);

similar heuristic is used that adapts the local search intensity according to the current fitness evaluations, ($\Gamma_{elapsed}$) when the individuals are collected from the subpopulations for the meme to be refined (Figure 1).

$$\texttt{calculateLSI()} \quad = lsf - \frac{(lsf \times \Gamma_{elapsed})}{\Gamma_{max}} \qquad (1)$$

Equation 1 takes into consideration the (lsf) and maximum fitness evaluations (Γ_{max}).

B. *Concurrent Meme Collection*

This meme collection strategy *concurrently* collects a list of the fittest individuals from the subpopulations at the end of the exploration phase as shown in Figure 2. Each meme is refined with a fixed local search intensity. This algorithm follows the same *initialization* procedure as Algorithm 1. In the *meme collection* step (Algorithm 2 - Line 7), we begin by performing global search where subpopulations are independently evolved and evaluated (Γ_{max}). After each phase of global search, individuals in the respective subpopulations are ranked by fitness. In the *refinement* step (Algorithm 2 -

Algorithm 2: *Concurrent* Strategy

1 **Step 1**: Population Initialization
2 s = h + o; $\Gamma_{elapsed} = 0$;
3 **for** $y \in \{1,..., s\}$ **do**
4   $L(y)$ = `rand`(μ, w_{max}, w_{min});
5   $F(y)$ = `eval`($L(y)$);
6   $\Gamma_{elapsed} = \Gamma_{elapsed} + |L(y)|$;

7 **Step 2**: Collection (Global Search)
8 **while** $\Gamma_{elapsed} < \Gamma_{max}$ **do**
9   **while** $(\Gamma_{elapsed} - e_{counter}) < (lsf + 1)$ **do**
10   **for** $y \in \{1,..,s\}$ **do**
11   **for** $j \in \{1,..,\mu\}$ **do**
12         $L(y)$ =`evolve`($L(y)$); $\Gamma_{elapsed}$ += μ x (γ + 1);

13 $n = \frac{\Gamma_{elapsed}}{lsf}$;
14 L = `orderAsc`(L);
15 mc = `getTopSolutions`(L, n);

16 **Step 3**: Refinement (Local Search)
17 **for** $u \in \{1,..,\mu\}$ **do**
18   ε_t = `bpnn`($\delta^*, \lambda, mc(u), lsi$);
19 `evalMemes`(mc);

TABLE I
SYMBOL TABLE

Variable	Description	Variable	Description
α	Mutation Rate	ε_t	Test Accuracy.
μ	Population Size	lsf	LS Frequency.
Γ_{max}	Max Evaluations.	lsi	LS Intensity.
$\Gamma_{elapsed}$	Total Evaluations.	δ^*	Best Meme.
λ	Learning Rate.	L	sp Set.
γ	Optimization Time	sp	Sub-population.
i	# Input Neurons.	w_{min}	Lower Weight Limit.
h	# Hidden Neurons.	w_{max}	Upper Weight Limit.
o	# Output neurons.	mc	Meme Collection.
ε_{min}	Required Minimum ε_t.	$e_{counter}$	Elapsed Counter
ec	Elapsed Times	n	Top # of Memes
F	Sub-population Fitness		

Line 16), each meme is refined with the given local search intensity. Finally, the best meme is selected according to fitness (`evalMemes(`mc`)`), decomposed and incorporated back into the respective subpopulations.

III. Simulation and Analysis

This section presents the experiments that evaluates the proposed meme collection strategies. The results are compared with our previous implementation [22]. The software code and dataset used for the experiments is given online [1,2].

A. Experiment Design

The subpopulations in cooperative neuroevolution employ a pool of 300 individuals (μ) that feature the generalized generation gap with parent-centric crossover (G3-PCX) evolutionary algorithm [25]. It uses a pool size of 2 offspring and 2 parents used for generating news solutions. The respective subpopulations are initialized by $\{w_{min}$:-5, w_{max}:5$\}$. The global search duration is ($\Gamma_{max} = 100,000$) fitness evaluations. The meme that implements backpropagation uses a learning rate ($\lambda = 0.1$).

In the *sequential* strategy, the (lsi) is varied according to Equation 1 at every $lsf = 5000$ fitness evaluations. In the *concurrent* strategy, $lsi = 2000$ epochs. At the end of global search, subpopulations are ranked according to fitness and the top n solutions are taken into the collection. This study sets n to 10.

It is important to explore different features of time series data during pre-processing for improved prediction accuracy. Taken's theorem is used here which reconstructs original time series data into a phase space that can be used for prediction training [26]. Time lag *T* determines the interval at which data points are selected while the embedding dimension *D* specifies the size of the sliding window over the data points. The embedding dimensions *D* also determines the number of input neurons for the feedforward neural network. The pre-processing information for each dataset is shown in Table II. We use 5 hidden and 1 output neuron, since we consider one step-ahead prediction as done in previous studies [15], [22].

B. Datasets

This study employs five chaotic time series datasets consisting of of real-world and simulated problems. These consist of Sunspot [27], Mackey-Glass [28], Lazer [29], Lorenz [30] and Taiwan Trading Index Exchange (TWI Exchange) [31]. Each problem is divided into a 60/20/20 % partition (training/validation/test). The root mean squared error (*RMSE*) is used for evaluation (Equation 2), where y_i refers to observed data, $\hat{y}_i$ the predicted data, $\bar{y}_i$ the average of observed data, and *N* is the size of the data.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2}$$

[1]https://github.com/gary-wong-fiji/Meme-Collection-SEQ
[2]https://github.com/gary-wong-fiji/Meme-Collection-CON

IV. Results

The results are presented in Tables III to VII which highlight the mean, best, worst accuracy (RMSE) along with total fitness evaluations. The *sequential* strategy obtained the best performance in terms of accuracy whereas the *concurrent* strategy obtained the second best accuracy in most of the problems tested. The concurrent strategy, however, obtained the worst accuracy in the Lorenz problem. This indicates that diversity was maintained in terms of the collection scheme solutions in different parts of the solution space. The same applies to MCNE which had the least accuracy in almost all the problems tested. Note that the experiment design limited MCNE to 100,000 fitness evaluations in our previous work, hence comparison is not fair, however, we provide the results to present a baseline [22].

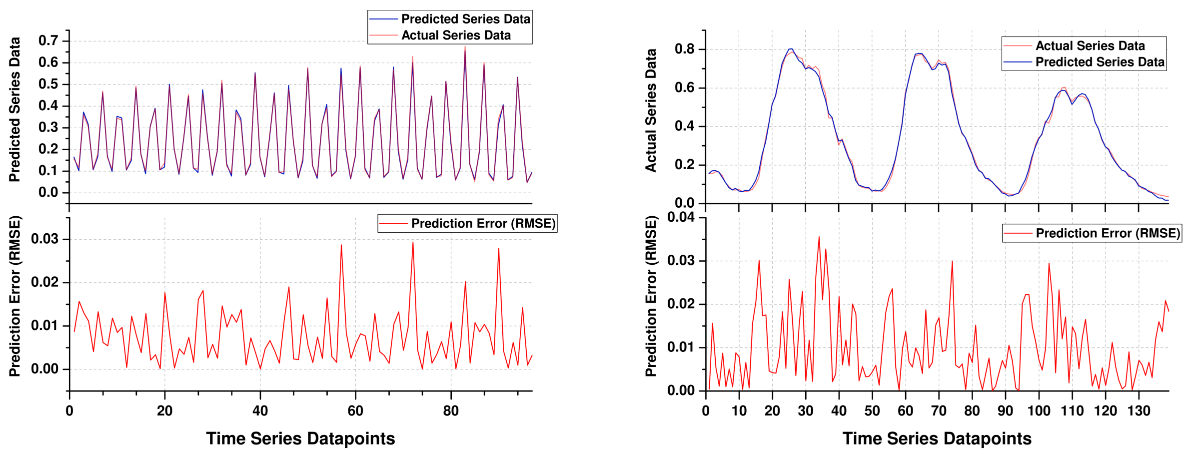
V. Discussion

The results, in general, show that the sequential strategy had the best generalization performance. This can be attributed to the refinement of diverse solutions from the subpopulations by the memes during evolution when compared to the concurrent strategy that featured memes only at the end of the evolution phase. The adaptation of the local search intensity of the sequential strategy seems to have been useful in providing a better balance for solutions collected at different points of the global search. Both the meme collection strategies proved useful in reducing local search interference by fully separating the exploration and exploitation processes. The exploration process defined as the global search operation while the exploitation process is defined by the local search operations. The evolutionary CC provided a diversity of solutions that were not refined by local search until the end of evolution. The meme employs gradient descent based weight updates that features knowledge of the network structure which differs with the search with evolutionary operators in the subpopulations that view the neural network as a black-box.

The trend is similar to the results of Chandra *et. al* [20] where an adaptive local search intensity proportional to the fitness evaluations was used. The authors ensured that local search intensity increased in the later stages to refine solutions found by global search features. For this study, an adaptive lsi was used to enable more refinement of the meme during the initial stage which is different when compared to [20] where more refinement time was used in the later stage. The computation time it takes to refine all collected memes is substantial when compared to our previous work, however

TABLE II
Datasets information

Dataset	Samples	Dim. and Time Lag	*Train/Val/Test*
Laser	1000	D: 3, T: 2	194 / 166 / 166
Lorenz	1000	D: 3, T: 2	299 / 99 / 99
MackeyGlass	1000	D: 3, T: 2	299 / 99 / 99
Sunspot	2000	D: 5, T: 3	399 / 132 / 132
TWIExchange	304	D: 5, T: 1	177 / 55 / 55



Laser problem

Sunspot problem

Fig. 3. The prediction accuracy (RMSE) of a single experimental run for the test dataset using sequential strategy on the Laser [29] and Sunspot problems [27].

TABLE III
PERFORMANCE FOR THE **SUNSPOT** PROBLEM

Method	**Mean** ε_t	**Best** ε_t	**Worst** ε_t	**Eval.** $\Gamma_{elapsed}$
Sequential	0.0127696	0.0107341	0.0195412	205,039
Concurrent	0.019353	0.014647	0.02512103	121,200
MCNE [22]	0.0478444	0.0246412	0.0671124	100,000

TABLE IV
PERFORMANCE FOR THE **TWI EXCHANGE** PROBLEM [31]

Method	**Mean** ε_t	**Best** ε_t	**Worst** ε_t	**Eval.** $\Gamma_{elapsed}$
Sequential	0.0394227	0.035412	0.0412148	272,318
Concurrent	0.0397674	0.0363142	0.0432614	121,200
MCNE [22]	0.0852743	0.0745214	0.0912457	100,000

TABLE V
PERFORMANCE FOR THE **SANTA FE LASER** PROBLEM [29]

Method	**Mean** ε_t	**Best** ε_t	**Worst** ε_t	**Eval.** $\Gamma_{elapsed}$
Sequential	0.069533	0.0571243	0.072142	269,421
Concurrent	0.0768557	0.0634781	0.0793412	121,200
MCNE [22]	0.194982	0.147142	0.2188464	100,000

TABLE VI
PERFORMANCE FOR THE **MACKEY GLASS** PROBLEM [28]

Method	**Mean** ε_t	**Best** ε_t	**Worst** ε_t	**Eval.** $\Gamma_{elapsed}$
Sequential	0.00454625	0.00192641	0.0057482	271,031
Concurrent	0.00595269	0.00320041	0.00671213	121,200
MCNE [22]	0.0252556	0.012321489	0.03451222	100,000

TABLE VII
PERFORMANCE FOR THE **LORENZ** PROBLEM [30]

Method	**Mean** ε_t	**Best** ε_t	**Worst** ε_t	**Eval.** $\Gamma_{elapsed}$
Sequential	0.073145	0.071354	0.078321	260,668
Concurrent	0.34457	0.32148871	0.3811421	121,200
MCNE [22]	0.0747062	0.075321	0.0793321	100,000

this seems worthwhile since better accuracy is obtained. The concurrent strategy on the other hand, used lower computation cost with similar generalization performance for most of the problems.

VI. CONCLUSIONS AND FUTURE WORK

In this study, memetic cooperative neuroevolution was extended using two strategies that took advantage of global search via the subpopulations of cooperative coevolution and gradient descent for meme refinement. The performance of the strategies were promising in terms of prediction accuracy for most of the problems. The *sequential* strategy did much better when compared to the *concurrent* strategy and our previous implementation given in the literature. This strengthens the notion that reducing local search interference in the initial evolution phase is better for convergence. The proposed strategies are flexible and can be easily modified to incorporate other local search methods.

Future work can focus on using a heuristic to implement a collaboration and competition mechanism where a pool of memes are used that use different types of local search. Application of the proposed approach can be extended to other architectures that include recurrent neural networks.

REFERENCES

[1] N. Krasnogor and S. Gustafson, “Toward truly ”memetic” memetic algorithms: discussion and proofs of concept,” in *Advances in Nature-Inspired Computation: The PPSN VII Workshops*, 2002.

[2] A. Sinha and D. E. Goldberg, “A survey of hybrid genetic and evolutionary algorithms,” *IlliGAL report*, vol. 2003004, 2003.

[3] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, “Real-coded memetic algorithms with crossover hill-climbing,” *Evolutionary Computation*, vol. 12, no. 3, pp. 273–302, 2004.

[4] F. Neri and C. Cotta, “Memetic algorithms and memetic computing optimization: A literature review,” *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.

[5] Y.-S. Ong, M. H. Lim, and X. Chen, “Memetic computationpast, present & future [research frontier],” *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.

[6] R. Dawkins, *The Selfish Gene*. Oxford University Press, 1976.
[7] P. Gancarski and A. Blansche, "Darwinian, lamarckian, and baldwinian (co)evolutionary approaches for feature weighting in k-means-based algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 617–629, Oct 2008.
[8] R. Meuth, M.-H. Lim, Y.-S. Ong, and D. C. Wunsch, "A proposition on memes and meta-memes in computing for higher-order learning," *Memetic Computing*, vol. 1, no. 2, pp. 85–100, 2009.
[9] G. Iacca, F. Neri, E. Mininno, Y.-S. Ong, and M.-H. Lim, "Ockhams razor in memetic computing: three stage optimal memetic exploration," *Information Sciences*, vol. 188, pp. 17–43, 2012.
[10] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. 1, pp. 1–29, 2000.
[11] Y. Liu, X. Yao, Q. Zhao, and T. Higuchi, "Scaling up fast evolutionary programming with cooperative coevolution," in *Evolutionary Computation, Proceedings of the 2001 Congress on*, San Diego, CA, USA, Jun. 2001, pp. 1101–1108.
[12] L. M. Antonio and C. A. Coello Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," *2013 IEEE Congress on Evolutionary Computation*, pp. 2758–2765, 2013.
[13] 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.
[14] ——, "Multi-objective cooperative coevolution of artificial neural networks (multi-objective cooperative networks)," *Neural Netw.*, vol. 15, no. 10, pp. 1259–1278, 2002.
[15] R. Chandra and M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction," *Neurocomputing*, vol. 186, pp. 116 – 123, 2012.
[16] R. Salomon, "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms," *Biosystems*, vol. 39, no. 3, pp. 263 – 278, 1996.
[17] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Accelerated neural evolution through cooperatively coevolved synapses," *J. Mach. Learn. Res.*, vol. 9, pp. 937–965, 2008.
[18] 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.
[19] R. Chandra, Y. S. Ong, and C. K. Goh, "Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction," *Neurocomputing*, vol. 243, pp. 21–34, 2017.
[20] 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.
[21] R. Chandra, "Memetic cooperative coevolution of elman recurrent neural networks," *Soft Comput.*, vol. 18, no. 8, pp. 1549–1559, 2014.
[22] G. Wong, R. Chandra, and A. Sharma, "Memetic cooperative neuro-evolution for chaotic time series prediction," in *Neural Information Processing*, A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, and D. Liu, Eds. Cham: Springer International Publishing, 2016, pp. 299–308.
[23] G. Wong and R. Chandra, "Enhancing competitive island cooperative neuro-evolution through backpropagation for pattern classification," in *International Conference on Neural Information Processing*. Springer, 2015, pp. 293–301.
[24] 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.
[25] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," *Evol. Comput.*, vol. 10, no. 4, pp. 371–395, 2002.
[26] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980*, ser. Lecture Notes in Mathematics, 1981, pp. 366–381.
[27] 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/
[28] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.
[29] A. S. Weigend and N. A. Gershenfeld, "Laser problem dataset: The santa fe time series competition data," 1994. [Online]. Available: http://www-psych.stanford.edu/∼andreas/Time-Series/SantaFe.html
[30] E. Lorenz, "Deterministic non-periodic flows,," *Journal of Atmospheric Science*, pp. 267–285, 1963.
[31] Admin, "Exchange rate (twi). may 1970 aug 1995." Feb 2014. [Online]. Available: https://datamarket.com/data/set/22tb