Journal of Intelligent Information Systems, 25:1, 95–121, 2005

Fuzzy Cognitive Maps Learning Using Particle Swarm Optimization

ELPINIKI I. PAPAGEORGIOU* epapageo@ee.upatras.gr
Department of Electrical and Computer Engineering, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26500 Patras, Greece

KONSTANTINOS E. PARSOPOULOS kostasp@math.upatras.gr
Computational Intelligence Laboratory (CI Lab), Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece

CHRYSOSTOMOS S. STYLIOS stylios@teleinfom.teiep.gr
Department of Communications, Informatics and Management, TEI of Epirus, Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-47100, Artas, Greece

PETROS P. GROUMPOS groumpos@ee.upatras.gr
Department of Electrical and Computer Engineering, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26500 Patras, Greece

MICHAEL N. VRAHATIS vrahatis@math.upatras.gr
Computational Intelligence Laboratory (CI Lab), Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece

Received April 22, 2003; Accepted August 4, 2004

Abstract. This paper introduces a new learning algorithm for Fuzzy Cognitive Maps, which is based on the application of a swarm intelligence algorithm, namely Particle Swarm Optimization. The proposed approach is applied to detect weight matrices that lead the Fuzzy Cognitive Map to desired steady states, thereby refining the initial weight approximation provided by the experts. This is performed through the minimization of a properly defined objective function. This novel method overcomes some deficiencies of other learning algorithms and, thus, improves the efficiency and robustness of Fuzzy Cognitive Maps. The operation of the new method is illustrated on an industrial process control problem, and the obtained simulation results support the claim that it is robust and efficient.

Keywords: Fuzzy Cognitive Maps, Particle Swarm Optimization, swarm intelligence, soft computing

1. Introduction

Fuzzy Cognitive Maps (FCMs) constitute a modeling methodology that combines fuzzy logic and neural networks (Kosko, 1997). They were developed by Kosko as an expansion

*Author to whom all correspondence should be addressed.

of cognitive maps (Axelrod, 1976; Kosko, 1986), and they belong to the class of neuro–fuzzy systems, which are able to incorporate human knowledge and adapt it through learning procedures. FCMs are designed by experts through an interactive procedure of knowledge acquisition (Hagiwara, 1992; Stylios and Groumpos, 2000).

FCMs have a wide field of application, including modeling of complex and intelligent systems (Jang et al., 1997), decision analysis (Khan et al., 1999), and graph behavior analysis (Hagiwara, 1992). They have also been used for planning and decision–making in the fields of international relations and social systems modeling (Taber, 1991,1994), as well as in management science, operations research and organizational behavior (Craiger and Coovert, 1994). Dickerson and Kosko have used FCMs to construct virtual worlds (Dickerson and Kosko, 1994). Furthermore, FCMs have been proposed for modeling supervisory systems (Groumpos and Stylios, 2000; Papageorgiou et al., 2004a), decision-making in radiation therapy planning systems (Papageorgiou et al., 2003b; Parsopoulos et al., 2004), as well as for grading urinary bladder and tumor characterization (Papageorgiou et al., 2004b, 2004c).

The wide recognition of FCMs as a promising modeling and simulation methodology for complex systems, characterized by abstraction, flexibility and fuzzy reasoning, promoted the research on new concepts and learning algorithms for FCMs. However, the existing learning algorithms for FCMs still require enhancement, stronger mathematical justification, and further testing on systems of higher complexity. Moreover, the elimination of deficiencies, such as the abstract estimation of the initial weight matrix, as well as the development of techniques that could further refine the experts' knowledge, will significantly improve the functionality and applicability of FCMs. In this context, the development of learning algorithms is a stimulating research topic.

Up-to-date, a few algorithms have been proposed for FCM learning (Kosko, 1997; Papageorgiou et al., 2004b, 2004c; Papageorgiou and Groumpos, 2004). The main task is to find proper values of the FCM's weights that lead the FCM to a desired steady state. This is achieved through the minimization of a properly defined objective function. Established algorithms are mainly dependent on the initial weight matrix approximation, which is provided by the experts. Recently, a different approach has been proposed for FCM learning, which is based on the application of Evolution Strategies for the computation of proper weight matrices (Koulouriotis et al., 2001).

This paper proposes a new approach for FCM learning, which is based on a swarm intelligence algorithm. More specifically, the Particle Swarm Optimization (PSO) method is used for the determination of proper weight matrices for the system through the minimization of a properly defined objective function. PSO is selected due to its efficiency and effectiveness on a plethora of applications in science and engineering, as well as its straightforward applicability (Abido, 2002; Agrafiotis and Cedeno, 2002; Cockshott and Hartman, 2001; Fourie and Groenwold, 2002; Kennedy and Eberhart, 2001; Laskari et al., 2002a, 2002b; Lu et al., 2002; Ourique et al., 2002; Parsopoulos et al., 2003, 2004; Papageorgiou et al., 2004a; Parsopoulos and Vrahatis, 2002b, 2002c, 2003, 2004; Ray and Liew, 2002; Saldam et al., 2002; Tandon et al., 2002). The proposed approach is illustrated on an industrial process control problem, with promising results.

The rest of the paper is organized as follows: in Section 2 the main principles underlying FCMs are described. In Section 3, the PSO algorithm is briefly presented; Section 4 is devoted to the description and analysis of the proposed learning algorithm. The process control problem, on which the proposed algorithm is tested, is described in Section 5, while the obtained results are reported and discussed in Section 6. Section 7 closes the paper, with conclusions and ideas for future research.

2. Description of fuzzy cognitive maps

FCMs have been introduced by Kosko (1986) as signed directed graphs for representing causal reasoning and computational inference processing. FCMs exploit a symbolic representation for the description and modeling of a system. Concepts are utilized to represent different aspects of the system, as well as, their behavior. The dynamics of the system are simulated by the interaction of concepts. FCMs are used to represent both qualitative and quantitative data. The construction of an FCM requires the input of human experience and knowledge on the system under consideration. Thus, FCMs integrate the accumulated experience and knowledge concerning the underlying causal relationships amongst factors, characteristics, and components that constitute the system.

An FCM consists of nodes-concepts,

$$C_i, \quad i = 1, \ldots, N,$$

where N is the total number of concepts. Each node–concept represents one of the key-factors of the system and it is characterized by a value,

$$A_i \in [0, 1], \quad i = 1, \ldots, N.$$

The concepts are interconnected through weighted arcs, which imply the relations among them. A simple FCM with five nodes and ten weighted arcs is illustrated in figure 1. Each interconnection between two concepts, C_i and C_j, has a weight, W_{ij}, which is analogous

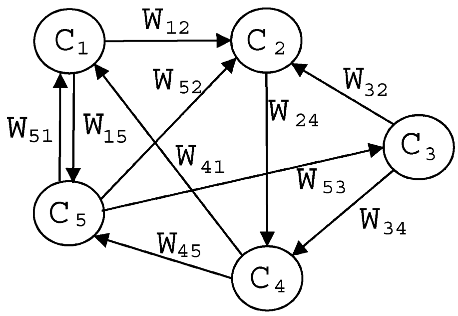

Figure 1. A simple Fuzzy Cognitive Map.

to the strength of the causal link between C_i and C_j. The sign of W_{ij} indicates whether the relation between the 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,

$$\begin{cases} W_{ij} > 0, & \text{expresses positive causality,} \\ W_{ij} < 0, & \text{expresses negative causality,} \\ W_{ij} = 0, & \text{expresses no relation.} \end{cases}$$

Human knowledge and experience on the system determines the type and number of nodes, as well as the initial weights of the FCM. The value, A_i, of a concept, C_i, expresses the quantity of its corresponding physical value and it is derived by the transformation of the fuzzy values assigned by the experts to numerical values. Having assigned values to the concepts and weights, the FCM converges to a steady state through the interaction process subsequently described. At each step, the value A_i of a concept is influenced by the values of concepts–nodes connected to it and it is updated according to the scheme (Kosko, 1997),

$$A_i(k+1) = f\left(A_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^{n} W_{ji} A_j(k)\right), \tag{1}$$

where k stands for the iteration counter; and W_{ji} is the weight of the arc connecting concept C_j to concept C_i. The function f is the sigmoid function,

$$f(x) = \frac{1}{1 + e^{-\lambda x}}, \tag{2}$$

where $\lambda > 0$ is a parameter that determines its steepness in the area around zero. In our approach, the value $\lambda = 1$ has been used. This function is chosen since the values A_i of the concepts, by definition, must lie within [0, 1]. The interaction of the FCM results in a steady state after a few iterations, i.e., the values of the concepts are not modified further. Desired values of the output concepts of the FCM guarantee the proper operation of the simulated system.

The design of an FCM is a process that heavily relies on the input from experts (Stylios et al., 1999; Stylios and Groumpos, 2000). At the beginning, experts are pooled to determine the relevant factors that will be represented in the map as concepts. Then, they are individually asked to describe the causal relationships among the concepts, using a linguistic notion. First, experts determine the influence of a concept on another as “negative”, “positive” or “no influence”. Then, linguistic weights, such as “strong”, “weak”, etc., are assigned to each arc. The linguistic variables that describe each arc, for each expert, are defined in Cox (1999), and they are characterized by the fuzzy sets whose membership functions are shown in figure 2. The linguistic variables are combined, and the aggregated linguistic variable is transformed to a single linguistic weight, through the SUM technique (Lin and Lee, 1996). Finally, the Center of Area (CoA) defuzzification method (Kosko, 1992; Lin and Lee, 1996) is used for the transformation of the linguistic weight to a numerical value within the range

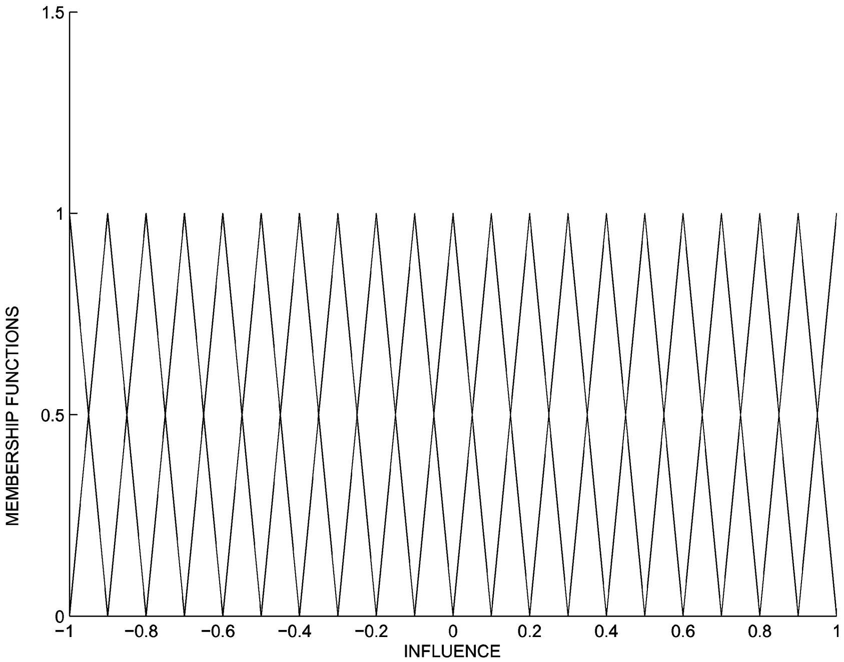

Figure 2. Membership functions for fuzzy weights of FCMs.

$[-1, 1]$. This methodology has the advantage that experts are not required to assign directly numerical values to causality relationships, but rather to describe qualitatively the degree of causality among the concepts. Thus, an initial weight matrix,

$$W^{\text{initial}} = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1N} \\ W_{21} & W_{22} & \cdots & W_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{NN} \end{pmatrix},$$

with $W_{ii} = 0$, $i = 1, \ldots, N$, is obtained. Using the initial concept values, A_i, and the matrix W^{initial}, the FCM interacts through the application of the rule of Eq. (1), and it stops to a potential steady state.

The potential convergence to undesired steady states is a major deficiency of FCMs. Also, techniques that could further refine the experts' knowledge enhance significantly their performance. Learning procedures constitute means to increase the efficiency and robustness of FCMs by updating the weight matrix so as to avoid convergence to undesired steady states. Up-to-date, there are just a few FCM learning algorithms and they are mostly based on ideas coming from the field of artificial neural networks training. Kosko has developed the first algorithm, named Differential Hebbian Learning (Dickerson and Kosko, 1994).

Two different proposed algorithms are the Active Hebbian Learning rule (Papageorgiou et al., 2004d), which derives from the unsupervised Hebbian learning algorithm for neural networks (Hebb, 1949), and the Nonlinear Hebbian Rule (Papageorgiou et al., 2003a). The aforementioned algorithms start from an initial state and an initial weight matrix, W^{initial}, of the FCM, and adapt the weights, in order to compute a weight matrix that leads the FCM to a desired state of the output concepts. The desired state is characterized by values of the FCM's output concepts that are accepted by the experts, *ex post*. The main drawback of this approach is the dependence of the final weights on the initial weight matrix. Wrong initial estimation of the weights or large deviation among the experts' suggestions may lead in reduced efficiency of the algorithms and/or in undesired states of the system.

A novel learning procedure that alleviates the problem of the potential convergence to an undesired steady state is proposed in this paper. This approach is based on a swarm intelligence algorithm, which is briefly presented in the next section.

3. The particle swarm optimization method

Particle Swarm Optimization (PSO) is a stochastic, population-based optimization algorithm. It belongs to the class of *swarm intelligence* algorithms, which are inspired from the social dynamics and emergent behavior that arise in socially organized colonies (Bonabeau et al., 1999; Eberhart et al., 1996; Kennedy and Eberhart, 1995, 2001; Parsopoulos and Vrahatis, 2002c, 2004). Swarm intelligence is related to the field of evolutionary computation, which consists of algorithms inspired by natural evolution and genetic dynamics, such as Genetic Algorithms (Michalewicz, 1994); Genetic Programming (Banzhaf et al., 1998; Koza, 1992); Evolution Strategies (Bäck, 1996; Beyer, 2001; Rechenberg, 1994; Schwefel, 1995); and Evolutionary Programming (Fogel, 1996).

PSO exploits a population, called a *swarm*, of individuals, called *particles*, to probe the search space. Each particle moves with an adaptable velocity within the search space, and retains a memory of the best position it ever encountered. In the *global* variant of PSO, the best position ever attained by all individuals of the swarm is communicated to all the particles. In the *local* variant, each particle is assigned to a neighborhood consisting of prespecified particles. In this case, the best position ever attained by the particles that comprise the neighborhood is communicated among them (Eberhart et al., 1996; Kennedy and Eberhart, 2001).

Assume a D-dimensional search space, $S \subset \mathbb{R}^D$, and a swarm consisting of M 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,

$$V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^\top.$$

The best previous position encountered by the i-th particle is a point in S, denoted by

$$P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^\top \in S.$$

Usually, the particles are considered to lie on a ring topology, i.e., X_M and X_2 are considered the immediate neighbors of X_1. In this case, the neighborhoods of X_i consist of the particles $X_{i-r}, \ldots, X_i, \ldots, X_{i+r}$, where r is the neighborhood's radius. Assume g_i to be the index of the particle that attained the best previous position among all the particles in the neighborhood of X_i, and t to be the iteration counter. Then, the swarm is manipulated by the equations (Clerc and Kennedy, 2002),

$$V_i(t+1) = \chi \left[V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 \big(P_{g_i}(t) - X_i(t) \big) \right], \tag{3}$$
$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{4}$$

where $i = 1, \ldots, M$; χ is a parameter called *constriction factor*; c_1 and c_2 are two parameters called *cognitive* and *social* parameter, respectively; r_1, r_2, are random vectors with elements uniformly distributed within [0, 1]; and g_i is 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). All vector operations in Eqs. (3) and (4) are performed componentwise.

Alternatively, a different version of the algorithm, which incorporates a parameter called *inertia weight*, has been proposed (Eberhart and Shi, 1998; Shi and Eberhart, 1998a, 1998b),

$$V_i(t+1) = w V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 \big(P_{g_i}(t) - X_i(t) \big), \tag{5}$$
$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{6}$$

where w is the inertia weight.

Both the constriction factor and the inertia weight are mechanisms for controlling the magnitude of velocities. However, there are some major differences regarding the way these two are computed and applied. The constriction factor is derived analytically through the formula (Clerc and Kennedy, 2002),

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \tag{7}$$

for $\phi > 4$, where $\phi = c_1 + c_2$, and $\kappa = 1$. Different configurations of χ, as well as a thorough theoretical analysis of the derivation of Eq. (7), can be found in Clerc and Kennedy (2002), Trelea (2003). On the other hand, the inertia weight, w, in Eq. (5), is employed to manipulate the impact of the previous history of velocities on the current velocity. Therefore, w resolves the trade-off between the exploration (wide-ranging search) and exploitation (more refined local search) abilities of the swarm. A large inertia weight encourages exploration, while a small one promotes exploitation. A suitable value for w provides the desired balance in the algorithm and improves its effectiveness. Experimental results suggest that it is preferable to initialize the inertia weight to a large value, giving priority to global exploration of the search space, and gradually decrease it, so as to obtain refined solutions (Shi and Eberhart, 1998a, 1998b). This finding is intuitively very appealing. In conclusion, an initial value of w around 1.0 and a gradual decline towards 0 is considered a proper choice for w.

In general, the constriction factor version of PSO is faster than the one with the inertia weight, although in some applications its global variant suffers premature convergence. Proper fine–tuning of the parameters c_1 and c_2 results in faster convergence and alleviation of local minima (Kennedy, 1998). The default values, $c_1 = c_2 = 2$, have been proposed, but experimental results indicate that alternative configurations, depending on the problem at hand, can produce superior performance (Clerc and Kennedy, 2002; Parsopoulos et al., 2001; Parsopoulos and Vrahatis, 2002c).

The initialization of the swarm and the velocities is usually performed randomly and uniformly in the search space, although more sophisticated initialization techniques can enhance the overall performance of the algorithm (Parsopoulos and Vrahatis, 2002a).

4. The proposed approach

The present work focuses on the development of an FCM learning procedure based on PSO. The purpose is to determine the values of the cause-effect relationships among the concepts, i.e., the values of the weights of the FCM that produce a desired behavior of the system. The determination of the weights is of major significance and it contributes towards the establishment of FCMs as a robust methodology. The desired behavior of the system is characterized by output concept values that lie within desired bounds prespecified by the experts. These bounds are in general problem dependent.

The learning procedure is, to some extent, similar to that of neural networks training. Let

$$C_1, \dots, C_N,$$

be the concepts of an FCM, and let

$$C_{\text{out}_1}, \dots, C_{\text{out}_m}, \quad 1 \leq m \leq N,$$

be the output concepts, 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,

$$A_{\text{out}_i}^{\min} \leq A_{\text{out}_i} \leq A_{\text{out}_i}^{\max}, \quad i = 1, \dots, m,$$

predetermined by the experts, which are crucial for the proper operation of the modeled system. Thus, the main goal is to detect a weight matrix,

$$W = [W_{ij}], \quad i, j = 1, \dots, N,$$

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, we consider the following objective function,

$$F(W) = \sum_{i=1}^{m} H\big(A_{\text{out}_i}^{\min} - A_{\text{out}_i}\big)\big|A_{\text{out}_i}^{\min} - A_{\text{out}_i}\big| + \sum_{i=1}^{m} H\big(A_{\text{out}_i} - A_{\text{out}_i}^{\max}\big)\big|A_{\text{out}_i}^{\max} - A_{\text{out}_i}\big|, \tag{8}$$

where H is the well-known Heaviside function

$$H(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0, \end{cases}$$

and A_{out_i}, $i = 1, \ldots, m$, are the steady state values of the output concepts that are obtained through the application of the procedure of Eq. (1), 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, i.e., all output concepts are bounded within the desired regions. The objective function suits straightforwardly the problem, however, it is non–differentiable and, thus, gradient–based methods are not applicable for its minimization. On the other hand, in the proposed approach, PSO is used for the minimization of the objective function defined by Eq. (8). The non–differentiability of F poses no problems in our approach since PSO, like all evolutionary algorithms, requires function values solely, and can be applied even on discontinuous functions.

The weight matrix W is represented by a vector, which consists of the rows of W in turn, excluding the elements of its main diagonal, $W_{11}, W_{22}, \ldots, W_{NN}$, which are by definition equal to zero,

$$X = \Big[\underbrace{W_{12}, \ldots, W_{1N}}_{\text{row }1}, \underbrace{W_{21}, \ldots, W_{2N}}_{\text{row }2}, \ldots, \underbrace{W_{N1}, \ldots, W_{N,N-1}}_{\text{row }N}\Big].$$

Thus, an FCM with N fully interconnected concepts (i.e., each concept interacts with all other concepts) corresponds to an $N(N-1)$-dimensional minimization problem. If some interconnections are missing, then their corresponding weights are zero and they can be omitted, thereby reducing the dimensionality of the problem. This is most often the case, since the FCMs provided by experts are rarely fully connected.

Each interconnection of an FCM has a specific physical meaning, and, thus, several constraints are posed by the experts on the values of the weights. Constraints are provided in the form of negative or positive relations between two concepts. So, if two concepts C_i and C_j are negatively related, then the weight W_{ij} takes values in the range $[-1, 0]$, while if they are positively related, it takes values within $[0, 1]$. More strict constraints may be additionally posed on some weights, either by the experts, or by taking into consideration the convergence regions obtained through the application of the learning algorithm, as illustrated in Section 6. Such constraints may enhance the overall performance of the algorithm.

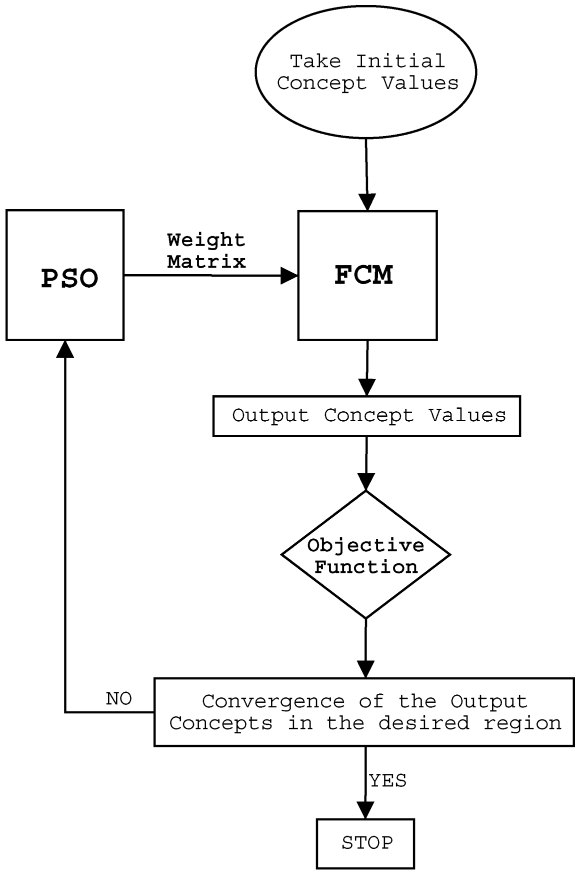

Figure 3. Flowchart of the proposed learning procedure.

The application of PSO for the minimization of the objective function F, starts with an initialization phase, where a swarm of M particles (weight matrices in vector form),

$$\mathbb{S} = \{X_1, \ldots, X_M\},$$

is generated randomly and it is evaluated using F. Then, Eqs. (3) and (4) or, alternatively, (5) and (6) are used to evolve the swarm. As soon as a weight configuration that globally minimizes F is reached, the algorithm is terminated. A flowchart of this procedure is depicted in figure 3.

There is, in general, a plethora of weight matrices that lead to convergence of the FCM to the desired regions of the output concepts. PSO is a stochastic algorithm, and, thus, it is quite natural to obtain such suboptimal matrices that differ in subsequent experiments. All these matrices are proper for the design of the FCM and follow the constraints of the problem, though, each matrix may have different physical meaning for the system. Statistical analysis of the obtained weight matrices may help in the better understanding of the system's dynamic as it is implied by the weights, as well as in the selection of the most appropriate suboptimal matrix.

The aforementioned procedure uses only primitive information from the experts. However, any information available *a priori*, may be incorporated to enhance the procedure, either by modifying the objective function in order to exploit the available information or by imposing further constraints on the weights. The proposed approach has proved to be very efficient in practice. In the following section, its operation is illustrated on an industrial process control problem.

5. Application on an industrial process control problem

A simple process control problem encountered in chemical industry is selected to illustrate the workings of the proposed learning algorithm (Stylios and Groumpos, 1998). The process control problem, illustrated in figure 4, consists of one tank and three valves that influence the amount of a liquid in the tank. Valve 1 and Valve 2 pour two different liquids into the tank. During the mixing of the two liquids, a chemical reaction takes place in the tank, and a new liquid is produced. Valve 3 empties the tank when the amount of the produced liquid reaches a specific level. A sensor is placed inside the tank to measure the specific gravity of the produced liquid. When the value, G, of the specific gravity lies in a range $[G_{\min}, G_{\max}]$, the desired liquid has been produced. There is also a limit on the height, T, of the liquid in the tank, i.e., it cannot exceed a lower limit, $T_{\min}$, and an upper limit, $T_{\max}$. The control target is to keep the two variables, T and G, within their bounds,

$$T_{\min} \leq T \leq T_{\max},$$
$$G_{\min} \leq G \leq G_{\max}.$$

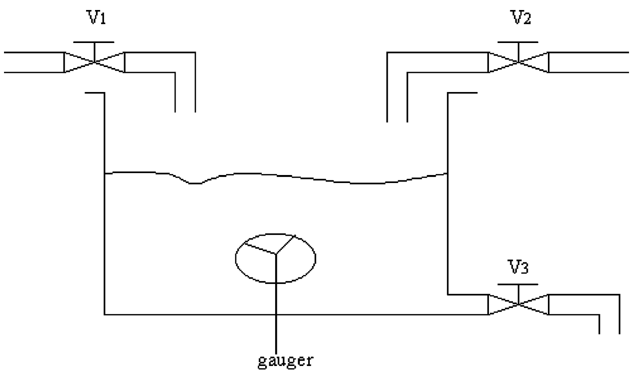

Figure 4. Illustration of a process control problem from industry.

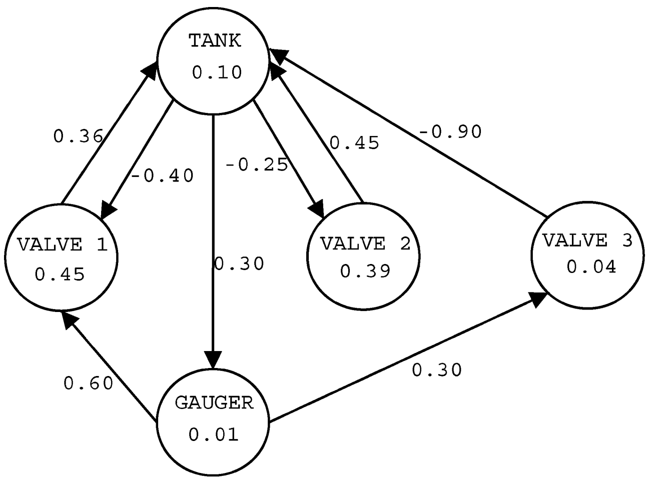

Figure 5. The FCM that corresponds to the problem of figure 4.

A group of experts constructs the FCM for the simulation of this system, following the procedure described in Section 2. The FCM that models and controls the specific system is depicted in figure 5. It consists of five concepts which are defined as:

- *Concept* 1—the amount of the liquid in the tank. It depends on the operational state of Valves 1, 2 and 3;
- *Concept* 2—the state of Valve 1 (closed, open or partially opened);
- *Concept* 3—the state of Valve 2 (closed, open or partially opened);
- *Concept* 4—the state of Valve 3 (closed, open or partially opened);
- *Concept* 5—the specific gravity of the produced liquid in the tank.

There is a consensus among the experts regarding the direction of the arcs among the concepts. For each weight, the overall linguistic variable and its corresponding fuzzy set are also determined by the experts. The ranges of the weights implied by the fuzzy regions are:

$$\begin{aligned} -0.50 &\le W_{12} \le -0.30,\\ -0.40 &\le W_{13} \le -0.20,\\ 0.20 &\le W_{15} \le 0.40,\\ 0.30 &\le W_{21} \le 0.40,\\ 0.40 &\le W_{31} \le 0.50,\\ -1.00 &\le W_{41} \le -0.80,\\ 0.50 &\le W_{52} \le 0.70,\\ 0.20 &\le W_{54} \le 0.40, \end{aligned} \tag{9}$$

and the initial weight matrix derived through the CoA defuzzification method is,

$$W^{\text{initial}} = \begin{pmatrix} 0.00 & -0.40 & -0.25 & 0.00 & 0.30 \\ 0.36 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.45 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.90 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.60 & 0.00 & 0.30 & 0.00 \end{pmatrix}.$$

All experts agreed on the same range for the weights W_{21}, W_{31}, and W_{41}, and most of them agreed on the same range for the weights W_{12} and W_{13}. However, there was no such agreement on the cases of the weights W_{15}, W_{52}, and W_{54}, where their opinions varied significantly.

PSO is applied to update the eight nonzero weight values of the FCM. To avoid physically meaningless weights, the bounds $[-1, 0]$ or $[0, 1]$, implied by the directions of the corresponding arcs of the FCM, are imposed on each weight.

The output concepts for this problem are C_1 and C_5. The desired regions for the two output concepts, which are crucial for the proper operation of the modeled system, have been defined by the experts,

$$0.68 \leq C_1 \leq 0.70, \tag{10}$$
$$0.78 \leq C_5 \leq 0.85. \tag{11}$$

In the next section, the simulation results are reported and analyzed.

6. Simulation results

Two main scenarios have been considered for the simulations. The first scenario investigates the behavior of the system accepting, initially, all the constraints that are imposed on the weights by the experts. The second scenario considers only the constraints for which there was an unanimous agreement amongst the experts. The results are very interesting and provide insight regarding the appropriateness of the experts' suggestions as well as suboptimal weight matrices that lead the FCM to a desired steady state.

For each scenario, 100 independent experiments have been performed using the local variant of the constriction factor PSO version with neighborhood radius equal to 3. This version was selected due to its fast convergence rates and efficiency. Swarm size has been set equal to 20, for all experiments, since it proved sufficient to detect global minimizers of the objective function effectively and efficiently. Moreover, further experiments with larger swarms and different PSO versions did not result in significantly different convergence rates, in terms of the required number of function evaluations. The constriction factor as well as the cognitive and the social parameters have been set to their optimal values, $\chi = 0.729$, $c_1 = c_2 = 2.05$ (Clerc and Kennedy, 2002; Trelea, 2003). The accuracy for the determination of the global minimizer of the objective function has been equal to 10^{-8}.

The mean number of function evaluations required, varied from 40 to 620, depending on the considered scenario.

6.1. First scenario

The first scenario initially considers the weights that lie in the ranges defined by Relation (9), which are derived by the fuzzy regions proposed by the experts. A hundred experiments were performed using the proposed approach and the eight constraints for the weights. However, no solution was detected, indicating that the suggested ranges for the weights, as well as the initial weight matrix, W^{initial}, provided by the experts are not proper and do not lead the FCM to a desired steady state. The best weight matrix detected in these regions, in terms of its objective function value (i.e., the matrix that corresponds to the smallest objective function value) is,

$$W = \begin{pmatrix} 0.00 & -0.35 & -0.20 & 0.00 & 0.40 \\ 0.40 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.80 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.75 & 0.00 & 0.20 & 0.00 \end{pmatrix},$$

which led the FCM to the steady state,

$$C_1 = 0.6723,\ C_2 = 0.7417,\ C_3 = 0.6188,\ C_4 = 0.6997,\ C_5 = 0.7311,$$

that clearly violates the constraints for both C_1 and C_5, defined in Relations (10) and (11).

Since the consideration of all eight constraints on the weights prohibits the detection of a suboptimal matrix, some of the constraints were omitted. Specifically, the constraints for the three weights W_{15}, W_{52}, and W_{54}, for which the experts' suggestions regarding their values varied widely, were omitted, one by one at the beginning, and subsequently in pairs. The corresponding weights were allowed to assume values in the range $[-1, 0]$ or $[0, 1]$, in order to avoid physically meaningless weight matrices. Despite this, no solutions were detected in these cases. However, suboptimal matrices were detected after omitting all three constraints. The statistics of the weights' values for this case are reported in Table 1 and depicted in the boxplot of figure 6. A boxplot is a box and whisker statistical plot. The box has lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers. Notches represent a robust estimate of the uncertainty about the medians for box to box comparison. As shown in the figure, the weights W_{21}, W_{31}, and W_{41}, converged to almost the same value in each experiment; a value which is close to the bounds defined by the experts. The weights W_{13} and W_{54} converged also in very small ranges, while the remaining

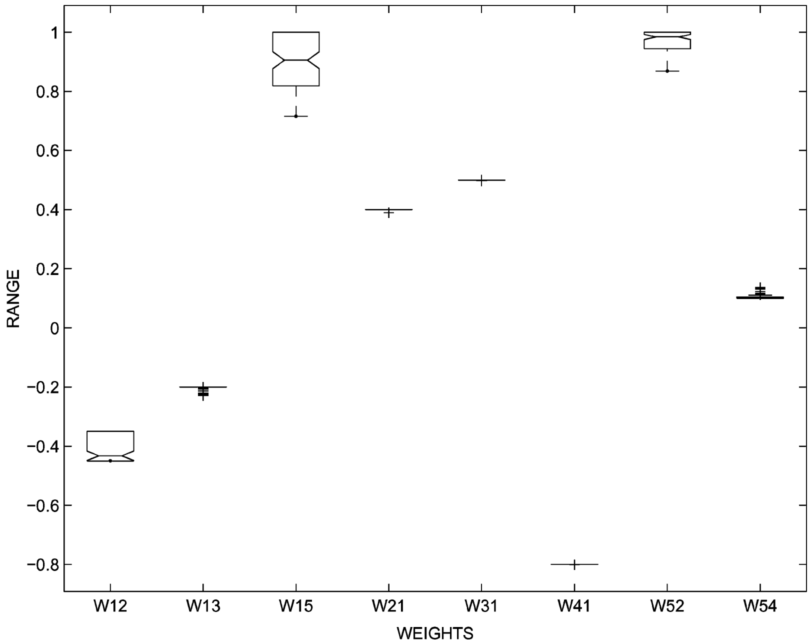

Figure 6. Boxplot of the obtained results for the weights for the first scenario.

weights assumed values in wider regions. Moreover, the three unconstrained weights W_{15}, W_{52}, and W_{54}, converged in regions significantly different than those determined by the experts.

The ranges of the output concepts' values for the obtained suboptimal matrices are depicted in figure 7. The output concept C_1 converges to almost the same value for each suboptimal matrix, while C_5 takes a wide range of values, always within the desired bounds. Regarding the remaining concepts, C_3 and C_4, they converge to almost the same values, while the values of C_2 vary slightly. The obtained values for these three concepts are physically meaningful and appropriate for the operation of the system.

Table 1. Statistical analysis of the results for the first scenario.

	W_{12}	W_{13}	W_{15}	W_{21}	W_{31}	W_{41}	W_{52}	W_{54}
Mean	−0.4027	−0.2016	0.8991	0.3999	0.5000	−0.8000	0.9659	0.1043
Median	−0.4329	−0.2000	0.9050	0.4000	0.5000	−0.8000	0.9837	0.1000
St.dev.	0.0487	0.0056	0.0909	0.0011	0.0003	0.0002	0.0420	0.0090
Min	−0.4500	−0.2291	0.7156	0.3889	0.4971	−0.8014	0.8685	0.1000
Max	−0.3500	−0.2000	1.0000	0.4000	0.5000	−0.8000	1.0000	0.1363

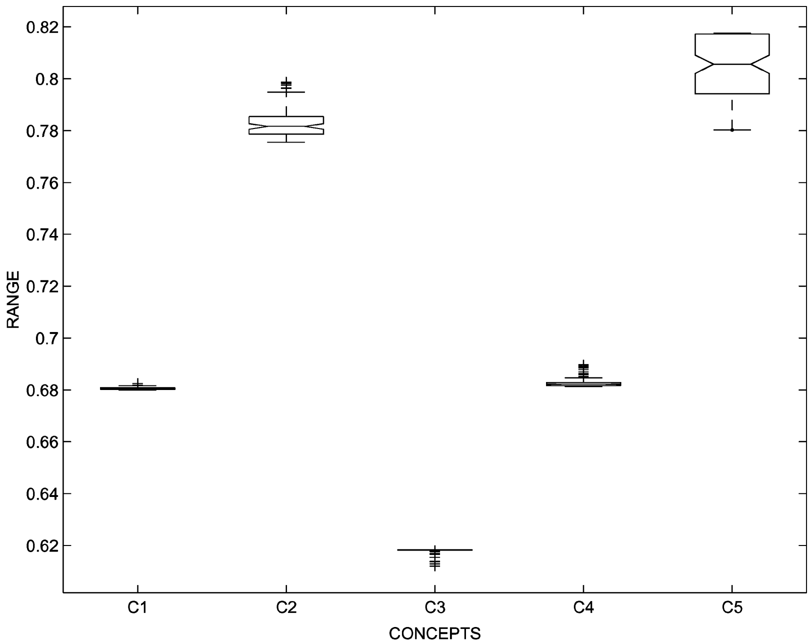

Figure 7. Boxplot of the obtained results for the concepts for the first scenario.

One of the obtained suboptimal matrices is the following,

$$W = \begin{pmatrix} 0.00 & -0.45 & -0.20 & 0.00 & 0.84 \\ 0.40 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.80 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.99 & 0.00 & 0.10 & 0.00 \end{pmatrix},$$

which leads the FCM to the desired steady state,

$$C_1 = 0.6805,\ C_2 = 0.7798,\ C_3 = 0.6176,\ C_4 = 0.6816,\ C_5 = 0.7967.$$

6.2. Second scenario

In this scenario, only the constraints for the three weights W_{21}, W_{31}, and W_{41}, for which all the experts agreed regarding their bounds, are considered. The remaining weights are unconstrained within the ranges $[-1, 0]$ or $[0, 1]$. The statistics of the obtained weight matrices as well as the corresponding boxplot are given in Table 2 and figure 8, respectively.

Table 2. Statistical analysis of the results for the second scenario.

	W_{12}	W_{13}	W_{15}	W_{21}	W_{31}	W_{41}	W_{52}	W_{54}
Mean	−0.2832	−0.1595	0.9198	0.3994	0.4994	−0.8000	0.8915	0.1216
Median	−0.2389	−0.1166	0.9611	0.4000	0.5000	−0.8000	0.9599	0.1000
St.dev.	0.1847	0.0805	0.0917	0.0035	0.0035	0.0005	0.1397	0.0438
Min	−0.6662	−0.3965	0.7143	0.3753	0.4679	−0.8048	0.5196	0.1000
Max	−0.1000	−0.1000	1.0000	0.4000	0.5000	−0.8000	1.0000	0.2770

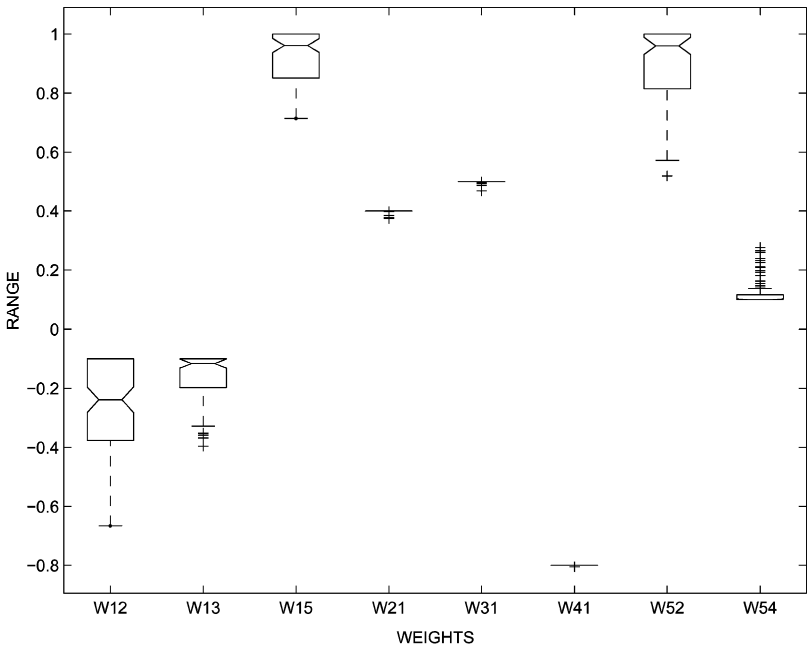

Figure 8. Boxplot of the obtained results for the weights for the second scenario.

It is observed that, the three constrained weights take values in a small subset of their initial bounds, namely,

$$\begin{aligned} 0.38 \leq\ & W_{21} \leq 0.40, \\ 0.47 \leq\ & W_{31} \leq 0.50, \\ -0.81 \leq\ & W_{41} \leq -0.80. \end{aligned} \tag{12}$$

This is an indication that their ranges can be further shortened. The convergence regions of the concepts are depicted in figure 9. Based on the obtained results, three cases are considered.

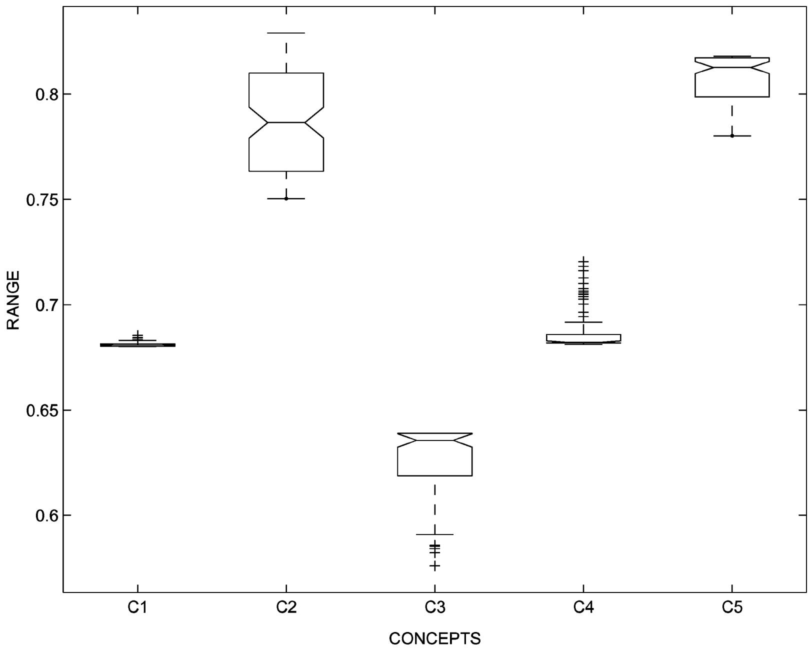

Figure 9. Boxplot of the obtained results for the concepts for the second scenario.

6.2.1. Case A. In this case, the three weights W_{21}, W_{31}, and W_{41}, are constrained in the bounds defined in Relation (13), while the remaining weights are unconstrained within $[-1, 0]$ or $[0, 1]$. The statistics of the obtained results for this case, are reported in Table 3 and a boxplot for the weights is given in figure 10. The corresponding convergence regions of the concepts are depicted in figure 11. We observe that the weights W_{21}, W_{31}, and W_{41}, take almost the same value in each experiment, while the remaining weights lie in the following regions,

$$-0.57 \leq W_{12} \leq -0.10,$$
$$-0.31 \leq W_{13} \leq -0.10,$$

Table 3. Statistical analysis of the results for the Case A.

	W_{12}	W_{13}	W_{15}	W_{21}	W_{31}	W_{41}	W_{52}	W_{54}
Mean	−0.2328	−0.1443	0.9106	0.4000	0.4991	−0.8100	0.8828	0.1132
Median	−0.1952	−0.1006	0.9475	0.4000	0.5000	−0.8100	0.9349	0.1000
St.dev.	0.1408	0.0651	0.0976	0.0000	0.0051	0.0000	0.1295	0.0297
Min	−0.5742	−0.3182	0.7141	0.4000	0.4700	−0.8100	0.6045	0.1000
Max	−0.1000	−0.1000	1.0000	0.4000	0.5000	−0.8100	1.0000	0.2324

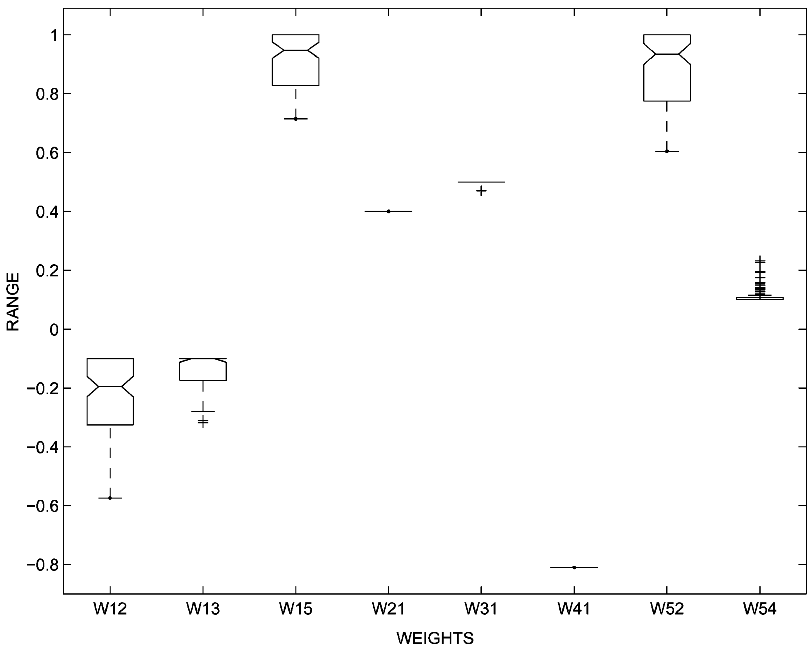

Figure 10. Boxplot of the obtained results for the weights for the Case A.

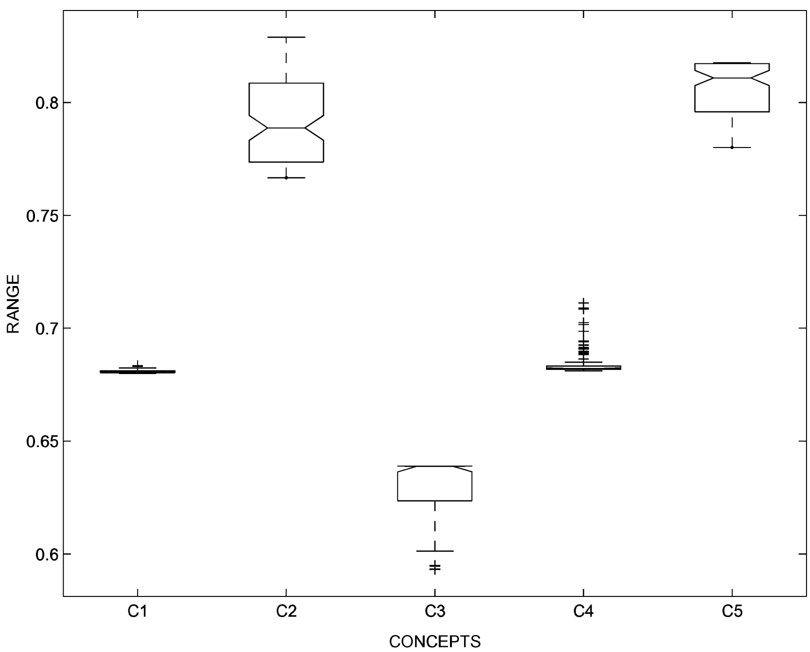

Figure 11. Boxplot of the obtained results for the concepts for the Case A.

$$\begin{aligned} 0.71 \leq\ & W_{15} \ \leq 1.00, \\ 0.60 \leq\ & W_{52} \ \leq 1.00, \\ 0.10 \leq\ & W_{54} \ \leq 0.23. \end{aligned} \tag{13}$$

One of the obtained suboptimal matrices is the following,

$$W = \begin{pmatrix} 0.00 & -0.44 & -0.10 & 0.00 & 1.00 \\ 0.40 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.81 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.13 & 0.00 \end{pmatrix},$$

which leads the FCM to the desired steady state,

$$C_1 = 0.6805,\ C_2 = 0.7872,\ C_3 = 0.6390,\ C_4 = 0.6898,\ C_5 = 0.8172.$$

6.2.2. Case B. Since in the previous case the three weights W_{21}, W_{31}, and W_{41}, assume almost fixed values, we consider them fixed and equal to their mean values reported in Table 3, while the remaining weights are constrained within the ranges determined in Relation (13). The statistics of the obtained results are reported in Table 4. We observe that the weights W_{12}, W_{13}, W_{15}, and W_{54}, take values in the same regions determined by their bounds, while only the weight W_{52} takes values in a smaller region, namely [0.72, 1]. The convergence regions of the concepts are depicted in figure 12. Obviously their convergence regions are small and have an acceptable physical meaning. A suboptimal matrix for this

Table 4. Statistical analysis of the results for the Case B.

	W_{12}	W_{13}	W_{15}	W_{21}	W_{31}	W_{41}	W_{52}	W_{54}
Mean	−0.2366	−0.1349	0.8637	0.4000	0.5000	−0.8100	0.9124	0.1214
Median	−0.2220	−0.1223	0.8674	0.4000	0.5000	−0.8100	0.9223	0.1166
St.dev.	0.0880	0.0375	0.0805	0.0000	0.0000	0.0000	0.0695	0.0232
Min	−0.5700	−0.3040	0.7147	0.4000	0.5000	−0.8100	0.7226	0.1000
Max	−0.1000	−0.1000	1.0000	0.4000	0.5000	−0.8100	1.0000	0.2050

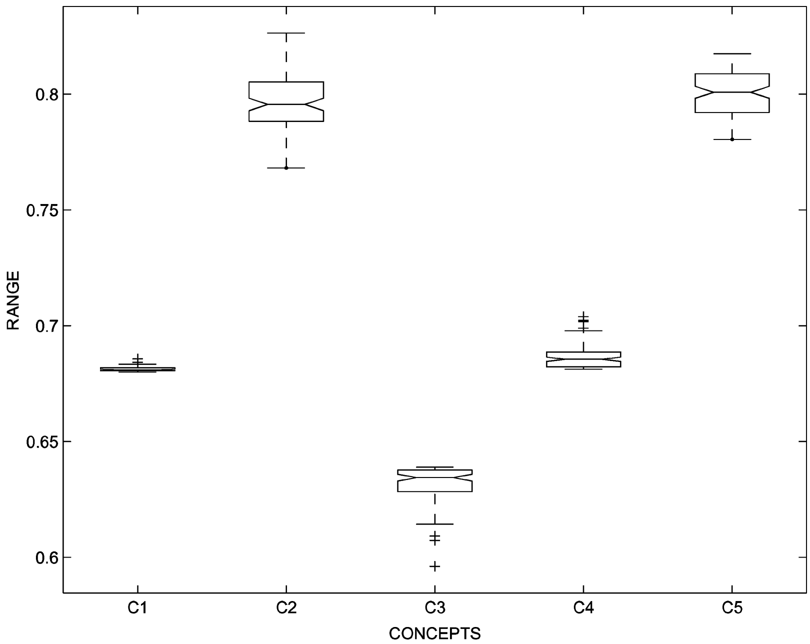

Figure 12. Boxplot of the obtained results for the concepts for the Case B.

case is,

$$W = \begin{pmatrix} 0.00 & -0.27 & -0.20 & 0.00 & 1.00 \\ 0.40 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.81 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 & 0.10 & 0.00 \end{pmatrix},$$

and its steady state is,

$$C_1 = 0.6816, \; C_2 = 0.8090, \; C_3 = 0.6174, \; C_4 = 0.6822, \; C_5 = 0.8174.$$

6.2.3. Case C. In this final case the three weights W_{21}, W_{31}, and W_{41}, are fixed and equal to the same values as in Case B, while the remaining weights are constrained within the ranges defined by their

$$\text{Mean Value} \pm \text{Standard Deviation},$$

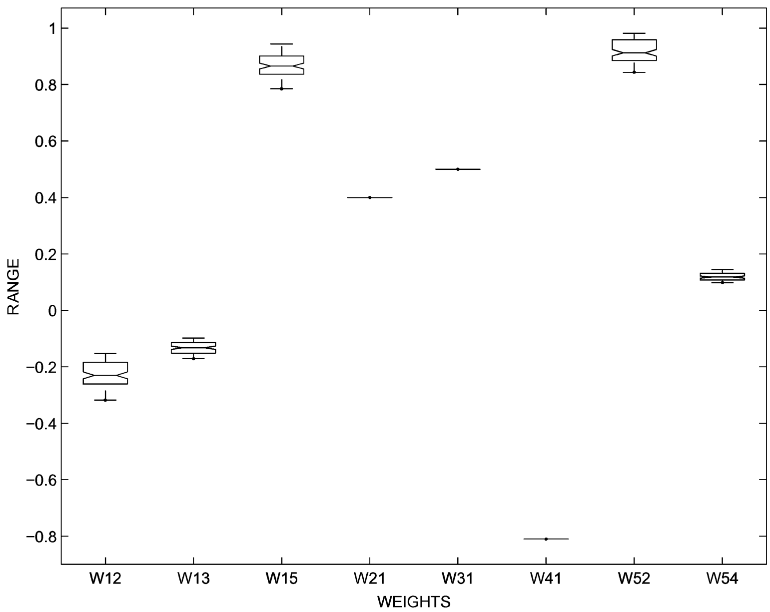

Figure 13. Boxplot of the obtained results for the weights for the Case C.

where the mean values and standard deviations reported in Table 4 are used, i.e., the weights are constrained in the following regions,

$$\begin{aligned} -0.32 \le\ & W_{12} \le -0.15, \\ -0.17 \le\ & W_{13} \le -0.10, \\ 0.78 \le\ & W_{15} \le 0.94, \\ 0.84 \le\ & W_{52} \le 0.98, \\ 0.10 \le\ & W_{54} \le 0.14. \end{aligned} \tag{14}$$

The obtained results are reported in Table 5 and a boxplot for the weights is depicted in figure 13. The convergence regions of the weights W_{12}, W_{13}, W_{15}, W_{52}, and W_{54}, are

Table 5. Statistical analysis of the results for the Case C.

	W_{12}	W_{13}	W_{15}	W_{21}	W_{31}	W_{41}	W_{52}	W_{54}
Mean	−0.2268	−0.1321	0.8659	0.4000	0.5000	−0.8100	0.9165	0.1192
Median	−0.2298	−0.1320	0.8654	0.4000	0.5000	−0.8100	0.9127	0.1180
St.dev.	0.0486	0.0212	0.0422	0.0000	0.0000	0.0000	0.0402	0.0135
Min	−0.3177	−0.1714	0.7848	0.4000	0.5000	−0.8100	0.8433	0.0983
Max	−0.1525	−0.0976	0.9440	0.4000	0.5000	−0.8100	0.9818	0.1440

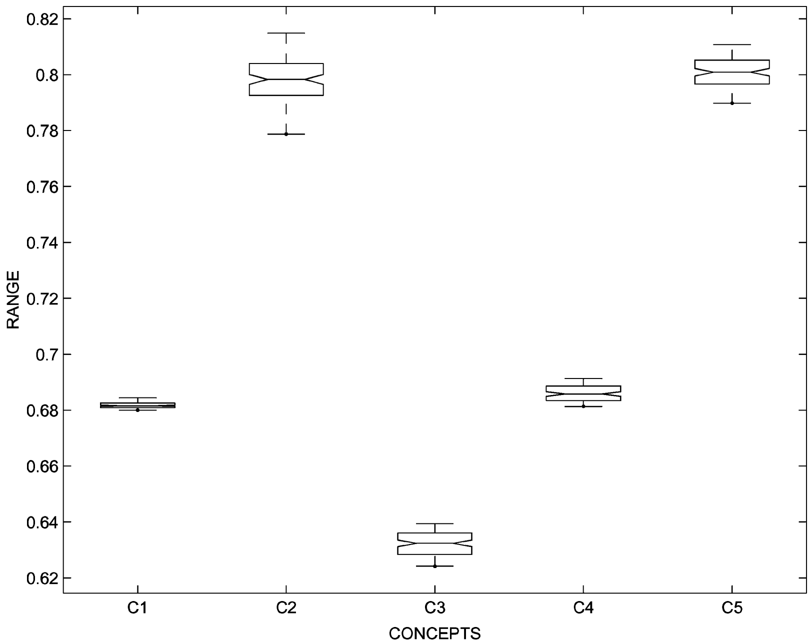

Figure 14. Boxplot of the obtained results for the concepts for the Case C.

almost the same with the regions defined in Relation (14). Further experiments have been performed, using the new mean values and the new standard deviations, reported in Table 5, to constrain the weights, but significantly different convergence regions of the weights, were not obtained. The convergence regions of the concepts are depicted in the boxplot of figure 14. An obtained suboptimal matrix for this case is,

$$W = \begin{pmatrix} 0.00 & -0.23 & -0.13 & 0.00 & 0.86 \\ 0.40 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.81 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.92 & 0.00 & 0.12 & 0.00 \end{pmatrix},$$

and the corresponding steady state is,

$$C_1 = 0.6817,\ C_2 = 0.7985,\ C_3 = 0.6323,\ C_4 = 0.6860,\ C_5 = 0.8007.$$

From the different scenarios examined, it is clear that there is a significant divergence of some weights from the initial weight values suggested by the experts. Specifically, the weights W_{13}, W_{15}, W_{52}, and W_{54}, always converge to regions significantly different than the fuzzy regions suggested by the experts. The weights W_{21}, W_{31}, and W_{41}, take almost identical values in every experiment, near the initial bounds suggested by the experts. Finally, the weight W_{12} deviates slightly from its initial region. Thus, the proposed learning algorithm is able to provide proper weight matrices for the design of the FCM, efficiently and effectively, alleviating deficiencies caused by deviation in the experts' suggestions. Exploiting a priori information, such as constraints posed by the experts on weights, enhances its performance. Moreover, a primitive statistical study of the obtained results provides an intuition on the operation and the dynamics of the modeled system.

7. Conclusions

Fuzzy Cognitive Maps (FCMs) are widely used to successfully model and analyze complex systems. The need to improve the functional representation of FCMs has been outlined. A new learning algorithm for determining suboptimal weight matrices for Fuzzy Cognitive Maps with fixed structures, in order to reach a desired steady state, is introduced. The proposed approach is based on the minimization of a properly defined objective function through the Particle Swarm Optimization algorithm. The new learning approach for the determination of the FCM's weight matrix is formulated and explained.

An industrial process control problem is used for the illustration of the proposed learning algorithm, and different scenarios are investigated. The results appear to be very promising, verifying the effectiveness of the learning procedure. The physical meaning of the obtained results is retained. The proposed approach also provides a robust solution in the event of divergent opinions of the experts concerning the system.

Future work will also consider the automatic selection of the FCM's arcs through swarm intelligence algorithms, as well as the application of the proposed approach on systems of higher complexity.

Acknowledgment

This work was supported by the "PYTHAGORAS II" research grant co-funded by the European Social Fund and National Resources.

References

Abido, M.A. (2002). Optimal Design of Power System Stabilizers Using Particle Swarm Optimization. *IEEE Trans. Energy Conversion*, 17(3), 406–413.

Agrafiotis, D.K. and Cedeno, W. (2002). Feature Selection For Structure–Activity Correlation Using Binary Particle Swarms. *Journal of Medicinal Chemistry*, 45(5), 1098–1107.

Axelrod, R. (1976). *Structure of Decision: the Cognitive Maps of Political Elites*. Princeton, NJ: Princeton University Press.

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. New York: Oxford University Press.

Banzhaf, W., Nordin, P., Keller, R.E., and Francone, F.D. (1998). *Genetic Programming–An Introduction*. San Francisco: Morgan Kaufman.

Beyer, H.-G. (2001). *The Theory of Evolution Strategies*. Berlin: Springer.

Bonabeau, E., Dorigo, M., and Théraulaz, G. (1999). *From Natural to Artificial Swarm Intelligence*. New York: Oxford University Press.

Clerc, M. and Kennedy, J. (2002). The Particle Swarm–Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Trans. Evol. Comput.*, 6(1), 58–73.

Cockshott, A.R. and Hartman, B.E. (2001). Improving the Fermentation Medium for *Echinocandin B* Production Part II: Particle Swarm Optimization. *Process Biochemistry*, 36, 661–669.

Cox, E. (1999). *The Fuzzy Systems Handbook*. Cambridge, MA: Academic Press.

Craiger, P. and Coovert, M.D. (1994). Modelling Dynamic Social and Psychological Processes with Fuzzy Cognitive Maps. In *Proc. 3rd IEEE Conf. Fuzzy Systems*.

Dickerson, J.A. and Kosko, B. (1994). Adaptive Cognitive Maps in Virtual Worlds. In *Annual Meeting World Congress Neural Networks*.

Eberhart, R.C. and Shi, Y. (1998). Comparison Between Genetic Algorithms and Particle Swarm Optimization. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben (Eds.), *Evolutionary Programming* (pp. 611–616). Vol. VII. Springer.

Eberhart, R.C., Simpson, P., and Dobbins, R. (1996). *Computational Intelligence PC Tools*. Academic Press.

Fogel, D. (1996). *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press.

Fourie, P.C. and Groenwold, A.A. (2002). The Particle Swarm Optimization Algorithm in Size and Shape Optimization. *Struct. Multidisc. Optim.*, 23, 259–267.

Groumpos, P.P. and Stylios, C.D. (2000). Modelling Supervisory Control Systems Using Fuzzy Cognitive Maps. *Chaos, Solutions and Fractals*, 11, 329–336.

Hagiwara, M. (1992). Extended Fuzzy Cognitive Maps. In *Proc. IEEE Int. Conf. Fuzzy Systems* (pp. 795–801).

Hebb, D.O. (1949). *The Organization of Behaviour: A Neuropsychological Theory*. John Wiley.

Jang, J.S., Sun, C.T., and Mizutani, E. (1997). *Neuro–Fuzzy and Soft Computing*. Upper Saddle River, NJ: Prentice Hall.

Kennedy, J. (1998). The Behavior of Particles. In: V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben (Eds.), *Evolutionary Programming* (pp. 581–590). Vol. VII. Springer.

Kennedy, J. and Eberhart, R.C. (1995). Particle Swarm Optimization. In: *Proc. IEEE Int. Conf. Neural Networks* (pp. 1942–1948). Vol. IV, Piscataway, NJ. IEEE Service Center.

Kennedy, J. and Eberhart, R.C. (2001). *Swarm Intelligence*. Morgan Kaufmann Publishers.

Khan, S., Chong, A., and Gedeon, T.A. (1999). Methodology for Developing Adaptive Fuzzy Cognitive Maps for Decision Support. In *Proc. 3rd Australia–Japan Evolutionary Systems* (pp. 93–100). Canberra.

Kosko, B. (1986). Fuzzy Cognitive Maps. *Int. J. Man–Machine Studies*, 24, 65–75.

Kosko, B. (1992). *Neural Networks and Fuzzy Systems*. New Jersey: Prentice Hall.

Kosko, B. (1997). *Fuzzy Engineering*. New York: Prentice Hall.

Koulouriotis, D.E., Diakoulakis, I.E., and Emiris, D.M. (2001). Learning Fuzzy Cognitive Maps Using Evolution Strategies: A Novel Schema For Modeling and Simulating High–Level Behavior. In *Proc. 2001 IEEE Cong. Evol. Comp.* Seoul, Korea.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Laskari, E.C., Parsopoulos,, K.E., and Vrahatis, M.N. (2002a). Particle Swarm Optimization for Integer Programming. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation* (pp. 1576–1581). Hawaii (HI), USA. IEEE Press.

Laskari, E.C., Parsopoulos, K.E., and Vrahatis, M.N. (2002b). Particle Swarm Optimization for Minimax Problems. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation* (pp. 1582–1587). Hawaii (HI), USA. IEEE Press.

Lin, C.T. and Lee, C.S. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, N.J: Prentice Hall.

Lu, W.Z., Fan, H.Y. Leung, A.Y.T., and Wong, J.C.K. (2002). Analysis of Pollutant Levels in Central Hong Kong Applying Neural Network Method With Particle Swarm Optimization. *Environmental Monitoring and Assessment*, 79, 217–230.

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer.

Ourique, C.O., Biscaia, E.C., and Carlos Pinto, J. (2002). The Use of Particle Swarm Optimization for Dynamical Analysis in Chemical Processes. *Computers and Chemical Engineering*, 26, 1783–1793.

Papageorgiou, E.I. and Groumpos, P.P. (2004). A Weight Adaptation Method for Fine–Tuning Fuzzy Cognitive Map Causal Links. *Soft Computing Journal* (in press).

Papageorgiou, E.I., Parsopoulos, K.E., Groumpos, P.P., and Vrahatis, M.N. (2004a). Fuzzy Cognitive Maps Learning Through Swarm Intelligence. *Lecture Notes in Computer Science (LNAI)*, 3070, 344–349.

Papageorgiou, E.I., Spyridonos, P., Ravazoula, P., Stylios, C.D., Groumpos, P.P., and Nikiforidis, G. (2004b). Grading Urinary Bladder Tumors Using Unsupervised Hebbian Algorithm for Fuzzy Cognitive Maps. *Biomedical Soft Computing and Human Sciences*, 9(2), 33–39.

Papageorgiou, E.I., Spyridonos, P., Stylios, C.D., Nikiforidis, G., and Groumpos, P.P. (2004c). The Challenge of Using Soft Computing Techniques for Tumor Characterization. *Lecture Notes in Computer Science (LNAI)*, 3070, 1031–1036.

Papageorgiou, E.I., Stylios, C.D., and Groumpos, P.P. (2003a). Fuzzy Cognitive Map Learning based on Nonlinear Hebbian Rule. *Lecture Notes in Computer Science (LNAI)*, 2903, 254–266.

Papageorgiou, E.I., Stylios, C.D., and Groumpos, P.P. (2003b). An Integrating Two–Level Hierarchical System for Decision Making in Radiation Therapy Using Fuzzy Cognitive Maps. *IEEE Transactions on Biomedical Engineering*, 50(12), 1326–1339.

Papageorgiou, E.I., Stylios, C.D., and Groumpos, P.P. (2004d). Active Hebbian Learning Algorithm to Train FCMs. *International Journal of Approximate Reasoning*, 37(3), 219–249.

Parsopoulos, K.E., Laskari, E.C., and Vrahatis, M.N. (2003). Particle Identification by Light Scattering Through Evolutionary Algorithms. In *Proceedings of the 1st International Conference for Mathematics and Informatics for Industry* (pp. 97–108). Thessaloniki, Greece.

Parsopoulos, K.E., Papageorgiou, E.I., Groumpos, P.P., and Vrahatis, M.N. (2004). Evolutionary Computation Techniques for Optimizing Fuzzy Cognitive Maps in Radiation Therapy Systems. *Lecture Notes in Computer Science (LNCS)*, 3102, 402–413.

Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., and Vrahatis, M.N. (2001). Objective Function "Stretching" to Alleviate Convergence to Local Minima. *Nonlinear Analysis, Theory, Methods & Applications*, 47(5), 3419–3424.

Parsopoulos, K.E. and Vrahatis, M.N. (2002a). Initializing the Particle Swarm Optimizer Using the Nonlinear Simplex Method. In A. Grmela and N. Mastorakis (Eds.), *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation* (pp. 216–221). WSEAS Press.

Parsopoulos, K.E. and Vrahatis, M.N. (2002b). Particle Swarm Optimization Method for Constrained Optimization Problems. In P. Sincak, J. Vascak, V. Kvasnicka, and J. Pospichal (Eds.), *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies* (pp. 214–220). Vol. 76 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

Parsopoulos, K.E. and Vrahatis, M.N. (2002c). Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization. *Natural Computing*, 1(2–3), 235–306.

Parsopoulos, K.E. and Vrahatis, M.N. (2003). Computing Periodic Orbits Of Nondifferentiable/Discontinuous Mappings Through Particle Swarm Optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium* (pp. 34–41). Indianapolis (IN), USA.

Parsopoulos, K.E. and Vrahatis, M.N. (2004). On the Computation of All Global Minimizers Through Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), 211–224.

Ray, T. and Liew, K.M. (2002). A Swarm Metaphor for Multiobjective Design Optimization. *Engineering Optimization*, 34(2), 141–153.

Rechenberg, I. (1994). Evolution Strategy. In J.M. Zurada, R.J. Marks II, and C. Robinson (Eds.), *Computational Intelligence: Imitating Life* (pp. 147–159). Piscataway, NJ: IEEE Press.

Saldam, A., Ahmad, I., and Al-Madani, S. (2002). Particle Swarm Optimization for Task Assignment Problem. *Microprocessors and Microsystems*, 26, 363–371.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. New York: Wiley.

Shi, Y. and Eberhart. R.C. (1998a). A Modified Particle Swarm Optimizer. In *Proceedings IEEE Conference on Evolutionary Computation* (pp. 69–73). Anchorage, AK. IEEE Service Center.

Shi, Y. and Eberhart, R.C. (1998b). Parameter Selection in Particle Swarm Optimization. In V.W. Porto, N. Saravanan, D. Waagen, and A.E. Eiben (Eds.), *Evolutionary Programming* (pp. 591–600). Vol. VII. Springer.

Stylios, C.D., Georgopoulos, V., and Groumpos, P.P. (1999). Fuzzy Cognitive Map Approach to Process Control Systems. *J. Adv. Comp. Intell.*, 3(5), 409–417.

Stylios, C.D. and Groumpos, P.P. (1998). The Challenge of Modelling Supervisory Systems Using Fuzzy Cognitive Maps. *J. Intelligent Manufacturing*, 9, 339–345.

Stylios, C.D. and Groumpos, P.P. (2000). Fuzzy Cognitive Maps in Modeling Supervisory Control Systems. *J. Intelligent & Fuzzy Systems*, 8(2), 83–98.

Taber, R. (1991). Knowledge Processing With Fuzzy Cognitive Maps. *Expert Systems With Applications 2* (pp. 83–87).

Taber, R. (1994). Fuzzy Cognitive Maps Model Social Systems. *AI Experts*, 9, 8–23.

Tandon, V., El-Mounayri, H., and Kishawy, H. (2002). NC End Milling Optimization Using Evolutionary Computation. *Int. J. Machine Tools & Manufacture*, 42, 595–605.

Trelea, I.C. (2003). The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection. *Information Processing Letters*, 85, 317–325.