

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Learning FCM with Simulated Annealing

M.Ghazanfari and S. Alizadeh

Industrial Engineering Department, Iran University of Science and Technology (IUST) Narmak, Tehran, Iran

1. Introduction

In decision making process, there are some critical components. In most of the time, numbers of these critical components are numerous and they affect each other, so analyzing them is not easy. The efficiency of decision-making depends largely on the ability of decision-makers to analyze the complex cause and effect relationships and take productive actions based on the analysis. In complex systems, different components affect each other, and these cause and effect relations show system behavior. Cause and effect are two different concepts. Causes tell the reason why something happened, whereas effects are the results of that happening. In most of the systems, managers draw a system conceptualization graph to understand all of the system aspect. This diagram shows the cause and effect relations between system components. The information about these relations generated and enriched over time with the experience of managers who are expert in that field. There are two big challenges, at first, if there is no expert to construct the above mental model how this must be drawn and secondly, if there is a way to construct that diagram with more components, how they could be analyzed. Therefore, a new mechanism must be used to bridge these two gaps and constituted with experts in first case and cluster the components into similar categories based on their behaviors for the second one. This article organized as follows: This paper is organized as follows: section 2 states some basic concepts and definitions of Fuzzy Cognitive Map (FCM) and history of FCM Automatic construction. The proposed learning model is presented in section 3. While, section 4 presents the experimental evaluation and discussion of the achieved results and model effectiveness. Finally, Section 5 covers conclusions and future research directions.

2. Theoretical background

2.1 Fuzzy cognitive map

Cognitive Maps contain components and their corresponding relations, which may be positive, negative, or neutral. A cognitive Map is a directed graph that its nodes correspond to relevant concepts and the edges state the relation between these two nodes by a sign. A positive sign implies a positive relation; moreover, any increase in its source value leads to increase in its target value. A negative sign presents negative relation and any increase or decrease in its source value leads to reverse effect to its target value. In a cognitive map if there is no edge between two nodes it means that, there is no relation between them.

Source: Simulated Annealing, Book edited by: Cher Ming Tan, ISBN 978-953-7619-07-7, pp. 420, February 2008, I-Tech Education and Publishing, Vienna, Austria

Cognitive Maps were initially introduced by Robert Axelrod in 1976 and applied in political science [1]. Also it was used in numerous areas of application such as analysis of electrical circuits [2], medicine [3], supervisory systems [4, 5, 6], organization and strategy planning [7], [8], analysis of business performance indicators [9], software project management [10,11], Information retrievals[12], modeling of plant control [13], system dynamics and complex systems [14, 15, 16, 17, 18, 19, 20, 21] and modeling virtual world [22], etc.

In 1988, Kosko introduced a new extension concept for Cognitive Map and named it fuzzy cognitive maps (FCM) [23, 24, 25, 26]. In a FCM, the relation between two nodes is determined by taking a value in interval $[-1, 1]$. While -1 corresponds to the strongest negative, $+1$ corresponds to strongest positive one. The other values express different levels of influence. This model can be presented by a square matrix called *connection matrix*. The value of relation between two nodes is set in their corresponding cell. In the connection matrix, row and column are associated with a source node and a target node, respectively. An FCM consists of nodes, $C_i, i = 1 \dots N$, where N is the total number of concepts. Each arc between two nodes C_i and C_j , has a weight F_{ij} , which is the strength of the causal link between C_i and C_j . The sign of F_{ij} indicates whether the relation between two concepts is direct or inverse. The direction of causality indicates whether the concept C_i causes the concept C_j or vice versa. Thus, there are three types of weights:

$F_{ij} > 0$ express positive causality,

$F_{ij} < 0$ express negative causality,

$F_{ij} = 0$ express no relation,

A simple FCM with five nodes and ten weighted arcs is depicted in Fig.1.

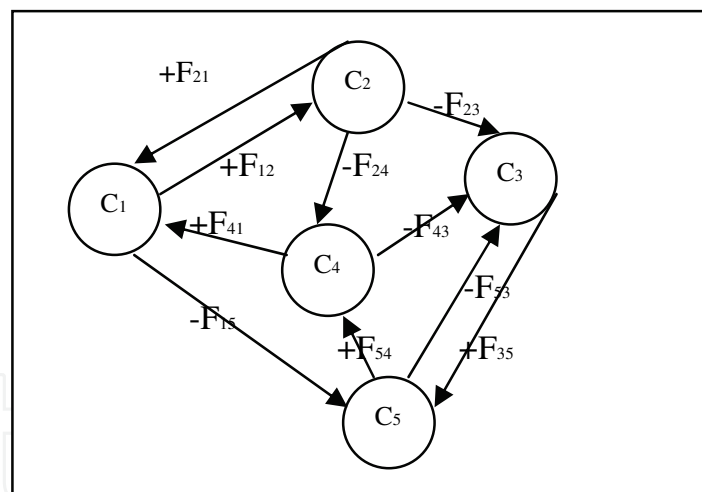


Fig. 1. A simple Fuzzy Cognitive Map (FCM)

Experts develop a FCM or a mental model manually based on their knowledge in the area under study. At first, they identify key domain issues or concepts. Secondly, they identify the causal relationships among these concepts and thirdly, they estimate causal relationships strengths. The achieved graph (FCM) shows not only the components and their relations but also the strengths.

A group of experts can be utilized to improve the results. All experts are asked to determine the relevant factors in a brain storm meeting. They discuss about main characteristics of the system, for example, number and kinds of concepts and relation between nodes, which are in the FCM. Then, they determine the structure and the interconnections of the network

using fuzzy conditional statements or fuzzy rules. Each expert may draw his own individual FCM, which can be different from the others. In order to deal with these diagrams, the assigned weights by each expert can be considered and a new FCM will be constructed by all experts. Thus, this constructed FCM will represent the knowledge and experience of all related experts. [27], [28]

FCMs can be produced by expert manually or generated by other source of information computationally. They named manual FCMs and automated FCMs.

In Fuzzy Cognitive Maps like Cognitive Map, the influence of a concept on the others is considered as “negative”, “positive” or “neutral”, but all relations are expressed in fuzzy terms, e.g. very weak, weak, medium, strong and very strong. The following set of linguistic variables is also considered:

{negatively very strong, negatively strong, negatively medium, negatively weak, zero, positively weak, medium, positively strong and positively very strong}.

The corresponding membership functions for these terms are shown in Fig. 2:

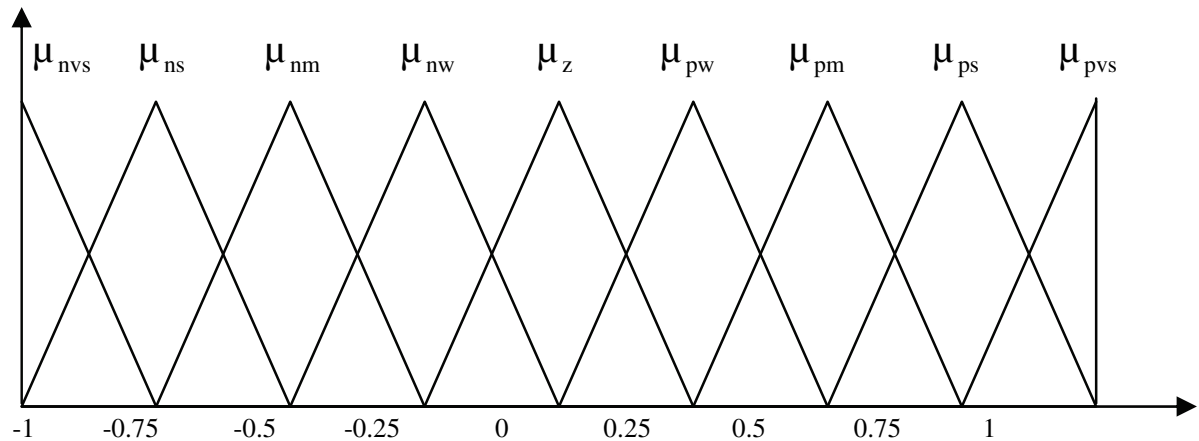


Fig. 2. Membership functions

In a FCM, all fuzzy variables are mapped into interval [-1, 1]. A simple way is to map fuzzy expression to numerical value in a range of [-1, 1]. For example, positively weak is mapped to 0.25, negatively medium to -0.5, positively strong to 0.75. [29] Then, all the suggested linguistic variables, are considered and an overall linguistic weight is obtained, with the defuzzification method of Centre of Gravity (COG) [30], is transformed to a numerical weight belonging to the interval [-1, 1].

In general, 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:

- There is no expert to define a FCM.
- The experts' knowledge is different with each other and they draw different FCM.
- There are large amount of concepts and 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 example designing a new method could eliminate the existing weakness. The related knowledge can be extracted by analyzing past information about the given systems.

2.2 Automated FCM constructin (related works)

When the experts are not able to express their expertise or even there is no expert in the area under studied to add some expression based on her/his expertise, therefore a new way should be defined. For these reasons, the development of computational methods for learning FCM is necessary [31]. In this method, not only casual relations between nodes, but also the strength on each edge must be achieved based on historical data. The required knowledge is extracted from historical data by means and new computational procedures. Many methods for learning FCM model structure have been recently proposed. In general, these methods are categorized in two min groups:

- Hebbian algorithm
- Genetic algorithm

Soft computing approach such as neural networks and genetic algorithm can be used to discover appropriate knowledge from historical data in the form of a graph or a FCM. Many researches worked on these areas by investigating FCM learning methods using historical data.

Kosko proposed a new model by use of simple Differential Hebbian Learning law (DHL) in 1994, but he used this model to learning FCMs without any applications [32]. This learning process modified weights of edges existing in a FCM in order to find the desired connection matrix. In general, when the corresponding concept changes, the value of the related edges for that nodes will be modified too.

In 2002, Vazquez introduced a new extension to DHL algorithm presented by Kosko. He used a new idea to update edge values in a new formula [33]. Another method of learning FCMs based on the first approach (Hebbian algorithm), was introduced in 2003 by Papageorgiou et al. He developed another extension to Hebbian algorithm, called Nonlinear Hebbian Learning (NHL) [34]. Active Hebbian Algorithm (AHL) introduced by Papageorgiu et al. in 2004. In the recent method, experts not only determined the desired set of concepts, initial structure and the interconnections of the FCM structure, but also identified the sequence of activation concepts [35].

Another category in learning connection matrix of FCM is application of genetic algorithms or evolutionary algorithms. Koulouriotis et al. applied the Genetic Strategy (GS) to learn FCM's structure In 2001 [36]. In mentioned model, they focused on the development of an ES-based procedure that determines the values of the cause-effect relationships (causality). Parsopoulos et al also published other related papers in 2003. They tried to apply Particle Swarm Optimization (PSO) method, which belongs to the class of Swarm Intelligence algorithms, to learn FCM structure [37, 38]. Khan and Chong worked on learning initial state vector of FCM in 2003. They performed a goal-oriented analysis of FCM and their learning method did not aim to compute the connection matrix, and their model focused on finding initial state vector for FCM [39]. In 2005, Stach et al. applied real-coded genetic algorithm (RCGA) to develop FCM model from a set of historical data in 2005 [28].

Other work to train a FCM was done by Konar in 2005. He worked on reasoning and unsupervised learning in a FCM. In this paper, a new model was introduced for unsupervised learning and reasoning on a special type of cognitive maps realized with Petri nets [40]. In 2006, Parsopoulos et al combined these two categories and published a paper about using evolutionary algorithms to train Fuzzy Cognitive Maps. In their model, they

investigated a coupling of differential evolution algorithm and unsupervised Hebbian learning algorithm [29]. Our model based on Simulated Annealing and genetic algorithm is a new method to learn connection matrix rapidly. Table 1 shows a comparison between some existing methods. The table compares the methods based on several factors, such as learning goal, kind of input historical data, type of transformation function, size of FCM model, type of learning strategy and whether experts are involved in model or not. In this table *Single* historical data consisting of one sequence of state vectors and *multiple* historical data consisting of several sequences of state vectors, for different initial conditions. When initial human intervention is equal “Yes&No” it means that human interaction is necessary but later when applying the algorithm there is no human intervention needed. This study aims to provide a learning method, which avoids disadvantages of the existing methods. It uses Simulated Annealing to develop FCM connection matrix based on data consisting of one sequence of state vectors. In contrast, the approach introduced in [36], requires a set of such sequences. The proposed method is fully automatic, i.e. in contrast to NHL and AHL methods it does not require input from a domain expert. This algorithm learns the connection matrix for a FCM that uses continuous transformation function, which is a more general problem than the one considered in [33]. The quality of RCGA algorithm [28] deteriorates with the increasing size of the maps. In general, the RCGA method achieved maps up to 6 nodes while; our algorithm is satisfied for learning FCM with nodes more than 6 with excellent quality. In next section, it shows that this algorithm is better than RCGA algorithm which proposed by Stach in learning FCM.

Algorithm	learning goal	Human Intervention	type of data used	transformation Function	NO of node	learning algorithm
DHL	Connection matrix	No	Single	N/A	N/A	Hebbian
BDA	Connection matrix	No	Single	Binary	5 7 9	Modified Hebbian
NHL	Connection matrix	Yes&No	Single	Continuous	5	Modified Hebbian
AHL	Connection matrix	Yes&No	Single	Continuous	8	Modified Hebbian
GS	Connection matrix	No	Multiple	Continuous	7	Genetic
PSO	Connection matrix	No	Multiple	Continuous	5	Swarm
GA	Initial vector	N/A	N/A	Continuous	11	Genetic
RCGA	Connection matrix	No	Single	Continuous	4,6,8,10	Genetic
SA (This paper)	Connection matrix	No	Single	Continuous	Any Number	SA

Table 1. Overview of some learning approaches applied to FCMs

3. The proposed learning method by SA

This new method proposed a solution for automatic construction of Fuzzy Cognitive Map by using Simulated Annealing. The focus of this model is to determine cause-effect relationships (causality) and their strength.

3.1 Problem definition

As mentioned before, a cause-effect relation is specified by a related Connection matrix. The elements of this matrix are the values of edges in the FCM. The aim of the proposed method is to find these elements. The relations between nodes and edges are calculated as:

$$C_i(t+1) = f \left(\sum_{j=1}^n e_{ji} C_j(t) \right)$$

where e_{ij} 's are the elements of the matrix and f is a transform function which includes recurring relation on $t \geq 0$ between $C(t+1)$ and $C(t)$ that can be presented by a logistic function like:

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

Eq. (1) and Eq. (2) can be expressed by Eq.(3):

$$Output_i(t_{n+1}) = E \times Input_i(t_n)$$

$Input_i(t_n)$ is input data for node i , $Output_i(t_{n+1})$ is its corresponding output data and E is the Connection matrix of FCM. Eq. (3) implies that for each node i its corresponding output can be calculated. E (Related connection Matrix) is a vital factor in Eq. (3) which should be determined in the FCM learning process. The proposed FCM learning methods forms structure of a FCM and is able to generate state vector sequences that transform the input vectors into the output vectors. When all real input and output values of a FCM are in hand, the most important step is to find a new solution for the FCM and calculate the estimated output related to this new FCM.

$$Output_i^{estimated}(t_{n+1}) = E^{proposed} \times Input_i(t_n)$$

According to Eq. (4), $Output_i^{estimated}(t_{n+1})$ is the estimated output and $Input_i(t_n)$ is its corresponding input for the i th node. $E^{proposed}$ is the new proposed matrix. The real output is $Output_i^{real}(t_{n+1})$ and the difference between real and estimated outputs is calculated by:

$$Error = Output_i^{estimated}(t_{n+1}) - Output_i^{real}(t_{n+1})$$

By using the later two equations, the objective is defined as minimizing the difference between real and estimated outputs. This objective is defined for all nodes as:

$$Total_Error = \sum_{n=1}^K \sum_{i=1}^N Output_i^{estimated}(t_{n+1}) - Output_i^{real}(t_{n+1})$$

Where N is the number of nodes and K is the iteration.

$$Input_i(t_n) \rightarrow Output_i(t_{n+1}) \quad \forall t = 0, \dots, K - 1$$

If $Input_i(t_n)$ defined as an *initial vector*, and $Output_i(t_{n+1})$ as *system response*, $K-1$ pairs in the form of $\{initial\ vector, system\ response\}$ can be generated from the input data. As mentioned in section 3, there are many methods for automatic constructing FCM matrix, for example, Stach et al. constructed this matrix by a Real Code Genetic Algorithm (RCGA) with simple operators. In this paper, we concentrate on simulated annealing as a heuristic model in learning FCM. Fig.3 shows the outline of the proposed method:

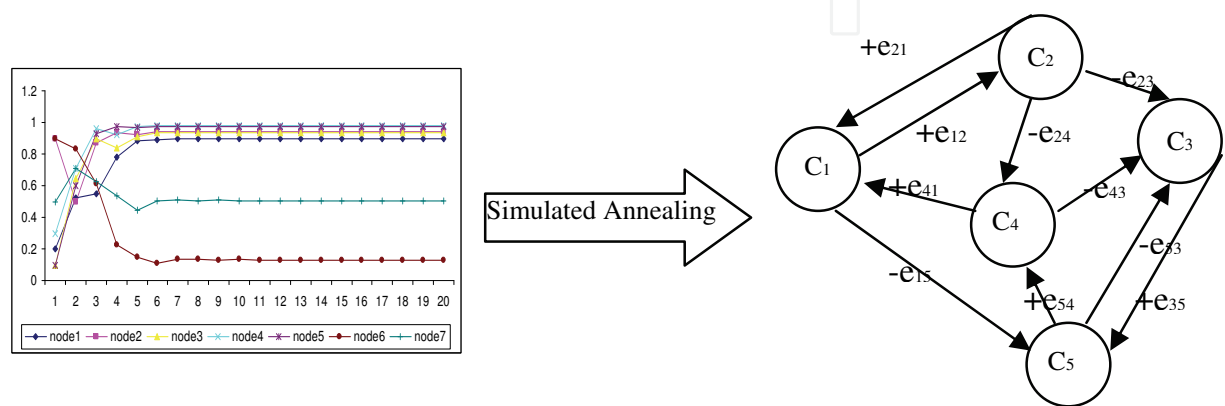


Fig. 3. the diagram of new model

The proposed learning model uses simulated annealing is used to escape the local minimum solution and to improve the optimum solution. The following sections provide details about simulated annealing and compare it with GA. It assumed that readers are familiar with GA and SA. A useful summary about relevant GA and SA can be found in [41, 42, 43]. Also, it is tried to demonstrate all essential elements of propose method, including structure of solution coding (chromosomes), generation of initial solution, initial temperature, fitness function, stopping condition, genetic operators in GA, Neighboring solutions in SA, and selection strategy.

3.2 A proposed SA method for learning FCM

In this section, SA algorithm for learning FCM is introduced. Simulated annealing is an algorithm for discrete optimization backs to the early 1980s. It was originally developed as a simulation model for a physical annealing process and hence it is referred to as simulated annealing. In simulated annealing, a problem starts at an initial solution, and a series of moves (i.e., changing the values of decision variables) are made according to a user-define annealing schedule. It terminates, when either the optimal solution is attained or the problem becomes frozen at a local optimum that cannot be improved. To avoid freezing at a local optimum, the algorithm moves slowly (with respect to the objective value) through the solution space. This controlled improvement of the objective value is accomplished by accepting non-improving moves with a certain probability that decrease as the algorithm progresses. Important parts of the simulated annealing algorithm for learning FCM are explained as follows:

SA algorithm

* Initialization: Select an initial solution x_1 in X

Initialize the best value F^{best} of F and the corresponding solution x^{best} :

$$F^{best} = F(x_1)$$
$$x^{best} = x_1$$

Initialize N, T, α

Do while not stopping condition is fulfilled $T_{n+1} > \alpha$

Do (n < N) and (Not Change)

n=n+1

Consider $x^{neighbour}$ at random in the neighbourhood

If $F(x^{neighbour}) = F(x)$ then counter=counter+1 and change=false

If $F(x^{neighbour}) < F(x)$ then $x \leftarrow x^{neighbour}$

If $F(x^{neighbour}) < F^{best}$ the $F^{best} \leftarrow F(x^{neighbour})$ and $x^{best} \leftarrow x^{neighbour}$

If $F(x^{neighbour}) > F(x^{best})$

Consider R= a random number in [0 , 1],

$$\Delta = F(x^{neighbour}) - F^{best}$$
$$\alpha = \frac{1}{1 + e^{\frac{-\Delta}{T}}}$$

If $R < \alpha$ Then $x \leftarrow x^{neighbour}$

End loop

$$T_{n+1} = \frac{1}{1 + e^{\frac{-\Delta}{T}}} * T_n, \quad N=0, \quad \text{Change=false}$$

End loop

For designing the SA algorithm, many principle factors considered and introduced here:
Solution coding
The solution coding for SA is considered in Figure (4).

e11	e12	E13	E14	E15
e21	e22	E23	E24	E25
e31	e32	E33	E34	E35
e41	e42	E43	E44	E45
e51	e52	E53	E54	E55

Fig. 4. the solution code structure

Fitness function

As mentioned before, the total difference between real and estimated outputs for all nodes is defined in Eq(7). This error can be used as the core of fitness function.

$$\text{Fitness function} = h \left(\alpha \left(\sum_{n=1}^K \sum_{i=1}^N \text{Output}_i^{\text{estimated}}(p_{n+1}) - \text{Output}_i^{\text{real}}(p_{n+1}) \right)^2 \right)$$

α is the parameter used to normalize error rate, which equal to $1/((K-1).N)$ (K and N were explained before) h is an auxiliary function. The auxiliary function h was introduced for two main reasons:

- To ensure that better individuals correspond to greater fitness function values. Argument of this function is the summed error rate, and thus needs to be inversed.
- To embed non-linearity that aims to reward solution code, which are closer to the desired solution.

The following function h was proposed: $h(x) = \frac{1}{ax + 1}$ where parameter a is established

experimentally. The fitness function is normalized to the $(0, 1]$ where It is zero for worse case or it is equal to one for an ideal chromosome, which results is exactly the same state vector sequence as the input data. The mathematical modeling of this problem is presented here: $\text{Max } Z = h(x)$

Initial solution

An initial solution is a starting solution (point) that will be used in the search process and considered as a random solution. In this research, the initial solution generates randomly.

Initial temperature and cooling schedule

An initial temperature T_0 and a cooling schedule (α) are used to control the series of moves in the SA search process. In general, the initial temperature should be high enough to allow all candidate solutions to be accepted. Cooling schedule (α) is the rate at which temperature is reduced. In this paper, a classical schedule is represented in figure (5). Starting from T_0 , the temperature is decreased through multiplication by a fixed factor (α) ($0 < \alpha < 1$).

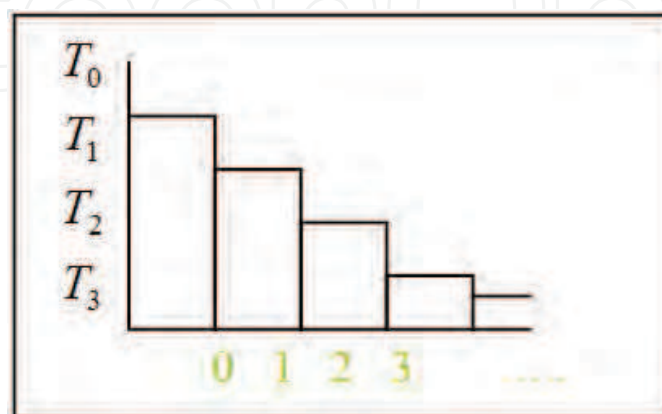


Fig. 5. temperature schedule

Neighboring solutions

Neighboring solutions are the set of feasible solutions that can be generated from the current solution. Each feasible solution can be directly reached from current solution by a move (like genetic operations mutation or inversion) and resulted neighboring solution.

Stopping criteria

Here are four criteria for stopping the algorithms (GA, SA) as follows:

- Maximum number of the established generation (G).
- Least variance of the generation (μ)
- Maximum run time for the algorithm (MRT).
- The number of temperature transitions is used as a stopping criterion. Furthermore, the SA algorithm can be terminated, when the term ($T_{n+1} > \varepsilon$) or stopping condition is satisfied. ε can be a constant or calculated by other parameters.

4. Computational results

In our experiment, the problem data were used to construct FCM by using SA on a PC Pentium IV, 1.6 GHz. The meta-heuristic algorithms were developed using Visual Basic 6. Two algorithms, the genetic algorithm which used by the others and Simulated Annealing ran with mentioned data and the Error and time consuming saved. Table 2 shows the essential parameters for these algorithms. The aim of the experiments is to assess quality of the proposed method for learning FCMs. Two algorithms, the genetic algorithm, Simulated Annealing ran with different Node numbers: 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and for every run the Error and time consuming saved. Each considered FCM, in terms of the number of nodes, was simulated 100 times with the two algorithms. The obtained results are shown in table 3.

Parameter name	Value	Comments
Generation	300	The Max Number of Generation
Population	1000	The Number of population in each Generation
p_c	0.95	Probability of crossover
p_m	0.90	Probability of mutation
δ	0.1	For stopping criterion (when algorithm $T_{n+1} > \delta$ stops)
T_n	$T_{n+1} = \alpha * T_n$	Value of temperature in transition (n)
T_0	5000	The first temperature
α	$\alpha = \frac{1}{1 + e^{\frac{-\Delta}{T}}}$	α denotes the temperature and cooling schedule in SA $\Delta = F(x^{neighbour}) - F^{best}$
μ	10	Least variance of the generation
MRT	0.5 hour	Maximum run time for the algorithm
a	10000	$h(x) = \frac{1}{ax + 1}$ parameter a is established experimentally

Table 2. Parameters in two algorithms

NO	Node	Genetic algorithm		Simulated Annealing	
		Fitness Function L2 (Error)	Time(sec)	Fitness Function L2 (Error)	Time(sec)
	2	1.7998E-06	40	3.5000E-05	2.96
	3	1.8308E-06	40	3.8308E-05	2.68
	4	3.0140E-06	40	4.8140E-05	2.39
	5	5.6714E-06	40	5.2714E-05	2.18
	6	1.5381E-05	40	5.4381E-05	1.85
	7	2.4262E-05	40	6.4262E-05	1.64
	8	3.1149E-05	40	7.1432E-05	1.51
	9	6.4874E-05	40	7.4320E-05	1.25
	10	1.0917E-04	40	7.6345E-05	1.09
	11	1.5211E-04	40	8.2345E-05	0.96
	12	1.9053E-04	40	8.7432E-05	0.75
	13	2.6694E-04	40	1.0060E-04	0.6
	14	3.2980E-04	40	1.0604E-04	0.48
	15	4.0198E-04	40	1.2029E-04	0.43

Table 3. Experiment results with different fitness functions and times

The results of these experiments show that these algorithms gradually converge into a high-quality candidate FCM. Two examples of FCM learning experiments based on GA and SA are plotted in Fig.6-1 and Fig.6-2.

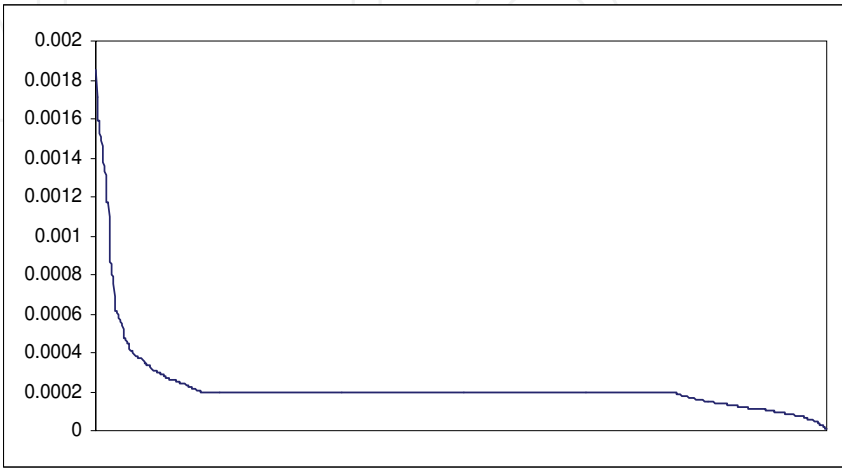


Fig. 6-1. An example of SA fitness Function which show that error covers to near zero

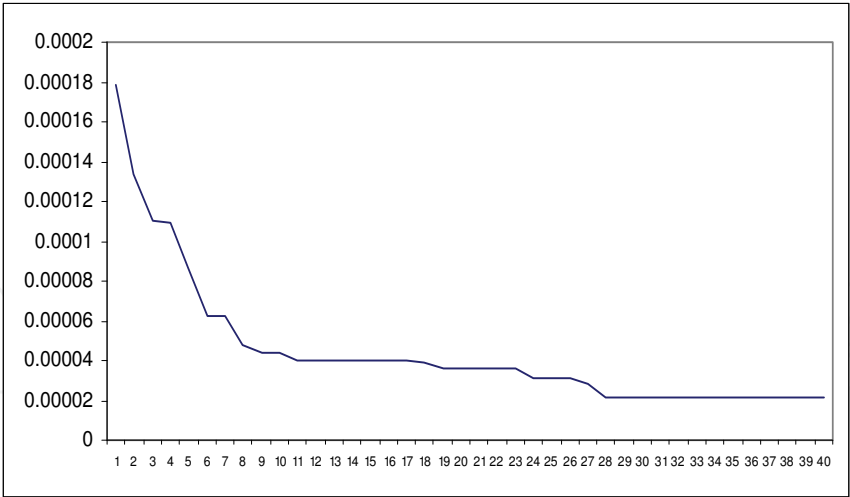


Fig. 6-2. An example of GA fitness Function which show that error coversges to near zero

Fig. 7. shows the error of GA and SA algorithms. This figure shows that SA algorithm produce better solution with less error for FCM with node number more than 10. That means SA is appropriate for learning FCM with nodes more than 10.

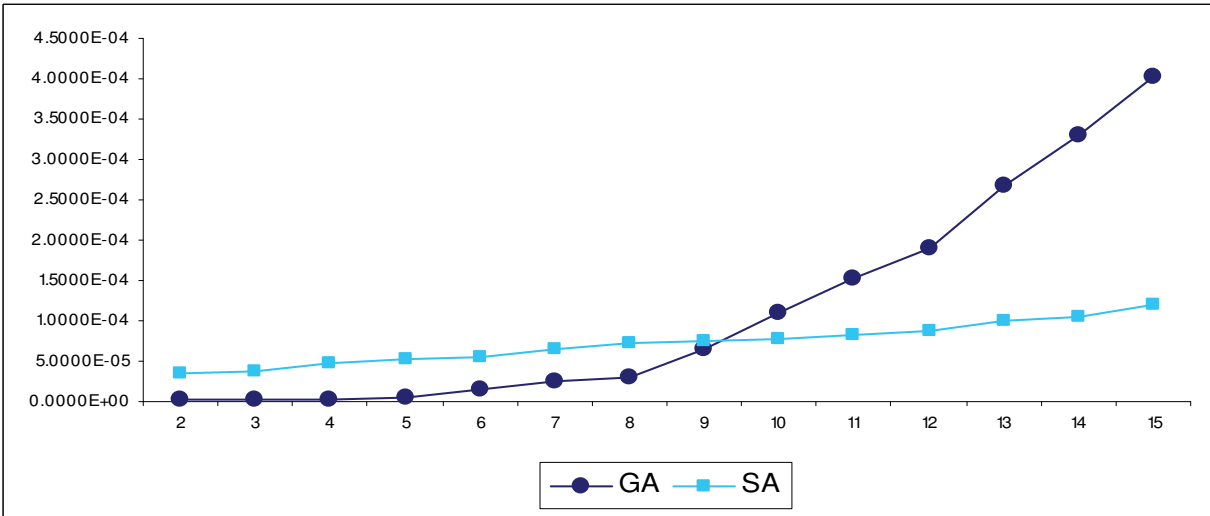


Fig. 7. GA and SA errors for different nodes

Fig. 7. compares the time consuming with GA and SA algorithms. This figure shows that simulated annealing algorithm, as a learning algorithm for FCM is faster than genetic algorithms in the same problem. Therefore, The SA algorithm found related solutions in less computational times than GA.

In these two algorithms, the error of GA in FCM with little Node is less that Simulated annealing. However, in FCM with more nodes the error of SA as a new learning algorithm is less that GA. Considering the results of Tables (3) shows that the presented metaheuristic algorithms are able to find and report the near-optimal and promising solutions in a reasonable computational time. This indicates the success of the proposed method in learning FCM. In general, we can conclude that the SA algorithm meanly found better solutions than GA in less computational times for nodes more than 10.

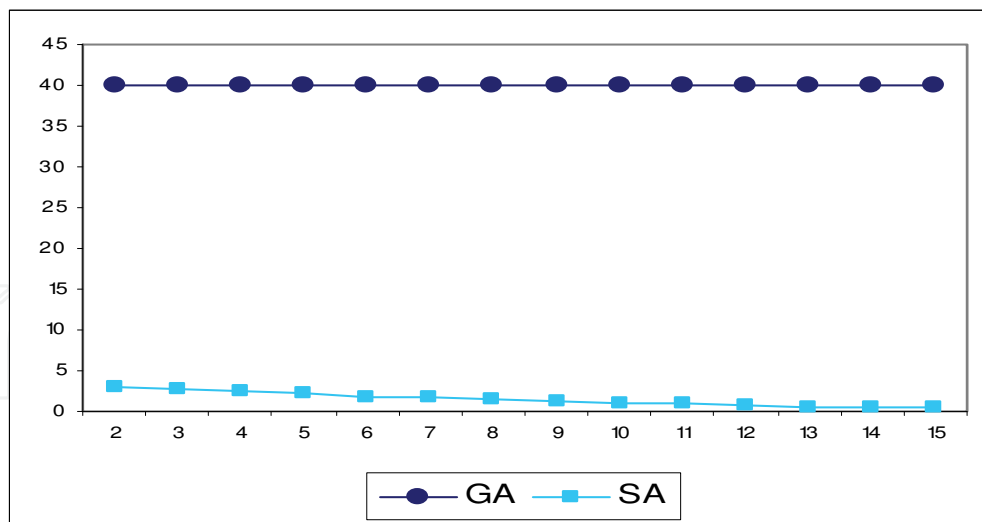


Fig. 8. Time consuming with GA and SA algorithms

5. Conclusion

In this paper, we have developed a new method for learning FCMs by SA algorithm. It has been shown that SA not only can improve the speed of learning process, but also can improve the quality of learning FCMs with nodes more than 10. The quality of learning method based on GA deteriorates with the increasing size of the maps but SA over comes this difficulty and when the size of maps increase the GA algorithms replaced with SA algorithms. According to these properties, a new method proposed. The results show this new method is very effective, and generates FCM models that can almost perfectly represent the input data. In general, the proposed method achieves excellent quality for maps in every size of FCM. The produced results could provide some guidelines for other learning methods. The future work will concern on the improvement of the proposed learning method. One of interesting and open issues is using the other heuristic methods for learning FCMs and comparing them with the others.

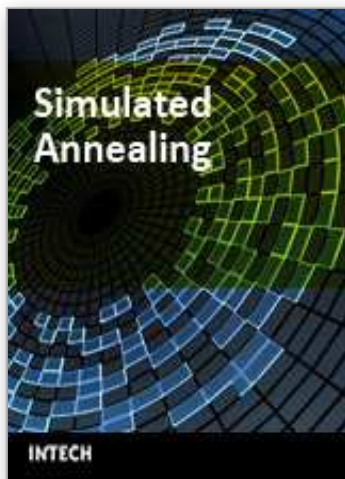
6. References

- R.Axelrod, Structure of Decision: The Cognitive Maps of Political Elites, Princeton University Press, Princeton, NJ, 1976.
- M.A. Styblinski, B.D. Meyer, Signal flow graphs versus fuzzy cognitive maps in application to qualitative circuit analysis, Internet. J. Man Mach. Studies 35 (1991) 175–186.
- V.C. Georgopoulos, G.A. Malandraki, C.D. Stylios, A fuzzy cognitive map approach to differential diagnosis of specific language impairment, J. Artif. Intel. Med. 29 (3) (2003) 261–278.
- C.D. Stylios, P.P. Groumpos, The challenge of modeling supervisory systems using fuzzy cognitive maps, J.Intel. Manuf. 9 (4) (1998) 339–345.
- C.D. Stylios, P.P. Groumpos, Fuzzy cognitive maps: a model for intelligent supervisory control systems, Comput. Ind. 39 (3) (1999) 229–238.

- C.D. Stylios, P.P. Groumpos, Fuzzy cognitive map in modeling supervisory control systems, *J. Intel. & Fuzzy Systems* 8 (2) (2000) 83–98.
- M. G. Bougon, "Congregate Cognitive Maps: a Unified Dynamic Theory of Organization and Strategy," *Journal of Management Studies*, 29:369-389, (1992)
- K.C. Lee, W.J. Lee, O.B. Kwon, J.H. Han, P.I. Yu, Strategic planning simulation based on fuzzy cognitive map knowledge and differential game, *Simulation* 71 (5) (1998) 316–327.
- D. Kardaras, G. Mentzas, Using fuzzy cognitive maps to model and analyze business performance assessment, in: J. Chen, A. Mital (Eds.), *Advances in Industrial Engineering Applications and Practice II*, 1997, pp. 63–68.
- W. Stach, L. Kurgan, Modeling software development project using fuzzy cognitive maps, *Proc. 4th ASERC Workshop on Quantitative and Soft Software Engineering (QSSE'04)*, 2004, pp. 55–60.
- W. Stach, L. Kurgan, W. Pedrycz, M. Reformat, Parallel fuzzy cognitive maps as a tool for modeling software development project, *Proc. 2004 North American Fuzzy Information Processing Society Conf. (NAFIPS'04)*, Banff, AB, 2004, pp. 28–33.
- A. R. Montazemi, D. W. Conrath, "The Use of Cognitive Mapping for Information Requirements Analysis," *MIS Quarterly*, 10:44-55, (1986)
- K. Gotoh, J. Murakami, T. Yamaguchi, Y. Yamanaka, Application of fuzzy cognitive maps to supporting for plant control, *Proc. SICE Joint Symp. 15th Systems Symp. and Tenth Knowledge Engineering Symp.*, 1989, pp. 99–104.
- Carvalho, J.P., Tomé, J.A., "Rule Based Fuzzy Cognitive Maps and Fuzzy Cognitive Maps - A Comparative Study", *Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society, NAFIPS99*, New York
- Carvalho, J.P., Tomé, J.A., "Rule Based Fuzzy Cognitive Maps- Fuzzy Causal Relations", *Computational Intelligence for Modelling, Control and Automation*, Edited by M. Mohammadian, 1999
- Carvalho, J.P., Tomé, J.A., "Fuzzy Mechanisms for Causal Relations", *Proceedings of the Eighth International Fuzzy Systems Association World Congress, IFSA'99*, Taiwan
- Carvalho, J.P., Tomé, J.A., "Rule Based Fuzzy Cognitive Maps - Qualitative Systems Dynamics", *Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society, NAFIPS2000*, Atlanta
- S. Alizadeh, M. Ghazanfari, M. Jafari, "An approach for solving fuzzy system dynamics problems", *21st International system dynamics conference*, U.S.A, July 2003.
- D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, E.N. Antonidakis, I.A. Kaliakatsos, Efficiently modeling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps (Invited Paper), *Internat. J. Comput. Cognition* 1 (2) (2003) 41–65.
- D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, C.D. Zopounidis, Development of dynamic cognitive networks as complex systems approximators: validation in financial time series, *Applied Soft Computing* 5 (2005) 157–179
- C.D. Stylios, P.P. Groumpos, Modeling complex systems using fuzzy cognitive maps, *IEEE Trans. Systems Man, Cybern. Part A: Systems Humans* 34 (1) (2004).

- D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, Anamorphosis of fuzzy cognitive maps for operation in ambiguous and multi-stimulus real world environments, 10th IEEE Internet. Conf. on Fuzzy Systems, 2001, pp. 1156–1159.
- B. Kosko, Hidden patterns in combined and adaptive knowledge networks, Internet. J. Approx. Reason. 2 (1988) 377–393.
- B. Kosko, Fuzzy cognitive maps, Internat. J. Man-Mach. Studies 24 (1986) 65–75.
- B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.
- B. Kosko, Fuzzy Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1997.
- M. Khan, M. Quaddus, Group decision support using fuzzy cognitive maps for causal reasoning, Group Decision Negotiation J. 13 (5) (2004) 463–480.
- Wojciech Stach, Lukasz Kurgan, Witold Pedrycz, Marek Reformat Genetic learning off uzzy cognitive maps, Fuzzy Sets and Systems 153 (2005) 371–401
- E Papageorgiou, C. Stylios P. Groumpos, Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links, Int. J. Human-Computer Studies 64 (2006) 727–743
- Lee C. C, Fuzzy logic in control systems: Fuzzy logic controller, Part 1, 2 IEEE Trans. Syst. Man Cybernet, 20 (2), 404–435.
- M. Schneider, E. Shnaider, A. Kandel, G. Chew, Automatic construction of FCMs, Fuzzy Sets and Systems 93 (2) (1998) 161–172.
- J.A. Dickerson, B. Kosko, Fuzzy virtual worlds, Artif. Intel. Expert 7 (1994) 25–31.
- 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.
- E. Papageorgiou, C.D. Stylios, P.P. Groumpos, Fuzzy cognitive map learning based on nonlinear Hebbian rule, Australian Conf. on Artificial Intelligence, 2003, pp. 256–268.
- E. Papageorgiou, C.D. Stylios, P.P. Groumpos, Active Hebbian learning algorithm to train fuzzy cognitive maps, Internat. J. Approx. Reason. 37 (3) (2004) 219–249.
- 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, IEEE Congr. On Evolutionary Computation (CEC2001), 2001, pp. 364–371.
- E. Papageorgiou, K.E. Parsopoulos, C.D. Stylios, P.P. Groumpos, M.N. Vrahatis, Fuzzy cognitive maps learning using particle swarm optimization, J. Intel. Inform. Systems. 2005
- K.E. Parsopoulos, E.I. Papageorgiou, P.P. Groumpos, M.N. Vrahatis, A first study of fuzzy cognitive maps learning using particle swarm optimization, Proc. IEEE 2003 Congr. On Evolutionary Computation, 2003, pp. 1440–1447.
- M. Khan, A. Chong, Fuzzy cognitive map analysis with genetic algorithm, Proc. 1st Indian Internat. Conf. on Artificial Intelligence (IICAI-03), 2003
- Amit Konar, Uday K. Chakraborty, Reasoning and unsupervised learning in a fuzzy cognitive map, Information Sciences 170 (2005) 419–441
- Duc Truong Pham, Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks.

- Marc Pirlot, General local search methods, European journal of operational research 92, 1996 , 493-511
- van Laarhoven, and Aarts, E. (1987): Simulated Annealing: Theory and Applications. Dordrecht: Reidel.
- R.Tavakkoli-Moghaddam, M.B. Aryanezhad, N.Safaei, A. Azaron, Solving a dynamic cell formation problem using metaheuristics, Applied Mathematics and Computation 170 (2005) 761-780



Simulated Annealing

Edited by Cher Ming Tan

ISBN 978-953-7619-07-7

Hard cover, 420 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

This book provides the readers with the knowledge of Simulated Annealing and its vast applications in the various branches of engineering. We encourage readers to explore the application of Simulated Annealing in their work for the task of optimization.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

M.Ghazanfari and S. Alizadeh (2008). Learning FCM with Simulated Annealing, Simulated Annealing, Cher Ming Tan (Ed.), ISBN: 978-953-7619-07-7, InTech, Available from:

http://www.intechopen.com/books/simulated_annealing/learning_fcm_with__simulated__annealing

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen