

The procedure of system dynamics modeling could be broadly broken into these major steps:

- Problem definition.
- System conceptualization via causal loop diagrams or influence diagrams.
- Model formulation.
- Simulation and validation of model.
- Policy analysis and improvement.
- Policy implementation.

Fig.1 shows the relation among these phases: [5].

2-2. Data mining and Knowledge Discovery

Data mining or knowledge discovery in databases (KDD) is the process of discovering useful knowledge from large amount of data stored in databases, data warehouses, or other information repositories. [6,7] Data mining is a hybrid disciplinary that integrates technologies of databases, statistics, artificial intelligent. Recently, a number of data mining applications and prototypes have been developed for a variety of domains, including marketing, banking, finance, manufacturing, and health care other types of scientific data.[8] The more common model functions in current data mining practice include the following [9]:

- Classification: classifies a data item into one of several predefined categorical classes.
- Clustering: maps a data item into one of several clusters
- Discovering association rules: describes association relationship among different attributes.
- Summarization: provides a compact description for a subset of data.
- Dependency modeling: describes significant dependencies among variables.
- Sequence analysis: models sequential patterns, like time-series analysis.

The other soft computing methodologies like fuzzy sets, neural networks, and Genetic Algorithms are most widely applied in the data mining. Fuzzy sets copes with uncertainty in data mining process. Neural networks are used for classification and rule generation. Genetic algorithms (GAs), simulated Annealing (SA), Tabu search (TS) are involved in various optimization and search processes [10]. All of these functions and methodologies tried to discover knowledge from historical data. This knowledge represented in the form of *rules* most of the time.

This paper focused on discovering knowledge with the form of graph. Therefore, in this paper investigated how soft computing methodologies and data mining functions can discover this kind of knowledge, which named Fuzzy Cognitive Map (FCM).

2-3. Fuzzy Cognitive Map (FCM) and Causal Loop Diagram

Cognitive maps were initially introduced by Robert Axelrod in 1976 and applied in political science

[11]. Also it was used in numerous areas of application such as analysis of electrical circuits [12], medicine [13], supervisory systems [14,15,16], organization and strategy planning [17], [18], analysis of business performance indicators [19], software project management [20,21], Information retrievals [22] modeling of plant control [23], system dynamics and complex systems [24,25,26,27,28,29] and modeling virtual world [30,31].

This model contains 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 relations between every 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. In 1988, Kosko introduced a new extension to cognitive map [32]. It named fuzzy cognitive maps. In a simple fuzzy cognitive map, 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 is associated with a source node and a target node, respectively. A simple FCM with five nodes and ten weighted arcs is depicted in Fig.2.

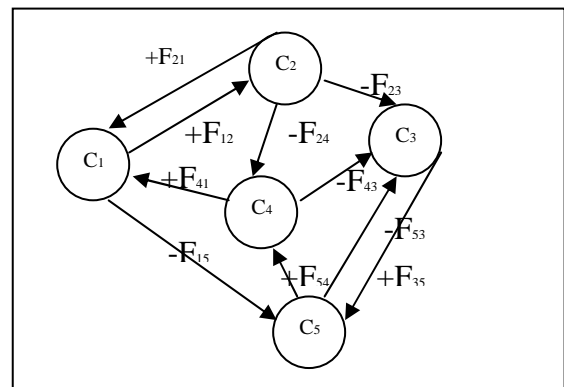


Fig. 2. A simple Fuzzy Cognitive Map (FCM)

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, number and kind of concepts, which should be contained in the FCM. Then, they determine the structure and the interconnections of the network using fuzzy conditional statements. 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' expertise. Thus, this FCM will represent the knowledge and experience of all related experts [33,34].

FCMs can be produced by expert manually or generate by other source of information computationally. Experts developed a FCM or a mental model manually based on their knowledge in related area. 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.

This achieved graph (FCM) shows not only the components and their relations but also the strengths. In fuzzy diagrams, the influence of a concept on the others is considered as "negative", "positive" or "neutral". All relations are expressed in fuzzy terms, e.g. very weak, weak, medium, strong and very strong. Here, the following set of linguistic variables is 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. 3 and they are:

$$\mu_{nvs}, \mu_{ns}, \mu_{nm}, \mu_{nw}, \mu_z, \mu_{pw}, \mu_{pm}, \mu_{ps}$$

and μ_{pvs}

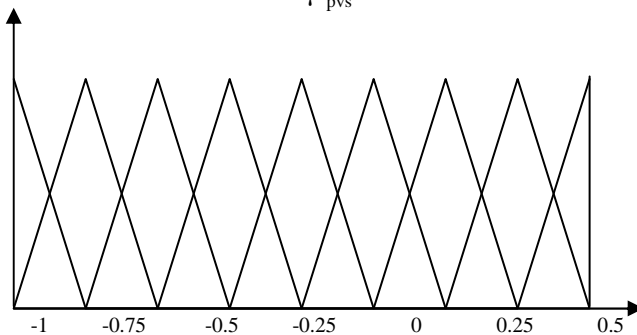


Fig. 3. Membership functions

In a simple 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. Then, all the suggested linguistic variables, are aggregated and an overall linguistic weight is obtained, with the defuzzification method of Centre of Gravity (COG) [35], 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 studied. In some situations, a FCM could not construct 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, developing a FCM manually becomes very difficult and experts' intervention could not resolve the problem. When the experts were not able to express their expertise or even there is no expert in the related area, therefore there is a gap. For these reasons, the development of computational methods for learning FCM is necessary.

For this purpose, not only the edge or casual relations between nodes, but also the strength on each edge must be achieved based on historical data. In this way, the expert knowledge is substituted by the knowledge, which produced from historical data by means and new computational procedures. Many algorithms for learning FCM model structure have been recently proposed. In general, four main categories of these algorithms are used:

- Association rules algorithms
- Distance based algorithms (classification)
- Hebbian algorithm (soft computing)
- Genetic algorithm (soft computing)

The application of association rules in FCM(s) used by Lee et al. in 2002 for the first time [36]. They published a paper about FCM(s) and their relation with web mining inference amplification. In their paper, a fuzzy cognitive map (FCM) used to amplify inference results of Web mining as a dramatic usage of the Internet for a wide variety of daily management activities.

They stated that causal knowledge is similar to IF_THEN rules with many differences. The causal knowledge seems natural and more understandable but richer in interpretation. Their propose model was composed of three major phases. At first, extract association rules from related historical data, secondly, transform into FCM causal knowledge based, and thirdly inference amplification. In their model, all rules discovered by association rule function and eliminate rules redundancy and search for the directly and indirectly rules.

The other category in learning connection matrix of FCM is distance-based algorithm. In 1998 M. Schneider and el worked on constructing fuzzy cognitive maps automatically. [37] Their method found not only the degree of similarity between any two variables (represented by numerical vectors), but also the relations between variables. They used the fuzzy expert system tool (FEST) which determined the causality among variables. D. Kardaras et al. presented another similar method in 1998. [38] In their paper assumed that every concept in an FCM could be represented by a numerical vector (V), whereas each

element (v) of the vector represents a measurement of the concept. Experts can determine an upper threshold (a_u) and a lower threshold (a_l) for every vector, so that:

$$\forall v (v \geq a_u) \Rightarrow (\mu_v = 1) \quad (1)$$

$$\forall v (v \leq a_l) \Rightarrow (\mu_v = 0) \quad (2)$$

Where (μ_v) is the membership degree of element (v). For all the elements (v) between the thresholds, a formula is applied to project all elements into the interval (0, 1) proportionally.

Therefore, every concept is a fuzzy set. Both the polarity and the strength of the relationship between two concepts are based on the concept of the similarity between two vectors. More specifically, in the case where two vectors (v_1) and (v_2) are monotonically increasing (direct relation), the distance (d_i) between two elements (v_i) of the vectors is defined as:

$$d_i = |\chi_1(v_i) - \chi_2(v_i)| \quad (3)$$

Where χ_1 and χ_2 is the degree of membership for the (i) element of the vectors (v_1) and (v_2) respectively. In the case of monotonically decreasing vectors (reverse relation) the distance of the two elements (v_i) is defined as:

$$d_i = |\chi_1(v_i) - (1 - \chi_2(v_i))| \quad (4)$$

Let AD be the average distance the vectors then:

$$AD = \frac{\left(\sum_{i=1..n} |d_i| \right)}{n} \quad (5)$$

Once the average distance is calculated then the similarity (S) of the two vectors is defined as:

$$S = 1 - AD \quad (6)$$

The similarity is calculated twice, once based on direct relation and second based on the reverse relation. The higher similarity determines the polarity (1 or 2) and the strength of the relationships between the two fuzzy sets (v_1) and (v_2). Soft computing like neural network and Genetic Algorithm help data mining to discover appropriate knowledge in the form of Graph or Fuzzy Cognitive map (FCM) from historical data. Many scientists work on this area and investigated that FCM and its related connection matrix are learned and discovered by historical data.

Soft computing approach such as neural networks and genetic algorithm can be used to discover appropriate knowledge from historical data in the form of graph or FCM. Many researchers 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 [39]. 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. [40] Another method of learning FCMs based on the first approach (Hebbian algorithm), was introduced by Papageorgiou et al. in 2003. He developed another extension to Hebbian algorithm, called Nonlinear Hebbian Learning (NHL) [41]. 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. [42]

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 structure in 2001 [30]. In mentioned model, they focused on the development of an ES-based procedure that determines the values of the cause-effect relationships (causality). Other related papers were also published by Parsopoulos et al. in 2003. They tried to apply Particle Swarm Optimization (PSO) method, which belongs to the class of Swarm Intelligence algorithms, to learn FCM structure [43, 44]. 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 [45]. In 2005, Stach et al. applied real-coded genetic algorithm (RCGA) to develop FCM model from a set of historical data in 2005 [34].

In 2005, 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 [46]. Other work to train a FCM was done by Konar in 2005. He worked on reasoning and unsupervised learning in a FCM. In that paper, a new model was introduced for unsupervised learning and reasoning on a special type of cognitive maps realized with Petri nets [47]. In 2007 M.Ghazanfari et al. published a paper about using Simulated Annealing and genetic algorithm in FCM learning [48]. In that paper, they show that SA algorithm is better than GA in FCM with more nodes and introduced a new method to learn connection matrix rapidly. In this research, the other heuristic algorithms are used to learn FCM matrix.

3. A New Model for learning FCM in System Dynamics Simulation

As mentioned, in system dynamics a casual loop diagram is needed to simulate the interactions among components. If there is no expert to define these relationship, this method can define them based on their historical behavior. This methodology focused on automatic casual loop diagram construction and the other steps are like formal system dynamics steps. Here are steps of this methodology:

- Description the problem.
- Define the most important components and parameters in that problem
- Collecting related data about these parameters during the time. Normalize and show them in the form of time series.

After defining the most important parameters, their related data must collected. As these data show the components behavior during time, they can be shown with the form of time series. But they must be preprocessed and normalized to be prepared for learning. These data are normalized according this formula:

$$Normal(c_i) = \frac{c_i - c_{Min}}{c_{Max} - c_{Min}} \quad (7)$$

In this formula every element (c_i) normalized between $[0,1]$. c_{Min} is minimum value and c_{Max} is the maximum value.

(a). Learning FCM based on their historical behavior.

In this step, a solution for automatic construction of Fuzzy Cognitive map is found by using Simulated Annealing, Genetic Algorithm and Tabu search. The focus of this model is to determine cause-effect relationships (causality) and their strength.

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:

$$C_i(t+1) = f \left(\sum_{j=1}^n e_{ji} C_j(t) \right) \quad (8)$$

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:

$$f(x) = \frac{1}{1 + e^{-cx}} \quad (9)$$

Eq. (8) and Eq. (9) can be expressed by Eq. (10):

$$Output_i(t_{n+1}) = E \times Input_i(t_n) \quad (10)$$

$Input_i(t_n)$ 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. (10) implies that corresponding output for every node can be calculated. E (Related connection Matrix) is a vital factor in Eq. (10) which should be determined in the FCM learning process. The proposed FCM learning method 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.

$$Output_i^{estimated}(t_{n+1}) = E^{proposed} \times Input_i(t_n) \quad (11)$$

According to Eq. (11), $Output_i^{estimated}(t_{n+1})$ is the estimated output and $Input_i(t_n)$ is its corresponding input for the i th node. $E^{proposed}$ is new proposed matrix. The real output is $Output_i^{real}(t_{n+1})$ and the difference between real and estimated outputs is calculated by Eq.(12):

$$Error = Output_i^{estimated}(t_{n+1}) - Output_i^{real}(t_{n+1}) \quad (12)$$

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

$$Total_Error = \sum_{n=1}^K \sum_{i=1}^N Output_i^{estimated}(t_{n+1}) - Output_i^{real}(t_{n+1}) \quad (13)$$

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

$$Input_i(t_n) \rightarrow Output_i(t_{n+1}) \quad \forall t = 0, \dots, K-1 \quad (14)$$

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 2.3, there are many methods to construct FCM matrix automatically, for example, Stach et al. constructed this matrix by a Real Code Genetic Algorithm (RCGA) with simple operators and Ghazanfari et al. constructed this matrix by SA. In this paper, we compared GA, SA and TS to learn FCM and the best FCM is chosen. The proposed learning model uses Genetic algorithm to find the near optimum solution. Simulated annealing or Tabu search are used to escape the local minimum solution and to improve the optimum solutions. In this paper assumed that readers are familiar with GA and SA and TS algorithms. A useful summary about relevant GA and SA and TS can be found in [49,50,51].

(b) To Continue the Other System Dynamic Steps in Order to Simulate and Policy Making.

After building FCM or casual loop diagram the other System Dynamics steps can be done. The other steps are “Model formulation”, “Simulation and validation of model”, “Policy analysis and improvement” and “Policy implementation”. As mentioned before in this paper we focus on the learning casual loop diagram and the other step don’t consider.

4. Illustration

In this paper, a purchase planning system is considered and tried to draw the relationship between its components. This system determines details of purchase planning. For example, it defines how a special purchasing must be done, who are the best suppliers, when is the best time to start in order to decrease the cost and so on. According to the proposed methodology, at first, the problem must be described and the related parameters must be defined. Table(1) shows the most important parameters in the purchase system which consider here:

Tab. 1. Parameters of the purchase planning system

<u>1-Total cost:</u> The total cost that influence to inventory cost	<i>CTotal</i>
<u>2-Cost of repair:</u> It is the repair cost and it is related to quality of parts.	<i>CR</i>
<u>3-Cost of parts:</u> It is cost of parts purchased from the suppliers.	<i>Cpart</i>
<u>4-Cost of inventory:</u> It is the cost of inventory.	<i>Cinv</i>
<u>5-Quality:</u> It is quality of parts, which is purchased from the suppliers.	<i>Q</i>
<u>6-Product Cost:</u> Cost of production.	<i>P_Cost</i>
<u>7- Holes in product lines:</u> This holes in production lines force a Cost of line sleep: It is the cost of delay in production line.	<i>L_Sleep</i>
<u>8-Lead-Time:</u> This is the time, which in that interval purchase must be done.	<i>LT</i>
<u>9-Supplier number:</u> It is the number of suppliers, which have the acceptable parameters for our purchase. For example, their quality must be good and their cost must be low.	<i>Supp- No</i>
<u>10-Transportation Cost:</u> Cost Of part transportation to inventory.	<i>T_Cost</i>
<u>11-Customer satisfaction:</u> The level of customer satisfaction	<i>CS</i>

• Collecting Related Data.

After defining the most important parameters, their related data must be collected. As these data show the components behavior during time, they can be shown with the form of time series. But they must be preprocessed and normalized to be prepared for learning. For this system these data are gathered and normalized according this formula:

$$Normal(c_i) = \frac{c_i - c_{Min}}{c_{Max} - c_{Min}}$$

In this formula every element (c_i) normalized between [0,1]. c_{Min} is minimum value and c_{Max} is the maximum value.

In this paper, the behaviour of related components gathered in a period of 20 weeks.

• Definition of the Effects of the System Components on each Other by Learning FCM From Historical Data

In this step, a FCM is learned by means of historical data. Some heuristic models like GA, SA and TS are used for learning methods. All essential elements of these proposed methods, including structure of solution coding (chromosomes), generation of initial solution, initial temperature and fitness function, stopping condition, genetic operators in GA, Neighboring solutions in SA and TS, and selection strategy are demonstrated here:

Genetic Algorithms

For designing the GA, many principle factors are considered as follows:

Solution coding (chromosome structure)

The chromosome structure or a solution for FCM is formed as a matrix. In GA, the initial solution generates random.

Fitness value

The fitness value is a criterion for the quality measurement of a chromosome or feasible solution. An offspring or a new solution is accepted when its objective function value is minimum as compared with its parents. The objective function is:

Objective Function =

$$\sum_{n=1}^K \sum_{i=1}^N Output_i^{estimated}(t_{n+1}) - Output_i^{real}(t_{n+1}) \tag{10}$$

Mating pool selection strategy

For creating the new generation, it is necessary to select some chromosomes (mating pool) with the latest fitness in the current generation for recombining or creating chromosomes related to the new generation. In this paper, at first, the population sort from the best to the worst. Secondly, the selection probability assigned to each chromosome. Thirdly, they selected according to their ranking numbers.

GA operators

In this paper, the chromosome structure is formed as a matrix. Thus, the GA linear operators cannot be used to a matrix type as the traditional forms. These operators

should be improved proportional to the matrix type. Therefore, considering the nature of the matrix, each of three operators called crossover, mutation, and inversion are considered as follows:

In horizontal operator, at first two positions in row selected randomly and operation is exercised over this selected area. Vertical operator is similar to the horizontal operator but two positions selected in column randomly. In diametric operator, two numbers in the relevant matrix column or row limits and one of the directions of primary or secondary are selected randomly.

Stopping criteria

The stopping criteria condition can be defined by two different ways:

- *Number of generations*, In this case, the algorithm terminates, if the number of generations exceeds the specific number.
- *Best Solutions Found*, In this case, the algorithm terminates, if the best solution find and the error function does not change after a period.

Simulated Annealing for Learning FCM

For designing the SA algorithm, many principle factors considered and introduced here:

Solution coding

The solution coding for SA is equal to GA, which mentioned before.

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, When SA algorithm is used to optimize FCM structure alone, its initial solution generates random and in combined GA and SA model, the best solution defined by GA algorithm is used as initial solution in SA model.

Initial temperature

An initial temperature T_0 and a cooling schedule α are used to control the series of moves in the 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 the temperature is reduced. In this paper, the parameter α is obtained as follows:

$$\Delta = F(x^{neighbour}) - F^{best} \quad (16)$$

$$T_{n+1} = \frac{1}{1 + e^{\frac{-\Delta}{T}}} * T_n \quad (17)$$

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 (as genetic operations—mutation or inversion) and resulted neighboring solution.

Stopping criteria

The number of temperature transitions is used as a stopping criterion. Furthermore, the SA algorithm can be terminated, when the term ($T_{n+1} > \alpha$) or stopping condition is satisfied. α can be a constant or calculated by other parameters.

Tabu Search for Learning FCM

For designing the TS, many principle factors are considered as follows:

Solution coding

The solution coding for TS is similar to GA and SA that mentioned before.

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, When TS algorithm is used to optimize FCM structure alone, its initial solution generates in a random way.

Neighboring solutions and move

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 (as genetic operations mutation or inversion) and resulted neighboring solution. The solutions admitted to $N^*(x)$, the new neighborhood, are determined through using special memory structures (Tabu List).

The search progresses by iteratively moving from a solution x to a solution x' in $N^*(x)$. To prevent the search from cycling, attributes of recently visited solutions are memorized in a *tabu list* for a number of iterations (*tabu duration*).

Stopping criteria

Tabu search uses a local or neighborhood search procedure to iteratively move from a solution x to a solution x' in the neighborhood of x , until some stopping criterion has been satisfied. In this paper stopping criteria is number of iterations or time.

Computational result

In our experiment, to compare meta-heuristic results, some test problems were solved by using GA, SA on a PC Pentium IV, 1.6 GHz. The meta-heuristic algorithms were developed by using Visual Basic 6. These algorithms ran under mentioned conditioned and the Error was saved. Table 2 shows these conditions:

Tab. 2. Parameters in two algorithms

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	0.9	α denotes the temperature and cooling schedule in SA
μ	10	Least variance of the generation
MRT	0.5 hour	Maximum run time for the algorithm
Tabu List	20	To prevent the search from cycling a list was constructed

Three algorithms, Genetic Algorithm (GA), Simulated Annealing (SA) and Tabu Search (TS) ran on related data. In each run, stopping condition changed and the error and time consuming saved. Each considered FCM was simulated 100 times with the three algorithms, which totaled in 900 experiments.

The obtained results are shown in table (3,4,5). Considering the results of Tables (3,4,5) the presented meta-heuristic 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 algorithms. In general, we can conclude the following results:

All heuristic algorithms found the near-optimal solutions in less computational times than the optimal algorithm. The results of these experiments show that these algorithms gradually converge into a high-quality candidate FCM. Three examples of FCM learning experiments based on GA, SA and TS are plotted in Figures (4-1), (4-2) and (4-3).

Tab. 3. Stopping criteria: Cpu Time =100 (MIN)

Method	Fitness Value		Cpu Time
	F_average	Var(F_average)	
GA	2.98919E-05	9.38E-11	100
SA	0.000217225	6.67E-09	1.89
TS	2.33031E-06	4.07E-13	56.3

Tab. 4. Stopping criteria: Cpu Time =40(MIN)

Method	Function Value		Cpu Time
	F_average	Var(F_average)	
GA	3.15603E-05	6.01E-10	40
SA	0.000217225	6.67E-09	1.89
TS	2.06654E-06	1.18E-12	40

Tab. 5. Stopping criteria: Cpu Time =6(MIN)

Method	Function Value		Cpu Time
	F_average	Var(F_average)	
GA	0.000248822	2.97E-08	6
SA	0.000217225	6.67E-09	1.89
TS	2.52243E-06	5.82E-12	6

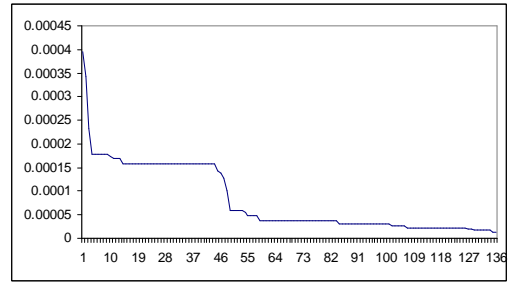


Fig. 4-1. GA fitness Function which show that error converges to near zero

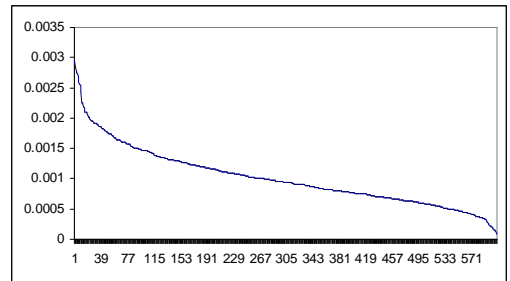


Fig. 4-2. SA fitness Function which show that error converges to near zero

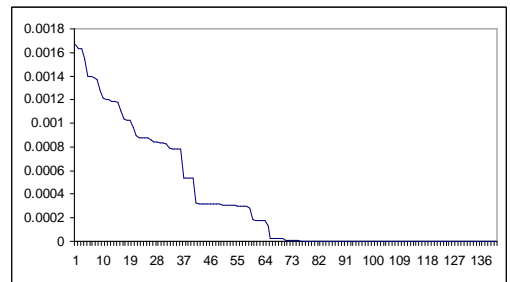


Fig. 4-3. TS fitness Function which show that error converges to near zero

- The TS algorithm meanly found better solutions with less error than GA and SA.
- The SA algorithm meanly found solutions in less computational times than GA and TS.

As explained before, TS algorithm was a good way to learn FCM with less error. The connection matrix shown in table.6 is the output produced by TS algorithm. This matrix is corresponding with a graph which exhibits the relation between components of

mentioned purchase system. This graph is a complete graph and is too complicated to show all of the relations in a figure.

5. Conclusion

In this study, a comprehensive learning method has been proposed to construct FCMs based on historical data. Some meta-heuristics (GA, SA, and TS) have been used to extract FCMs in the proposed method and their results have been compared. As a case study, a FCM has generated based on historical data of a real purchase system using the proposed method. Comparison of the results in this case study show that TS is more effective, generating FCMs which almost perfectly represent the system's behavior. 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. Another interesting direction concerns the use of the learned FCM in ranking components in complex systems.

References

- [1] Bingchiang, Jeng., Jian-xun Chen, Ting-peng Liang, *Applying Data Mining to Learn System Dynamics in a Biological Model*, Expert system with application 2005.
- [2] Forrester, J.W., *Industrial Dynamics*, The MIT Press, Cambridge, MA, 1961.
- [3] J.W. Forrester, *World Dynamics*, Wright-Allen Press, Cambridge, MA, 1971.
- [4] Steerman, J., "Business Dynamics System Thinking and Modeling for a Complex World", Irwin/mcgraw Hill, Boston, 2000.
- [5] Sushil, "System Dynamics a Practical Approach for Managerial Problems", center for management studies: Wiley Eastern Limited, 2002.
- [6] Fayyad, U., Uthurusamy, R., "Data Mining and Knowledge Discovery in Databases," Commun. ACM, Vol. 39, 1996, pp. 24–27.
- [7] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., Eds., *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI/MIT Press, 1996.
- [8] Pal, S.K., Mitra, S., *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*. New York: Wiley, 1999.
- [9] Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques*, Morgan kaufman publishers, 2006.
- [10] Hussein Aly Abbass, Ruhul Amin, Sarker, Charles S., Newton. *Data Mining: a Heuristic Approach*, Idea Group Publishing, 2002.
- [11] Axelrod, R., *Structure of Decision: The Cognitive Maps of Political Elites*, Princeton University Press, Princeton, NJ, 1976.
- [12] Styblinski, M.A., Meyer, B.D., *Signal Flow Graphs Versus Fuzzy Cognitive Maps in Application to Qualitative Circuit Analysis*, Internat. J. Man Mach. Studies 35, 1991, pp. 175–186.
- [13] Georgopoulos, V.C., Malandraki, G.A., Stylios, C.D., *A Fuzzy Cognitive Map Approach to Differential Diagnosis of Specific Language Impairment*, J. Artif. Intel. Med. 29 (3), 2003, pp. 261–278.
- [14] Stylios, C.D., Groumpos, P.P., *The Challenge of Modelling Supervisory Systems Using Fuzzy Cognitive Maps*, J. Intel. Manuf. 9 (4), 1998, pp. 339–345.
- [15] Stylios, C.D., Groumpos, P.P., *Fuzzy Cognitive Maps: a Model for Intelligent Supervisory Control Systems*, Comput. Ind. 39 (3), 1999, pp. 229–238.
- [16] Stylios, C.D., Groumpos, P.P., *Fuzzy Cognitive Map in Modeling Supervisory Control Systems*, J. Intel. & Fuzzy Systems 8 (2), 2000, pp. 83–98.
- [17] Bougon, M.G., "Congregate Cognitive Maps: a Unified Dynamic Theory of Organization and Strategy," Journal of Management Studies, 29: 1992, pp. 369–389.
- [18] Lee, K.C., Lee, W.J., Kwon, O.B., Han, J.H., Yu, P.I., *Strategic Planning Simulation Based on Fuzzy Cognitive Map Knowledge and Differential Game*, Simulation 71 (5), 1998, pp. 316–327.
- [19] Kardaras, D., Mentzas, G., *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.
- [20] Stach, W., Kurgan, L., *Modeling Software Development Project Using Fuzzy Cognitive Maps*, Proc. 4th ASERC Workshop on Quantitative and Soft Software Engineering (QSSE'04), 2004, pp. 55–60.
- [21] Stach, W., Kurgan, L., Pedrycz, W., Reformat, M., *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.
- [22] Montazemi, A.R., Conrath, D.W., "The Use of Cognitive Mapping for Information Requirements Analysis," MIS Quarterly, 10: 1986, pp. 44–55.
- [23] Gotoh, K., Murakami, J., Yamaguchi, T., Yamanaka, Y., *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.
- [24] 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, 1999.
- [25] Carvalho, J.P., Tomé, J.A., "Rule Based Fuzzy Cognitive Maps- Fuzzy Causal Relations", Computational Intelligence for Modeling, Control and Automation, Edited by M. Mohammadian, 1999.
- [26] Carvalho, J.P., Tomé, J.A., "Fuzzy Mechanisms for Causal Relations", Proceedings of the Eighth International Fuzzy Systems Association World Congress, IFSA'99, Taiwan, 1999.
- [27] 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, 2000.
- [28] Koulouriotis, D.E., Diakoulakis, I.E., Emiris, D.M., Antonidakis, E.N., Kaliakatsos, I.A., *Efficiently Modeling and Controlling Complex Dynamic Systems Using Evolutionary Fuzzy Cognitive Maps (Invited Paper)*, Internet. J. Comput. Cognition 1 (2), 2003, pp. 41–65.
- [29] Stylios, C.D., Groumpos, P.P., *Modeling Complex Systems using Fuzzy Cognitive Maps*, IEEE Trans. Systems Man, Cybern. Part A: Systems Humans 34 (1), 2004.
- [30] 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 Behavior*, IEEE Congr. On Evolutionary Computation (CEC2001), 2001, pp. 364–371.
- [31] Koulouriotis, D.E., Diakoulakis, D.E., Emiris, D.M., *Anamorphous 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.
- [32] Kosko, B., *Hidden Patterns in Combined and Adaptive Knowledge Networks*, Internet. J. Approx. Reason. 2, 1988, pp. 377–393.
- [33] Khan, M., Quaddus, M., Group decision support using fuzzy cognitive maps for causal reasoning, Group Decision Negotiation J. 13 (5), 2004, pp. 463–480.
- [34] Stach, W., Lukasz Kurgan, Witold Pedrycz, Marek Reformat *Genetic learning off uzzy cognitive maps*, Fuzzy Sets and Systems 153, 2005, pp. 371–401.
- [35] Lee, C.C., *Fuzzy Logic in Control Systems: Fuzzy Logic Controller*, Part 1,2 IEEE Trans. Syst.Man Cybernet, 20(2), 2001, pp. 404-435.
- [36] Kun Chang Lee, Jin Sung Kim, Nam Ho Chung, Soon Jae Kwon, *Fuzzy Cognitive Map Approach to Web Mining Inference Amplification*, Expert System with Applications 22, 2002, pp. 197-211.
- [37] Schneider, M., Shnaider, E., Kandel, A., Chew, G., *Automatic construction of FCMs*, Fuzzy Sets and Systems 93 (2), 1998, pp. 161–172.
- [38] Kardaras, D., Karakostas, B., "The use of Fuzzy Cognitive Maps to Simulate the Information Systems Strategic Planning Process", Information and Software Technology 41, 1999, pp. 197–210.
- [39] Dickerson, J.A., Kosko, B., *Fuzzy virtual worlds*, Artif. Intel. Expert 7, 1994, pp. 25–31.
- [40] Vazquez, A., *A Balanced Differential Learning Algorithm in Fuzzy Cognitive Maps*, Technical Report, Department de Llenguatges I Sistemes Informatics, Universitat Politecnica de Catalunya (UPC), 2002.
- [41] Papageorgiou, E., Stylios, C.D., Groumpos, P.P., *Fuzzy Cognitive Map Learning Based on Nonlinear Hebbian Rule*, Australian Conf. on Artificial Intelligence, 2003, pp. 256–268.
- [42] Papageorgiou, E., Stylios, C.D., Groumpos, P.P., *Active Hebbian Learning Algorithm to Train Fuzzy Cognitive Maps*, Internat. J. Approx. Reason. 37 (3), 2004, pp. 219–249.
- [43] Parsopoulos, K.E., Papageorgiou, E.I., Groumpos, P.P., Vrahatis, M.N., *A First Study of Fuzzy Cognitive Maps Learning using Particle Swarm Optimization*, Proc. IEEE 2003 Congr. on Evolutionary Computation, 2003, pp. 1440–1447.
- [44] Papageorgiou, E.I., Parsopoulos, K.E., Stylios, C.D., Groumpos, P.P., Vrahatis, M.N., *Fuzzy Cognitive Maps Learning using Particle Swarm Optimization*, J. Intel. Inform. Systems, in press, 2002.
- [45] Khan, M., Chong, A., *Fuzzy Cognitive Map Analysis with Genetic Algorithm*, Proc. 1st Indian Internat. Conf. on Artificial Intelligence (IICAI-03), 2003.
- [46] Papageorgiou, E., Chrysostomos Stylios, Peter P. Groumpos, *Unsupervised Learning Techniques for Fine-Tuning Fuzzy Cognitive Map Causal Links*, International Journal of Human-Computer Studies ,Volume 64 , Issue 8, August 2006, pp: 727-743.
- [47] Amit Konar, Uday K., Chakraborty , *Reasoning and Unsupervised Learning in a Fuzzy Cognitive Map*, Information Sciences 170, 2005, pp. 419–441.
- [48] Ghazanfari, M., Alizadeh, S., Fathian, M., Koulouriotis, D.E., *Comparing Simulated Annealing and Genetic Algorithm in Learning FCM*, Applied Mathematics and Computation (2007), doi:10.1016/j.amc.2007.02.144.
- [49] Marc Pirlot, *General local search methods*, European journal of operational research 92, 1996, pp. 493-511.
- [50] Duc Truong Pham, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*, 2003.
- [51] van Laarhoven, P., Aarts, E., *Simulated Annealing: Theory and Applications*. Dordrect: Reidel, 1987.