Expert-Based and Computational Methods for Developing Fuzzy Cognitive Maps

Wojciech Stach, Lukasz Kurgan, and Witold Pedrycz

Abstract. Development of Fuzzy Cognitive Maps (FCMs) that accurately describe a given dynamic system is a challenging task which in many cases cannot be fully completed based solely on human expertise. Some of the reasons behind this limitation include potential bias of the human experts and excessive size of the problem itself. However, due to the lack of automated or semi-automated methods that would replace or support designers, most of existing FCMs were developed using expert-based approaches. Interestingly, in the recent years we have witnessed the development of algorithms that support learning of FCMs from data. The learning corresponds to the construction of connection matrices based on historical data presented in the form of multivariate time series. Since the FCM may include feedback loops and they incorporate nontrivial transformation functions, forming these models from data is a complex task that requires searching through a large solution space. The existing automated learning methods are based either on the Hebbian learning or they apply evolutionary algorithms. This chapter formulates the task of learning FCMs and describes the corresponding design challenges. We present a comprehensive survey of the current expert-based and semi-automated/automated methods for learning FCMs. The leading learning methods are described and analyzed both analytically and experimentally with the help of a case study. We also contrast computational approaches versus expert-based methods and outline future research directions.

1 Introduction

Fuzzy Cognitive Maps (FCMs), which were introduced by Kosko (Kosko 1986) as an extension to Cognitive Maps (Axelrod 1976), are a powerful machinery for modeling of dynamic systems. They define a system as a collection of interconnected concepts where connections reflect cause-effect relationships between the

---

Wojciech Stach · Lukasz Kurgan · Witold Pedrycz
Department of Electrical and Computer Engineering,
University of Alberta, Edmonton, Canada
e-mail: {wstach, lkurgan, pedrycz}@ece.ualberta.ca

M. Glykas (Ed.): Fuzzy Cognitive Maps, STUDFUZZ 247, pp. 23–41, 2010.
springerlink.com

concepts. The relationships are represented as directed edges of the graph of the FCM and quantify the strength of causal effects between the concepts. FCMs have been recognized as a useful and flexible technique in problem solving where many decision variables are causally interrelated. In addition, they have a convenient graph representation, which consists of nodes (concepts) and weighted casual edges (relationships) representing knowledge that is easy to visualize and manipulate (Aguilar 2005). These advantages motivated application of FCMs to diverse domains including engineering (Stach et al. 2004b), medicine (Innocent and John 2004), economics (Xirogiannis and Glykas 2004), e-business (Xirogiannis and Glykas 2007), financial organizations (Glykas and Xirogiannis 2005), human management (Xirogiannis et al 2008), environmental sciences (Giordano et al. 2005), politics (Andreou et al. 2005), to name just a few. In parallel to the widespread applications, the last decade observed significant research efforts into building methodologies and tools for the development, aggregation, simulation, and analysis of the FCMs (Aguilar 2005). This chapter is entirely devoted to methods related to the development of Fuzzy Cognitive Maps.

Figure 1 shows an example process control problem which was discussed by Papageorgiou et al. (Papageorgiou et al. 2003). Two valves, *valve 1* (*V1*) and *valve 2* (*V2*), supply two different liquids into the tank. The liquids are mixed and a chemical reaction takes place. The control objective is to maintain the desired level of liquid and its specific gravity. *Valve 3* (*V3*) is used to drain liquid from the tank.

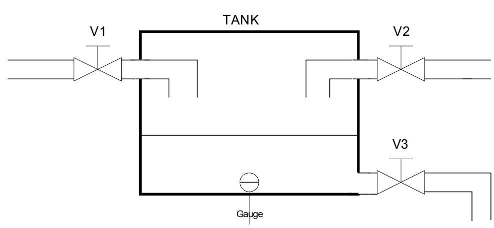


Fig. 1 Process control problem

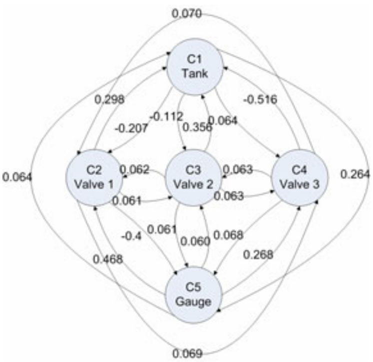
The corresponding Fuzzy Cognitive Map that describes this system and allows for its simulation can be developed using a wide range of methods described in Sections 2 and 3. At this point, we introduce the background of FCMs based on the analysis of the FCM model that was developed for this control system by Papageorgiou et al. We also present a brief description of how an FCM model can be used to perform various tasks of analysis and simulations in order to obtain useful knowledge about the system being modeled.

The FCM model for the system presented in Figure 1 involves the following five concepts:

C1 – the amount of the liquid in the tank
C2 – the state of valve 1
C3 – the state of valve 2

C4 – the state of valve 3
C5 – the specific gravity of the liquid in the tank

These concepts are connected as illustrated in the form of a graph shown in Figure 2. The figure also shows a connection matrix which stores weights associated with directed connections between all pairs of the concepts; the matrix is equivalent to the corresponding FCM graph.



	C1	C2	C3	C4	C5
C1	0	-0.207	-0.112	0.064	0.264
C2	0.298	0	0.061	0.069	0.067
C3	0.356	0.062	0	0.063	0.061
C4	-0.516	0.070	0.063	0	0.068
C5	0.064	0.468	0.060	0.268	0

Fig. 2 FCM graph along with its connection matrix for the process control problem

Using either the graph or the connection matrix, one can perform *static analysis* of the model using techniques of graph theory, as adopted by Tsadiras et al. (Tsadiras et al. 2001). This analysis includes identification of cycles to uncover nontrivial relationships between concepts, calculation of the model density to obtain an indication of its complexity, and an analysis of importance of individual nodes. For instance, the importance can be quantified by adding up absolute values of all weights for connections both entering and leaving a given concept (node of the graph). Consequently, the corresponding values for the control process model from Figure 2 are C1=1.88, C2=1.30, C3=0.84, C4=1.18, and C5=1.32. The two concepts with the highest values are C1 and C5, which suggests they play the most important role in this system. This conclusion is consistent with the explanation of the system provided in the paper by the experts, i.e., these concepts define the control target task, which is to maintain desired liquid gravity (C5) by keeping the liquid level (C1) within a defined range. The reader may refer to the original paper or other relevant resources that describe methods for the static analysis (see e.g., Gross and Yellen 1998) for further details.

The usage of different FCM development methods may result in different maps. Static comparison of such maps, which relies solely on the values of the connection matrices, is based on the following sum of the average absolute differences reported between the corresponding weight values (Stach et al. 2004a).

$$Matrix-error=\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\left|e_{ij}^{1}-e_{ij}^{2}\right| \tag{1}$$

where e_{ij}^{1}, e_{ij}^{2} are the weights for relation from concept C_i to C_j in the FCM models 1 and 2, respectively.

The *dynamic analysis* of the FCM model allows the user to draw additional observations and conclusions concerning the underlying system, which are not available through the static analysis. The dynamic analysis is concerned with the simulation of the FCM system as a whole and the simulation of its constituent components. This provides insights into existence, interactions and dependencies between the concepts in successive iterations of the simulation. This type of analysis allows exploring "what-if" scenarios by performing simulations when imposing different initial conditions on concepts. It offers description of dynamical behavior of the underlying system, which can be used to support decision making (Stylios et al. 2008) and/or predictions about its future states (Stach et al. 2008b).

Dynamic analysis of FCMs is based on an execution model which calculates concepts *activation levels* in successive iterations. Activation levels determine degrees of presence of a given concept in the system and are represented by floating-point numbers between 0 (inactive) and 1 (active). For the control system example, the activation level of each valve determines degree to which it is open. The value of 0 means that a given valve is closed, value of 1 means that it is fully opened, and the remaining values represent the valve being partially opened. The simulation also requires defining initial values of all concepts (also called *initial condition* or initial *state vector*). To calculate successive values of all concepts (Kosko 1986), called *system state* (state vector) we use the following expression:

$$\forall j \in \{1,\dots,N\}, C_j(t+1) = f\left(\sum_{i=1}^{N} e_{ij}C_i(t)\right) \tag{2}$$

where $C_j(t)$ is the activation level of concept j^{th} at iteration t, e_{ij} is the weight for relationship from concept C_i to C_j, and f is the transformation (transfer) function.

The transformation function is used to normalize concepts' values to the range [0,1], which allows for comparison of activation levels between different concepts. The most popular functions are continuous, although in some research work binary functions were used. The binary functions limit the dynamic analysis of concepts just to two values that correspond to two linguistic terms, inactive and active. Comparison of different transfer functions for FCMs was recently carried out by Tsadiras (Tsadiras 2008).

Figure 3 shows a result of a simulation of the control model completed when using continuous transfer function and the initial state vector suggested in the original paper, i.e., **C**[0] = [0.4, 0.708, 0.612, 0.717, 0.3].

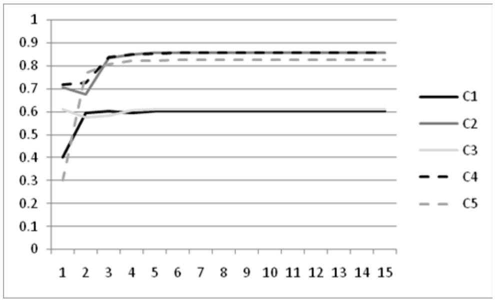


Fig. 3 Sample FCM simulation result of the control system model from Fig 2

Typically, the concept values in an FCM simulation either converge to particular state (referred to as the *fixed-point-attractor*) or they keep cycling between a fixed set of states (referred to as the *limit-cycle*). The dynamic analysis usually investigates several aspects, such as activation levels of concepts at the final state (if there is any) or cycles (intervals, concepts activation levels within the cycle), changes/trends in the activation levels throughout the simulation for either all concepts or a subset of concepts that is of interest to the user. The simulation from Figure 3 follows the fixed-point-attractor. In order to achieve the desired final values of concepts C1 and C5, when compared to their initial state, the following actions on valves are taken: the opening of the valves 1 and 3 is increased by 21% and 19%, respectively, whereas the opening of valve 2 is reduced by 1%. This leads the system to the stable state in as little as four iterations after some adjustments are made within the first three iterations. Detailed dynamic analysis of this particular model is presented in (Papageorgiou et al. 2003). A more sophisticated FCM dynamic analysis can be performed using time-series analysis (e.g. Hamilton 1994).

Similar to the *matrix-error* measure that was introduced for the static analysis, the most commonly used formula to compare two simulation results is based on the sum of the absolute average differences between the corresponding concept values normalized per concept per iteration (Stach et al. 2005b) as described below

$$Simulation-error = \frac{1}{(K-1)\cdot N}\sum_{t=1}^{K-1}\sum_{i=1}^{N}\left|C_i^1(t)-C_i^2(t)\right| \tag{3}$$

where $C_i^1(t), C_i^2(t)$ are values of i^{th} node at t^{th} iteration obtained from simulation of the FCM models 1 and 2, respectively, K is the number of available iterations to compare (we ignore the initial state vector since it is always the same for both models), and N is the number of concepts.

Following that, we will focus on the central theme of this chapter which is the design of FCMs models, including deciding on the concepts relevant to a given system and defining relationships (weights) between the selected concepts. Generally speaking, Fuzzy Cognitive Maps models can be developed *by experts* and/or *computationally* (in either *automated or semi-automated* fashion) (Stach et al. 2005a). The expert-based approach can be classified as *deductive modeling* and involves application of human expertise in a given domain. In contrary, the computational methods can be classified as *inductive modeling* and they use available data and a learning algorithm to develop or to support development of an FCM model for a given system. In spite of their shortcomings, FCM models were developed almost exclusively using expert-based methods. The main reason for that is that the computational methods were introduced relatively recently (Stach et al. 2005a).

In this chapter, we present a comprehensive review of different approaches to develop FCMs. Next section is devoted to the expert-based methods and includes explanation of the steps that are performed by an expert to create an FCM model. We also describe how to combine multiple maps that are created by different experts for the same underlying system. Section 3 provides a systematic survey of computational methods and includes both semi-automated (which require some involvement of a human expert) and automated methods. We also demonstrate working of the considered FCM development methods using the example control system model and we provide a side-by-side comparison of these methods. Finally, conclusions and future research directions are outlined in Section 4.

2 Expert-Based Methods

2.1 Overview

Expert-based development of Fuzzy Cognitive Maps relies entirely on human expertise and domain knowledge. The relatively simple model representation makes it possible to simply manually draw the graph that corresponds to an FCM using only a pencil and a sheet of paper. The experts are also required to have a rudimentary knowledge of the FCM theory to understand the meaning of the weights and the direction of the causal effects. In order to increase credibility of the model, a group of experts instead of a single person may be involved in the development process. Experts can work together or design individual maps that represent their own understanding of a given system. In the latter case the individual maps can be combined into a single model.

2.2 Development of FCMs by a Single Expert

The expert-based development of FCMs usually consists of the following three steps (Kosko 1986, Khan and Quaddus 2004)

1. Identification of important concepts.
2. Identification of causal relationships among these concepts.
3. Estimation of the strength of the causal relationships.

In the first step, the decision which from among all available concepts should be included in the model has to be made. The most intuitive strategy is to create a list of all relevant concepts and remove the insignificant ones. In the second step, all cause-effect direct relationships between the remaining concepts have to be identified, including their directions. Usually this is accomplished by focusing on one pair of concepts at a time, since then the expert is relieved of the task of coming up with hidden or indirect cause–effect relationships. These relationships become apparent later through analyses carried out using the completed FCM. These first two steps result in a structural design which consists of a graph with nodes and directed edges.

The main challenge in expert-based development of FCMs is to accurately estimate the strength of the relationships. We note that the number of weights exhibits quadratic growth with the number of concepts, which may lead to difficulties in developing maps with several dozens of concepts. Following the original paper (Kosko 1986), each relationship strength value (weight) is expressed by a real number from the [-1,1] interval. The value of 0 denotes no relationship and is implicitly assigned at the end of the second step. Higher absolute values represent stronger relationships, whereas the sign defines the type: *promoting* (positive numbers) or *inhibiting* (negative numbers). Theoretically, each weight can take on an infinite numbers of values. Consequently, this step is potentially susceptible to subjective judgment of a given expert. A common practice to facilitate the estimation of the weight values is to first describe each relationship by a linguistic term and next to transform these terms into numerical values. The corresponding work can be divided into the following three steps (Kosko 1986, Khan and Quaddus 2004)

1. Determining the sign of each relationship.
2. Describing each relationship by means of linguistic terms, e.g. *weak, medium, strong* and *very strong.*
3. Mapping the linguistic terms to numerical values, e.g. *weak* to 0.25, *medium* to 0.5, *strong* to 0.75, and *very strong* to 1.0.

The use of the linguistic expressions to describe the degrees of causality in relationships allows the experts to avoid the difficult task of specifying the precise numerical values before a draft model is established. Additionally, analytical procedures, such as *Analytical Hierarchy Process* (Saaty 1980), may be helpful to find the numerical values used in the last step of the weight estimation procedure.

2.3 FCM Development by a Group of Experts

Fuzzy Cognitive Maps allow for a relatively simple aggregation of knowledge obtained from several experts. The aggregation should improving reliability of the final model which is less susceptible to potentially erroneous beliefs of a single expert. There are a couple of procedures for combining multiple FCMs into a single, final model. They involve simple matrix operations, such as summations and multiplications by a number (Kosko 1988), which are computed using the connection matrices developed by individual experts.

It is not uncommon that experts decide on different number of concepts. Consequently, the sizes of corresponding matrices may not be the same and/or the corresponding rows/columns may concern different concepts. In such a case, the first step towards combining the maps is to equalize their sizes. The connection matrices are augmented by including any missing concept(s), when considering all concepts in all input maps, through addition of extra rows and columns of all zeros. In other words, the omitted concepts are added “superficially” by assigning them with no incoming and outgoing relationships with other concepts. If the total number of distinct concepts over all input FCMs equals N, then each connection matrix is augmented to the matrix of $N \times N$ size (Khan and Quaddus 2004).

Assuming no additional information on the credibility of individual experts or assuming that all experts are equally credible, the simplest method for combining the maps is to calculate average of each relationship weight across all experts. Therefore, for k experts, the connection matrix of the final FCM is established by the following expression (Kosko 1988):

$$E = \frac{1}{k}\left(E_1 + E_2 + \ldots + E_k\right) \tag{4}$$

In this approach, each expert contributes equally to the final model. This basic formula can be easily modified to accommodate credibility of different experts by assigned a weight w_i that quantifies credibility of the i^{th} expert. These weights take value from [0,1] range and their sum is usually normalized, that is . The final, combined model is calculated using the following weighted average:

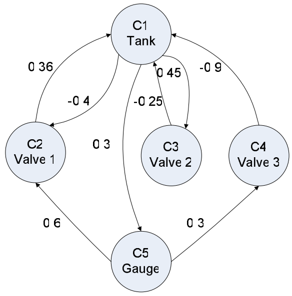
$$E = \frac{1}{\sum_{i=1}^{k} w_i}\left(w_1 E_1 + w_2 E_2 + \ldots + w_k E_k\right) \tag{5}$$

As a result, the experts with higher credibility have stronger influence on the structure of the final model than those with lower credibility. More detailed discussion on the assignment of the credibility weights can be found in the literature (e.g. Taber and Siegel 1987).

2.4 Example

Figure 4 shows the FCM model proposed by experts for the process control problem defined in Figure 1 (Papageorgiou et al. 2003). The model was developed by three experts who had good understanding of the modeled system. Firstly, they jointly agreed on the set of concepts used in the model. Secondly, each expert drew the relationships between the concepts and assigned weights for each relationship. Finally, the models were merged using the technique described in Section 3; the original paper does not mention whether credibility weights were used. This model was later updated to the final map shown in Figure 2.

Static comparison between the expert-derived map and the final map from Figure 2 shows that these maps are similar. The experts’ map is sparser, i.e., some of the weights that have small magnitude in the final model are rounded to zero. The weights that have larger magnitude have the same sign between the two maps. We also observe that the weight developed by experts have lower precision. The *matrix-error* between the two models is small and equals 0.09 (with the standard deviation of 0.07). Therefore, the static analysis of these two models gives very similar results. For instance, the most important concept in both models



	C1	C2	C3	C4	C5
C1	0	-0.4	-0.25	0	0.3
C2	0.36	0	0	0	0
C3	0.45	0	0	0	0
C4	-0.9	0	0	0	0
C5	0	0.6	0	0.3	0

Fig. 4 FCM model of the process control problem that was developed by three experts

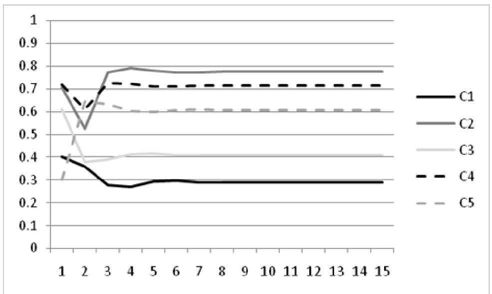


Fig. 5 Simulation result for the experts' map of the process control system from Fig. 4

is C1 with the corresponding importance measure equal 2.66 and 1.88 in the expert-based and final models, respectively. At the same time, the second and third important concept in the expert model are C2 and C5 with the values of 1.36 and 1.20, respectively, while in the final model they are C5 and C2 with the values of 1.32 and 1.30, respectively. The model proposed by the experts is structurally less complex since it has fewer non-zero connections between nodes. We observe that the expert-based models generated are usually sparsely connected (Stach et al. 2005b).

Although the expert and the final map are similar from the static analysis point of view, their simulations for the same initial conditions, which are shown in Figures 5 and 3, respectively, differ quite significantly. The *simulation-error* equals 0.19 (with the standard deviation of 0.08), which is relatively large considering that the error values range in-between 0 and 1. Therefore, the conclusions resulting from the dynamical analysis of the two models will be likely different. For instance, the amount of liquid in the tank (C1) after completing the simulation is approximately two times lower than for the final model, while the gravity of the liquid (C2) is only slightly lower. This demonstrates that models that are similar based on the static analysis may exhibit different dynamic behavior, while the experts usually do not consider the dynamic behavior in their modeling.

2.5 Summary

Expert-based methods for the development of Fuzzy Cognitive Maps are well established and have been extensively applied to real life modeling tasks in diverse domains (Aguilar 2005, Stach et al. 2005a). The popularity of the expert-based methods stems from at least two reasons. Firstly, these methods are relatively simple and straightforward. All the steps in the development process are clearly defined and described in the literature, the development does not requires sophisticated knowledge of the underlying modeling technique, and the expert knowledge is represented by a simple to comprehend graph. The second reason is the lack, until recently, of alternative methods for FCM development. These methods, which are described in the next Section, provide support to or replace experts from the model development task.

We also observe that expert-based methods aim at developing structure of the model that corresponds to expert(s) understanding of a given system. The experts virtually never simulate the model to verify whether its dynamic behavior is correct. Therefore, models created by expert(s) usually provide good static description of the system, and are better suited for the static analysis. On the other hand, the dynamic analysis (simulations) of the experts-derived maps may lead to inaccuracies when compared with the actual system.

3 Computational Methods

3.1 Overview

Computational methods utilize historical data available for a given system to establish FCM model. Semi-automated methods require a relatively limited human intervention, whereas fully automated approaches are able to compute the FCM model solely based on the historical data, i.e. without any human input. A number of algorithms for learning FCM structure from data have been recently proposed. They can be categorized into two groups based on the learning paradigm used, i.e., Hebbian-based learners and methods based on evolutionary algorithms. The following two subsections describe chronologically algorithms from the two groups.

3.2 Hebbian-Based Methods

In one of the first attempts, Dickerson and Kosko proposed a simple *Differential Hebbian Learning (DHL)* (Dickerson and Kosko 1993, Dickerson and Kosko 1994) method, which is based on Hebbian theory (Hebb 1949). During DHL learning the values of weights are iteratively updated until the desired structure is found. In general, the weights of outgoing edges for each concept in the connection matrix are modified only when the corresponding concept value changes,

$$e_{ij}(t+1)=\begin{cases} e_{ij}(t)+c_t\left[\Delta C_i \Delta C_i - e_{ij}(t)\right] & \text{if } \Delta C_i \neq 0 \\ e_{ij}(t) & \text{if } \Delta C_i = 0 \end{cases} \tag{6}$$

where e_{ij} denotes the weight for relation from concept C_i to C_j, ΔC_i represents the change in the C_i concept's activation value, t is the iteration number, and c_t is a learning coefficient. The learning coefficient is a small constant which values usually decrease as the learning progresses. The main drawback of this learning method is that the formula updates weights between each pair of concepts taking into account only these two concepts and ignoring the influence from other concepts.

An improved version of DHL learning was introduced by Huerga (Huerga 2002). The new algorithm, called *Balanced Differential Algorithm (BDA)* eliminates one of the limitations of DHL method by taking into account all the concept values that change at the same time when updating the weights. More specifically, the modified formula for $e_{ij}(t+1)$ takes into consideration changes in all concepts if they occur at the same iteration and have the same direction. Empirical comparison between DHL and BDA demonstrates that the latter method improves quality of the learned maps (Huerga 2002). On the other hand, the BDA algorithm was applied only to binary FCMs, i.e., maps with binary transfer functions, which limits its application areas.

One year later, Papageorgiou and colleagues introduced *Nonlinear Hebbian Learning (NHL)* algorithm (Papageorgiou et al. 2003). While this algorithm originates from the same learning principles, it uses a nonlinear extension to the basic Hebbian rule (Oja et al. 1991) by introducing modified weight update formula. The NHL learning method has been designed as a semi-automated approach that requires initial human intervention. Experts are required to suggest nodes that are directly connected and only these edges are updated during learning. In addition, the experts have to indicate sign of each edge according to its physical interpretation. The algorithm updates the corresponding weights while preserving their initial signs. In a nutshell, the NHL algorithm allows obtaining model that retains initial graph structure imposed by the expert(s), and therefore requires human intervention before the learning process starts. Also, the experts have to define output concepts and specify range of values that these concepts can take. The latter is used after every update of the learned model's weights to validate the model. The validation is based on checking whether the model state satisfies these constrains.

The same research group proposed *Active Hebbian Algorithm (AHL)* in 2004 (Papageorgiou et al. 2004). This approach introduces and exploits the task of determination of the sequence of activation concepts. Expert(s) determines the desired set of concepts, initial structure and the interconnections of the FCM structure as well as the sequence of activation concepts. A seven-step AHL procedure, which is based on Hebbian learning, is iteratively used to adjust the weights to satisfy defined predefined stopping criteria.

In a recent work, Stach and coworkers proposed an improved version of the NHL method (Stach et al. 2008a). The algorithm, called *Data-Driven Nonlinear Hebbian Learning (DD-NHL),* is based on the same learning principle as NHL, but it takes advantage of historical data (a simulation of the actual system) and uses output concepts to improve the learning quality. An empirical comparative study have shown that if historical data are available, then the DD-NHL method produces better FCM models when compared with those developed using the generic NHL method (Stach et al. 2008a).

3.3 Evolutionary Algorithms-Based Methods

In 2001, Koulouriotis and colleagues applied the Genetic Strategy (GS) to learn FCM's model structure, i.e., weights of relationships, from data (Koulouriotis et al. 2001). In their method, the learning process is based on a collection of input/output pairs, which are referred to as examples. The learning requires historical data consisting of multiple sequences of state vectors (multiple simulations of the system). The algorithm computes the structure of an FCM that is able to generate state vector sequences that transform the input vectors into the output vectors. The main drawback of this approach is that it requires multiple state vector sequences, which might be difficult to obtain in some of the application domains.

Particle Swarm Optimization (PSO) method, proposed by Parsopoulos and coworkers, belongs to the class of Swarm Intelligence algorithms (Parsopoulos et al. 2003). This method aims at learning FCM structure based on historical data that converge to a desired final state. PSO is a population based algorithm, which performs a search for the solution by maintaining and transforming a population of individuals. The learning requires human knowledge that is used to specify adequate constraints, which would guarantee that the relationships within the FCM model retain the physical meaning defined by the expert(s).

The next algorithm proposed by Khan and Chong aims to accomplish a different learning objective (Khan and Chong 2003). Instead of learning the structure of the FCM model, their goal was to find an initial state vector (initial condition) that leads a given model to the specified end state. Their method employed genetic algorithms to find the initial state.

A fully automated method for learning FCMs, which is based on real-coded genetic algorithms (RCGA), was introduced by Stach and colleagues in 2005 (Stach et al. 2005b). The RCGA is a floating-point extension (Herrera et al. 1998) to genetic algorithms (Goldberg 1989). This extension was used to allow finding floating point weights instead of weights that take on a limited set of values. The core of this approach is a learning module which exploits RCGA to find FCM structure that is capable of mimicking a given input historical data. This approach is flexible in terms of the input data as it can use either one or multiple sets of concepts values over successive iterations. A follow-up of this work includes analysis of the quality of the RCGA-based learning depending on the amount of the available historical data (Stach et al. 2004a). It demonstrates that the RCGA-based method can generate FCM models that are identical to models proposed by domain expert given the input data of sufficient size, and that increasing the amount of the input data improves accuracy of the learning.

Recently, the same research group introduced a parallel RCGA-based method that targets learning of large maps that consist of dozens of concepts (Stach et al. 2007). The method was reported to be up to four times faster than the sequential

RCGA learning when executed on eight processors. It allows learning maps that include several dozens of concepts in a few hours.

3.4 Example

We use the process control system from Figure 1 to investigate the quality of models learned using state-of-the-art learning methods. The data from Figure 3 were used to learn the FCM model and we tested three learning algorithms that include NHL and DD-NHL, two most recently proposed Hebbian learning based methods, and RCGA, which is the most recent genetic-based method.

Since the RCGA method is initialized with a 100 randomly generated maps, whereas the two other methods use just a single map, the experiments for both Hebbian-based methods were repeated 100 times using the 100 initial maps generated for the RCGA method. The final output was selected as the map that provides simulations with the lowest value of the *simulation-error*.

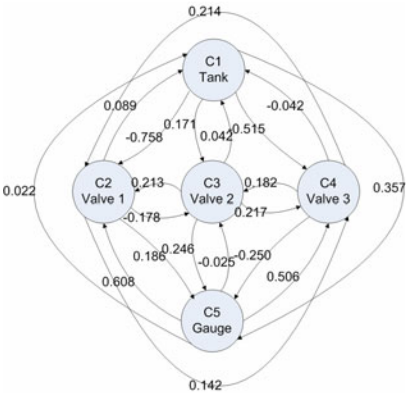
Table 1 presents a summary of the results. We computed both *matrix-error* and *simulation-error* to quantify the quality with respect to both the static and the dynamic analysis, respectively. In addition, the last column gives the learning time for each method in seconds. We report the average values together with the corresponding standard deviations (shown in brackets).

Table 1 Experimental results concerning comparison of the quality of FCM models for the process control system developed by the experts and learned using computational methods including NHL, DD-NHL and RCGA

Learning method	*Matrix-error*	*Simulation-error*	Time [s]
NHL	0.236 (0.162)	0.064 (0.053)	8
DD-NHL	0.245 (0.145)	0.056 (0.063)	9
RCGA	0.225 (0.154)	0.003 (0.004)	68
Expert-based	0.092 (0.078)	0.187 (0.077)	N/A

We show detailed results, i.e., the resulting map and its simulation, only for the RCGA method since it outperformed the two Hebbian-based methods on both quality criteria. Figures 6 and 7 present the FCM model and its simulation result for the same initial condition as in Figure 3, respectively. We note that the *simulation-error* values obtained with maps learned using the NHL and DD-NHL methods are substantially lower than the error of the expert-based method.

In spite of the relatively different connection matrix generated by RCGA (comparison with the final map, i.e., comparison of models from Figures 2 and 7, reveals that the *matrix-error* of RCGA equals 0.255 and that 25% of the weights have opposite signs), the *simulation-error* is very small (equals 0.003 and it reduces to 0.001 if we consider only the final state). This observation can be generalized, based on experiments reported in the literature (e.g. Stach et al. 2004a), to a statement that the structurally different maps can generate very similar simulations. The chances that the RCGA method will find suboptimal solutions, i.e., solutions with low *simulation-error* and relatively high *matrix*-error, decrease



	C1	C2	C3	C4	C5
C1	0	-0.758	0.171	-0.515	0.357
C2	0.089	0	-0.178	0.142	0.186
C3	0.042	0.213	0	0.217	0.246
C4	-0.042	0.214	0.182	0	-0.250
C5	0.022	0.608	-0.025	0.506	0

Fig. 6 FCM model of the process control system learned using RCGA method

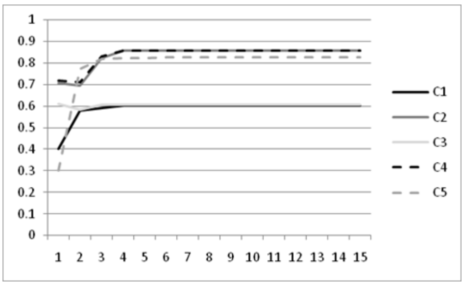


Fig. 7 Simulation result of the FCM for the process control system learned using the RCGA method

as the amount of the available input data increases (see Stach et al. 2004a for details). One has to be aware that suboptimal solutions can be generated by the fully automated learning approaches and in these cases results of the static analysis could be inaccurate.

In our example of dynamic analysis we compare the final states for the three simulations from Figures 3 (the actual system), 5 (expert-derived model), and 7 (best-performing automated computational model). The comparison of the difference between the final values of the two most important concepts, C1 and C5 (see Section 1), shows that

- For C1, the expert-derived model underestimates the final state value by 51% while the model developed with RCGA makes only 0.1% error
- For C5, the error equals 26% and 0.2% for the expert-based and RGCA-based learning, respectively.

While the RCGA-based method provides satisfactory performance, the expert-based solution may lead to problems. The incorrect liquid gravity (C5) would likely produce wrong setups of the valves. Additionally, substantial underestimation of the liquid level (C1) would inevitably lead to exceeding its maximum level by making incorrect control decision regarding the opening of the valves.

On the other hand, example of static analysis shows that the expert-derived map correctly identifies C1 as the most important concept with C2 and C5 as the second and third most important concepts when compared to the correct order of C1, C5 and C2. Similar analysis applied to the map generated by the RCGA method shows that the top three important concepts are C2, C5 and C4 the corresponding importance measures equal 2.39, 2.20 and 2.07, respectively.

3.5 Summary

Computational methods, when compared to the expert-based methods, are a relatively new branch of research devoted to the Fuzzy Cognitive Maps. They emerged to eliminate the drawbacks and limitations of the expert-based methods. Several years of the research resulted in computational methods that are summarized and compared in Table 2. The table performs a side-by-side comparison that includes several aspects, such as learning goal, involvement of a domain expert, input data type, and learning strategy used.

Table 2 Comparative analysis of computational methods used for learning FCMs

Method	Reference	Learning Goal	Expert input	Data used[1)]	Transformation function	# concepts	Learning Algorithm
DHL	(Dickerson and Kosko 1994)	Connection matrix	No	Single	N/A	N/A	Hebbian
BDA	(Huerga 2002)	Connection matrix	No	Single	Binary	5,7,9	Modified Hebbian
NHL	(Papageorgiou et al. 2003)	Connection matrix	Yes and No[2)]	Single	Continuous	5	Modified Hebbian
AHL	(Papageorgiou et al. 2004)	Connection matrix	Yes and No[2)]	Single	Continuous	8	Modified Hebbian
DD-NHL	(Stach et al. 2008a)	Connection matrix	Yes and No[2)]	Single	Continuous	5	Modified Hebbian
GS	(Koulouriotis et al. 2001)	Initial vector	No	Multiple	Continuous	7	Genetic
PSO	(Parsopoulos et al. 2003)	Connection matrix	No	Multiple	Continuous	5	Swarm
Genetic	(Khan and Chong 2003)	Initial vector	N/A	N/A	Continuous	11	Genetic
RCGA	(Stach et al. 2005b)	Connection matrix	No	Single	Continuous	4,6,8,10	Genetic
Parallel RCGA	(Stach et al. 2007)	Connection matrix	No	Single	Continuous	5,10,20, 40,80	Parallel Genetic

1) Single – historical data consisting of one sequence of state vectors, Multiple – historical data consisting of several sequences of state vectors for different initial conditions

2) Initial human intervention is necessary but later when applying the algorithm the human input is not needed

The choice of a particular computational learning method is affected by several factors. One needs to consider the type of available data since some methods require multiple simulations (state vectors). The Hebbian-based methods are faster since evolutionary optimization requires complex and time-consuming calculations. On the other hand, methods from the latter group provide better quality of the learned models in the context of the similarity of their dynamic behavior defined as the *simulation-error*. Semi-automated methods are preferred if some structural constraints can be imposed on the map by the expert. Otherwise, if the only criterion is the quality of model's dynamic behavior, then the fully automated genetic optimization seems to be the best option.

4 Conclusions and Future Directions

4.1 Conclusions

Fuzzy Cognitive Maps have gained a well-deserved attention in the recent years. Numerous successful applications in various research and industrial domains clearly imply the effectiveness of this modeling technique. The unquestionable advantages of FCMs, such as simplicity and adaptability to a given application area, encourage researchers and practitioners to apply this method. However, it seems that further development of FCMs is somewhat constrained by deficiencies that are present in their underlying theoretical framework. One of the issues that have been recently investigated is to provide a systematic approach to efficient design of FCMs.

The two categories of approaches to develop FCMs include the *expert-based* and the *computational* methods. The advantages, disadvantages and additional characteristics of these two types of methods are summarized in Table 3.

Table 3 Comparison of expert-based and computational methods for the development of FCMs

	Expert-based	Computational
Type of modeling	Deductive	Inductive
Main objective	To create a model that is structurally understandable	To create a model that provides accurate simulations
Main application	Static analysis	Dynamic analysis
Main shortcoming	Dynamic analysis could be inaccurate	Static analysis could be inaccurate and more difficult

The expert-based methods which are fairly well established have been used for a relatively long time. Their main advantage is the easiness of the representation that is used by the expert(s) to develop the maps. Nevertheless, models developed by experts are vulnerable to subjectivity of expert(s) beliefs and could be difficult to develop for large problems that involve dozens of concepts. Moreover, although maps developed by experts provide accurate static analysis of the FCM model, they may lead to inaccurate dynamic analysis. These limitations motivated researchers towards alternative learning strategies that would provide models that accurately represent the dynamics of the modeled system.

The computational methods aim at learning FCMs from data and therefore at providing models suitable to perform accurate dynamic analysis. Two main methodologies used for computational learning of FCM include approaches based on Hebbian learning rule and methods that exploit genetic algorithms. Unfortunately, fully automated computational methods may fail to provide models that allow for accurate static analysis. A partial solution to this problem is provided by the semi-automated methods, in which the experts supervise the learning. However, the existing semi-automated learning methods often provide models that are not as good for the static analysis as the models obtained from the expert-based methods, and worse in the context of the dynamic analysis when compared with the fully automated methods.

4.2 Future Directions

The progress in research towards finding an efficient approach to develop Fuzzy Cognitive Maps that has been observed within last few years provides a solid foundation for future investigations. Recent interest in computational methods suggests that this will be the main direction of future research. Even though the first step towards automation of FCM development from data was done, there are still problems that need to be overcome.

One of the main challenges is to provide solution to the main drawback of automated methods, which generate solutions that are hard or impossible to interpret and which may lead to incorrect static analysis. The ultimate solution should be fully automated as only then FCMs could be applied to model large problems such as those encountered in systems biology. At the same time, the intermediate steps will likely involve developing efficient and accurate semi-automated algorithms. Another pressing issue is that the currently most accurate automated methods which are based on genetic optimization cannot scale to work on problems exceeding several dozens of concepts. Although some work was done towards improving scalability of these methods, further research in this direction is required. The final challenge facing the researchers working on the computational methods is to popularize their work. This could be accomplished by freely providing implementations of the developed methods to the relevant research and development communities and by creation of links between the developers of the methods and the practitioners who build the FCM models.

References

Aguilar, J.: A survey about fuzzy cognitive maps papers. Int. J. Comp. Cogn. 3(2), 27–33 (2005)

Andreou, A.S., Mateou, N.H., Zombanakis, G.A.: Soft computing for crisis management and political decision making: The use of genetically evolved fuzzy cognitive maps. Soft. Comput. 9(3), 194–210 (2005)

Axelrod, R.: Structure of decision: the cognitive maps of political elites, Princeton, NJ (1976)

Dickerson, J.A., Kosko, B.: Virtual worlds as fuzzy cognitive maps. In: VRAIS 1993, pp. 471–477 (1993)

Dickerson, J.A., Kosko, B.: Virtual worlds as fuzzy cognitive maps. Presence 3(2), 173–189 (1994)

Giordano, R., Passarella, G., Uricchio, V.F., Vurro, M.: Fuzzy cognitive maps for issue identification in a water resources conflict resolution system. Phys. Chem. Earth 30(6-7), 463–469 (2005)

Glykas, M., Xirogiannis, G.: A soft knowledge modeling approach for geographically dispersed financial organizations. Soft. Comput. 9(8), 579–593 (2005)

Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1989)

Gross, J.L., Yellen, J.: Graph theory and its applications. CRC Press, Boca Raton (1998)

Hamilton, J.D.: Time series analysis. Princeton University Press, Princeton (1994)

Hebb, D.O.: The organization of behavior. Wiley, NY (1949)

Herrera, F., Lozano, M., Verdegay, J.L.: Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. Artif. Intell. Rev. 12(4), 265–319 (1998)

Huerga, A.V.: A balanced differential learning algorithm in fuzzy cognitive maps. In: QR 2002, poster (2002)

Innocent, P.R., John, R.I.: Computer aided fuzzy medical diagnosis. Inf. Sci. 162(2), 81–104 (2004)

Khan, M., Quaddus, M.: Group decision support using fuzzy cognitive maps for causal reasoning. Group Decis. Negotiation J. 13(5), 463–480 (2004)

Khan, M.S., Chong, A.: Fuzzy cognitive map analysis with genetic algorithm. In: IICAI 2003 (2003)

Kosko, B.: Fuzzy cognitive maps. Int. J. Man Mach. Stud. 24, 65–75 (1986)

Kosko, B.: Hidden patterns in combined and adaptive knowledge networks. Int. J. Approximate Reasoning 2, 377–393 (1988)

Koulouriotis, D.E., Diakoulakis, I.E., Emiris, D.M.: Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behaviour. In: CEC 2001, pp. 364–371 (2001)

Oja, E., Ogawa, H., Wangviwattam, J.: Learning in nonlinear constrained Hebbian networks. In: Kohonen, T., Makisara, K., Simula, O., Kangas, J. (eds.) Artificial Neural Networks. Elsevier, Amsterdam (1991)

Papageorgiou, E., Stylios, C., Groumpos, P.: Fuzzy cognitive map learning based on nonlinear Hebbian rule. In: Gedeon, T.(T.) D., Fung, L.C.C. (eds.) AI 2003. LNCS (LNAI), vol. 2903, pp. 256–268. Springer, Heidelberg (2003)

Papageorgiou, E., Stylios, C.D., Groumpos, P.P.: Active Hebbian learning algorithm to train fuzzy cognitive maps. Int. J. Approximate Reasoning 37(3), 219–249 (2004)

Parsopoulos, K.E., Papageorgiou, E.I., Groumpos, P.P., Vrahatis, M.N.: A first study of fuzzy cognitive maps learning using particle swarm optimization. In: CEC 2003, pp. 1440–1447 (2003)

Saaty, T.L.: The analytic hierarchy process: planning, priority setting, resource allocation. McGraw-Hill, New York (1980)

Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Learning fuzzy cognitive maps with required precision using genetic algorithm approach. Electron. Lett. 40(24), 1519–1520 (2004a)

Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Parallel fuzzy cognitive maps as a tool for modeling software development project. In: NAFIPS 2004, pp. 28–33 (2004b)

Stach, W., Kurgan, L.A., Pedrycz, W.: A survey of fuzzy cognitive map learning methods. In: Grzegorzewski, P., Krawczak, M., Zadrozny, Z. (eds.) Issues in Soft Computing: Theory and Applications. Exit (2005a)

Stach, W., Kurgan, L., Pedrycz, W., Reformat, M.: Genetic learning of fuzzy cognitive maps. Fuzzy Sets Syst. 153(3), 371–401 (2005b)

Stach, W., Kurgan, L., Pedrycz, W.: Parallel learning of large fuzzy cognitive maps. In: IJCNN 2007, pp. 1584–1589 (2007)

Stach, W., Kurgan, L.A., Pedrycz, W.: Data-driven nonlinear Hebbian learning method for fuzzy cognitive maps. In: WCCI 2008, pp. 1975–1981 (2008a)

Stach, W., Kurgan, L.A., Pedrycz, W.: Numerical and linguistic prediction of time series with the use of fuzzy cognitive maps. IEEE Trans. Fuzzy Syst. 16(1), 61–72 (2008b)

Stylios, C.D., Georgopoulos, V.C., Malandrakic, G.A., Chouliara, S.: Fuzzy cognitive map architectures for medical decision support systems. Appl. Soft Comput. 8(3), 1243–1251 (2008)

Taber, W.R., Siegel, M.A.: Estimation on expert weights using fuzzy cognitive maps. In: ICNN 1986, vol. 2, pp. 319–325 (1987)

Tsadiras, A.K.: Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. Inf. Sci. 178(20), 3880–3894 (2008)

Tsadiras, A.K., Kouskouvelis, I., Margaritis, K.G.: Making political decisions using fuzzy cognitive maps: The FYROM crisis. In: PCI 2001, vol. 2, pp. 501–510 (2001)

Xirogiannis, G., Glykas, M.: Fuzzy cognitive maps in business analysis and performance driven change. IEEE Trans. Eng. Manage 51(3), 334–351 (2004)

Xirogiannis, G., Glykas, M.: Intelligent modeling of e-business maturity. Expert Syst. Appl. 32(2), 687–702 (2007)

Xirogiannis, G., Chytas, P., Glykas, M., Valiris, G.: Intelligent impact assessment of HRM to the shareholder value. Expert Syst. Appl. 35(4), 2017–2031 (2008)