2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Learning Large-scale Fuzzy Cognitive Maps using a Hybrid of Memetic Algorithm and Neural Network

Yaxiong Chi, Jing Liu*
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education
International Research Center for Intelligent Perception and Computation
Xidian University, China
*Corresponding author, email: neouma@mail.xidian.edu.cn

***Abstract*—Fuzzy cognitive maps (FCMs) are cognition fuzzy influence graphs, which are based on fuzzy logic and neural network. In this paper, we propose a novel method combining Memetic Algorithms (MAs) and Neural Networks (NNs) to learn large-scale FCMs, which is labeled as MA-NN-FCM. In MA-NN-FCM, MAs are used to determine the regulatory connections in the network from multiple observed response sequences and NNs are used to calculate the interactions between concepts. In the experiments, the performance of MA-NN-FCM is validated on synthetic data with different number of nodes. The experimental results demonstrate the efficiency of our method, and show MA-NN-FCM can construct FCMs with high accuracy without expert knowledge. The performance of MA-NN-FCM is better than that of other FCM learning algorithms, such as ant colony optimization, non-linear Hebbian learning, and real-coded genetic algorithm.**

Keywords: **Fuzzy cognitive maps, memetic algorithms, neural networks.**

I. INTRODUCTION

Fuzzy cognitive maps (FCMs), introduced by Kosko [1], are a kind of effective understand tools for creating models of complex systems. An FCM is a fuzzy-graph representing causal reasoning which consists of nodes and weighted edges. Nodes in the graph stand for real world concepts (events, actions, values, goals, etc.) and weighted edges represent the relations between nodes. FCMs have several advantages in terms of abstraction, flexibility, adaptability, and fuzzy reasoning than traditional modeling techniques such as expert systems and neural networks. Therefore, they have been proposed and applied in a variety of applications such as medical diagnosis [2, 3, 4], time series analysis [5, 6], control [7], political and social sciences [8], business [9], information technology [10], and modeling of software development project [11, 12].

Papageorgiou *et al.* in [13] reviewed the learning algorithms which have been used to improve the quality of FCMs systematically. All these learning approaches are concentrated mainly on modifying the FCM weight matrix, which is updated on the basis of experts' knowledge and/or historical data. On the basis of the type of knowledge used, the most important learning approaches can be classified into three types, i.e., Hebbian-based, evolutionary-based, and hybrid of Hebbian-based and evolutionary-based learning algorithms, where evolutionary algorithms are a kind of stochastic global optimization methods inspired by the biological mechanism of evolution and heredity, and have been successfully used to solve various problems [14-19].

Among the Hebbian-based learners, one representative method is the nonlinear Hebbian learning (NHL) algorithm proposed by Papageorgiou *et al.* in [20, 21]. But the NHL method requires experts to build an initial network with expert knowledge. However, some application areas are difficult for experts to build FCMs and this method is not available to reconstruct the regulatory networks with only limited prior knowledge. Therefore, evolutionary-based learning algorithms such as genetic algorithm [22], particle swarm optimization [23] and artificial bee colony algorithm [24] have been proposed to learn FCMs directly from data. But most of the evolutionary-based learning methods optimizing the floating point weights in a network are slow.

Therefore, we propose a hybrid method for learning FCMs by combining memetic algorithms (MAs) and neural networks (NNs), which is named as MA-NN-FCM. The MA is an evolutionary-based algorithm that was first introduced by Moscato in his technical report [25] in 1989. The method is widely used as a synergy of evolutionary with separate local improvement procedures for problem search. MAs have been proved to be more efficient than conventional evolutionary algorithms in a variety of problem. The advantage of MAs lies in its effective exploitation both in global and local search space.

In our method, one chromosome represents a subset of nodes in the FCM model. Appropriate crossover, mutation, and local search operators are employed to search for the best combination of nodes. The fitness of each chromosome is evaluated by a signal-layer NN which is used to calculate the influence of a concept affect another. As the network structure is unknown, the value of interconnection weight between nodes is difficult to be determined. Our method can not only find the plausible combinations of nodes, but also optimize the interconnection weights matrix rapidly. In the experiments, MA-NN-FCM is applied on various FCM models with 20 to 100 nodes. The results show that the

This work is partially supported by the National Natural Science Foundation of China under Grants 61271301 and 61103119, the Research Fund for the Doctoral Program of Higher Education of China under Grant 20130203110010, and the Fundamental Research Funds for the Central Universities under Grant K5051202052.

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

proposed method is capable of learning FCM models effectively. The rest of this paper is organized as follows: Section 2 gives an introduction on FCMs, while our learning algorithm is presented in Section 3. In Section 4, experimental results are reported. Finally, Section 5 concludes the paper.

II. Fuzzy Cognitive Maps

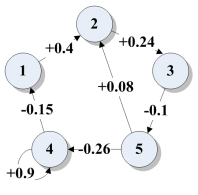
Fuzzy cognitive maps are cognitive maps within which the relations between the elements (e.g. concepts, events, project resources) of a "mental landscape" can be used to compute the "strength of impact" of these elements [1]. An FCM is a signed fuzzy digraph, structured as N_n concept nodes, and the state values of these nodes are denoted as a vector $\boldsymbol{C}$,

$$\boldsymbol{C} = \left[C_1,\ C_2,\ ...,\ C_{N_n} \right] \tag{1}$$

where $C_i \in [0, 1]$, i=1, 2, ..., N_n. That is, each concept node has a state value ranging from 0 to 1. For example, if the concept representing "Appreciation of Scary Movies", we use 1 to express strongest appreciation, and 0.5 to express neutral emotion, and 0 to express maximum disgusting. And the casual relationships between concept nodes are defined as an $N_n \times N_n$ weight matrix $\boldsymbol{w}$,

$$\boldsymbol{w} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N_n} \\ w_{21} & w_{22} & \cdots & w_{2N_n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N_n1} & w_{N_n2} & \cdots & w_{N_nN_n} \end{bmatrix} \tag{2}$$

where w_{ij} is in the range of [-1, 1], and represents the relationship between concepts i and j, i, j=1, 2, …, N_n. Each edge has a weight which indicates how much one concept affects another. Values range from -1 (strongest negative impact), through 0 (no impact), to +1 (strongest positive impact). Fig.1(a) shows a simple example with 5 concept nodes, and the corresponding weight matrix is shown in Fig.1(b). For example, w_{12}=0.4 denotes there is a positive excitatory pointing from node 1 to 2 with 0.4 strength. w_{13}=0 means there is no relationship between nodes 1 and 3. Similarly, w_{44}=0.9 suggests that node 4 has a positive feedback on itself.



$$\boldsymbol{w} = \begin{bmatrix} 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0.24 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1 \\ -0.15 & 0 & 0 & 0.9 & 0 \\ 0 & 0.08 & 0 & -0.26 & 0 \end{bmatrix}$$

(a) (b)

Fig.1. An FCM with 5 concept nodes, (a) the topological structure, (b) the corresponding weight matrix.

The state value of a concept node at the (t+1)th iteration is determined by the strength of the relationship and state values of related concept nodes at the tth iteration. Thus, the dynamics of FCMs is determined by the following equation,

$$C_i^{t+1} = g\left(\sum_{j=1}^{N_n} w_{ji} C_j^t \right) \tag{3}$$

where C_i^t is the state value of node i at the tth iteration in one response, and $g(\cdot)$ is a transfer function that is used to bound the state value to the range of [0, 1], namely the range of allowable state values. Various transfer functions can be used. According to the comparison study in [26], the sigmoid transfer function outperforms the others in general. Thus, the following sigmoid transfer function is employed,

$$g(x) = \frac{1}{1 + e^{-\lambda x}} \tag{4}$$

where λ is a parameter used to characterize the steepness of the function around zero, and its value should be chosen according to the nature of the problem. Usually a small λ is suitable for highly nonlinear system. In this paper, λ is set to 5 because this is a widely used value in many FCM learning research [27].

III. Proposed Method

The proposed approach is based on taking a node for learning and updating target node for the next learning step. A similar method named as "Concept by Concept (CbC) learning" which use the Big Bag-Big Crunch learning method as the global learning method is also presented in [28]. In this paper we proposed a hybrid method combines memetic algorithms with a single-layer NN. As an evolutionary-based learning algorithm, MA is fast in finding a set of regulated nodes, but slow in searching for the floating interconnection weights. And Neural Networks are a kind of efficient tools to compute values from inputs by feeding information through the network. Therefore, the neural component uses the gradient descent strategy to determine the weights. The proposed algorithm is named as MA-NN-FCMs. The details of each step are described as follows,

STEP 1

If all nodes have been optimized, output the interaction matrix learned; otherwise choose a node that has not been optimized as the target node.

STEP 2

Initialize the population of chromosomes. Each gene, which represents one of the node in the network (the ith gene represents for the ith node), is initialized with the state value chosen from the historical data of temporal pairs. As we have no idea about the structure of the FCM, we initialize each gene a sign (0 or 1) randomly to indicate whether this node is "open", and the sign follows the gene recombination, mutation and so on. For example, the sign of ith gene in a chromosome is 1 means the ith node is allowed to affect the target node, whereas the sign of ith gene in a chromosome is 0 means the ith node is forbidden to affect the target node. A

chromosome and the target node form a training set of input-output pairs. The population is a set of chromosomes and each chromosome is evaluated by assigning to it a fitness function value (see STEP 4).

STEP 3

Use the gradient descent method to calculate the value of weights between the "open" nodes in a chromosome and the target node (the weights between "closed" nodes and the target node are set to 0). This process needs historical data that consist of multiple sequences of state vectors. If the historical data is observed from state 1 to N, we can get N-1 training sets to calculate the values of weights. This technique used the back propagation equations:

$$\Delta w_{ij} = \eta \sum_{p=1}^{K} (\mathrm{d}_p - \mathrm{y}_p)\, \mathrm{y}_p (1 - \mathrm{y}_p)\, \mathrm{x}_{ip} \tag{5}$$

where d_p and y_p is the observed value and estimated value of the pth training set, respectively, x_{ip} is the ith gene of the chromosome in the pth training set. K is the number of training sets. η is the learning coefficient which defines the learning capability of a neural network during training. After several iterations, the weight values between target node and input nodes can be determined.

STEP 4

Calculate the estimated data sequences of the target node according to (3). Suppose we have N_s response sequences of each gene, and the length of each sequence is N_t. Then, the problem can be formulated as a numerical optimization problem, and the objective is to minimize the following function,

$$Output\_Error = \frac{1}{N_s (N_t - 1)} \sum_{\substack{1 \le t \le N_t - 1 \\ 1 \le s \le N_s}} (C^t_{\text{target}}(\mathrm{s}) - \hat{C}^t_{\text{target}}(\mathrm{s}))^2 \tag{6}$$

where $C^t_{target}(\mathrm{s})$ is the tth value of the target node in the sth observed data series. $\hat{C}^t_{target}$ is the tth value of the target node in the sth estimated data series. *Output_Error* is used to measure the differences between the observed data sequence and the estimated data sequence, and is the fitness value for each chromosome.

STEP 5

Use the standard crossover and mutation operators on all chromosomes in the current population to vary the combination of nodes.

STEP 6

Select 10% individuals which have higher fitness in the population, and conduct the local search on these individuals as follows. For each selected chromosome, change the sign of an arbitrary single gene. If the change produces a chromosome with better fitness, then keep this change; otherwise, reject this change and change the sign of another arbitrary single gene, repeating the above process until the stop criteria of local search is met.

STEP 7

2-tournament selection is employed to generate the next population.

STEP 8

If the termination criteria are met, record the calculated weights of the interconnection matrix based on the best chromosome, and go to STEP 1; otherwise go to STEP 3.

Since the historical data which is used to train MA-NN-FCM need to be split into several input-output pairs; that is to say, the current state vector depends only on the state vector at the preceding iteration and does not depend on any other state vectors, even if the historical data matrix has missing rows, MA-NN-FCM is still capable of handling it.

IV. Experiments

We used the method in [29] to generate simulated data to test the performance of MA-NN-FCM. First, the interconnection matrix of FCM is constructed by random values in the interval [-1, 1]. The value will be set to 0 if the absolute value of the generated random number is less than 0.05. This is because causal relation with strength less than 0.05 usually has no significance in practical problems [30]. In consideration to the FCM model are always sparse in actual world, only a small number of weights are random numbers and the others are set to 0. Second, initial state values in the range of [0, 1] are created randomly and assigned to each node. The response sequence is generated by (3). The parameters of MA-NN-FCM are presented in Table I.

TABLE I
THE PARAMETER SETTING OF MA-NN-FCM

Parameters	Values
Crossover	*Single point*, 0.9
Mutation	*Random*, 0.1
Selection	*Binary Tournament*
Local-search	*Hill climbing method*
Gradient descent	*Standard/Step*
Learning coefficient	0.9
Iterations	20-100

A. Performance Measures

Data Error, *Model Error*, and *SS Mean* are calculated to evaluate the performance. *Data Error* is used to measure the differences between the observed data sequence and estimated data sequence and is defined as

$$Data\ Error = \frac{1}{N_n N_s (N_t - 1)} \sum_{\substack{1 \le t \le N_t - 1 \\ 1 \le n \le N_n \\ 1 \le s \le N_s}} (\mathrm{C}^t_n(\mathrm{s}) - \hat{\mathrm{C}}^t_n(\mathrm{s}))^2 \tag{7}$$

Model Error is used to compare the weight matrix of learned FCM model and the target FCM model directly, and is defined as

$$Model\ Error = \frac{1}{N_n{}^2} \sum_{i=1}^{N_n} \sum_{j=1}^{N_n} \left| w_{ij} - \hat{w}_{ij} \right| \tag{8}$$

In order to predict the existence of an edge between two nodes, the learned weight need to be transformed to binary one based on the rule described in [29]; that is, if the absolute value of weight less than 0.05, it is set to 0 and defined as positive results; otherwise set to 1 and defined as negative result. *SS Mean* is also calculated to identify the plausible causal relations (existence or non-existence) among the FCM nodes. *SS Mean* is calculated by the following equation,

$$SS\ mean = \frac{2 \times Specificity \times Sensitivity}{Specificity + Sensitivity} \tag{9}$$

where

$$Specificity = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{10}$$

$$Sensitivity = \frac{N_{TN}}{N_{TN} + N_{FP}} \tag{11}$$

where N_{TP} is the number of true positives, N_{FN} is the number of false negatives, N_{TN} is the number of true negatives, N_{FP} is the number of false positives. They are defined in Table II.

TABLE II. The definition of *TP*, *FN*, *TN*, and *FP*.

		Target Networks	
		0	1
Learned Networks	0	*TP*	*FP*
	1	*FN*	*TN*

B. Experimental Results

We test the proposed algorithm on the data sets with various scales and properties. The number of nodes (N_n) is set to 20, 40, and 100, respectively, and the edge density is set to 20% and 40% for each scale. Three sets of experiments are conducted. The first set of experiments is conducted on one sequence with 20 time points per node (N_s=1, N_t=20). The second set of experiments is conducted on 5 response sequences with 4 time points per sequence (N_s=5, N_t=4). The third set of experiments is conducted on 40 response sequences with 10 time points per sequence (N_s=40, N_t=10). The experimental results are compared with those of NHL [20], DDNHL [31], RCGA [22], ACO [29], ACO_{RD} [32] in terms of *Data Error*, *Model Error*, and *SS Mean*. The results are reported in Tables III and IV. All results are averaged over 30 independent runs for FCMs with 20-40 nodes and 15 independent runs for FCMs with 100 nodes.

As can be seen from the results, in terms of *Data Error*, ACO_{RD} performs well when the number of nodes is 20, but among the six cases for 40 nodes, the performance of MA-NN-FCM is the best in 3 cases and performs better than ACO_{RD} in 5 cases. In terms of *Model Error*, MA-NN-FCM outperforms ACO, RCGA, DDNHL, and NHL in all cases, and the performance is better than ACO_{RD} in most cases for 20, 40, and 100 nodes.

We also compare the *SS mean* of different algorithms to show the ability of MA-NN-FCM in predicting the existence of links. The results show that MA-NN-FCM is better than ACO_{RD}, ACO, RCGA, DDNHL, and NHL in all cases except three.

TABLE III
The comparison in terms of *Data Error*, *Model Error*, and *SS mean* with the density 20% (Average±Standard Deviation)

Algorithm(N_s-N_t)	*Data Error*	*Model Error*	*SS mean*
	#Nodes = 20, Density = 20%		
MA-NN-FCM(1-20)	**0.001**±0.000	**0.141**±0.028	**0.47**
ACO_{RD}(1-20)	0.004±0.003	0.299±0.033	0.24
ACO(1-20)	0.002±0.001	0.513±0.026	0.13
MA-NN-FCM(5-4)	0.013±0.008	**0.103**±0.010	**0.62**
ACO_{RD} (5-4)	**0.008**±0.004	0.242±0.040	0.25
ACO(5-4)	0.011±0.005	0.461±0.019	0.16
MA-NN-FCM(40-10)	0.008±0.004	0.093±0.037	0.69
ACO_{RD} (40-10)	**0.000**±0.000	**0.014**±0.021	**0.97**
ACO(40-10)	0.009±0.002	0.348±0.017	0.21
RCGA	/	0.426	0.16
DDNHL	/	0.464	0.14
NHL	/	0.461	0.13
	#Nodes = 40, Density = 20%		
MA-NN-FCM(1-20)	**0.001**±0.001	**0.150**±0.007	**0.37**
ACO_{RD}(1-20)	0.033±0.008	0.388±0.098	0.14
ACO(1-20)	0.006±0.006	0.525±0.011	0.13
MA-NN-FCM(5-4)	**0.012**±0.002	**0.113**±0.007	**0.48**
ACO_{RD} (5-4)	0.144±0.011	0.390±0.054	0.14
ACO(5-4)	0.020±0.005	0.508±0.010	0.12
MA-NN-FCM(40-10)	**0.017**±0.011	**0.094**±0.006	**0.49**
ACO_{RD} (40-10)	0.107±0.009	0.205±0.065	0.35
ACO(40-10)	0.019±0.006	0.448±0.021	0.15
RCGA	/	0.453	0.15
DDNHL	/	0.468	0.12
NHL	/	0.489	0.15
	#Nodes = 100, Density = 20%		
MA-NN-FCM(1-20)	**0.065±0.010**	0.469±0.042	**0.48**
ACO_{RD}(1-20)	0.283±0.008	**0.424**±0.031	0.14
ACO(1-20)	/	/	/
MA-NN-FCM(5-4)	**0.026±0.002**	**0.116**±0.002	**0.32**
ACO_{RD} (5-4)	0.170±0.012	0.400±0.035	0.15
ACO(5-4)	/	/	/
MA-NN-FCM(40-10)	**0.054±0.004**	**0.106**±0.003	**0.41**
ACO_{RD} (40-10)	0.300±0.021	0.286±0.042	0.26
ACO(40-10)	/	/	/

V. CONCLUSION

A novel learning algorithm, which combines memetic algorithm and neural network, is proposed for learning the interconnection weight matrix of an FCM. The experiment results show that the proposed algorithm is efficient to learn FCMs with 100 nodes. Future work will include the application of the hybrid method on more complex problems as well as practical problems.

REFERENCES

[1] B. Kosko, "Fuzzy cognitive maps", *International Journal of Man-Machine Studies*, vol. 24, pp. 65-75, 1986.

[2] E. I. Papageorgiou, W. Froelich, "Application of evolutionary fuzzy cognitive maps for prediction of pulmonary infections", *IEEE Transactions on Information Technology in Biomedicine*, 16(1), 143-149 (2012).

[3] W. Froelich and A. Wakulicz-Deja, "Mining temporal medical data using adaptive fuzzy cognitive maps," in *Proc. Conf. Human Syst. Interact.*, Catania, Italy, 2009, pp. 13-20.

[4] E. I. Papageorgiou, "A new methodology for decisions in medical informatics using fuzzy cognitive maps based on fuzzy rule-extraction techniques," *Appl. Soft Comput.*, 11, pp.500-513, 2011.

TABLE IV
The comparison in terms of *Data Error*, *Model Error*, and *SS mean* with the density 40% (Average±Standard Deviation)

Algorithm(N_s-N_t)	*Data Error*	*Model Error*	*SS mean*
	#Nodes = 20, Density = 40%		
MA-NN-FCM(1-20)	**0.001**±0.001	**0.252**±0.023	**0.42**
ACO_{RD}(1-20)	0.003±0.002	0.367±0.011	0.16
ACO(1-20)	**0.001**±0.001	0.517±0.022	0.14
MA-NN-FCM(5-4)	0.019±0.007	**0.196**±0.013	**0.54**
ACO_{RD} (5-4)	0.009±0.005	0.326±0.078	0.18
ACO(5-4)	**0.006**±0.002	0.473±0.020	0.14
MA-NN-FCM(40-10)	0.012±0.004	0.145±0.012	0.63
ACO_{RD} (40-10)	**0.000**±0.000	**0.027**±0.012	**0.93**
ACO(40-10)	0.006±0.002	0.328±0.031	0.24
RCGA	/	0.413	0.14
DDNHL	/	0.436	0.13
NHL	/	0.468	0.10
	#Nodes = 40, Density = 40%		
MA-NN-FCM(1-20)	0.057±0.099	**0.240**±0.008	**0.39**
ACO_{RD}(1-20)	0.023±0.008	0.416±0.054	0.14
ACO(1-20)	**0.004**±0.002	0.562±0.010	0.12
MA-NN-FCM(5-4)	0.027±0.014	**0.211**±0.024	**0.44**
ACO_{RD} (5-4)	0.073±0.005	0.420±0.061	0.12
ACO(5-4)	**0.014**±0.005	0.538±0.007	0.13
MA-NN-FCM(40-10)	0.036±0.007	0.194±0.008	0.43
ACO_{RD} (40-10)	0.040±0.006	**0.188**±0.022	**0.44**
ACO(40-10)	**0.019**±0.004	0.437±0.012	0.16
RCGA	/	0.436	0.15
DDNHL	/	0.465	0.12
NHL	/	0.498	0.12
	#Nodes = 100, Density = 40%		
MA-NN-FCM(1-20)	**0.069±0.010**	0.489±0.010	**0.49**
ACO_{RD}(1-20)	0.105±0.032	**0.434**±0.024	0.14
ACO(1-20)	/	/	/
MA-NN-FCM(5-4)	**0.035±0.008**	**0.210**±0.003	**0.32**
ACO_{RD} (5-4)	0.057±0.028	0.438±0.029	0.14
ACO(5-4)	/	/	/
MA-NN-FCM(40-10)	**0.059±0.006**	**0.204**±0.001	**0.29**
ACO_{RD} (40-10)	0.170±0.011	0.309±0.028	0.24
ACO(40-10)	/	/	/

[5] H. Song, C. Miao, W. Roel, Z. Shen, and F. Catthoor, "Implementation of fuzzy cognitive maps based on fuzzy neural network and application in prediction of time series," *IEEE Trans. Fuzzy System*, 18(2), pp. 233-250, 2010.

[6] W. Stach, L. A. Kurgan, and W. Pedrycz, "Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps," *IEEE Trans. Fuzzy System*, 16(1), pp.61-72, 2008.

[7] J. L. Gonzalez, L. T. Aguilar, O. Castillo, "A cognitive map and fuzzy inference engine model for online design and self fine-tuning of fuzzy logic controllers", *International Journal of Intelligent System* 24(11), 1134-1173 (2009).

[8] A. S. Andreou, N. H. Mateou, G. A. Zombanakis, "Soft computing for crisis management and political decision making: the use fo genetically evolved fuzzy cognitive maps", *Soft Computing Journal* 9(3), 194-210 (2006).

[9] M. Glykas, "Fuzzy cognitive strategic maps in business process performance measurement" *Expert Systems with Applications* 40(1), 1-14 (2013).

[10] K. C. Lee, S. Lee, "A causal knowledge-based expert system for planning an internet-based stock trading system", *Expert Systems With Applications* 39(10), 8626-8635 (2012).

[11] W. Stach and L. Kurgan, "Modeling software development project using fuzzy cognitive maps", *Proc. 4th ASERC Workshop on Quantitative and Soft Software Engineering*, 2004, pp. 55-60.

[12] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, "Parallel fuzzy cognitive maps as a tool for modeling software development project", *Proc. 2004 North American Fuzzy Information Processing Society Conf. (NAFIPS'04)*, Banff, AB, 2004, pp. 28-33.

[13] E. I. Papageorgiou, "Learning algorithms for fuzzy cognitive maps - a review study," *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 42(2), pp.150-163, 2012.

[14] W. Zhong, J. Liu, X. Xue, and L. Jiao, "A multiagent genetic algorithm for global numerical optimization," *IEEE Trans. on Syst., Man, and Cybern., Part B*, 34(2), 1128-1141, 2004.

[15] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for constraint satisfaction problems," *IEEE Trans. on Syst., Man, and Cybern., Part B*, 36(1), 54-73, 2006.

[16] J. Liu, W. Zhong, and L. Jiao, "Moving block sequence and organizational evolutionary algorithm for general floorplanning with arbitrarily shaped rectilinear blocks," *IEEE Trans. on Evolutionary Computation*, 12(5), 630-646, 2008.

[17] L. Jiao, J. Liu, and W. Zhong, "An organizational coevolutionary algorithm for classification," *IEEE Trans. on Evolutionary Computation*, 10(1), 67-80, 2006.

[18] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for combinatorial optimization problems," *IEEE Trans. on Syst., Man, and Cybern., Part B*, 40(1), 229-240, 2010.

[19] J. Liu, H.A. Abbass, D.G. Green, and W. Zhong, "Motif difficulty (MD): A predictive measure of problem difficulty for evolutionary algorithms using network motifs," *Evolutionary Computation Journal (MIT)*, 20(3): 321-347, 2012.

[20] E. I. Papageorgiou, C. D. Stylios, and P. P. Groumpos, "Fuzzy cognitive map learning based on nonlinear hebbian rule", in *Proc. Australian Conference on Artificial Intelligence*, 2003, pp.256-268.

[21] E. I. Papageorgiou and P. P. Groumpos, "A weight adaptation method for fine-tuning fuzzy cognitive map causal links," *Soft Comput,* pp. 846-857, J, 9, 2005.

[22] W. Stach, L. Kurgan, W. Pedrycz and M. Reformat. "Genetic learning of fuzzy cognitive maps". *Fuzzy Sets and Systems,* 153, 3 (2005), 371-401.

[23] E. I. Papageorgiou, K. E. Parsopoulos, C. S. Stylios, P. P. Groumpos and M. N. Vrahatis. "Fuzzy cognitive maps learning using particle swarm optimization". *Journal of Intelligent Information Systems,* 25(2005), 95-121.

[24] E. Yesil, C. Ozutrk, M. F. Dodurka, A. Sakalli, "Fuzzy cognitive maps learning using artificial bee colony optimization", In *IEEE Int. Conf. Fuzzy Systems* (2013).

[25] P. Moscato. "On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms". *Caltech Concurrent Computation Program* C3P Report, 826, 1989.

[26] K. Tsadiras, "Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps", *Information Sciences*, vol. 178, pp. 3880-3894, 2008.

[27] W. Stach, "Learning and aggregation of fuzzy cognitive maps – an evolutionary approach", PhD Dissertation, *University of Alberta*, 2010.

[28] M. F. Dodurka, E. Yesil, C. Ozturk, A. Sakalli, C. Guzay, "Concept by concept learning of fuzzy cognitive maps", *The Artificial Intelligence Applications and Innovations Conference* (*AIAI 2013*), Paphos, Cyprus, 2013.

[29] Y. Chen, L. J. Mazlack, and L. J. Lu, "Learning fuzzy cognitive maps from data by ant colony optimization", In: *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pp. 9-16, 2012.

[30] W. Stach, L. Kurgan, W. Pedrycz, and M. Reformat, "Evolutionary development of fuzzy cognitive maps", In *Proceedings of the IEEE International Conference on Fuzzy Systems*, *FUZZ-IEEE* 2005 United states, pp.619-624, 2005.

[31] W. Stach, L. Kurgan, and W. Pedrycz, "Data driven nonlinear hebbian learning method for fuzzy cognitive maps", in *Proc. IEEE World Congr. Comput. Intell.*, Hong Kong 2008. pp.1975-1981.

[32] Y. Chen, L. J. Mazlack, and L. J. Lu, "Inferring fuzzy cognitive map models for gene regulatory networks from gene expression data", *IEEE International Conference on Bioinformatics and Biomedicine* (*IEEE BIBM* 2012), 589-601, Philadelphia, USA, October, 2012.