

Training Fuzzy Cognitive Maps by Using Hebbian Learning Algorithms: A Comparative Study

G.A. Papakostas, A.S. Polydoros, D.E. Koulouriotis and V.D. Tourassis

Department of Production and Management Engineering

Democritus University of Thrace (DUTH)

Xanthi, Greece

gpapakos@ee.duth.gr, {athapoly,jimk,vtourasi}@pme.duth.gr

Abstract—A detailed analysis of the Hebbian-like learning algorithms applied to train Fuzzy Cognitive Maps (FCMs) is presented in this paper. These algorithms aim to find appropriate weights between the concepts of the FCM so the model equilibrates to a desired state. For this manner, four different types of Hebbian learning algorithms have been proposed in the past. Along with the theoretical description of these algorithms, their performance in system modeling problems is investigated in this work. The algorithms are studied in a comparative fashion by using appropriate performance indices and useful conclusions about their training capabilities are experimentally derived.

Keywords—fuzzy cognitive maps; hebbian learning; training; system modeling

I. INTRODUCTION

An increased interest about the theory and application of the Fuzzy Cognitive Maps (FCMs) [1] in engineering science is noted recently. FCMs are characterized by their ability to model the dynamics of complex systems by incorporating the casual relationships between the main concepts that describe the system. They have been used initially to model complex social and financial systems [2,3], where analytical descriptions do not exist or can not be derived.

The design of the FCMs is highly dependent on the knowledge of the experts, which define the model's concepts and their interconnections among them. This dependency may significantly affect the modeling accuracy due to the imprecise expert's knowledge.

In order to enhance the knowledge storing capabilities of the FCM models, learning algorithms that search for the best weight values of the concepts' interconnections have been proposed. Mainly, there are two categories of learning algorithms: (1) algorithms that use the Hebbian adaptation rule (Hebbian-like) [5-11] and (2) algorithms that use evolutionary optimization mechanisms (Evolutionary-like) [12-14].

Although the algorithms of the second category ensure near-optimum solutions in the weights search space, the Hebbian-like algorithms of the first category are studied in this work, due to their popularity and quick convergence to desirable FCM states.

The first usage of the Hebbian rule in learning FCMs was performed by Dickerson and Kosko [5], who proposed the *Differential Hebbian Learning* (DHL) algorithm by using the generalized Hebbian rule introduced by Oja [15]. Since there, three other types of Hebbian learning algorithms were proposed, the *Nonlinear Hebbian Learning* (NHL) [7,8], *Data-Driven Nonlinear Hebbian Learning* (DD-NHL) [9] and *Active Hebbian Learning* (AHL) [10,11] ones.

Despite of the review of the Hebbian-like learning algorithms from the literature, the main contribution of this work is the comparison of these algorithms under the same conditions. On the author's best knowledge this kind of comparison is missing from the literature, although such an analysis has to be performed whether a new algorithm is proposed.

The paper is organized, by describing shortly the fundamental theory of the FCMs in section 2, by discussing the Hebbian-like learning algorithms in section 3, by comparing the learning capabilities of the algorithms in system modeling problems in section 4 and finally, by summarizing some to useful conclusions in section 5.

II. FUZZY COGNITIVE MAPS

Fuzzy Cognitive Maps (FCMs) are fuzzy signed directed graphs with feedback [4]. They were proposed by Kosko [6] as a modeling methodology of complex systems, able to describe the casual relationships between the main factors (concepts) that determine the dynamic behaviour of a system.

In their graphical presentation, FCMs have the form of a diagram with nodes having interconnections between them, as illustrated in the following Fig. 1.

As it can be seen from Fig. 1, a FCM consists of nodes (concepts), C_i , $i=1,2,3,\dots,N$, where N is the total number of concepts, which are characteristics, main factors, or properties of the system being modeled. The concepts are interconnected through arcs having weights that denote the cause and effect relationship that a concept has on the others. There are three possible types of casual relationships between two concepts C_i and C_j (i) positive, meaning that an increase or decrease in the value of a cause concept results in the effect concept moving in the same direction and it is described by a positive weight W_{ij} , (ii) negative, meaning that changes in the cause and effect

concepts take place in opposite directions, and the weight W_{ij} has negative sign and (iii) no relation, with the interconnection weight being zero. The weight value, e.g. W_{ij} , denotes the strength of how casual concept C_i causes C_j and takes values in the range $[-1,1]$.

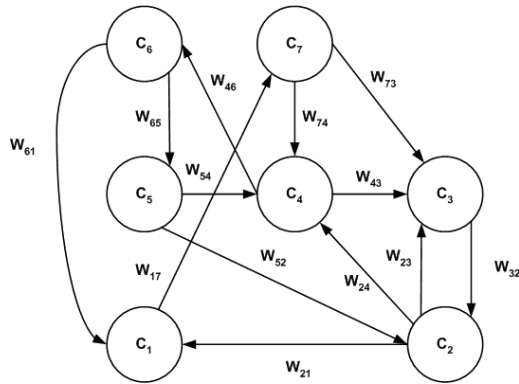


Figure 1. A simple Fuzzy Cognitive Maps

At each time, the value of each concept A_i is calculated, by summarizing the influences of all the other concepts to this and by squashing the overall impact using a barrier function f , according to the following rule

$$A_i^{(t+1)} = f \left(A_i^{(t)} + \sum_{j=1, j \neq i}^N W_{ji} A_j^{(t)} \right) \quad (1)$$

where A_i^{t+1} and A_i^t are the values of concept C_i at times $t+1$ and t respectively, A_j^t the value of concept C_j at time t , W_{ji} the weight value of the interconnection with direction from concept C_j to concept C_i , and f the barrier function used to restrict the concept value into a specific range, usually in the range $[0,1]$.

In each step, a new state of the concepts is derived according to (1) and after a number of iterations, the FCM may arrive in one of the following states: (1) a fixed equilibrium point, (2) a limited cycle and (3) a chaotic behaviour.

When the FCM arrives in a fixed equilibrium point, we can conclude that the FCM has converged and the final state corresponds to the actual system state in which the system concludes, when the initial values of concepts are applied.

The design of the FCMs is significantly based on the experience of some experts, who have the knowledge of the system being modeled and provide the weight values for the interconnection between concepts. In more flexible FCM structures these weights are being found during a learning procedure, in a similar way as in the case of neural networks training. For this reason many researchers have borrowed and adjusted algorithms from the neural network theory and proposed new learning procedures [5-9].

In the present paper, the Hebbian-like learning algorithms [5-11] is used in order to find the optimal weight set of the FCMs under investigation and their performance is compared to each other.

III. HEBBIAN LEARNING ALGORITHMS

Hebbian learning constitutes an unsupervised technique initially applied on the training of artificial neural networks [16]. The main feature of this learning rule is that the change of a synaptic is computed by taking into account the presynaptic and postsynaptic signals flow towards each processing unit (neuron) of a neural network and is defined by the following Eq.(2).

$$\Delta w_{ij} = \eta y_i(n) x_j(n) \quad (2)$$

where η is a positive constant that determines the *rate of learning* and x_j and y_i is the presynaptic and postsynaptic signals. The above rule has been modified by Oja [15] in order to solve the stability problems imposed in Eq.(2) by introducing a generalized form of the Hebbian rule called *Oja's rule*, having the following form:

$$\Delta w_{ij} = \eta y_i(n) (x_j(n) - y_i(n) w_{ij}(n)) \quad (3)$$

Based on the Oja's rule, several Hebbian like learning algorithms have been proposed in the literature, initially for the case of neural networks and recently, after appropriate modifications, for training FCMs.

A short description of the Hebbian like algorithms, having the form of Eq.(2), proposed as learning procedures that find an appropriate set of interconnection weights (connection matrix) subject to a system's requirements is presented hereafter.

A. Differential Hebbian Learning (DHL)

The DHL algorithm was the first Hebbian learning algorithm proposed by Dickerson and Kosko [5] in order to find better interconnection weights than those provided by the experts participated to the FCM design. This algorithm correlates the changes in the concept nodes of the model in order to modify their weights, according to the following Eq.(4).

$$w_{ij}^{(t+1)} = \begin{cases} w_{ij}^{(t)} + \mu_t (\Delta A_i^{(t)} \Delta A_j^{(t)} - w_{ij}^{(t)}), & \Delta A_i^{(t)} \neq 0 \\ w_{ij}^{(t)} & , \Delta A_i^{(t)} = 0 \end{cases} \quad (4)$$

where

$$\Delta A_i^{(t)} = A_i^{(t)} - A_i^{(t-1)} \quad (5)$$

Moreover, parameter μ_t refers to the learning rate which decreases with the algorithm's iteration as follows:

$$\mu_t = 0.1 \left[1 - \frac{t}{1.1N} \right] \quad (6)$$

where t is the current iteration and N is a constant that ensures the learning rate not to take negative values. An acceptable value of this constant is the maximum number of the algorithm's iterations.

B. Nonlinear Hebbian Learning (NHL)

The next Hebbian learning algorithm that mostly likes with the Oja's rule is the NHL proposed by Papageorgiou et al. [7], defined as:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta A_j^{(t)} (A_i^{(t)} - A_j^{(t)} w_{ij}^{(t)}) \quad (7)$$

where η is the learning rate. Besides the above basic formulation of the NHL algorithm two more extensions were introduced by Papageorgiou et al. [8], having the following forms:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \eta A_j^{(t)} (A_i^{(t)} - \text{sgn}(w_{ij}^{(t)}) A_j^{(t)} w_{ij}^{(t)}) \quad (8)$$

where $\text{sgn}(\cdot)$ corresponds to the sign function that returns the sign of a quantity and it is used to maintain the sign of the corresponding weight. The last NHL form is derived by adding a weight decay parameter γ in the update rule, as follows:

$$w_{ij}^{(t+1)} = \gamma w_{ij}^{(t)} + \eta A_j^{(t)} (A_i^{(t)} - \text{sgn}(w_{ij}^{(t)}) A_j^{(t)} w_{ij}^{(t)}) \quad (9)$$

In order to decrease the number of the free parameters, only the NHL version of Eq.(8) will be considered in the experimental section of the paper.

Despite of the new weight update rule of the NHL algorithm, the usage of two termination criteria [8] of the learning procedure was also a significant novel feature. These two criteria have the following form:

$$F_1 = \sqrt{\sum_{i=1}^n (OC_i^{(t)} - T_i)^2} \quad (10)$$

$$F_2 = |OC_i^{(t+1)} - OC_i^{(t)}| < e \quad (11)$$

where $OC_i^{(t)}$ is the i^{th} output concept at iteration t and T_i the target value of the output concept, which for the case of a constrained concept lies in a range of $[T_i^{\min}, T_i^{\max}]$ takes the

value $T_i = \frac{T_i^{\min} + T_i^{\max}}{2}$. The objective of the NHL learning

algorithm is to find a set of interconnection weights that minimizes the cost functions F_1 and F_2 (subject to a tolerance e), so the FCM model can reach a desired state. While the first criterion shows the error of the output concepts from the desired values, the second one ensures that the FCM model equilibrates to a final state.

A short description of the processing steps of the algorithm is as follows [7]:

Step 1: For a given initial state $\mathbf{A}^0$ and weight set $\mathbf{W}^0$.

Step 2: For each iteration k .

Calculate the concept values $\mathbf{A}^k$ by using Eq.(1).

Update weights $\mathbf{W}^k$ by using Eq.(8).

Until the termination criteria F_1 and F_2 are met.

Step 3: Return the final weights $\mathbf{W}^{\text{final}}$.

C. Data-Drive Nonlinear Hebbian Learning (DD-NHL)

A modified version of the NHL algorithm was proposed by Stach et al. [9], which incorporated historical data of the model into the learning procedure. This algorithm called Data-Driven Nonlinear Hebbian Learning (DD-NHL) and differs to the original NHL in two steps:

Difference 1: Instead of computing the concept values in each iteration by using Eq.(1), the concept values are provided by the historical data. These data is generated by applying the NHL algorithm [9] in a first stage.

Difference 2: The termination criterion F_1 is replaced by the measuring the output concepts that lie inside the desired target values. More precisely, the candidate weight set is used to simulate the FCM model and if the output concepts equilibrated to a desired state the learning is completed (with simultaneously satisfaction of the F_2 criterion), otherwise the procedure continues.

D. Active Hebbian Learning (AHL)

A common feature of the previous learning algorithms is that in each iteration the concepts of the FCM model are updated by using Eq.(1), synchronously.

The Active Hebbian Learning (AHL) algorithm considers that the concept nodes are activated asynchronously by a specific sequence. In this way the equilibrium point is reached by considering different times of nodes' activation, a mechanism that is useful in systems where the concepts are

activated based on a specific activation sequence. Moreover, unlike the previous algorithms where only the non-zero weights are being updated, in the case of the AHL algorithm all weights except those of the diagonal are updated.

Based on the AHL algorithm, the nodes of the FCM model is distinguished to *Activated* and *Activation* concepts, by the former ones being the nodes that firstly activated by causing the activation of the latter. In this case the activated concepts are updated through the following rule:

$$A_i^{(t+1)} = f \left(A_i^{(t)} + \sum_{j=1, j \neq i}^N W_{ji} A_j^{act(t)} \right) \quad (12)$$

where A_i are the value of the C_i activated concept and A_j^{act} the values of the activation concepts that influence the concept C_i . The weights are updated according to the following rule:

$$w_{ij}^{(t+1)} = (1 - \gamma^{(t)}) w_{ij}^{(t)} + \eta^{(t)} A_i^{(t)} (A_j^{(t)} - w_{ij}^{(t)} A_i^{act(t)}) \quad (13)$$

where the learning rate η and the weight decay γ are decreased exponentially with the algorithm's iteration number (activation-simulation cycles) as follows:

$$\begin{aligned} \eta^{(t)} &= b_1 e^{(-\lambda_1 t)} \\ \gamma^{(t)} &= b_2 e^{(-\lambda_2 t)} \end{aligned} \quad (14)$$

where $0.01 < b_1 < 0.09$, $0.1 < \lambda_1 < 1$ and b_2, λ_2 are positive constants determined by trial and error. Furthermore, the two termination criterion of the NHL is also applied in the case of AHL in order to derive a desired equilibrium point of the FCM model.

The DHL, NHL, DD-NHL and AHL algorithms will be used to train the same FCM models that describe three different systems and their performance will be compared to each other, in the hereafter section.

IV. COMPARATIVE STUDY

In order to compare the performance of the previously described learning algorithms, in finding the appropriate weights set that give desired output concepts' values, two performance indices are used.

The first one (Mean Squared Error - MSE) measures the mean error between the values of the output concepts (OCs) and the desired values defined by the problem's specifications and is defined as:

$$MSE = \frac{1}{N} \sum_{n=1}^N (OC_n - T_n)^2 \quad (15)$$

where N is the number of the output concepts, OC_i and T_i the value the i^{th} simulated output concept and the corresponding desired value, respectively.

Furthermore, in order to measure the capability of the proposed weight set to converge to an acceptable output values when different initial states are applied (generalization capabilities), the following metric (Generalization Error – GE) is used.

$$GE = \frac{1}{NM} \sum_{k=1}^M \sum_{n=1}^N (OC_n^k - T_n)^2 \quad (16)$$

where M is the number of the different initial states.

A. Problem I

The first problem under study, is a benchmark system modeling problem commonly used in the literature [8,9]. It is a process control example consisting of one tank and three valves that control the amount of a liquid in the tank, as depicted in Fig.2.

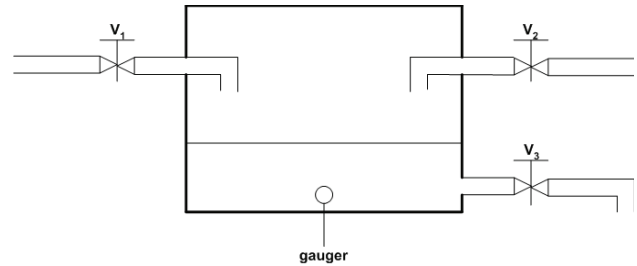


Figure 2. A simple process control problem

The main objective is to keep the height of the liquid and the specific gravity (measured by a sensor – gauger) of the liquid inside of predefined ranges. The FCM model for this process control problem consists of five concepts:

Concept 1 – C1: the amount of the liquid in the tank.

Concept 2 – C2: the state of valve 1

Concept 3 – C3: the state of valve 2

Concept 4 – C4: the state of valve 3

Concept 5 – C5: the specific gravity of the liquid in the tank

The initial weights provided by the experts are:

$$W^{initial} = \begin{bmatrix} 0 & -0.4 & -0.25 & 0 & 0.3 \\ 0.36 & 0 & 0 & 0 & 0 \\ 0.45 & 0 & 0 & 0 & 0 \\ -0.9 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0.3 & 0 \end{bmatrix}$$

By taking into account that the constraints regarding the desired regions of the output concepts (C_1, C_5) are

$$\begin{aligned} 0.68 \leq A_1 \leq 0.74 \\ 0.74 \leq A_5 \leq 0.80 \end{aligned} \quad (17)$$

the four algorithms are applied with the settings of Table I and the following weight sets are derived in each case:

TABLE I. ALGORITHMS SETTINGS

Algorithms	Parameters
DHL	N=100, e=0.005
NHL	e=0.005, $\eta=0.04$
DD-NHL	e=0.005, $\eta=0.04$
AHL	e=0.005, $b_1=0.02$, $b_2=0.04$, $\lambda_1=0.1$, $\lambda_2=1$

$$W^{DHL} = \begin{bmatrix} 0 & -0.214 & -0.133 & 0 & 0.164 \\ 0.192 & 0 & 0 & 0 & 0 \\ 0.241 & 0 & 0 & 0 & 0 \\ -0.481 & 0 & 0 & 0 & 0 \\ 0 & 0.320 & 0 & 0.160 & 0 \end{bmatrix}$$

$$W^{NHL} = \begin{bmatrix} 0 & -0.197 & -0.017 & 0 & 0.477 \\ 0.547 & 0 & 0 & 0 & 0 \\ 0.578 & 0 & 0 & 0 & 0 \\ -0.825 & 0 & 0 & 0 & 0 \\ 0 & 0.710 & 0 & 0.5 & 0 \end{bmatrix}$$

$$W^{DD-NHL} = \begin{bmatrix} 0 & -0.244 & -0.068 & 0 & 0.445 \\ 0.515 & 0 & 0 & 0 & 0 \\ 0.557 & 0 & 0 & 0 & 0 \\ -0.839 & 0 & 0 & 0 & 0 \\ 0 & 0.690 & 0 & 0.467 & 0 \end{bmatrix}$$

$$W^{AHL} = \begin{bmatrix} 0 & -0.152 & -0.025 & 0.213 & 0.441 \\ 0.465 & 0 & 0.168 & 0.205 & 0.192 \\ 0.538 & 0.167 & 0 & 0.210 & 0.197 \\ -0.738 & 0.072 & 0.073 & 0 & 0.080 \\ 0.131 & 0.627 & 0.128 & 0.409 & 0 \end{bmatrix}$$

The initial values of the concepts used by the algorithms are $A^0=[0.4 \ 0.7 \ 0.6 \ 0.7 \ 0.3]$, while the final state of the FCM model after applying each one of the above weight matrix is presented in the following Table II.

TABLE II. EXPERIMENTAL RESULTS – PROBLEM I

	Algorithms			
	DHL	NHL	DD-NHL	AHL
A1	0.645	0.706	0.692	0.697
A2	0.682	0.759	0.749	0.628
A3	0.634	0.656	0.645	0.654
A4	0.690	0.756	0.749	0.827
A5	0.689	0.747	0.740	0.794
Iterations	6	16	14	17
MSE	5.42×10^{-3}	2.72×10^{-4}	6.05×10^{-4}	3.76×10^{-4}
GE	5.42×10^{-3}	2.72×10^{-4}	6.05×10^{-4}	3.76×10^{-4}

The above table also includes some additional experimental results such as the algorithm's numbers of iterations needed to converge the algorithms, along with the two performance indices defined in Eq.(15) and Eq.(16).

In this table the bold faced concept values indicate that these outputs concepts lie inside the pre-defined range of desired values of Eq.(17). Comparing the equilibrium points of each algorithm it is deduced that only the DHL algorithm fails to converge in a desirable state, while the rest ones are behave desirable. Among those well converged algorithms the NHL shows the best performance since it equilibrates to a point that is closest to the desired central of the concept's range. This is also justified by examining the MSE value in each case. Moreover, the iterations of the algorithms are almost the same, but as the overall algorithms' execution time is concerned, it has to be mentioned that the DD-NHL algorithm is the most time consumed due to the historical data construction procedure.

Furthermore, it has to be noted that all the algorithms show a remarkable stability on the different initial concepts' state. This perceived from the GE index, which has been calculated by using 10 different random initial state vectors, which remains the same (in fact the final states are differ beyond the fifth decimal digit) with the corresponding MSE value for the case of all the algorithms.

B. Problem II

The second problem studied here, is a part of a chemical plant consisting of two tanks [10], three valves, a heating element and two thermometers and is depicted in the following Fig.3. The objective of the control system is to keep the height of the liquid in both tanks between some desired values, by simultaneously preserving the temperature of the two tanks between predefined limits.

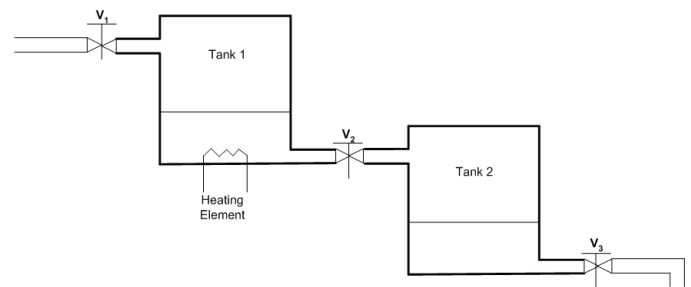


Figure 3. A chemical process control problem

The FCM model for this chemical process control problem consists of eight concepts:

Concept 1 – C1: the height of the liquid in tank 1

Concept 2 – C2: the height of the liquid in tank 2

Concept 3 – C3: the state of valve 1

Concept 4 – C4: the state of valve 2

Concept 5 – C5: the state of valve 3

Concept 6 – C6: the temperature of the liquid in tank 1

Concept 7 – C7: the temperature of the liquid in tank 2

Concept 8 – C8: the heating element operation

The initial weights provided by the experts are:

$$W^{initial} = \begin{bmatrix} 0 & 0 & 0.21 & 0.38 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.70 & 0.60 & 0 & 0 & 0 \\ 0.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.80 & 0.70 & 0 & 0 & 0 & 0 & 0.09 & 0 \\ 0 & -0.42 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.40 & 0 & 0 & 0 & 0 & 0.50 \\ 0 & 0 & 0 & 0.30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.40 & 0 \end{bmatrix}$$

By taking into account that the constraints regarding the desired regions of the output concepts (C_1, C_2, C_6, C_7) are

$$\begin{aligned} 0.55 &\leq A_1 \leq 0.75 \\ 0.75 &\leq A_2 \leq 0.80 \\ 0.75 &\leq A_6 \leq 0.82 \\ 0.65 &\leq A_7 \leq 0.75 \end{aligned} \quad (18)$$

the values of the algorithm's free parameters are determined by trial and error and the applied settings are summarized in the following Table III.

TABLE III. ALGORITHMS SETTINGS

Algorithms	Parameters
DHL	N=30, e=0.005
NHL	e=0.005, $\eta=0.03$
DD-NHL	e=0.005, $\eta=0.03$
AHL	e=0.005, $\eta=0.005$, $\gamma=0$ (constant)

The resulted weight sets provided by the four Hebbian-like learning algorithms are as follows:

$$W^{DHL} = \begin{bmatrix} 0 & 0 & 0.146 & 0.262 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.483 & 0.414 & 0 & 0 & 0 \\ 0.517 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.549 & 0.551 & 0 & 0 & 0 & 0 & 0.063 & 0 \\ 0 & -0.287 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.276 & 0 & 0 & 0 & 0 & 0.365 \\ 0 & 0 & 0 & 0.207 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.276 & 0 & 0 \end{bmatrix}$$

$$W^{NHL} = \begin{bmatrix} 0 & 0 & 0.266 & 0.420 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.720 & 0.635 & 0 & 0 & 0 \\ 0.780 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.762 & 0.831 & 0 & 0 & 0 & 0 & 0.177 & 0 \\ 0 & -0.362 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.453 & 0 & 0 & 0 & 0 & 0.571 \\ 0 & 0 & 0 & 0.357 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.454 & 0 & 0 \end{bmatrix}$$

$$W^{DD-NHL} = \begin{bmatrix} 0 & 0 & 0.272 & 0.424 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.722 & 0.639 & 0 & 0 & 0 \\ 0.783 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.757 & 0.835 & 0 & 0 & 0 & 0 & 0.186 & 0 \\ 0 & -0.356 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.458 & 0 & 0 & 0 & 0 & 0.575 \\ 0 & 0 & 0 & 0.363 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.460 & 0 & 0 \end{bmatrix}$$

$$W^{AHL} = \begin{bmatrix} 0 & 0.020 & 0.224 & 0.392 & 0.019 & 0.017 & 0.017 & 0.017 \\ 0.020 & 0 & 0.022 & 0.705 & 0.609 & 0.021 & 0.021 & 0.021 \\ 0.753 & 0.021 & 0 & 0.020 & 0.021 & 0.018 & 0.018 & 0.018 \\ -0.765 & 0.805 & 0.020 & 0 & 0.022 & 0.019 & 0.107 & 0.019 \\ 0.020 & -0.385 & 0.021 & 0.022 & 0 & 0.020 & 0.020 & 0.020 \\ 0.017 & 0.021 & 0.411 & 0.019 & 0.021 & 0 & 0.018 & 0.538 \\ 0.018 & 0.022 & 0.019 & 0.314 & 0.021 & 0.018 & 0 & 0.018 \\ 0.017 & 0.021 & 0.018 & 0.019 & 0.021 & 0.411 & 0.018 & 0 \end{bmatrix}$$

The initial values of the concepts used by the algorithms are $A^0 = [0.480 \ 0.570 \ 0.580 \ 0.680 \ 0.580 \ 0.590 \ 0.520 \ 0.580]$, while the final state of the FCM model after applying each one of the above weight matrix is presented in the following Table IV.

TABLE IV. EXPERIMENTAL RESULTS – PROBLEM II

	Algorithms			
	DHL	NHL	DD-NHL	AHL
A1	0.640	0.643	0.645	0.670
A2	0.723	0.771	0.773	0.806
A3	0.736	0.785	0.787	0.701
A4	0.813	0.876	0.877	0.730
A5	0.738	0.781	0.781	0.790
A6	0.714	0.750	0.751	0.679
A7	0.674	0.702	0.704	0.703
A8	0.729	0.768	0.768	0.679
Iterations	4	5	7	7
MSE	4.27×10^{-3}	6.46×10^{-4}	6.46×10^{-4}	6.28×10^{-3}
GE	4.27×10^{-3}	6.46×10^{-4}	6.46×10^{-4}	6.28×10^{-3}

The derived experimental results for the case of the second problem are almost the same with those of the first one. The NHL algorithm and its modified version DD-NHL, show the best performance by keeping the four output concepts inside the predefined ranges. The DHL and AHL algorithms have 50% accuracy, since they control satisfactory the two of the four outputs of the FCM model. Again, all the weight sets ensure robustness of the FCM model to the changes of the initial concepts' state as the GE index show.

C. Problem III

The third system that is being modeled by a FCM is a heat exchanger which constitutes a common part of the chemical and process industry [17]. The main objective in this case is the development of a model that simulates the behaviour of the system in order to control the water outlet temperature by adjusting the water flow rate.

The FCM model for this problem consists of five concepts:

Concept 1 – C1: the fan speed (S_f).

Concept 2 – C2: the water flow rate (F_w).

Concept 3 – C3: the water inlet temperature (T_{wi}).

Concept 4 – C4: the air inlet temperature (T_{ai}).

Concept 5 – C5: the water outlet temperature (T_{wo}).

The initial weights provided by the experts are:

$$W^{initial} = \begin{bmatrix} 0 & 0.625 & 0.75 & -0.25 & 0.625 \\ 0 & 0 & -0.75 & 0 & -0.75 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.75 & 0.125 & 0.125 & -0.75 & 0 \end{bmatrix}$$

The output concept is the water outlet temperature (C_5) which for the experimental needs has to satisfy the following constraint

$$0.74 \leq A_5 \leq 0.75 \quad (19)$$

The corresponding values of the algorithms' free parameters are determined by trial and error and the applied settings are summarized in the following Table III.

TABLE V. ALGORITHMS SETTINGS

Algorithms	Parameters
DHL	N=30, e=0.005
NHL	e=0.001, η =0.06
DD-NHL	e=0.001, η =0.06
AHL	e=0.01, b_1 =0.05, b_2 =0.02, λ_1 =0.1, λ_2 =1

The resulted weight sets provided by the four Hebbian-like learning algorithms are as follows:

$$W^{DHL} = \begin{bmatrix} 0.107 & 0.133 & 0.160 & -0.052 & 0.134 \\ 0 & 0.106 & -0.159 & 0 & -0.159 \\ 0 & 0 & 0.106 & 0 & 0 \\ 0 & 0 & 0 & 0.108 & 0 \\ -0.158 & 0.027 & 0.027 & -0.158 & 0.107 \end{bmatrix}$$

$$W^{NHL} = \begin{bmatrix} 0 & 0.655 & 0.729 & 0.271 & 0.736 \\ 0 & 0 & 0.009 & 0 & -0.037 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.380 & 0.651 & 0.646 & -0.368 & 0 \end{bmatrix}$$

$$W^{DD-NHL} = \begin{bmatrix} 0 & 0.653 & 0.734 & 0.241 & 0.734 \\ 0 & 0 & -0.071 & 0 & -0.106 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.414 & 0.630 & 0.624 & -0.405 & 0 \end{bmatrix}$$

$$W^{AHL} = \begin{bmatrix} 0 & 0.602 & 0.692 & -0.047 & 0.594 \\ 0.139 & 0 & -0.402 & 0.135 & -0.410 \\ 0.138 & 0.144 & 0 & 0.134 & 0.136 \\ 0.130 & 0.136 & 0.135 & 0 & 0.128 \\ -0.404 & 0.236 & 0.235 & -0.416 & 0 \end{bmatrix}$$

The initial values of the concepts used by the algorithms are $A^0 = [0.300 \ 0.650 \ 0.450 \ 0.150 \ 0.300]$, while the final state of the FCM model after applying each one of the above weight matrix is presented in the following Table VI.

TABLE VI. EXPERIMENTAL RESULTS – PROBLEM II

	Algorithms			
	DHL	NHL	DD-NHL	AHL
A1	0.628	0.567	0.563	0.735
A2	0.687	0.844	0.843	0.712
A3	0.662	0.845	0.840	0.718
A4	0.619	0.614	0.609	0.675
A5	0.652	0.750	0.744	0.745
Iterations	33	36	34	18
MSE	8.70×10^{-3}	2.04×10^{-5}	3.57×10^{-7}	5.71×10^{-8}
GE	8.56×10^{-4}	5.07×10^{-6}	4.51×10^{-10}	6.09×10^{-7}

Based on the above table it is deduced that the DHL algorithm fails to equilibrate to a desired point since the final value of the output concept lies outside the desirable range Eq. (19). On the other hand, the rest algorithms give a satisfied equilibrium state of the FCM model, especially the AHL algorithm, which gives the best output value in a shorter time (18 iterations). As far as the GE performance index is concerned, the DD-NHL algorithm shoes better generalization capabilities, since it equilibrates to the same state independently of the initial state of the model. Moreover, although the performance of the NHL algorithm is accepted, it is quite marginal.

Generally, the NHL algorithm exhibits a satisfactory behaviour since it fully controls the output concepts variations by keeping them between the system's specifications. While the DD-NHL presents similar performance, its dependency on historical data, gives them high execution time without showing significant improvement of its performance. The remaining algorithms the DHL and AHL, seem to perform worst for the three and the two problems under study respectively, but the feature of the AHL algorithm to search for new interconnections between the concepts, can help to design more reliable FCM models.

Finally, it has to be mentioned that all the Hebbian-like learning algorithms studied in this work, are dependent on some free parameters (η , γ , e , N), which significantly affect the convergence and termination of the algorithms to desired FCM model's states. Therefore special attention has to be paid to the selection of these parameters.

V. CONCLUSIONS

A detailed description of the Hebbian-like learning algorithms for training FCM models was presented in the previous sections. Appropriate experiments have been

conducted in order to analyze the performance of each algorithm under the same conditions. The results show that the NHL algorithm exhibits a satisfactory behaviour with low algorithmic complexity as compared with the other similar algorithms. While the comparison of the algorithms was realized by using some performance indices, there is a need to develop a methodology that permits the comparison of the different solutions in the real system's workspace.

The dependency of the Hebbian-like algorithms on some free parameters that significantly influence their performance is a disadvantage of these algorithms, since they control not only the final equilibrium point of the model but also the successfully termination of the algorithm. A novel algorithm with less sensitivity on such parameters and with high convergence rate on desirable states remains a major target of the scientists in this field.

REFERENCES

- [1] M. Glykas, "Fuzzy Cognitive Maps: Advances in theory, methodologies, tools and applications", *Studies in Fuzziness and Soft Computing*, vol. 247, Springer-Verlag, 2010.
- [2] J. Aguilar, "A survey about fuzzy cognitive maps papers (Invited Paper)", *International Journal of Computational Cognition*, vol. 3, no. 2, 2005.
- [3] D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, C.D. Zopounidis, "Development of dynamic cognitive networks as complex Systems approximators: Validation in financial time series", *Applied Soft Computing*, vol. 5, no. 2, 2005, pp. 157-179.
- [4] B. Kosko, "Fuzzy Cognitive Maps", *International Journal Man-Machine Studies*, vol. 24, 1986, pp. 65-75.
- [5] J.A. Dickerson and B. Kosko, "Virtual worlds as fuzzy cognitive maps", *Presence*, vol. 3, no. 2, 1994, pp. 173-189.
- [6] A.V. Hueriga, "A balanced differential learning algorithm in fuzzy cognitive maps", *Int. Workshop on Qualitative Reasoning*, 2002.
- [7] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, "Fuzzy cognitive map learning based on nonlinear Hebbian rule", In: T.D. Gedeon and L.C.C. Fung (Eds.): *AI 2003, LNAI vol. 2903*, Springer-Verlag, 2003, pp. 254-266.
- [8] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, "Unsupervised learning techniques for fine-tuning fuzzy cognitive map casual links", *International Journal of Human-Computer Studies*, vol. 64, no. 8, 2006, pp. 727-743.
- [9] W. Stach, L. Kurgan and W. Pedrycz, "Data-driven nonlinear Hebbian learning method for fuzzy cognitive maps", *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, 2008, pp. 1975-1981.
- [10] E.I. Papageorgiou, C.D. Stylios, P.P. Groumpos, "Active Hebbian learning algorithm to train fuzzy cognitive maps", *International Journal of Approximate Reasoning*, vol. 37, no. 3, 2004, pp. 219-249.
- [11] E.I. Papageorgiou and P.P. Groumpos, "A weight adaptation method for fuzzy cognitive maps to a process control problem", In: M. Bubak et al. (Eds.): *ICCS 2004, LNCS vol. 3037*, 2004, pp. 515-522.
- [12] G.A. Papakostas, Y.S. Boutalis, D.E. Koulouriotis and B.G. Mertzios, "Fuzzy cognitive maps for pattern recognition applications", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 22, no. 8, 2008, pp. 1461-1468.
- [13] W. Stach, L. Kurgan, W. Pedrycz, M. Reformat, "Genetic learning of fuzzy cognitive maps", *Fuzzy Sets and Systems*, vol. 153, no. 3, 2005, pp. 371-401.
- [14] 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*, vol. 25, no. 1, pp. 95-121, 2005.
- [15] E. Oja, "Neural networks, principal components and subspaces", *International Journal of Neural Systems*, vol. 1, pp. 61-68, 1989.
- [16] S. Haykin, "Neural networks: A comprehensive foundation", 2nd Ed., Prentice-Hall, 1999.
- [17] C.D. Stylios and P.P. Groumpos, "Modeling complex systems using fuzzy cognitive maps", *IEEE Trans. On Systems, Man and Cybernetics – Part A: Systems and Humans*, vol. 34, no. 1, 2004, pp. 155-162.