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

ICLA Imperialist Competitive Learning Algorithm for Fuzzy Cognitive Map: Application to Water Demand Forecasting

Sadra Ahmadi, Somayeh Alizadeh, Nafiseh Forouzideh, Chung-Hsing Yeh, Rodney Martin, Elpiniki Papageorgiou

Abstract—**In this paper, we develop a new Fuzzy Cognitive Map (FCM) learning method using the imperialistic competitive learning algorithm (ICLA). An FCM seems like a fuzzy signed directed graph with feedback, and models complex systems as a collection of concepts and causal relations between concepts. Conventional FCMs are mainly constructed by human experts who have experience in the specific problem domain. However, large problems need automated methods. We develop an automated method for FCM construction inspired by the socio-political behavior of countries as imperialists with colonies. In the real world imperialists extend their territories and change the socio attributes of their colonies. The ICLA is an evolutionary algorithm and simulates this behavior. We explain the algorithm for FCM learning and demonstrate its performance advantages through synthetic and real data of water demand. The results of the new algorithm were compared to that of a genetic algorithm, which is the most commonly used and well-known FCM learning algorithm.**

Keywords: Fuzzy Cognitive Maps (FCM), FCM learning, Imperialist Competitive algorithm (ICA)

I. Introduction

A Fuzzy Cognitive Map (FCM) is a directed weighted network with feedback loops [19]. This network type is used to model and analyze the behavior of dynamic and complex systems. This modeling technique creates a network which consists of all possible causal relationships between components of the system modeled. Nodes in this network are components of the system and arrows in this network represent the direction of relationship between nodes. Each relationship is associated with a weighted arrow which indicates the strength of relationship between two nodes [13].

There are several ways to develop an FCM model. Usually, experts' opinions are used to analyse a system and develop an FCM model. In this method, the robustness and the accuracy depend on the experts' knowledge. The manual method can become impractical as the size of the model increases. If historical time series data is available, then automated methods can be developed to build the FCM model. This issue is called "learning" of the FCM model.

The learning process builds the connection matrix (known as the weight matrix) representing the causal relationships between factors [24], by using the knowledge of human experts or from historical data [19].

There are three main groups of FCM learning methods [19]: (1) methods which use experts knowledge for developing and learning FCMs, (2) methods which rely on historical data about the given system and (3) hybrid methods which use both experts knowledge and historical data with a two-stage learning process. The first group is devoted to modelling issues/problems for where there is no historical data. These models are well established but are not usable in cases that there is limit or no expertise about the problem. The second group of FCM learning methods use population-based optimization algorithms applied to historical data to learn the FCM model.

The problem investigated in this study is the automatic construction of FCM using historical data. Previous solutions were developed by implementing several evolutionary or population based algorithms [19,29,33]. Common evolutionary optimization approaches used for FCM learning are genetic strategies (GS) [11], genetic algorithms (GA) [9], real coded generic algorithm – RCGA [30, 31, 33], Swarm Intelligence [25], Chaotic Simulated Annealing [2], Tabu search [3], game-based learning [14], Particle Swarm Optimization (PSO) [25], Memetic PSO (MPSO) [27], Ant Colony Optimization (ACO) [7], Extended Great Deluge algorithm [6], Big Bang-Big Crunch [38], Artificial Bee Colony (ABC) [37].

The goal of this study was to solve the problem of FCM learning or automatic construction of FCMs through the modification of the weight matrix by providing high performance models, generated from historical data, with short execution/number of function evaluations. This goal has been achieved with the development of a new learning method which we have called the "Imperialist Competitive Learning Algorithm (ICLA) for Fuzzy Cognitive Map".

The ICLA belongs to the group of evolutionary algorithms which are inspired by the cultural and visional evolution of humans and society. In certain situations these algorithms have shown a great performance in both convergence rate and better global optima achievement rather than other evolutionary algorithms [18]. The Imperialist Competitive

S. Ahmadi is with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (corresponding author to provide phone: +61399055190; email: Sadra.Ahmadi@monash.edu)

S. Alizadeh is with the Department of IT, Faculty of Industrial Engineering, K.N.Toosi University, Shahid Agahi Alley- Dabestan Street- Resalat Expressway- Seyed Khandan, Tehran 1631714191, Iran, (e-mail: S_alizadeh@kntu.ac.ir)

N. Forouzideh is with the Faculty of Engineering, Monash University, Clayton, VIC 3800, Australia (e-mail: Nafiseh.Forouzideh@Monash,edu)

C-H. Yeh is with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (email: Chunghsing.Yeh@monash.edu)

R. Martin is with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia (email: Rodney.Martin@monash.edu)

E. Papageorgiou is with the Center for Research & Technology Hellas—CERTH, 6th km Charilaou-Thermi Rd., Thermi 57001, Greece (email: epapageorgiou@iti.gr , epapageorgiou@teiste.gr)

Algorithm (ICA) has been used to solve multi-objective optimization methods and does not need the gradient of the function in its optimization process [4].

In the following sections of this paper, we first present the research background and describe FCM principals and existing learning algorithms. Following this, we explain the principals of the ICA and the proposed ICLA that is used for learning an FCM model. Finally, we conduct a number of experiments on real and synthetic data, which are used to test the new algorithm performance. We used genetic algorithms as a benchmark for comparison purposes, because GA's are a well-known method for FCM learning. In the last section, the conclusions are outlined.

II. Research Background

A. Fuzzy cognitive map

The idea of cognitive maps was developed during the 1970s [5]. A cognitive map is an oriented graph where the nodes are concepts and the arcs are relationships between these concepts. These relationships can be positive, negative or zero. A positive relationship means that an improvement in the source node causes an improvement in the destination node. Zero means no relationships between the two nodes [5].

In 1988, Kosko extended the concept of cognitive maps by using fuzzy logic and developed the new area of complex system analysis called Fuzzy Cognitive Maps (FCM). In this technique, fuzzy relationships between nodes are defined as numerical values in the interval [-1, 1]. A value of +1 means a completely positive relationship and a value of -1 means a completely negative relationship [10].

The relationships between factors in this model are represented as a matrix which is called the connection matrix. In this matrix, the numerical value of each pair of factors' relationship weight is shown in their related cell. Fig. 1 shows a simple FCM which is drawn with 5 nodes and 10 branches. The rows are the source nodes and the columns are the destination nodes.

	F_1	F_2	F_3	F_4	F_5
F_1	0	$+w_{12}$	0	0	$-w_{15}$
F_2	$+w_{21}$	0	$-w_{23}$	$-w_{24}$	0
F_3	0	0	0	0	$+w_{35}$
F_4	$+w_{41}$	0	$-w_{43}$	0	0
F_5	0	0	$-w_{53}$	$+w_{54}$	0

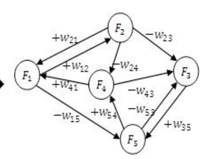

Fig. 1. A simple fuzzy cognitive map and its connection matrix

FCMs originated from the combination of fuzzy logic and neural networks [28]. FCMs use the concept of fuzzy logic when a system is intended to be modeled by using experts' knowledge. In this method decision makers define inter-relationships between factors using an if-then rule is used [23]:

If the value of factor F_i is changed by a {very small, small, medium, large, or very large amount}, then this will cause factor F_j to change by a {very small, small, medium, large, or very large amount}. The influence of factor F_i on Factor F_j can be one of 10 possible values in fuzzy set T given in Fig 2. The negative membership functions in set T are in the case of an increase in F_i causes a decrease in F_j.

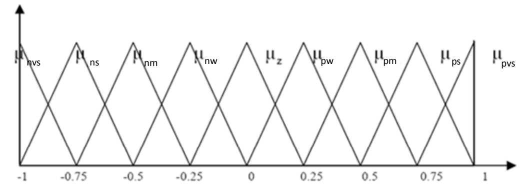

Fig. 2. The membership function of set T

FCMs also follow neural network regulations during the analysing process. A neural network consists of several neurons. The neuron's input signals $x_1, x_2, \ldots, x_n$ represent the output signals of other neurons and they are weighted by corresponding elements $w_1, w_2, \ldots, w_n$. Each weight corresponds to the strength of a biological synaptic connection. The output of multiplying an input by the weight of influence is called net input. The net input goes into an activation non-linear function f, which produces the neuron output y.

$$y = f(n) = f(w_1x_1 + w_2x_2 + \cdots + w_nx_n) \tag{1}$$

This activation function f keeps the neuron output between certain limits [8].

During the construction process, linguistic states are used to determine the initial strength of each factor or node. These linguistic states will then be defuzzified and converted to numerical values in the range [0-1].

When all nodes are activated by an initial value, they influence each other, based on their inter-relationships and a new value for each of them has to be calculated. The new values of the factors will influence each other again with a new calculated amount and this process will continue again and again until the values of factors do not change significantly between iterations. Eq.(2) shows how the new value of each factor is calculated.

$$A_i^{(t)} = f\left(\sum_{j=1}^{n} A_j^{(t-1)} \times W_{ji}\right) \tag{2}$$

Where $A_i^{(t)}$ represents values of node C_i in iteration t and $A_j^{(t-1)}$ shows the values of C_j in iteration $t-1$. In this equation, W_{ji} is the weight of interrelations of each pair of factors/nodes. f is a threshold function for converting the output of computation to a number in the interval [0, 1]. Nonlinear function f allows factors/nodes to gain rated values. There are several threshold functions but the most common function is the logistic function. Eq. (3).

$$f(x) = \frac{1}{1+e^{-cx}} \tag{3}$$

Where $c>0$ is the slope of the function.

B. Fuzzy cognitive maps learning algorithms

The main aim of FCM learning algorithms is to estimate the strength of the causal interrelationships between the system's concepts. Since the introduction of the FCM technique by Kosko [10], many learning algorithms have been developed. The developed algorithms according to their dependency on previous knowledge for learning the FCM model are categorized into three categories: (1) Hebbian-based algorithms, (2) Evolutionary-based algorithms, and (3) Hybrid algorithms. These types of learning algorithms are the most well-known and are widely used for training FCMs according to the existing literature [19].

Hebbian-based algorithms partly rely on experts' knowledge. In these algorithms, the initial connection matrix is developed by experts and the strength of weights in the connection matrix is modified by using mathematical techniques.

In order to enhance the learning capabilities of the Hebbian-based type of FCM learning, ensemble techniques, such as bagging and boosting, were integrated in the learning process. In the ensemble learning for FCMs, the model is trained using the non-linear Hebbian learning (NHL) algorithm and its performance is enhanced using ensemble techniques. This new gave results with higher classification accuracy than NHL on its own [21].

In what follows, we focus on the category of evolutionary based algorithms, where the experts are substituted by historical data. The corresponding learning algorithms or optimization algorithms are used to estimate the entries of the connection matrix. These evolutionary algorithms are usually oriented towards models that mimic the input data. They are often used to solve optimization problems. Several population-based algorithms, such as GS [11], (GA) [9], RCGA [30, 31, 33], Chaotic Simulated Annealing [2], TS [3], game-based learning [14], Particle Swarm Optimization (PSO) [25], Memetic PSO (MPSO) [27], ACO [7], Extended Great Deluge algorithm [6], Big Bang-Big Crunch [38], Artificial Bee Colony [37], cellular automata [16], immune algorithm [12], and supervised gradient-based algorithm [36] have been proposed for training FCMs. The first evolutionary learning algorithm was developed in 2001 by Koulouriotis et al. [11]. They applied a genetic strategy to learn an FCM's model structure from data. Later, the PSO method was proposed by Papageorgiou et al. [22], which belongs to the class of swarm intelligence algorithms. After that, a learning scheme was proposed by Khan et al. [9] to accomplish a different learning objective. In that method they used a genetic algorithm to find an initial state vector (initial condition) which leads a given model to the specified end state. Next, an efficient RCGA (real-coded genetic algorithms) was proposed for automated learning of FCMs from data An RCGA, as a floating-point extension to genetic algorithms, was used to allow finding the floating-point weights instead of weights that take on a limited set of values [32, 33]. Petalas et al. [26] proposed a swarm intelligence mimetic algorithm that combined PSO with both deterministic and stochastic local search schemes, for FCM learning tasks. Mateou et al. [17] proposed evolutionary FCMs that are used for multi objective decision making in order to increase their reliability and overcome the main weakness, which lies with the recalculation of weights corresponding to more than one concept every time a new multiple scenario is introduced.

A number of hybrid algorithms have also been developed. These use both historical data and experts' knowledge to learn the FCM models. Papageorgiou and Groumpos [20] developed NHL-Differential Evolution as a hybrid learning algorithm, which uses both Hebbian based learning and evolutionary learning. Yanchun and Wei [35] proposed the NHL-RCGA which uses the Hebbian based technique and a genetic algorithm. Other recently proposed algorithms are described in [21].

In the next section, we introduce the new evolutionary based algorithm which uses the Imperialist Competitive Algorithm applied for FCM learning.

III. New Learning Algorithm for FCM

Atashpaz-Gargari and Lucas [4] developed the Imperialist Competitive Algorithm (ICA) as a new evolutionary algorithm. The main aim of this algorithm is to find the optimum solution in various problems. This algorithm is inspired by the social political process of imperialism and imperialist competition; most other evolutionary algorithms are inspired by a nature-based evolutionary phenomenon. The main advantages of ICA are its high evolution speed and accuracy.

A. Parameters

To present the new FCM learning algorithm, we use the following notation and definitions:

TABLE I
ICA and ICLA Parameters

Parameter	ICA	ICLA
$P_{i,j}$	Colony i in empire j	Connection matrix i in group of connection matrix j
$f(P_{i,j})$	The cost function of colony i in empire j	The cost function of connection matrix i in group j
$d_{i,j}$	Distance between colony i and its related imperialist j	Distance between connection matrix i and the best connection matrix in corresponding group j
$\theta_{i,j}$	Deviation angle θ in moving colony i toward its related imperialist j	Deviation angle θ in moving connection matrix i toward its related group j
γ	Arbitrary parameter to adjust the deviation	Arbitrary parameter to adjust the deviation
$T.C._j$	Total cost of empire j	Total cost of group of connection matrix j
$N.T.C_n$	Normalized cost of empire j	Normalized cost of group of connection matrix j
P_{P_n}	Empire j probability of possessing new colony	Group j probability of possessing new connection matrix

B. Imperialist Competitive Algorithm

The ICA is inspired by imperialistic behaviour. This is where a government attempts to extend its power and rule beyond its own borders [1]. This algorithm is comprised of 7 steps - which start with an initial population. Following this, the best members of the initial population are selected to be the imperialists.

Countries are divided into imperialists and colonies. Each imperialist on the basis of its power, colonizes other countries and controls them. The assimilation and imperialist competitive policy constitutes the core of this algorithm. In this algorithm, the weakest empire which cannot extend its territory and power in competition against other empires will be eliminated from the competition. Then, other empires occupy colonies of the eliminated empire. Finally, a mechanism in this algorithm, which is called the collapse mechanism, converges all countries to be colonies of one empire. The robust empire would be our solution. In the rest of this section, the 7 steps will be explained in detail.

In the ICA, firstly, a set of empires are generated randomly and in each empire a set of countries are generated randomly. Each country has n attributes $(P_1, P_2, \ldots, P_n)$. These variables are specifications of each country such as its culture, language, economic structure and other specifications. Each country has variable called cost. In a set of countries in an empire, the country with the lowest cost is the imperialist and other countries are colonies of this imperialist.

The imperialist in an empire tries to assimilate colonies by changing their specifications. For instance, in the real world, colonies change their spoken language to speak in the same way as their corresponding imperialist. Thus, each colony is pushed toward its corresponding imperialist. Some colonies move directly toward the imperialist and some of them have some deviation in this movement. A sample of this process is shown in Fig. 3.

Sometimes, during a colonies' movement toward their corresponding imperialist a revolution will happen. The revolution is a sudden change in at least one specification of colony (e.g., cultural, religion). This specification helps the algorithm to avoid being trapped in a local optimum. In each empire, some colonies have a risk of a revolution.

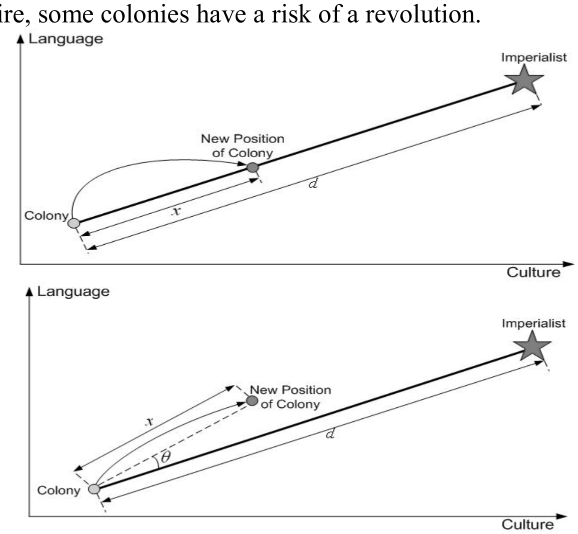

Fig. 3. Movement of a colony toward its related imperialist; (a) without deviation (b) with a deviation

Political history shows that although in most cases the assimilation policy of the imperialist destroys the political and social structures of its colonies, it sometimes has some positive consequences. For instance, some countries have achieved public confidence and after a while educated people (who have been educated in the imperialist culture) lead people to escape from the clutches of their imperialist. On the other hand, political history also shows that some countries that were once superior in political and military power, collapsed after time. In contrast, countries that were once colonies can rise to be the superior imperialists.

Colonies during movement toward their imperialist or facing a revolution may become much more powerful than the imperialist. In the ICA, it means that the cost of these colonies would be lower than their corresponding imperialist. In this case, the colony and the imperialist exchange their positions and the colony takes over the power in the empire.

In reality, empires fight against each other and try to extend their territory by occupying dominated regions of other imperialist and destroy other empires. In the ICA algorithm, empires try to increase their power and any of them who cannot do that will lose their power and will be gradually eliminated during the imperialist competitions. In this competition, powerful empires possess colonies in the weakest empire. In each iteration of the ICA, one or more colonies of the weakest empire are chosen and a competition among other empires is formed to possess these colonies. These colonies may not be possessed by the most powerful empire; however, they have the highest possibility to possess these colonies. This process will be continued until the most powerful empire eliminates all other empires and possesses all colonies. In the ICA, an empire with no colony is called a collapsed empire.

C. *ICA Based Learning Algorithm (ICLA)*

In the previous section, the ICA and its seven steps were explained. In this section, we show how we use the ICA for learning a fuzzy cognitive map and develop the ICLA.

To learn the FCM model by using any evolutionary algorithm like the ICA we need to use a set of historical data. As was explained, the FCM is usually used to model the behavior of a complex system. The historical data that we need to use here consists of numerical values of system components at several points in time. If a change happens in any component of a complex system, the components of the system interact with each other until the system reaches a steady state. Different values of system components during the time from the point that the change is applied until the system reaches the steady state form a time series. In this paper, we show how to use this time series and run the ICLA on it to lean the FCM model and find the weights of causal influences between components of system.

In the ICLA, a time series like Fig. 4 is the input. The result of applying the ICLA to the input is a connection matrix which contains causal interrelationships between components of the system. If the initial value of all components is multiplied by the connection matrix in an iterative process, new values of each concept after each iteration will be calculated. This process is called the FCM inference (Eqs. (2) and (3)). The total sum of differences between the values of components in each iteration from the historical data (initial time series) and the value of corresponding component in the

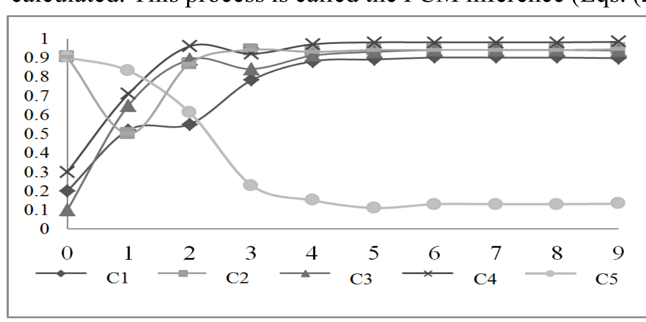

Fig. 4. A sample time series with 5 component and 10 time interval

corresponding iteration from new calculated values (new time series) is the overall error of the learning algorithm. Eq. (4) shows how to calculate this error.

$$Total\ Error=\sum_{n=1}^{K}\sum_{i=1}^{M}\left|C\text{-}value_i^{Estimated}(t_n)\text{-}C\text{-}value_i^{historical\ data}(t_n)\right| \quad (4)$$

where $C\text{-}value_i^{historical\ data}(t_n)$ is the values of component i at the time interval t_n in the historical data and $C\text{-}value_i^{Estimated}(t_n)$ is the values of component i at the corresponding time interval t_n in the time series resulting from applying the learning algorithm. M is the total number of components, n is the iteration number and K is the total number of time intervals.

Eq. (4) can be used as the main function to calculate the cost in the ICLA or any other FCM learning algorithm. In other FCM learning studies, this function is expanded and two equations (5) and (6) are used to estimate the performance of the FCM learning algorithm [33].

$$Errorl_{p1}=\frac{1}{(K\text{-}1)\times M}\sum_{n=1}^{K}\sum_{i=1}^{M}\left|C\text{-}value_i^{Historical\ data}(t_n)\text{-}C\text{-}value_i^{Estimated}(t_n)\right| \quad (5)$$

$$Errorl_{p2}=\frac{1}{(K\text{-}1)\times M}\sum_{n=1}^{K}\sum_{i=1}^{M}\left|C\text{-}value_i^{Historical\ data}(t_n)\text{-}C\text{-}value_i^{Estimated}(t_n)\right|^2 \quad (6)$$

where K is the input data length, M the number of components.

In what follows, the pseudo code of the ICLA is presented in Fig. 5. Next, the steps of the ICLA are explained and the Eqs. (5) and (6) are used to calculate the cost of errors.

Initialization of the algorithm: Generate some random connection matrix in the search space and create initial empires. (Initial Countries (connection matrices), Calculates the cost of countries, Create Initial Empires)

- *Assimilation: cells in connection matrices move towards cells in imperialist states in different in directions.*
- *Revolution: Random changes occur in the cells of some connection matrices.*
- *Calculates the cost of connection matrices*
- *Empire Possession: A connection matrix in an empire with a better position than the imperialist, take the control of empire by replacing the existing imperialist.*
- *Computation of Total Cost for Empires*
- *Uniting Similar Empires*
- *Imperialistic competition: All imperialists compete to take possession of colonies of each other.*
- *Eliminate the powerless empires: Weak empires lose their power gradually and they will finally be eliminated.*

If the stop condition is satisfied, stop, if not go to 2.

Fig. 5. ICLA pseudo code for learning FCM model

The steps of ICLA are:

Step 1: Create the initial empires

A set of empires and a set of colonies in each empire are generated randomly. Each colony is an $n\times n$ connection matrix. In each empire the cost of each member is calculated by using Eq. (5) and the member with the lowest cost is the imperialist and other members are its colonies. To learn FCM with ICA we generate 10 empires with 100 countries in each of them.

Step 2: Assimilation Policy

In this step, colonies move towards their imperialist with the lowest cost. As was explained, each colony is an $n\times n$ matrix where each cell in this matrix is an attribute of the colony. In the assimilation step, each attribute moves toward its corresponding attribute in the imperialist. In the ICLA, the new position of colonies' cells in the connection matrix is calculated by Eqs. (7) and (8).

$$Proposed\ cell\ position = cell\ position + \beta * (R*d) \quad (7)$$

$$New\ Position=Min(Max(Proposed\ position,-1)+1) \quad (8)$$

where β is a constant equal to 2, R is a random number in interval [0,1] and d is the distance between each cell in the colony's connection matrix and its corresponding cell in the imperialist connection matrix. The main aim of using Eq. (8) is to map all cells values in the range [-1,1].

During this step, all colonies move toward the imperialist. Since, this movement is interfered by a random number, it is possible the new error value of one colony to be lower than the imperialist error. In this situation the imperialist and colony's position will be exchanged.

Step 3: Revolution

The revolution in the ICLA is very similar to the mutation in the genetic algorithm. In the revolution, the numerical attributes of some colonies are changed suddenly. In this step, one column and row in a connection matrix are chosen randomly and they are replaced.

Step 4: Displacement of colonial and imperialist position

If after the revolution step, the error of one colony is lower that the error of its related imperialist, the position of these two members is exchanged and the colony will take the role of the imperialist.

Step 5: Calculating the total power of an empire

The total power of an empire depends on the power of its imperialist and colonies. In the ICLA, the total power of an empire is calculated by using Eq. (9).

$$T.C._n=Cost\left(Imprialist_n\right)+\xi\times mean\left\{Cost\left(Colonies\ of\ empire_n\right)\right\} \quad (9)$$

where $T.C._n$ is the total cost of empire n and ξ is a positive number between 0 and 1 and close to 0. Decreasing the value of ξ directly decreases the role of colonies in the calculation of total cost. In the most application of ICA using $\xi=0.05$ leads to acceptable results.

Step 6: Imperialist competition for constructing the FCM model

Any empire that cannot increase its power and expand its territory will lose its competition power and will be eliminated during the imperialist competitions. So, during time, the weakest colonies from the weakest empire are chosen and other empires according to their probability of possession compete with each other to take it over. In this competition, empires with the higher power have a higher chance of success.

In this step, Eqs. (10) and (11) are used to estimate the probability of possession for each empire during the competition.

$$N.T.C_n=\max_i\{T.C_i\}\text{-}T.C_n \quad (10)$$

where $T.C_n$ is the total cost of empire n and $N.T.C_n$ is the normalized cost of that empire. Any empire with lower $T.C_n$

has the higher $N.T.C_n$. $T.C_n$ is the total cost of one country and $N.T.C_n$ is its total power of possession. After calculating the normalized cost of each empire, the probability of possession for empire n is calculated by Eq.(11)

$$P_{P_n} = \left| \frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_i} \right| \tag{11}$$

To redistribute the set of colonies from the weakest empire to other empires the vector P is formed.

$$P=[P_{P_1},P_{P_2},\ P_{P_3},\ ...,\ P_{P_{N_{imp}}}] \tag{12}$$

The size of vector P is $1\times N_{imp}$ and is formed by using the probability of possession of empires. Following that, a new vector R with the same size as P is formed. Arrays in this vector are random numbers in [0,1] with the uniform distribution.

$$R=\left[r_1,r_2,r_3,...,\ r_{N_{imp}}\right] \qquad r_i\in U(0,1) \tag{13}$$

Then, a new vector D is generated by Eq.(13).

$$D=P\text{-}R=\left[D_1,D_2,D_3,...,\ D_{N_{imp}}\right]=\left[P_{P_1}\text{-}r_1,P_{P_2}\text{-}r_2,\ P_{P_3}\text{-}r_3,\ ...,\ P_{P_{N_{imp}}}\text{-}r_{N_{imp}}\right] \tag{14}$$

The empire with the greatest value of D will take over the colony. When the colony is owned by one of empires, this step of algorithm is finished.

Step 7: Convergence

Steps 2 to 6 are repeated several times and finally all empires except one are eliminated. All colonies of other empires are gathered in the only remaining empire and they are optimized. The imperialist of this empire is the optimum solution with the minimum error.

IV. ICLA PERFORMANCE ASSESSMENT

In order to show the performance of the new algorithm, we conduct a number of experiments. Actually, we used four FCM models to examine the performance of the ICLA. At first a real-life model developed by Tsadiras [34] was used. This model and has previously been used by Stach et al. [33] to test the performance of genetic-based FCM learning algorithms. Next, we used synthetic data to develop two other FCM models with 8 and 20 nodes. The ICLA was applied to learn these FCM models and their performance was compared against the RCGA. The forth model is devoted to represent a real case problem regarding the water demand forecasting in the island of Skiathos (Greece). The proposed ICLA was applied to learn this FCM model by considering real measurements of water demand as a set of historical data.

The first FCM model which consists of 7 nodes (C1 to C7) represents causal relationships between the concepts in the strategic planning process of an e-business company. The seven concepts of this FCM model are: (C1) e-business profits, (C2) e-business sales, (C3) prices cut offs, (C4) customer satisfaction, (C5) staff recruitments, (C6) impact from international e-business competition, and (C7) better e-commerce services. These concepts have positive and negative influences on each other. The input data for learning this model is shown in Fig. 6. The same input data was used in [33].

Firstly, the RCGA algorithm was used to learn the FCM and find weights of relationships between concepts. To compare the result of the ICLA against the RCGA, we used $Errorl_{p1}$ as the main performance criterion. Stach et al. [33] used the same criterion with a different name. The name of this criterion in [33] was "error-initial". They reported that the error-l_{p1}/error-initial (± stdev) using the RCGA was 0.004 (± 0.005).

In this study, we applied the ICLA to the same set of data series which were used by Stach et al. [33]. The error-l_{p1} calculated by the ICLA for 10 times of running the algorithm was 0.0011 (±8.88E-04). We also calculated error-l_{p2} (± stdev) for our research goals and it was 3.89E-06 (±1.12E-05). As can be seen, the amount of error in ICLA is lower than the RCGA. The ICLA has another advantage: the number of function evaluations (NFE) in ICLA is much lower

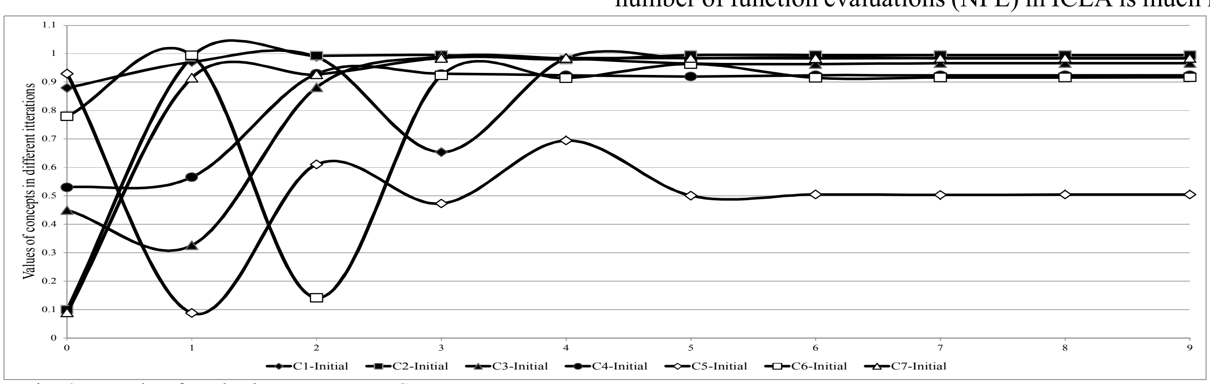

Fig. 6. Input data for e-business company FCM.

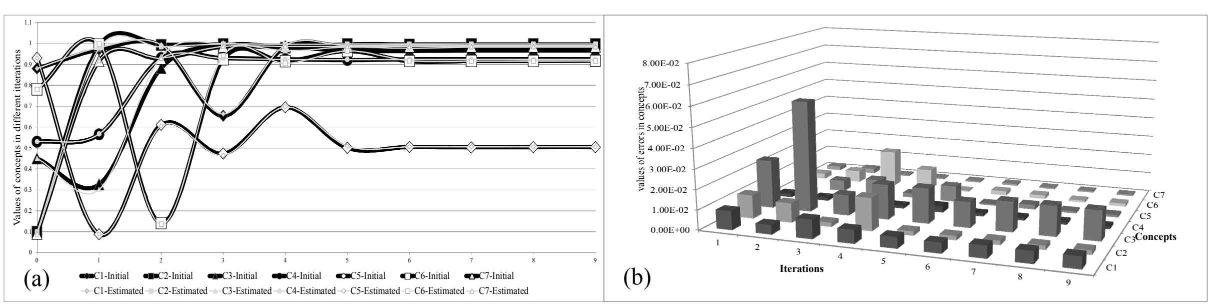

Fig. 7. Selected results achieved for the e-business FCM: (a) Comparison between state vector result from the CLA and the initial state vector; (b) This graph magnifies the errors between the nodes in (a) plotted them as a 3D bar chart.

than the RCGA. Stach et al. [33] reported that the NFE in RCGA was 3.00E+07; however the NFE in ICLA is 5.00E+05 when we used 5 empires with 100 countries and ran the algorithm for 1000 iterations. The detailed comparison between the ICLA and RCGA is shown in column "7 nodes" in Table II.

The NFE shows the overall number of execution of the cost functions during the whole process of running the algorithm. For instance, if the algorithm, in each iteration, generates 100 individuals and is executed for 1000 iteration, the NFE for this case is 1.00E+05.

The final result for learning the FCM model by applying the ICLA is shown in Fig. 7 (a). Each trace in Fig. 7 (a) actually comprises two traces - the results from both the estimated and the initial time series. The two traces are so close that they cannot be separated visually. Differences between initial time series and calculated outputs of ICLA are shown in Fig.6 (b). The highest difference appears in concept (3) iteration 2 and is approximately 5.58E-02. This error is about 6 % of the node value 0.8816 at that iteration.

In the case of FCM models produced from synthetic data we used two FCM models with 8 and 20 nodes. The same data sets were respectively used by Stach et al. [33] and Stach et al. [31]. These models were learnt by using the RCGA and divide and conquer RCGA. The performance comparisons between the two learning algorithms ICLA and RCGA are gathered in Table II.

TABLE II
COMPARISON BETWEEN ICLA AND RCGA

Algorithm	Performance Criterion		FCM models (number of nodes)		
			7 nodes	8 nodes	20 nodes
RCGA	error-l_{p1}	Min	Not reported	Not reported	Not reported
		Average	4.0E-03	5.53E-03	6.4E-03
		Max	0.004±0.005	Not reported	Not reported
	error-l_{p2}	Min	Not reported	Not reported	Not reported
		Average	Not reported	1.4E-05	Not reported
		Max	Not reported	Not reported	Not reported
	NFE		3.00E+07	3.00E+07	3.00E+07
ICLA	error-l_{p1}	Min	8.1E-04	1.0E-03	5.5E-03
		Average	1.1E-03	1.7E-03	6.7E-03
		Max	5.2E-03	2.3E-03	7.9E-03
	error-l_{p2}	Min	1.9E-06	2.2E-06	1.0E-04
		Average	3.9E-06	7.9E-06	1.7E-04
		Max	7.8E-06	1.3E-05	2.8E-04
	NFE		5.00E+05	5.00E+05	5.00E+05

Table II shows that in different FCM models with different number of nodes, the errors between the estimated time series and the initial state vectors are very small. In addition, it can be seen that in two FCM models with 7 and 8 nodes ICLA performs better than the RCGA and in the case of FCM model with 20 nodes, ICLA and RCGA have similar performance, however; the convergence speed of ICLA is much faster than the RCGA. In general terms, we can conclude from this that the ICLA preforms better than the RCGA algorithm in reported cases.

To further show the advantageous performance of ICLA algorithm, 200 measurements, concerning real water demand values from the island of Skiathos, was used to learn the FCM model which consists of four nodes. In this case, we tried to learn the FCM for water demand from real data, and next to proceed with the prediction using the learned FCM model.

The period on which the models are calibrated and validated is approximately three-year, from January 2011 to October 2013. The daily time series were normalized, after the outliers were removed. The scope was to learn the FCM model using the first 200 time series, by producing optimum weight matrix, and next with this weight matrix to predict the new 100 daily water demand values. The predicted values are compared with the real ones, using the performance criteria errors Lp1 and Lp2. The algorithm was run for a certain number of generations (3000 generations). The two reported performance criteria, bestCost (Error-Lp1) and bestCost_2 (Error-Lp2)) are the two considered criteria for the average best individuals calculated in the ten generations. The results of errors for the learning phase are avg_bestCost (Error-LP1) = 0.1255 (±0.0028), avg_bestCost_2 (Error-LP2) = 0.0242 (±0.0012), whereas the results of testing using the next 100 daily values, are avg_bestCost(Error-Lp1)=0.2346, avg_bestCost (Error-Lp2) = 0.0792.

This study is focused on FCM learning, thus we only present some preliminary results from prediction, which are considered not so sufficient regarding the produced errors, and especially the Error_Lp1 for the model performance. However, due to seasonality of data and other factors involved like weather, the predicted errors could be acceptable at this phase. The reported results from the learnt FCM model used for water demand forecasting are promising for our research continuation to this direction. Our future steps concern on the implementation of the proposed ICLA in time series prediction and forecasting problems, which is a new investigated area for the use of FCMs, thus to show further their performance and functionality in real situations.

V. CONCLUSION

All the previous evolutionary algorithms for learning FCM models have been inspired by the behaviour of animals in nature or the structure of genes inside the body of humans.

In this paper, a new powerful and innovative method for learning FCMs was developed by using the imperialist competitive algorithm. This algorithm is an alternative evolutionary one for FCM learning which is inspired by the behaviour of humans in their interactions in society. ICA is inspired by the social political process of imperialism and imperialist competition and it is a powerful and accurate algorithm for minimizing error and achieving optimum FCM learning. This new alternative algorithm is suited to FCM learning and through the testing has given very promising results. The ICLA improves the quality of learning FCMs in terms of convergence and accuracy and it has no limitation in learning performance when the number of nodes increases in FCM models.

The produced results provide a guideline for other learning methods. One of interesting and open issues is the use of the other heuristics methods which are inspired from humans' behaviour and interactions for learning FCMs and comparison of them with the others.

VI. Acknowledgment

The work of Elpiniki Papageorgiou was supported by the project ISS EWATUS--Integrated Support System for Efficient Water Usage and Resources Management--which is implemented in the framework of the 7th Framework Programme, Specific programme Cooperation Information and Communication Technologies; Grant Agreement Number 619228.

References

[1] S. Abdinnour-Helm, M.L. Lengnick-Hall, C.A. Lengnick-Hall, "Pre-implementation attitudes and organizational readiness for implementing an enterprise resource planning system," *European Journal of Operational Research*, Vol. 146, No. 2, pp. 258-273, 2003.

[2] S. Alizadeh, M. Ghazanfari, "Learning FCM by chaotic simulated annealing," *Chaos, Solitons & Fractal*, Vol. 41, No. 3, pp. 2282-1190, 2009.

[3] S. Alizadeh, M. Ghazanfari, M. Jafari, S. Hooshmand, "Learning FCM by Tabu Search," *International Journal of Computer Science*, Vol. 2, No. 2, pp. 142-149, 2007.

[4] E. Atashpaz-Gargari, C. Lucas. "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," *in Proc. IEEE Congress on Evolutionary Computation*. Singapore, pp. 4661 - 4667, 2007.

[5] R. Axelrod," *Structure of decision: the cognitive maps of political Elites*, New York: Princeton University Press, 1976.

[6] A. Baykasoglu, Z.D.U. Durmusoglu, V. Kaplanoglu, "Training Fuzzy Cognitive Maps via Extended Great Deluge Algorithm with applications," *Computers in Industry*, Vol. 62, No. 2, pp. 187-195, 2011.

[7] Y. Chen, L. Mazlack, L. Lu. "Learning fuzzy cognitive maps from data by ant colony optimization," *in Proc. The fourteenth international conference on Genetic and evolutionary computation conference (GECCO '12)*. New York, NY, USA, pp. 9-16, 2012.

[8] D. Graupe," *Principles of Artificial Neural Networks*, Second ed, Chicago, USA: World Sceintific, 2007.

[9] M. Khan, S. Khor, A. Chong, "Fuzzy cognitive map analysis with genetic algorithm," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 12, No. 2, pp. 31-42, 2004.

[10] B. Kosko, "Fuzzy cognitive maps," *International Journal of Man-Machine Studies*, Vol. 24, No. 1, pp. 65-75, 1986.

[11] D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris. "Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior," *in Proc. Evolutionary Computation*. COEX, Seoul, Korea, pp. 364-371, 2001.

[12] C. Lin, K. Chen, Y. He, "Learning fuzzy cognitive map based on immune algorithm," *WSEAS Transactions on Systems*, Vol. 6, No. 3, pp. 582-588, 2007.

[13] C. Lopez, J.L. Salmeron, "Dynamic risks modelling in ERP maintenance projects with FCM," *Information Sciences*, Vol. 256, No. 20, pp. 25-45, 2014.

[14] X. Luo, X. Wei, J. Zhang. "Game-based learning model using fuzzy cognitive map," *in Proc. the first ACM international workshop on Multimedia technologies for distance learning*. New York, NY, USA, pp. 67-76, 2009.

[15] S.S. Madeiro, F.J.V. Zuben. "Gradient-Based Algorithms for the Automatic Construction of Fuzzy Cognitive Maps," *in Proc. 11th International Conference on Machine Learning and Applications (ICMLA)*. Florida, USA, pp. 344-349, 2012.

[16] V.K. Mago, L. Bakker, E.I. Papageorgiou, A. Alimadad, P. Borwein, and V. Dabbaghiana, "Fuzzy cognitive maps and cellular automata: An evolutionary approach for social systems modelling," *Applied Soft Computing Journal*, Vol. 12, No. 12, pp. 3771–3784, 2012.

[17] N.H. Mateou, M. Moiseos, A.S. Andreou. "Multi-objective evolutionary fuzzy cognitive maps for decision support," *in Proc. The 2005 IEEE Congress on Evolutionary Computation*. Edinburgh, Scotland, pp. 824-830, 2005.

[18] S. Nazari-Shirkouhi, H. Eivazy, R. Ghodsi, K. Rezaie, E. Atashpaz-Gargaric, "Solving the integrated product mix-outsourcing problem using the Imperialist Competitive Algorithm," *Expert Systems with Applications*, Vol. 37, No. 12, pp. 7615–7626, 2010.

[19] E.I. Papageorgiou, "Learning Algorithms for Fuzzy Cognitive Maps—A Review Study," *IEEE Transactions on Systems Man and Cybernetics (SMC)-Part C*, Vol. 42, No. 2, pp. 150-163, 2012.

[20] E.I. Papageorgiou, P.P. Groumpos, "A new hybrid method using evolutionary algorithms to train Fuzzy Cognitive Maps," *Applied Soft Computing*, Vol. 5, No. 4, pp. 409-431, 2005.

[21] E.I. Papageorgiou, A. Kannappan, "Fuzzy Cognitive Map Ensemble learning paradigm to solve classification problems: application to autism identification," *Applied Soft Computing Journal*, Vol. 12, No. 12, pp. 3798–3809, 2012.

[22] E.I. Papageorgiou, K.E. Parsopoulos, C.D. Stylios, P.P. Groumpos, M.N. Vrahatis, "Fuzzy Cognitive Maps Learning Using Particle Swarm Optimization," *International Journal of Intelligent Information Systems*, Vol. 25, No. 1, pp. 95-121, 2005.

[23] E.I. Papageorgiou, J.L. Salmeron, "A Review of Fuzzy Cognitive Maps Research during the last decade," *IEEE Transactions on Fuzzy Systems*, Vol. 21, No. 1, pp. 66 - 79 2012.

[24] 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, pp. 219-249, 2004.

[25] K.E. Parsopoulos, E.I. Papageorgiou, P. Groumpos, M.N. Vrahatis. "A first study of fuzzy cognitive maps learning using particle swarm optimization," *in Proc. The IEEE Congress on Evolutionary Computation*. Canberra, Australia, pp. 1440-1447, 2003.

[26] Y.G. Petalas, E.I. Papageorgiou, K.E. Parsopoulos, G. P.P., M.N. Vrahatis. "Fuzzy cognitive maps learning using memetic algorithms," *in Proc. Int. Conference of Computational Methods in Sciences and Engineering, ICCMSE 2005*. Loutraki, Greece, pp. 1420-1423, 2005.

[27] Y.G. Petalas, K.E. Parsopoulos, M.N. Vrahatis, "Improving fuzzy cognitive maps learning through memetic particle swarm optimization," *Soft Comput*, Vol. 13, No. 1, pp. 77-94, 2009.

[28] L. Rodriguez-Repiso, R. Setchi, J.L. Salmeron, "Modelling IT projects success with Fuzzy Cognitive Maps," *Expert Systems with Applications*, Vol. 32, No. 2, pp. 543–559, 2007.

[29] W. Stach, L. Kurgan, W. Pedrycz, "A survey of fuzzy cognitive map learning methods," in: P. Grzegorzewski, M. Krawczak, and S. Zadrozny (Eds.), *Soft Computing: Theory and Applications*, Berlin: Springer, 2005, pp. 71-84.

[30] W. Stach, L. Kurgan, W. Pedrycz. "Parallel Learning of Large Fuzzy Cognitive Maps," *in Proc. International Joint Conference on Neural Networks*. Orlando, FL, pp. 1584 - 1589, 2007.

[31] W. Stach, L. Kurgan, W. Pedrycz, "A divide and conquer method for learning large Fuzzy Cognitive Maps," *Fuzzy Sets and Systems*, Vol. 161, No. 19, pp. 2515-2532, 2010.

[32] W. Stach, L. Kurgan, W. Pedrycz, "Expert-Based and Computational Methods for Developing Fuzzy Cognitive Maps," *Studies in Fuzziness and Soft Computing*, Vol. 247, No. pp. 23-41, 2010.

[33] W. Stach, L. Kurgan, W. Pedrycz, M. Reformat, "Genetic learning of fuzzy cognitive maps," *Fuzzy Sets and Systems*, Vol. 153, No. 3, pp. 371 - 401, 2005.

[34] A.K. Tsadiras. "Using fuzzy cognitive maps for e-commerce strategic planning," *in Proc. 9th Panhellenic conference on Informatics*. Thessaloniki, Greece, pp. 142-151, 2003.

[35] Z. Yanchun, Z. Wei. "An Integrated Framework for Learning Fuzzy Cognitive Map using RCGA and NHL Algorithm," *in Proc. 4th International Conference on Wireless Communications, Networking and Mobile Computing*. Dalian, pp. 1-5, 2008.

[36] A. Yastrebov, K. Piotrowska, "Simulation Analysis of Multistep Algorithms of Relational Cognitive Maps Learning," in: A. Yastrebov, B. Kuźmińska-Sołśnia, and M. Raczynska (Eds.), *Computer Technologies in Science, Technology and Education*: Institute for Sustainable Technologies - National Research Institute, 2012, pp. 126-137.

[37] E. Yesil, C. Ozturk, M.F. Dodurka, A. Sakalli. "Fuzzy Cognitive Maps Learning Using Artificial Bee Colony Optimization," *in Proc. IEEE International Conference on Fuzzy Systems*. Hyderabad, pp. 1 - 8, 2013.

[38] E. Yesil, L. Urbas, "Big Bang - Big Crunch Learning Method for Fuzzy Cognitive Maps," *World Academy of Science, Engineering and Technology*, Vol. 47, No. 1, pp. 816-825, 2010.