International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:11, 2010

Big Bang – Big Crunch Learning Method for Fuzzy Cognitive Maps

Engin Yesil, Leon Urbas

***Abstract*—**Modeling of complex dynamic systems, which are very complicated to establish mathematical models, requires new and modern methodologies that will exploit the existing expert knowledge, human experience and historical data. Fuzzy cognitive maps are very suitable, simple, and powerful tools for simulation and analysis of these kinds of dynamic systems. However, human experts are subjective and can handle only relatively simple fuzzy cognitive maps; therefore, there is a need of developing new approaches for an automated generation of fuzzy cognitive maps using historical data. In this study, a new learning algorithm, which is called Big Bang-Big Crunch, is proposed for the first time in literature for an automated generation of fuzzy cognitive maps from data. Two real-world examples; namely a process control system and radiation therapy process, and one synthetic model are used to emphasize the effectiveness and usefulness of the proposed methodology.

***Keywords*—**Big Bang-Big Crunch optimization, Dynamic Systems, Fuzzy Cognitive Maps, Learning.

I. Introduction

COGNITIVE maps were introduced by Axelrod (1976) [1] to represent crisp cause-effect relationships which are perceived to exist among the elements of a given environment. Fuzzy cognitive maps (FCM) are fuzzy signed directed graphs with feedbacks, and they model the world as a collection of concepts and causal relations between concepts [2]. Their main advantages are flexibility and adaptability to a given domain [3]. FCMs were applied to a significant number of domains such as analysis of electrical circuits, medicine, supervisory systems, organization and strategy planning, analysis of business performance indicators, software project management, information retrievals, modeling of plant control, system dynamics and complex systems and modeling virtual world [5].

In general, FCMs can be produced by expert manually or generated by other sources of information computationally. They are named manual FCMs and automated FCMs.

In most cases, the manual procedures for developing FCM have occurred, when at least there is one expert who has expertise in the area under study. In some situations, a FCM could not be constructed manually such as [5]:

a) There is no expert to define a FCM.
b) The experts' knowledge differs from each other and they even draw different FCM.
c) There are large amounts of concepts with connections between them, which could not be drawn without mistakes.

The above situation shows that in many cases, to develop a FCM manually becomes very difficult and experts' intervention could not resolve the problem. Therefore, a systematic way should be found in order to bridge this gap. For these reasons, the development of computational methods for learning FCM is necessary. Some methods for learning FCM model structure have been recently proposed. In general, these methods are categorized in two main groups: Hebbian learning and global optimization methods.

A simple differential Hebbian learning law (DHL) for FCM is introduced in [6], which has been extended by [7]. Further extensions, called nonlinear Hebbian learning (NHL) and Active Hebbian algorithm (AHL) are introduced in [8] and [9], respectively.

Another category in learning connection matrix of FCM is the application of global search strategies. In [10], the Genetic Strategy (GS) to learn FCM's structure is applied. In [11], a real-coded genetic algorithm (RCGA) is applied to develop a FCM model from a set of historical data. The aim of these studies was to compute the connection matrix. In [12], a goal-oriented analysis of FCM is performed and the aim of the proposed learning method was learning the initial state vector of FCM. Also, particle swarm optimization (PSO) [13], simulated annealing (SA) [14], chaotic simulated annealing [5] methods are proposed for learning of FCMs. Petalas et al. [16] propose memetic particle swarm optimization to improve the standard PSO.

One of the main disadvantages of the proposed genetic algorithm based learning methods is its mediocre scalability as the number of parameters to be established grows quadratically with the size of the FCM model. This is because the genetic optimization applied to this modeling is time consuming especially when dealing with large a number of variables. Therefore an approach to speed-up the learning process based on a "divide and conquer" strategy is proposed in [17].

The most important problem of the genetic algorithm based learning methods is the selection of the genetic algorithm operator parameters. The convergence and the time of

E. Yesil and L. Urbas are with Institute of Automation, Dresden University of Technology, D-01062 Dresden, Germany. (Corresponding author: E. Yesil phone: +90-2122853599; fax: +90-2122852920; e-mail: yesileng@itu.edu.tr)

E. Yesil is also with Faculty of Electrical and Electronics Engineering, Control Engineering Department, Istanbul Technical University, Maslak, TR-34469, Istanbul, Turkey. He is currently a visiting researcher with DAAD (Deutscher Akademischer Austausch-Dienst) Scholarship Award at Dresden University of Technology, Germany.

learning depend on this choice. Therefore, there is a need of a new simple global optimization based learning method.

In this study, Big Bang - Big Crunch (BB-BC) optimization [18] method is recommend for learning of FCMs as an alternative to the existing learning methods. This global optimization method is preferred since it has a low computational cost, a high convergence speed and just a few parameters which should be set by the designer for learning of FCMs. BB–BC optimization method is a natural evolutionary algorithm similar to the genetic algorithms, the ant colony optimization, the particle swarm optimizer and the harmony search [19]. The algorithm generates random points in the Big Bang (BB) phase and shrinks those points to a single representative point via a centre of mass or minimal cost approach in the Big Crunch (BC) phase. It is reported in [18] to be capable of quick convergence even in long, narrow parabolic shaped flat valleys or in the existence of several local minima. It has been applied to many areas including fuzzy model inversion [20], non-linear controller design [21], target motion analysis problem [22], genetic programming classifier design [23], design of space trusses [24], and size reduction of space trusses [25].

The remainder of the paper is organized as follows: Section 2 briefly describes the formulation of FCMs. Section 3 presents proposed Big Bang-Big Crunch FCM learning method based on historical data. Section 4 introduces the simulation experiment setup, and then Section 5 presents the obtained results. Finally, Section 6 provides the conclusion.

II. Fuzzy Cognitive Maps

A fuzzy cognitive map F is a 4-tuple (N, W, C, f) [12] where

1. $N=\{N_1, N_2, \ldots, N_n\}$ is the set of n concepts forming the nodes of a graph.

2. $W: (N_i, N_j) \rightarrow w_{ij}$ is a function of $N \times N$ to K associating w_{ij} to a pair of concepts (N_i, N_j), with w_{ij} denoting a weight of directed edge from N_i to N_j, if $i \neq j$ and w_{ij} equal to zero if $i=j$. Thus $W(N \times N) = (w_{ij}) \in K^{n \times n}$ is a connection matrix.

3. $C: N_i \rightarrow C_i$ is a function that at each concept N_i associates the sequence of its activation degrees such as for $t \in N$, $C_i(t) \in L$ given its activation degree at the moment t. $C(0) \in L^n$ indicates the initial vector and specifies initial values of all concept nodes and $C(t) \in L^n$ is a state vector at certain iteration t.

4. $f: R \rightarrow L$ is a transformation function, which includes recurring relationship on $t \geq 0$ between $C(t + 1)$ and $C(t)$.

The calculation rule that was initially introduced to calculate the value of each concept is based only on the influence of the interconnected concepts [2], [26], [27]

$$C_j(t+1) = f\left(\sum_{\substack{i=1 \\ i \neq j}}^{n} C_i(t) w_{ij}\right) \tag{1}$$

where n is the number of concepts, $C_j(t+1)$ is the value of concept C_j at time step t+1, $C_i(t)$ is the value of concept C_i at time step t, and w_{ij} is the weight of the causal interconnection from concept i^{th} toward concept j^{th}.

The transformation function is used to confine (clip) the weighted sum to a certain range, which is usually set to [0, 1]. The normalization hinders quantitative analysis, but allows for comparisons between nodes, which can be defined as active (value of 1), inactive (value of 0), or active to a certain degree (value between 0 and 1). Four most commonly used transformation functions are shown below:

1. bivalent

$$f(x) = \begin{cases} 0, & x \leq 0, \\ 1, & x \geq 0. \end{cases} \tag{2}$$

2. trivalent

$$f(x) = \begin{cases} -1, & x \leq -0.5, \\ 0, & -0.5 < x < 0.5, \\ 1, & x \geq 0.5. \end{cases} \tag{3}$$

3. sigmoid (logistic)

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

4. hyperbolic tangent

$$f(x) = \tanh(\lambda x) = \frac{e^{\lambda x} - e^{-\lambda x}}{e^{\lambda x} + e^{-\lambda x}} \tag{5}$$

where λ is a parameter used to determine proper shape of the function.

III. Proposed Learning Method: Big Bang – Big Crunch

A. Big Bang-Big Crunch Optimization Algorithm

The Big Bang-Big Crunch (BB-BC) optimization method developed by Erol and Eksin [18] consists of two main steps: The first step is the Big Bang phase where candidate solutions are randomly distributed over the search space and the next step is the Big Crunch phase where a contraction procedure calculates a center of mass for the population.

The initial Big Bang population is randomly generated over the entire search space similar to any other evolutionary search algorithm. All subsequent Big Bang phases are randomly distributed around the center of mass or the best fit individual in a similar fashion. In [18], the working principle of the Big Bang phase is explained as energy dissipation or the transformation from an ordered state (a convergent solution) to a disordered or chaotic state (new set of candidate solutions).

After the Big Bang phase, a contraction procedure is

applied during the Big Crunch. In this phase, the contraction operator takes the current positions of each candidate solution in the population and its associated cost function value and computes a centre of mass according to (6),

$$x_{COM} = \frac{\sum_{i=1}^{N} \frac{1}{f^i} x_i}{\sum_{i=1}^{N} \frac{1}{f^i}} \quad (6)$$

where x_{COM} is the position vector of the center of mass, x_i is the position vector of the i_{th} candidate, f^i is the cost function value of the i^{th} candidate, and N is the population size. The new generation for the next iteration Big Bang phase is normally distributed around x_{COM}. The new candidates around the centre of mass are calculated by adding or subtracting a normal random number whose value decreases as the iterations elapse. This can be formalized as

$$x^{new} = x_{COM} + \frac{r\alpha(x_{max} - x_{min})}{k} \quad (7)$$

where r is a normal random number, α is a parameter limiting the size of the search space, x_{max} and x_{min} are the upper and lower limits, and k is the iteration step. Since normally distributed numbers can be exceeding ±1, it is necessary to limit the population to the prescribed search space boundaries. This narrowing down restricts the candidate solutions into the search space boundaries. The procedure of the BB-BC optimization is given in the Table 1.

TABLE I
BIG BANG BIG CRUNCH ALGORITHM

Step 1 (Big Bang Phase)
An initial generation of N candidates is generated randomly in the search space.
Step 2
The cost function values of all the candidate solutions are computed.
Step 3 (Big Crunch Phase)
The center of mass is calculated. Either the best fit individual or the center of mass is chosen as the point of Big Bang Phase.
Step 4
New candidates are calculated around the new point calculated in Step 3 by adding or subtracting a random number whose value decreases as the iterations elapse.
Step 5
Return to Step 2 until stopping criteria has been met.

Instead of the centre of mass, other points like the best fit individual can also be chosen as the starting point in the Big Bang phase. In the experiments reported in this paper we apply an elitist strategy introduced by Camp [24]. The positions of new candidate solutions at the beginning of each Big Bang are normally distributed around a new point located between the center of mass and the best solution,

$$x^{new} = \beta.x_{COM} + (1-\beta).x_{BEST} + \frac{r\alpha(x_{max} - x_{min})}{k} \quad (8)$$

where β is the parameter controlling the influence of the global best solution x_{BEST} on the location of new candidate solutions. This modification of generating the new solution can be viewed as to be an elitist strategy, where the best solution influences the direction of the search.

B. Proposed Learning Methodology

The proposed BB-BC learning method develops a candidate FCM from input data as given in Fig. 1. The input data are given as time series and that consist of a sequence of state vectors which describe a given system at consecutive iteration. The number of these successive iterations of the given historical date is called as the data length. Because of the nature of FCMs, the data points are normalized to the unit interval [0, 1] and they correspond to the degree of presence of a given concept at a particular iteration. Given a system consisting of N concepts, the FCM model can be described fully by its connection matrix. The aim of the learning method is to establish the connection matrix that consists of N(N-1) variables assuming values in [-1, 1]. The proposed method uses the BB-BC algorithm and given input data to determine these values. In other words, the learning goal is to generate the same state vector sequence using the candidate FCM for the same initial vector as it is defined in the input data. Thereby, the candidate FCM generalizes the relations between the concepts, and it allows performing simulations from different initial state vectors in order to represent conclusions about the modeled system.

One of the most important considerations for BB-BC, similar to the other global optimization methods, is the design of a cost function, which is appropriate for a given problem. In literature many different cost functions are proposed. In this study the following cost function is found to be appropriate [11]:

$$J_1 = \frac{1}{(K-1).N} \sum_{t=1}^{K} \sum_{n=1}^{N} (C_n(t) - \hat{C}_n(t))^2 \quad (9)$$

where $C_n(t)$ is the given system response, $\hat{C}_n(t)$ is the candidate FCM response of the n^{th} concept for the initial state vector, K is the data length, and N is the number of concepts. In order to normalize and visualize the cost function (8) the following fitness function which has the value [0 1] is proposed:

$$f = \frac{1}{\theta.J_1 + 1} \quad (10)$$

where parameter θ is a positive scaling constant.

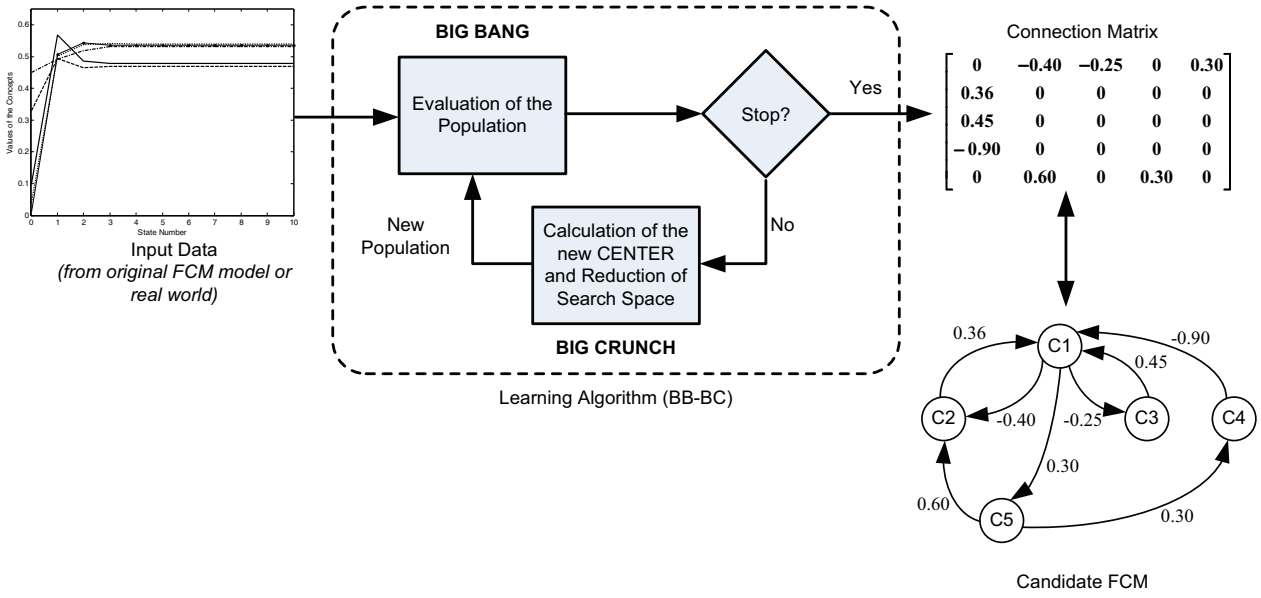


Fig. 1 Illustration of proposed BB-BC learning methodology

Another important condition for the BB-BC learning of FCM is the stopping criteria. This can be defined by following ways:

i. The algorithm terminates if time exceeds the specific time.
ii. The algorithm terminates if the number of BB-BC generations exceeds the specific number.
iii. The algorithm terminates if the best candidate FCM was not improved after a period.
iv. The algorithm terminates if the predefined fitness function value is reached with the best candidate FCM.

In this study, the second stopping criterion (number of generation) is used for experiments.

After the best candidate FCM is obtained using the proposed BB-BC learning method, its generalization capabilities are tested. For this purpose, the original FCM and the candidate FCM are simulated for R randomly chosen initial state vectors and the data are collected [11]. Then, the generalization capability is calculated using the following new criterion:

$$J_2 = \frac{1}{R.(K-1).N} \sum_{r=1}^{R} \sum_{t=1}^{K} \sum_{n=1}^{N} (C_n^r(t) - \hat{C}_n^r(t))^2 \qquad (11)$$

where $C_n^r(t)$ is the value of n^{th} node at iteration t for the data generated by original FCM model started from the r^{th} initial state vector, similarly, $\hat{C}_n^r(t)$ is the value of n^{th} node at iteration t for the data generated by candidate FCM model started from the r^{th} initial state vector, K is the data length, N is the number of nodes, and R is the number of randomly picked different initial state vectors.

IV. Experimental Setup

In this study, many systems are studied, but here three of these systems with different characteristics are discussed. The properties of the presented systems are summarized in Table 2. The first two systems have similar size and density that is defined as the ratio of non-zero weights to the total number of weights. The second system has an input node, which will be kept unchanged during the simulations. Furthermore all of the systems show different values for the transformation functions parameters (λ).

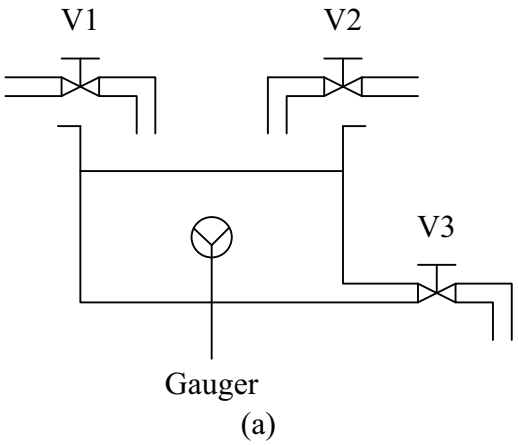
TABLE II
Parameters of the Studied Systems

System No	Number of nodes	Input node	λ in (3)	Density (%)
1	5	No	1	32
2	6	Yes	2	33
3	10	no	5	71

A. First System: A Process Control System

A process control problem [28] derived from chemical industry is used as our first case study. This process is used previously by several researchers as a benchmark system for different learning algorithms [29], [30], [16]. The system consists of a tank and three valves that control the amount of liquid in the tank. Two different liquids are poured and mixed into the tank through valve V1 and valve V2. During the mixing, a chemical reaction takes place and a new liquid is produced. Valve V3 empties the tank as soon as the amount of the produced liquid reaches a specific level. Meanwhile, the specific gravity (defined as the ratio of measured density to the density of a reference liquid) of the produced liquid is measured by a sensor placed inside the tank. When the

specific gravity value lies in a predefined range, the desired liquid has been produced. The process is illustrated in Fig.2a.



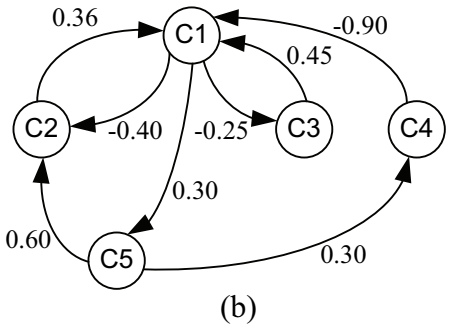


Fig. 2 Illustration of industrial tank-valve system

The constructed FCM for this problem is depicted in Fig.2b with the following five concepts:

C1: The amount of the liquid in the tank,
C2: The state of valve V1,
C3: The state of valve V2,
C4: The state of valve V3,
C5: The specific gravity of the produced liquid in the tank.

The connection matrix of the FCM built by the experts [28] is as follows:

$$W_1 = \begin{bmatrix} 0 & -0.40 & -0.25 & 0 & 0.30 \\ 0.36 & 0 & 0 & 0 & 0 \\ 0.45 & 0 & 0 & 0 & 0 \\ -0.90 & 0 & 0 & 0 & 0 \\ 0 & 0.60 & 0 & 0.30 & 0 \end{bmatrix} \quad (12)$$

The FCM is calculated according to (1). The transformation function is sigmoidal (logistical) with $\lambda=1$. The initial state vector C(0) is chosen for this study is as follows:

C(0)=[0.10 0.45 0.33. 0.04 0.01]

The values of the FCM concepts for 10 iterations for the given initial state vector are presented in Fig. 3. This data is used as the training set.

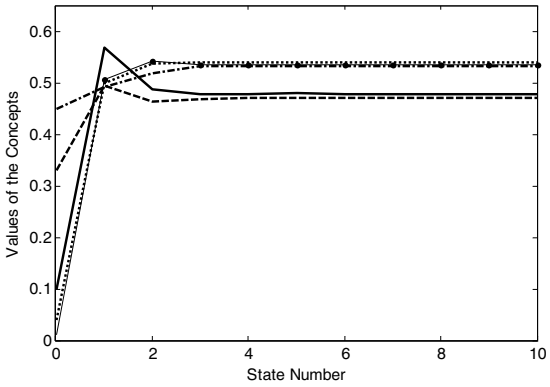


Fig. 3 Values of the concepts for the first system

B. Second System: Radiation Therapy Process

The second system concerns radiotherapy, which is used for cancer treatment. Radiation therapy is a complex process involving a large number of treatment variables. The objective of radiotherapy is to deliver the highest amount of radiation dose to the smallest possible volume that encloses the tumor, while minimizing the exposure of healthy tissues and critical organs to radiation. Treatment planning, which is also a complex process and doctor–computer interaction, is needed before the final treatment execution.

The radiation therapy process can be modeled and analyzed through a supervisor-FCM, constructed by experts [31]. The FCM consists of the following concepts:

C1: Tumor localization,
C2: Dose prescribed from the treatment planning,
C3: Machine factors,
C4: Human factors,
C5: Patient positioning and immobilization,
C6: Final dose received by the target volume.

The FCM constructed by experts for this problem is depicted in Fig.4.

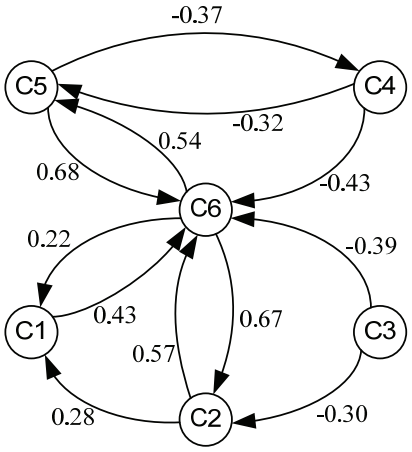


Fig. 4 Illustration of FCM for radiation therapy process

The connection matrix of the Radiation Therapy Process FCM built by the experts is as follows:

$$W_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.43 \\ 0.28 & 0 & 0 & 0 & 0 & 0.57 \\ 0 & -0.3 & 0 & 0 & 0 & -0.39 \\ 0 & 0 & 0 & 0 & -0.32 & -0.43 \\ 0 & 0 & 0 & -0.37 & 0 & 0.68 \\ 0.22 & 0.67 & 0 & 0 & 0.54 & 0 \end{bmatrix} \quad (13)$$

For this example, again, (1) is used as the calculation rule. The transformation function, this time, is sigmoidal (logistical) with λ=2 as stated in [31]. The initial state vector C(0) is chosen for this study is as follows:

C(0)= [0.30 0.65 0.50 0.10 0.80 0.10]

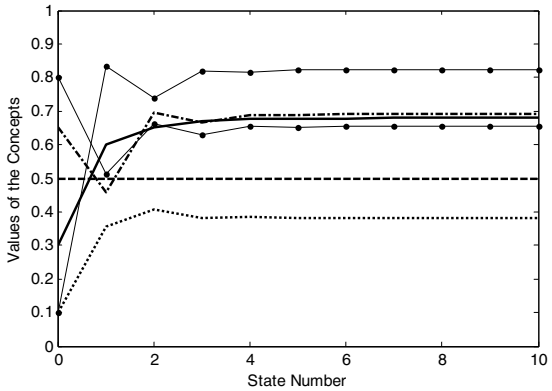


Fig. 5 Historical data for the second system

The values of the FCM concepts for 10 iterations for the given initial state vector are presented in Fig. 5. This data is used as the historical data for learning. Since concept C3 is an input node it is not affected by the other concepts and the initial state value will be constant for all iterations. This can also be seen from the connection matrix in (13), where the third column is a zero vector.

C. Third System: Randomly Generated FCM

In this study, two FCMs that are from real-world and constructed by experts are used. It becomes quite apparent that these FCMs are usually relatively small, and typically consist of 5–10 nodes [12]. Small size is a result of the manual development of such maps where we usually rely on expert knowledge. We note that mutual relationships among large number of concepts are hard to comprehend, analyze, and describe, which results in substantial difficulties in the construction of the corresponding maps.

Therefore, as a third system a synthetic FCM model is constructed randomly. This time, the randomly generated FCM has 10 concepts; consequently the connection matrix has 100 parameters. The aim of this example is to show the effectiveness of BB-BC learning method even the size of the FCM high.

The connection matrix of the randomly generated FCM is presented as (14).

For the randomly generated FCM example, (1) is used as the calculation rule. The transformation function, this time, is sigmoidal (logistical) with λ=5. The initial state vector C(0) is chosen for this study is as follows:

C(0)= [0.64, 0.28,0.89, 0.04, 0.33, 0.65, 0.58,0.81, 0.35, 0.06]

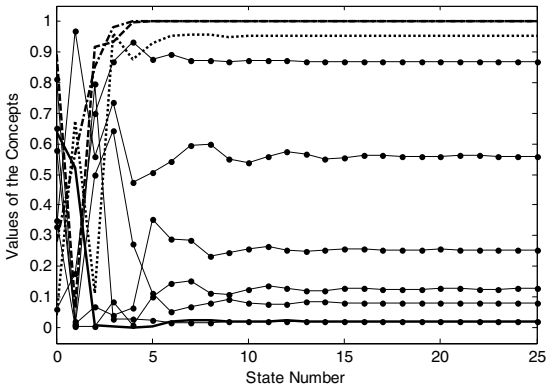


Fig. 6 Historical data for the third system.

$$W_3 = \begin{bmatrix} 0 & -0.7 & 0.3 & -0.82 & 0.73 & 0.37 & 0.15 & -0.92 & -0.64 & -0.67 \\ -0.86 & 0 & 0.53 & -0.06 & 0.15 & 0 & 0.46 & 0 & -0.25 & 0.88 \\ 0.95 & 0 & 0 & 0.63 & 0 & 0 & -0.58 & -0.56 & 0 & -0.61 \\ -0.58 & 0.84 & 0.82 & 0 & -0.77 & -0.54 & 0.12 & -0.14 & 0 & 0 \\ -0.12 & 0 & -0.71 & 0 & 0 & -0.49 & 0.03 & 0.63 & -0.78 & 0.06 \\ 0.2 & 0.6 & -0.21 & 0.04 & 0 & 0 & -0.47 & -0.65 & 0 & 0.26 \\ -0.49 & -0.21 & 0 & 0.32 & -0.31 & -0.81 & 0 & 0.43 & 0.36 & 0 \\ -0.85 & 0.48 & -0.48 & -0.27 & 0 & -0.42 & -0.12 & 0 & -0.84 & 0.28 \\ 0.98 & -0.68 & -0.56 & 0.38 & 1 & -0.5 & -0.2 & 0.23 & 0 & 0 \\ -0.39 & 0.81 & 0.9 & 0 & 0.57 & 0.78 & -0.74 & 0 & 0.43 & 0 \end{bmatrix} \quad (14)$$

The values of the FCM concepts for 25 successive iterations for the given initial state vector are presented in Fig. 6.

V. Results And Discussions

In order to show the effectiveness of the Big Bang-Big Crunch (BB-BC) learning method three FCMs with different sizes are studied. The study is divided into two phases: Learning phase, and generalization capability testing phase. All the parameters used in these two phases are summarized in Table 3.

TABLE III
Parameters Used in Learning and Generalization Capability Test Phases

System No	α in (8)	β in (8)	Population number (N)	Number of iterations	θ in (10)	R in (11)
1	10	0.25	20	5000	10^4	100
2	10	0.25	20	5000	10^4	100
3	10	0.25	20	5000	10^2	100

Since the proposed BB-BC learning algorithm is a stochastic method, many simulations for each system are performed. Firstly, the BB-BC algorithm is run ten times for each of the three systems during the learning phase and the cost function for each trial is calculated using (9). Then, out of ten candidates, the best candidate FCM that has the lowest cost function J_1 is chosen. The results obtained for the learning phase are presented in Table 4. Secondly, to calculate the generalization capability of the best candidate FCM, hundred random initial state vectors are generated for each example. Then, FCMs are simulated for these initial state vectors, and the generalization capability is calculated using (11). The results obtained for this phase are presented in Table 5. As a result, for this study, 330 FCM simulations are done.

Using our previous experience, the parameters of BB-BC algorithm, α and β in (8), are picked as 10 and 0.25, respectively. Also, in order to show the convergence of BB-BC learning method in a normalized way, the fitness function given in (10) with appropriate θ parameters are presented in Fig. 7. These plots show the change of best and average fitness functions for each system.

TABLE IV
Cost Function Values for Learning Phase

System No	Min J_1	Max J_1	Mean and Standard deviation
1	1.47×10^{-7}	1.57×10^{-6}	$4.97\times10^{-7} \pm 4.19\times10^{-7}$
2	3.91×10^{-6}	2.20×10^{-5}	$1.12\times10^{-5} \pm 6.39\times10^{-6}$
3	2.22×10^{-4}	3.88×10^{-3}	$2.03\times10^{-3} \pm 1.37\times10^{-3}$

As it is easily seen from Table 4 and Table 5, the two cost functions of both the first and second system are very low. The third system has nearly twice as much concepts as well as a high density value. Therefore the number of parameters for learning is much higher. As a result, the cost function values of the third system are one order of magnitude higher than the values of the first two systems. In order to show the capacity and effectiveness of the proposed BB-BC learning method for large FCMs, only the results of the third system are presented. In Fig. 8, the results obtained at the end of the learning phase are plotted for the comparison of the candidate FCM and original FCM. Then, for randomly generated hundred initial state vectors simulations are performed and the best and the worst results are illustrated in Fig. 9 and Fig. 10, respectively. The obtained results show that even for a large FCM, BB-BC learning algorithm effectively finds satisfactory results for the weight matrix with an appropriate precision.

In study [11], where real-coded genetic algorithm (RCGA) is proposed as learning method, systems that have 4, 6, 8, and 10 concepts and different densities are studied. These systems are similar to the ones presented in this paper. Even though, 150,000 iterations with 100 populations are needed for RCGA based learning, in this paper successful results are obtained just with 5,000 iterations and only 20 populations when the proposed BB-BC learning method is used.

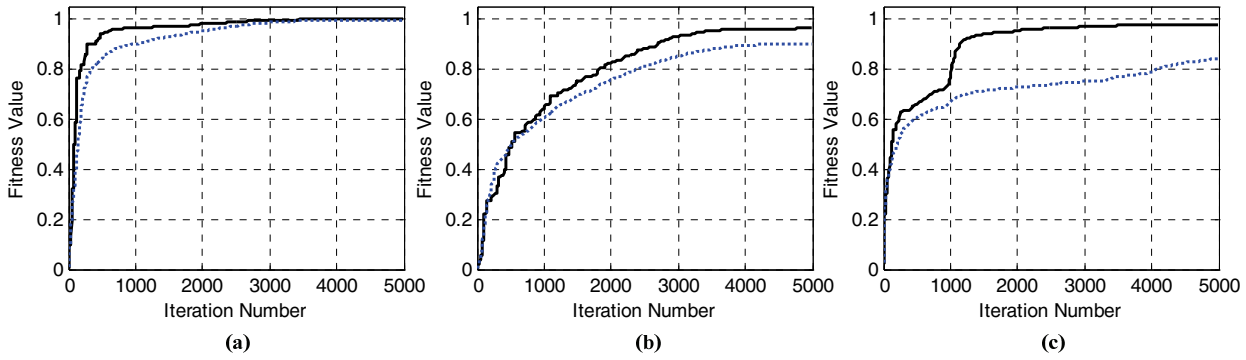


Fig. 7 Fitness functions (solid: best fitness, dashed: average fitness): (a) First system; (b) Second system; (b) Third system.

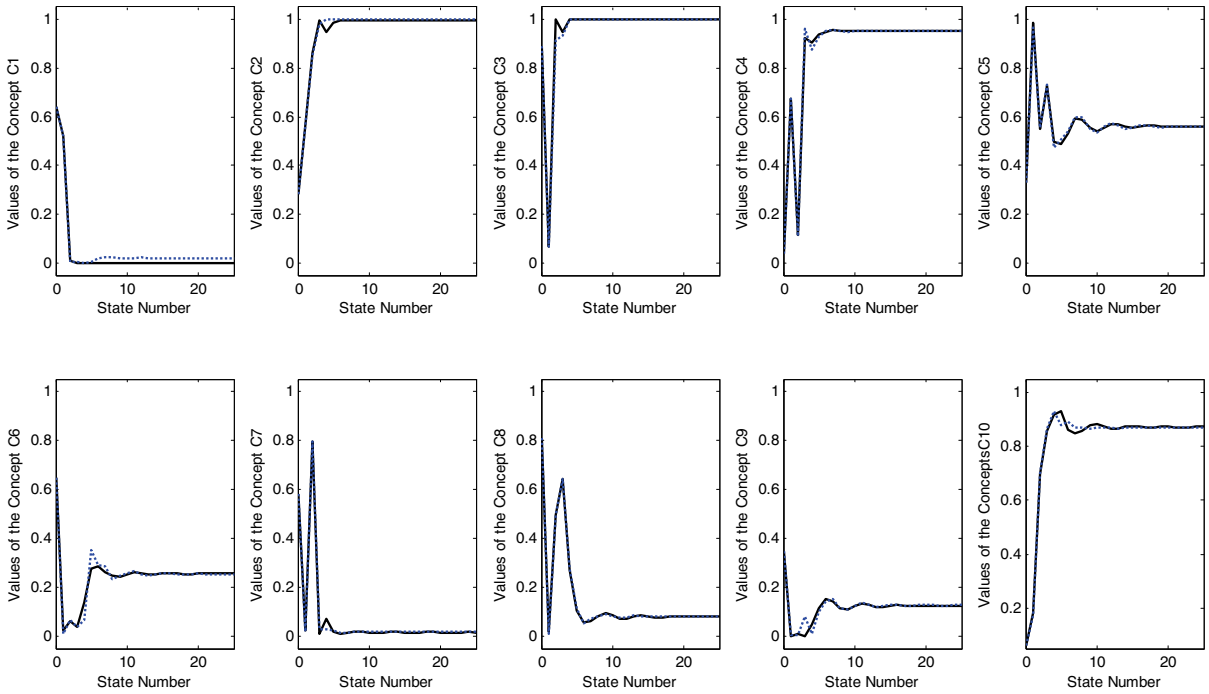


Fig. 8 Comparison of the input data and the data obtained from candidate FCM (solid: original FCM, dashed: best candidate FCM).

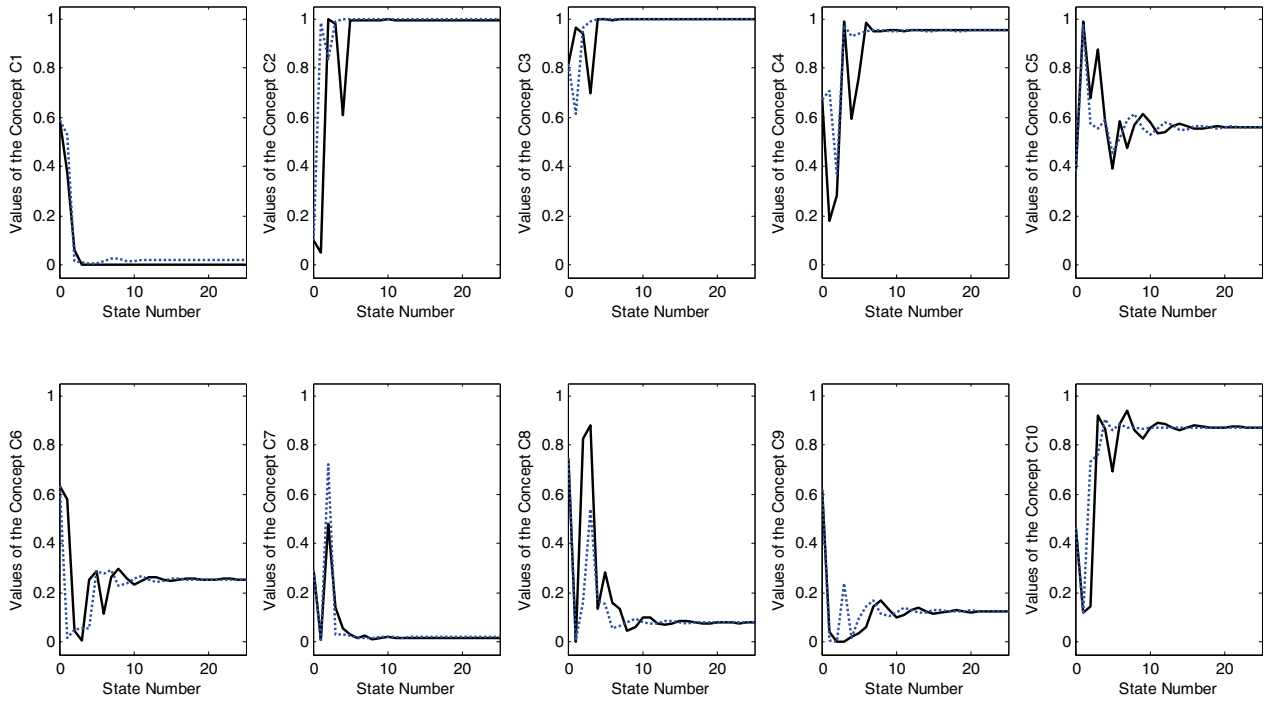


Fig. 9 Comparison of candidate and original FCM with a random initial state vector for the best simulation (solid: original FCM, dashed: candidate FCM)

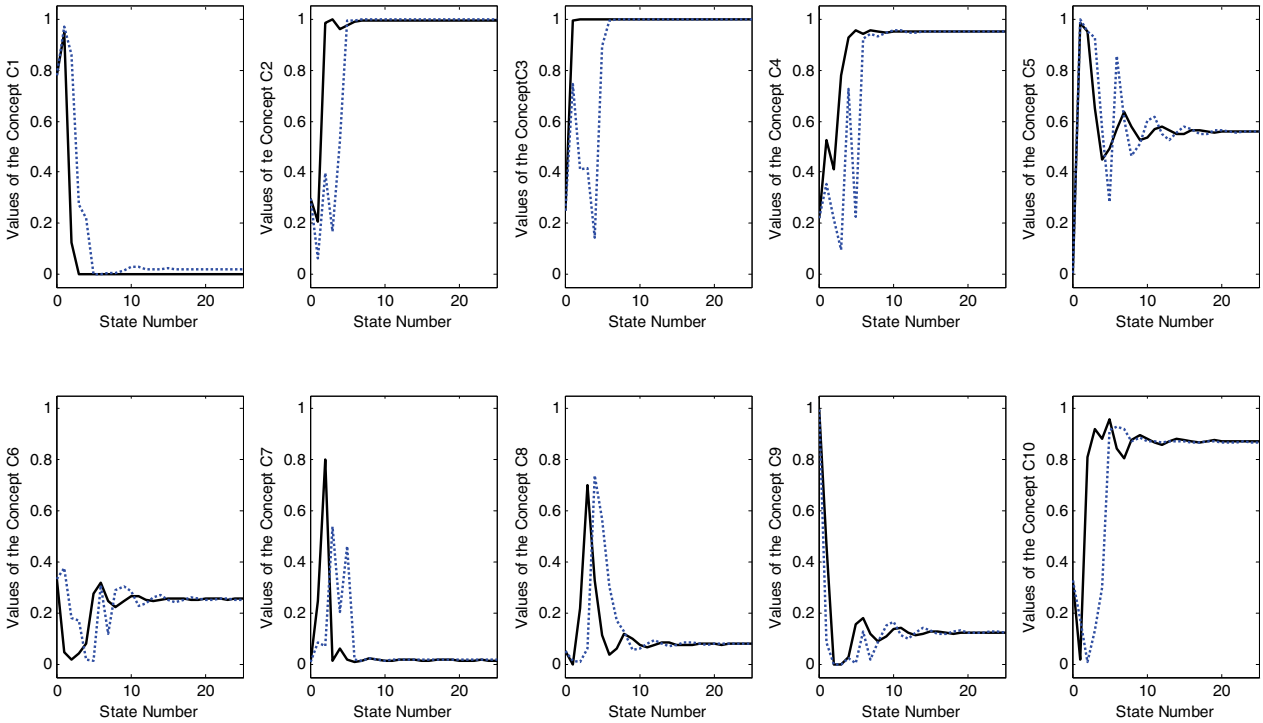


Fig. 10 Comparison of candidate and original FCM with a random initial state vector for the worst simulation (solid: original FCM, dashed: candidate FCM)

TABLE V
COST FUNCTION VALUES FOR GENERALIZATION CAPABILITY TESTING PHASE

System No	Min J_2	Max J_2	Mean and Standard deviation
1	3.58×10^{-6}	1.04×10^{-3}	$2.40\times10^{-4} \pm 2.25\times10^{-4}$
2	2.96×10^{-5}	2.86×10^{-3}	$7.91\times10^{-4} \pm 7.61\times10^{-4}$
3	6.67×10^{-3}	4.91×10^{-2}	$1.95\times10^{-2} \pm 6.81\times10^{-3}$

VI. Conclusion

In this study, a comprehensive learning method for the development of fuzzy cognitive maps is developed. For the first time in literature, a new but effective global optimization algorithm called Big Bang-Big Crunch is used for constructing FCMs on a basis of experimental historical data as an alternative to the existing method. This global optimization method is preferred since it is reported in previous studies that BB-BC has a low computational cost and a high convergence speed.

The proposed methodology is able to generate a FCM model from input data consisting of a single sequence of concept state vector values. Three systems, which have 5, 6 and 10 concepts, are studied to test the proposed methodology. The results show that the proposed learning method is very effective, and generates FCM models that can almost present the input data. Then, in order to test the generalization capability of the obtained FCM, several simulations are performed for randomly generated initial state vectors. It is obtained that the BB-BC based learning method deteriorates with the increasing size of the maps. For the first two systems with 5 and 6 concepts, the proposed method achieves excellent quality, while for the maps about 10 concepts the quality is still satisfactory.

Acknowledgment

The authors gratefully acknowledge DAAD (Deutscher Akademischer Austausch-Dienst, German Academic Exchange Service) for awarding the first author with a DAAD-Scholarship at the Institute of Automation, Dresden University of Technology, in Germany.

References

[1] R. Axelrod, *Structure of Decision: the Cognitive Maps of Political Elites*. Princeton University Press, Princeton, New Jersey, 1976.
[2] B. Kosko, "Fuzzy cognitive maps," *International Journal of Man-Machine Studies*, vol. 24, pp. 65–75, 1986.
[3] J. Aguilar, "A survey about fuzzy cognitive maps papers," *International Journal of Computational Cognition*, vol. 3(2), pp. 27–33, 2005.
[4] R. J. G. B. Campello and W. C. Amaral, "Towards true linguistic modeling through optimal numerical solutions," *Int. J. Syst. Sci.*, vol. 34 (2), pp. 139–157, 2003.
[5] S. Alizadeh and M. Ghazanfari, "Learning FCM by chaotic simulated annealing," *Chaos, Solutions & Fractals*, vol. 41(3), pp. 1182-1190, 2009.

[6] J.A. Dickerson and B. Kosko, "Fuzzy virtual worlds", *Artif. Intell. Expert*, vol. 7, pp. 25–31, 1994.
[7] A. Vazquez, *A balanced differential learning algorithm in fuzzy cognitive maps*, Technical report, Departament de Llenguatges I Sistemes Informatics, Universitat Politecnica de Catalunya (UPC), 2002.
[8] E. I. Papageorgiou, C. D. Stylios and P.P. Groumpos, "Fuzzy cognitive map learning based on nonlinear Hebbian rule", in *Australian conference on artificial intelligence*, 2003, pp. 256–68.
[9] E. I. Papageorgiou, C. D. Stylios and P.P. Groumpos, "Active Hebbian learning algorithm to train fuzzy cognitive maps", *Int. J. Approx. Reason.*, vol. 37(3), pp. 219–49, 2004.
[10] D. E. Koulouriotis, I. E. Diakoulakis and D. M. Emiris, "Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior," in *IEEE congress on evolutionary computation (CEC2001)*, 2001, pp. 364–71.
[11] W. Stach, L. Kurgan, W. Pedrycz and R. Marek, "Genetic learning of fuzzy cognitive maps", *Fuzzy Sets Syst*., vol. 153, pp. 371–401, 2005.
[12] M. Khan and A. Chong, "Fuzzy cognitive map analysis with genetic algorithm", in *Proceedings of the 1st Indian international conference on artificial intelligence (IICAI-03)*, 2003.
[13] Parsopoulos K.E, Papageorgiou E.I, Groumpos P.P. and Vrahatis M.N., "A first study of fuzzy cognitive maps learning using particle swarm optimization", in *Proceedings of the IEEE 2003 congress on evolutionary computation*, 2003. p. 1440–1447.
[14] M. Ghazanfari, S. Alizadeh, M. Fathian and D. E. Koulouriotis, "Comparing simulated annealing and genetic algorithm in learning FCM", *Appl. Math Comput.*, vol. 192(1), pp.56–68, 2007.
[15] E. Papageorgiou, C. Stylios and P. Groumpos, "Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links", *Int. J. Human Comput. Stud.*, vol. 64, pp. 727–743, 2006.
[16] Y. G. Petalas, , K. E. Parsopoulos and M. N. Vrahatis, "Improving Fuzzy Cognitive Maps Learning Through Memetic Particle Swarm Optimization", *Soft Computing*, vol. 13(1), pp. 77-94, 2009.
[17] W. Stach, L. Kurgana, and W. Pedrycz, "A divide and conquer method for learning large Fuzzy Cognitive Maps", *Fuzzy Sets and Systems*, to be published.
[18] O. K. Erol and I. Eksin, "A new optimization method: Big Bang-Big Crunch," *Advances in Engineering Software*, vol. 37, pp. 106-111, 2006.
[19] A. Kaveh and S. Talatahari, "Optimal design of Schwedler and ribbed domes via hybrid Big Bang-Big Crunch algorithm", *Journal of Construction Steel Research*, vol. 66(3), pp. 412-419, 2010.
[20] T. Kumbasar, I. Eksin, M. Guzelkaya and E. Yesil, "Big bang big crunch optimization method based fuzzy model inversion", *MICAI 2008, LNCS 5317*, pp. 732-740, 2008.
[21] M. Dogan and Y. Istefanopulos, "Optimal nonlinear controller design for flexible robot manipulators with adaptive internal model", *IET Control Theory and Applications*, vol. 1(3), pp.770-778, 2007.
[22] H. M. Genc and A. K. Hocaoglu, "Bearing-only target tracking based on Big Bang - Big Crunch algorithm", in *Proc. - The 3rd Int. Multi-Conf. Computing in the Global Information Technology, ICCGI 2008 in Conjunction with ComP2P*, pp. 229-233, 2008.
[23] A. Akyol and Y. Yaslan, O. K. Erol, "A Genetic Programming Classifier Design Approach for Cell Images", *ECSQARU 2007, LNCS 4724*, pp. 878-888, 2007.
[24] C. V. Camp, "Design of space trusses using big bang-big crunch optimization", *Journal of Structural Engineering*, vol. 133(7), pp. 999-1008, 2007.
[25] A. Kaveh and S. Talatahari, "Size optimization of space trusses using Big Bang–Big Crunch algorithm", *Computers and Structures*, vol. 87, pp. 1129-1140, 2009.
[26] B. Kosko, *Neural Networks and Fuzzy Systems*, Englewood Cliffs, NJ, Prentice-Hall, 1992.
[27] C. D. Stylios and P. P. Groumpos, "Modeling Complex Systems Using Fuzzy Cognitive Maps", *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 34(1), pp. 155-162, 2004.
[28] C. D. Stylios, P. P. Groumpos, "The challenge of modelling supervisory systems using fuzzy cognitive maps", J. Intell. Manufact., vol. 9, pp.339–345, 1998.
[29] P. P. Groumpos and C.D. Stylios, "Modeling Supervisory Control Systems using Fuzzy Cognitive Maps", Chaos, Solitons and Fractals, (2000), Vol.11, No 1-3, pp. 329-336
[30] E. I. Papageorgiou, C.D. Stylios, P.P. Groumpos, "Fuzzy Cognitive Map Learning based on Nonlinear Hebbian Rule", 16th Australian Joint Conference on Artificial Intelligence - AI'03, Perth-Western Australia, December 3-5, 2003; T.D. Gedeon and L.C.C. Fung (Eds.): AI 2003, LNAI 2903, pp. 254-266, 2003, Springer-Verlag Berlin Heidelberg 2003.
[31] K. E. Parsopoulos, E. I. Papageorgiou, P. P Groumpos, M. N. Vrahatis, "Evolutionary computation techniques for optimizing fuzzy cognitive maps in radiation therapy systems", *Lecture Notes in Computer Science (LNCS)*, vol. 3102, pp. 402–413, 2004.