Fuzzy Cognitive Maps Learning through Swarm Intelligence

E.I. Papageorgiou[1,3], K.E. Parsopoulos[2,3], P.P. Groumpos[1,3], and M.N. Vrahatis[2,3]*

[1] Department of Electrical and Computer Engineering, University of Patras, GR–26500 Patras, Greece, {epapageo, groumpos}@ee.upatras.gr
[2] Department of Mathematics, University of Patras, GR–26110 Patras, Greece, {kostasp, vrahatis}@math.upatras.gr
[3] University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR–26110 Patras, Greece

Abstract. A technique for Fuzzy Cognitive Maps learning, which is based on the minimization of a properly defined objective function using the Particle Swarm Optimization algorithm, is presented. The workings of the technique are illustrated on an industrial process control problem. The obtained results support the claim that swarm intelligence algorithms can be a valuable tool for Fuzzy Cognitive Maps learning, alleviating deficiencies of Fuzzy Cognitive Maps, and controlling the system's convergence.

1 Introduction

Fuzzy Cognitive Maps (FCMs) constitute a promising modeling methodology that provides flexibility on the simulated system's design, modeling and control. They were introduced by Kosko for the representation of causal relationships among concepts as well as for the analysis of inference patterns [1,2]. Up–to–date, FCMs have been applied in various scientific fields, including bioinformatics, manufacturing, organization behavior, political science, and decision making. Although FCMs constitute a promising modeling methodology, they have some deficiencies regarding the robustness of their inference mechanism and their ability to adapt the experts' knowledge through optimization and learning [1, 2]. These properties are crucial in several applications. Therefore, FCMs need further enhancement, stronger mathematical justification, and improvement of their operation. This can be attained through the development of new learning algorithms that alleviate the deficiencies and improve the performance of FCMs.

In this paper, an approach for FCMs learning, based on a swarm intelligence algorithm, is presented. In particular, the Particle Swarm Optimization (PSO) method is applied to determine an appropriate configuration of the FCM's weights, through the minimization of a properly defined objective function [3]. The technique is applied on a process control problem, with promising results.

* Partially supported by the "Pythagoras" research grant awarded by the Greek Ministry of Education and Religious Affairs as well as the European Union.

L. Rutkowski et al. (Eds.): ICAISC 2004, LNAI 3070, pp. 344–349, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The paper is organized as follows: the PSO algorithm is briefly presented in Section 2, while the basic principles of FCMs as well as the learning procedure are described in Section 3. In Section 4 the process control problem is described and the experimental results are reported and discussed. The paper concludes in Section 5.

2 The Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a population–based stochastic optimization algorithm. It belongs to the class of *swarm intelligence* algorithms, which are inspired from and based on the social dynamics and emergent behavior that arise in socially organized colonies [4,5]. In the context of PSO, the population is called a *swarm* and the individuals (search points) are called *particles*.

Assume a D–dimensional search space, $S \subset \mathbb{R}^D$, and a swarm consisting of N particles. The i–th particle is in effect a D–dimensional vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^\top \in S$. The velocity of this particle is also a D–dimensional vector, $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^\top \in S$. The best previous position encountered by the i–th particle is a point in S, denoted by $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^\top$. Assume g_i to be the index of the particle that attained either the best position of the whole swarm (global version) or the best position in the neighborhood of the i–th particle (local version). Then, the swarm is manipulated by the equations [6]:

$$V_i(t+1) = \chi \Big[V_i(t) + c_1\, r_1 \big(P_i(t) - X_i(t) \big) + \; c_2\, r_2 \big(P_{g_i}(t) - X_i(t) \big) \Big], \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{2}$$

where $i = 1, \ldots, N$; χ is a parameter called *constriction factor*; c_1 and c_2 are two parameters called *cognitive* and *social* parameters respectively; and r_1, r_2, are random numbers uniformly distributed within $[0, 1]$. The value of the constriction factor can be derived analytically [6]. The initialization of the swarm and the velocities, is usually performed randomly and uniformly in the search space.

3 Fuzzy Cognitive Maps Learning

FCMs combine properties of fuzzy logic and neural networks. An FCM models the behavior of a system by using concepts, C_i, $i = 1, ..., N$, that represent the states, variables or characteristics of the system. The system is then represented by a fuzzy signed directed graph with feedback, which contains nodes–concepts and weighted edges that connect the nodes and represent the cause and effect relations among them. The values, A_i, of the concepts lie within $[0, 1]$ and they are susceptible to change over time. The weights, W_{ij}, of the edges assume values in $[-1, 1]$, and represent the extent of the impact of the interconnected concepts on each other. The design of an FCM is a process that heavily relies on the input from a group of experts [7] and results in an initial weight matrix, $W^{\text{initial}} = [W_{ij}]$, with $W_{ii} = 0$, $i, j = 1, \ldots, N$. After the determination of its structure, the FCM is let to converge to a steady state by applying the rule,

$A_i(t+1) = f\left(A_i(t) + \sum_{\substack{j=1 \\ j\neq i}}^{n} W_{ji}A_j(t)\right)$, with arbitrary initial values of A_i [2], where t stands for the time counter. The function f is the threshold function, $f(x) = 1/(1+e^{-\lambda x})$, where $\lambda > 0$ is a parameter that determines its steepness. In the present study the value of λ was set to 1. A steady state of the FCM is characterized by concept values that are not further modified through the application of the aforementioned rule. After this stage, the FCM can simulate the system accurately. The heavy dependence on the experts' opinion regarding the FCM's design; the convergence to undesired steady states starting from the experts recommendations; as well as the need for specific initial values of the concepts, are significant weaknesses of FCMs, which can be addressed through learning procedures. Up–to–date, a few learning algorithms have been proposed [8,9], and they are mostly based on ideas coming from the field of neural network training. Recently, a new technique for FCMs learning, which is based on the minimization of a properly defined objective function using the Particle Swarm Optimization algorithm, has been developed [3]. For completeness purposes, this technique is outlined in the rest of this section.

The main goal of learning in FCMs is to determine the values of the weights of the FCM that produce a desired behavior of the system. The desired behavior of the system is characterized by values of the output concepts that lie within prespecified bounds, determined by the experts. These bounds are in general problem dependent. Let $C_{\mathrm{out}_1}, \ldots, C_{\mathrm{out}_m}$, $m \in \{1, 2, \ldots, N\}$, be the output concepts of the FCM, while the remaining concepts are considered input or interior concepts. The user is interested in restricting the values of these output concepts in strict bounds, $A_{\mathrm{out}_i}^{\min} \leqslant A_{\mathrm{out}_i} \leqslant A_{\mathrm{out}_i}^{\max}$, $i = 1, \ldots, m$, which are crucial for the proper operation of the modeled system. Thus, the main goal is to detect a weight matrix, $W = [W_{ij}]$, $i, j = 1, \ldots, N$, that leads the FCM to a steady state at which, the output concepts lie in their corresponding bounds, while the weights retain their physical meaning. The latter is attained by imposing constraints on the potential values assumed by weights. To do this, the following objective function is considered [3]:

$$F(W) = \sum_{i=1}^{m} H\left(Q_{\mathrm{out}_i}^{\min}\right)\left|Q_{\mathrm{out}_i}^{\min}\right| + \sum_{i=1}^{m} H\left(Q_{\mathrm{out}_i}^{\max}\right)\left|Q_{\mathrm{out}_i}^{\max}\right|, \tag{3}$$

where $Q_{\mathrm{out}_i}^{\min} = A_{\mathrm{out}_i}^{\min} - A_{\mathrm{out}_i}$; $Q_{\mathrm{out}_i}^{\max} = A_{\mathrm{out}_i} - A_{\mathrm{out}_i}^{\max}$; H is the well–known Heaviside function, i.e. $H(x) = 0$, if $x < 0$, and $H(x) = 1$ otherwise; and A_{out_i}, $i = 1, \ldots, m$, are the steady state values of the output concepts that are obtained using the weight matrix W. Obviously, the global minimizers of the objective function F are weight matrices that lead the FCM to a desired steady state. An FCM with N fully interconnected concepts, corresponds to an $N(N-1)$–dimensional minimization problem [3].

The application of PSO for the minimization of the objective function F, starts with an initialization phase, where a swarm of weight matrices is generated randomly, and it is evaluated using F. Then, (1) and (2) are used to evolve the swarm. When a weight configuration that globally minimizes F is reached, the algorithm stops. There is, in general, a plethora of weight matrices for which the FCM converges to the desired regions of the output concepts. PSO is a stochastic

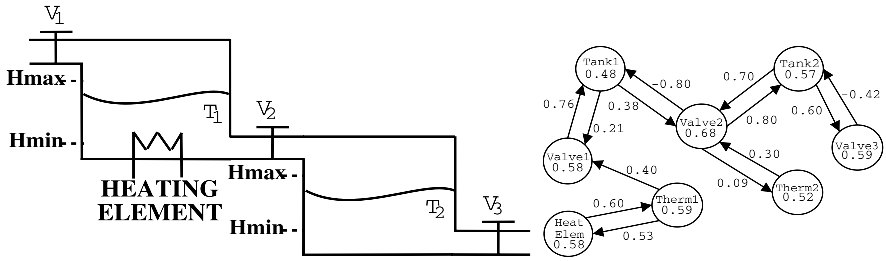

Fig. 1. The process control problem (left) and the corresponding FCM (right).

algorithm, and, thus, it is quite natural to obtain such suboptimal matrices that differ in subsequent experiments. The approach has proved to be very efficient in practice [3]. In the following section, its operation on an industrial process control problem, is discussed.

4 An Application to a Process Control Problem

The learning algorithm previously described, is applied on a complex industrial process control problem [7]. This problem consists of two tanks, three valves, one heating element and two thermometers for each tank, as depicted in Fig. 1. Each tank has an inlet valve and an outlet valve. The outlet valve of the first tank is the inlet valve of the second. The objective of the control system is to keep the height as well as the temperature of the liquid in both tanks, within prespecified bounds. The temperature, T^1, of the liquid in tank 1, is regulated by a heating element. The temperature, T^2, of the liquid in tank 2, is measured using a thermometer; if T^2 is decreased, then valve V^2 opens, and hot liquid from tank 1 is pured into tank 2. Thus, the main objective is to ensure that the relations $H^1_{\min} \leqslant H^1 \leqslant H^1_{\max}$, $T^1_{\min} \leqslant T^1 \leqslant T^1_{\max}$, $H^2_{\min} \leqslant H^2 \leqslant H^2_{\max}$, $T^2_{\min} \leqslant T^2 \leqslant T^2_{\max}$, hold, where H^1 and H^2 denote the height of the liquid in tank 1 and tank 2, respectively. An FCM that models this system has been developed in [7] and depicted in Fig. 1. The output concepts are C_1, C_2, C_6 and C_7. The sign and the weight of each interconnection have been determined by three experts [7]. All the experts agreed regarding the direction of the interconnections among the concepts, and they determined the overall linguistic variable and the corresponding fuzzy set for each weight. The final ranges for the weights, as implied by the fuzzy regions, are: $0.00 \leqslant W_{13} \leqslant 0.50$, $0.00 \leqslant W_{14} \leqslant 0.75$, $0.00 \leqslant W_{24} \leqslant 0.90$, $0.00 \leqslant W_{25} \leqslant 1.00$, $0.50 \leqslant W_{31} \leqslant 1.00$, $-1.0 \leqslant W_{41} \leqslant -0.25$, $0.25 \leqslant W_{42} \leqslant 1.00$, $-0.50 \leqslant W_{47} \leqslant 0.50$, $-0.75 \leqslant W_{52} \leqslant 0.75$, $0.00 \leqslant W_{63} \leqslant 0.75$, $0.25 \leqslant W_{68} \leqslant 0.75$, $0.00 \leqslant W_{74} \leqslant 0.60$, $0.00 \leqslant W_{86} \leqslant 0.90$, and the initial weights, derived through the CoA defuzzification method, are $W^{\text{initial}} = [0.21, 0.38, 0.70, 0.6, 0.76, -0.80, 0.80, 0.09, -0.42, 0.4, 0.53, 0.30, 0.60]$.

Two different scenarios have been considered to investigate the performance of our approach on the process control problem. For each scenario, 100 independent experiments have been performed using the global variant of a constriction

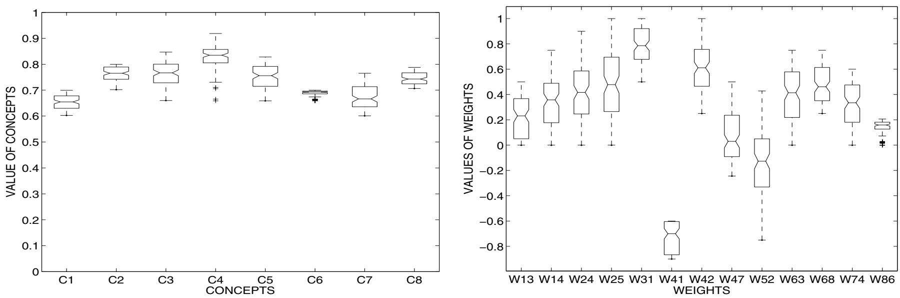

Fig. 2. Boxplots for the first scenario.

factor PSO. The swarm size was set to 5. The constriction factor as well as the cognitive and the social parameter have been set to their default values, $\chi = 0.729$, $c_1 = c_2 = 2.05$ [6].

The first scenario considers the constrained weights, and the following desired values for the four output concepts: $0.5 \leqslant C_1 \leqslant 0.7$, $0.7 \leqslant C_2 \leqslant 0.8$, $0.6 \leqslant C_6 \leqslant 0.7$, $0.6 \leqslant C_7 \leqslant 0.8$. The convergence regions of the concepts and weights are depicted in the boxplots of Fig. 2. A suboptimal weight vector is $W = [0.01, 0.36, 0.41, 0.82, 0.50, -0.60, 0.29, 0.39, 0.42, 0.22, 0.36, 0.11, 0.18]$, and the corresponding values of the output concepts are $C_1 = 0.62$, $C_2 = 0.79$, $C_6 = 0.69$, $C_7 = 0.74$. Comparing the derived convergence regions of the weights with the bounds provided by the experts, it can be observed that three weights, namely W_{47}, W_{52}, and W_{86} take values in ranges significantly smaller than their bounds. This can serve as an indication that the experts determined relatively wide initial bounds. The values of the remaining weights lie in their bounding regions. The mean number of required function evaluations was 32.85.

In the second scenario, the desired values for the output concepts are different: $0.5 \leqslant C_1 \leqslant 0.7$, $0.7 \leqslant C_2 \leqslant 0.8$, $0.73 \leqslant C_6 \leqslant 0.81$, $0.65 \leqslant C_7 \leqslant 0.75$. The convergence regions of the concepts and weights are depicted in the boxplots of Fig. 3. A suboptimal weight vector is $W = [0.21, 0.49, 0.01, 0.04, 0.51, -0.89, 0.61, 0.09, -0.40, 0.09, 0.29, 0.01, 0.84]$, and the corresponding values of the output concepts are $C_1 = 0.56$, $C_2 = 0.71$, $C_6 = 0.80$, $C_7 = 0.67$. Again, the weights W_{47}, W_{52}, and W_{86} assume values in ranges significantly smaller than their bounds, while the values of the remaining weights lie in their bounding regions. The mean number of required function evaluations was 15.35.

It is clear that the learning algorithm is capable of providing proper weight matrices for the FCM, efficiently and effectively. Moreover, the statistical analysis through the boxplots provides indications regarding the quality of the weights' bounds determined by the experts, which can be used in the future as a mechanism for the evaluation of the experts by taking into consideration the deviation of their suggestions from the obtained values.

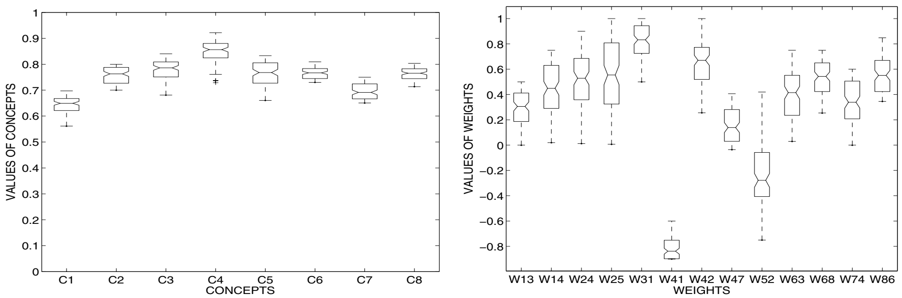

Fig. 3. Boxplots for the second scenario.

5 Conclusions

A methodology for determining the cause–effect relationships (weights) among the concepts of Fuzzy Cognitive Maps, has been presented. This approach is based on the minimization of a properly defined objective function through the Particle Swarm Optimization algorithm. A complex process control problem has been used to illustrate the algorithm. The results are very promising, verifying the effectiveness of the learning procedure. Moreover, the physical meaning of the obtained results is retained. This approach can provide a robust solution in the case of divergent opinions of the experts, and it will be considered, in a future work, as means for the evaluation of the experts.

References

1. Kosko, B.: Fuzzy cognitive maps. Int. J. Man–Machine Studies **24** (1986) 65–75
2. Kosko, B.: Fuzzy Engineering. Prentice Hall, New York (1997)
3. Parsopoulos, K.E., Papageorgiou, E.I., Groumpos, P.P., Vrahatis, M.N.: A first study of fuzzy cognitive maps learning using particle swarm optimization. In: Proc. IEEE 2003 Cong. Evol. Comput., (2003) 1440–1447
4. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing **1** (2002) 235–306
5. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (2001)
6. Clerc, M., Kennedy, J.: The particle swarm–explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6** (2002) 58–73
7. Stylios, C.D., Georgopoulos, V., Groumpos, P.P.: Fuzzy cognitive map approach to process control systems. J. Adv. Comp. Intell. **3** (1999) 409–417
8. Aguilar, J.: Adaptive random fuzzy cognitive maps. Lecture Notes in Computer Science., Vol. 2527, Springer (2002) 402–410
9. Papageorgiou, E.I., Stylios, C.D., Groumpos, P.P.: Active hebbian learning algorithm to train FCMs. Int. J. Approx. Reas. (2004) accepted for publication.