Soft Comput (2009) 13:77–94
DOI 10.1007/s00500-008-0311-2

ORIGINAL PAPER

Improving fuzzy cognitive maps learning through memetic particle swarm optimization

Y. G. Petalas · K. E. Parsopoulos · M. N. Vrahatis

Published online: 10 May 2008

Abstract Fuzzy cognitive maps constitute a neuro-fuzzy modeling methodology that can simulate complex systems accurately. Although their configuration is defined by experts, learning schemes based on evolutionary and swarm intelligence algorithms have been employed for improving their efficiency and effectiveness. This paper comprises an extensive study of the recently proposed swarm intelligence memetic algorithm that combines particle swarm optimization with both deterministic and stochastic local search schemes, for fuzzy cognitive maps learning tasks. Also, a new technique for the adaptation of the memetic schemes, with respect to the available number of function evaluations per application of the local search, is proposed. The memetic learning schemes are applied on four real-life problems and compared with established learning methods based on the standard particle swarm optimization, differential evolution, and genetic algorithms, justifying their superiority.

Keywords Fuzzy cognitive maps · Memetic algorithms · Particle swarm optimization · Local search · Machine learning

Y. G. Petalas · K. E. Parsopoulos · M. N. Vrahatis (✉)
Computational Intelligence Laboratory (CI Lab),
Department of Mathematics, University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, 26110 Patras, Greece
e-mail: vrahatis@math.upatras.gr

Y. G. Petalas
e-mail: petalas@math.upatras.gr

K. E. Parsopoulos
e-mail: kostasp@math.upatras.gr

1 Introduction

Fuzzy cognitive maps (FCMs) are simulation tools that draw ideas from neural networks and fuzzy logic. Their inherent ability for abstraction and adaptation that stems from their neuro-fuzzy representation, renders them a very useful tool for modeling and studying complex systems. Up-to-date, FCMs have been used in a plethora of applications in diverse scientific fields, including simulation of social and organizational systems (Craiger et al. 1996; Taber 1991, 1994), circuit design and analysis (Styblinski and Meyer 1988), industrial process control (Papageorgiou et al. 2005; Stylios et al. 1999), supervisory control systems (Groumpos and Stylios 2000; Stylios et al. 1999; Stylios and Groumpos 1998, 2000) and bioinformatics (Georgopoulos et al. 2003; Parsopoulos et al. 2004b).

Proper design of FCMs requires deep knowledge of the simulated system and its operation. For this purpose, a group of experts is responsible for the determination of the system's key concepts as well as their interactions. These features are represented on a directed graph with nodes and weighted edges, respectively. The weights on the interconnections among nodes, represent the magnitude of their interactions. These weights are computed through a fuzzification–defuzzification procedure, where the opinions of all experts are translated from linguistic to numerical values. This procedure prevents from possible errors that could be introduced in the weights through direct assignment of numerical values. However, in some cases the experts' opinions differ significantly and the corresponding weight setting does not result in appropriate simulation of the system (Papageorgiou et al. 2005). In such cases, *learning procedures* can be used to modify the weights properly.

Learning in FCMs is the procedure of modifying the weights in order to achieve a set of objectives that are usually problem-dependent. This is performed through the minimization of a properly defined objective function. There are just a few well-studied learning schemes in the literature and they can be classified in two categories. The first consists of approaches based on unsupervised neural networks training methods (Kosko 1997; Papageorgiou 2004b), while the latter refers to evolutionary computation and swarm intelligence approaches (Khan et al. 2004; Koulouriotis et al. 2003; Papageorgiou et al. 2005; Parsopoulos et al. 2004b; Stach et al. 2005).

Hybrid schemes that combine either two evolutionary algorithms or an evolutionary and a direct search component have been proposed and applied successfully in different applications, such as function optimization (Fan et al. 2004; Parsopoulos and Vrahatis 2002a,b), and learning in neuro-fuzzy systems (Juang 2004; Juang and Hsu 2005). Memetic Algorithms (MAs) are hybrid schemes that consist of a global and a local search component (Dawkins 1976; Moscato 1999). The global component is usually a stochastic population-based method that is competent to globally explore the search space (exploration), while the local component is responsible for more refined local search (exploitation). MAs have proved to be very efficient in a plethora of optimization problems. Recently, a Memetic Particle Swarm Optimization (MPSO) scheme equipped with the Random Walk with Directional Exploitation method proved to be very efficient in a plethora of numerical optimization problems (Petalas et al. 2007b). Also, an MPSO approach has been applied for FCMs learning (Petalas et al. 2005, 2007a), with preliminary results indicating its viability. Different learning methods, such as Differential Evolution (DE), Genetic Algorithms (GAs), and Evolution Strategies, have been also proposed and applied successfully for FCMs learning (Khan et al. 2004; Koulouriotis et al. 2001; Papageorgiou et al. 2005; Parsopoulos et al. 2004b; Petalas et al. 2005, 2007a; Stach et al. 2005).

The present work constitutes an extensive study of the recently proposed MPSO schemes on four real-life test cases. Additionally, a new technique for the adaptation of the number of function evaluations allocated to the local search component, is proposed. The considered memetic schemes incorporate the deterministic local search algorithm of Hooke and Jeeves (Hooke and Jeeves 1961; Rao 1992), as well as the stochastic method of Solis and Wets (Solis and Wets 1981), to the Particle Swarm Optimization (PSO) algorithm, which constitutes the global component. Their performance is compared with that of established PSO, DE and GA approaches, and the significance of their differences is verified through statistical tests.

The paper is organized as follows: Sect. 2 provides concise descriptions of FCMs, PSO, MAs, as well as the employed local search schemes. The considered MPSO scheme is introduced in Sect. 3. Descriptions of the studied problems are given in Sect. 4, and experimental results are reported in Sect. 5. The paper concludes in Sect. 6.

2 Background material

In the following sections, the basic concepts of FCMs, PSO, MAs as well as the employed local search schemes are briefly exposed.

2.1 Fuzzy cognitive maps

FCMs were introduced by Kosko (1986) as directional graphs with feedback. Similarly to traditional causal concept maps, FCMs consist of nodes that represent key concepts of the simulated system. The links among concepts are signed either as positive or negative, to represent their causal relationship. Feedback equips FCMs with the ability to simulate temporal causality, thereby providing causal explanation in dynamical models. On the other hand, determination of specific numerical weights for the interconnections among concepts by using fuzzy procedures, provides flexibility in the representation of causality degree among events. A simple FCM with 5 concepts and 7 weights is illustrated in Fig. 1.

Let N be the number of concepts, C_i, $i = 1, 2, \ldots, N$, of an FCM. Each concept, C_i, assumes a value, $A_i \in [0, 1]$, $i = 1, 2, \ldots, N$. An edge with direction from C_i to a different concept, C_j, is characterized by a weight, $w_{ij} \in [-1, 1]$. Positive weights denote positive causality, i.e., an increment in A_i triggers an increase in A_j, while a decrement in A_i results also in a decrement in A_j. On the other hand, negative weights imply negative causality, i.e., an increment in A_i results in a decrement in A_j, and vice versa. The absence of interconnection between C_i and C_j, can be considered as an edge with weight $w_{ij} = 0$. The values of the weights can be

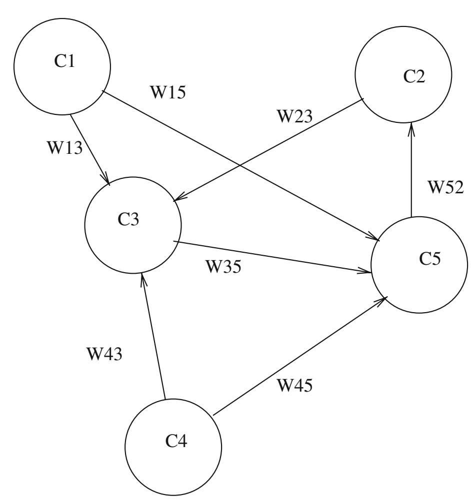

Fig. 1 A simple FCM with 5 concepts and 7 weights

organized in a matrix,

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

where the ith row represents the causality between concept C_i and the rest of the concepts in the map.

Both the design as well as the initial setting of the FCM's parameters is determined by a group of experts that have deep knowledge of the modeled system Stylios et al. (1999); Stylios and Groumpos (2000). Although experts can assign directly numerical values on concepts and weights, this procedure is error-prone and usually unnecessary. Instead, they can use linguistic modifiers, which are then converted into fuzzy functions, allowing the developer to capture more fine grain information about the representation. Also, experts can pose strict bounds on the values of concepts and weights, in order to retain their physical meaning.

After its initial configuration, the FCM behaves like a discrete dynamical system and it is let to converge to a stable state, i.e., a state where no further modification of the concepts' values is achieved, by applying the equation (Kosko 1997; Stylios and Groumpos 2004)

$$A_i(t+1) = f\left(A_i(t) + \sum_{\substack{k=1 \\ k \neq i}}^{N} w_{ki}(t) A_k(t) \right), \qquad (1)$$

where t denotes the iteration number; $A_i(t+1)$ is the value of concept C_i at time $t+1$; $A_k(t)$ is the value of C_k at time t; and f is usually a sigmoid function. After convergence, the FCM is assumed to be able to simulate the system accurately. However, this is not always achievable. Wide variations among experts' opinions can result in weights that do not lead the FCM to desirable stable states, resulting in undesirable values of the concepts. For this purpose, learning algorithms proved to be very useful.

A learning algorithm modifies further the weights of the FCM, such that the achieved stable state is desirable, with the concepts and weights lieing within the bounds posed by the experts. Established learning schemes are based either on the Hebbian rule for unsupervised neural networks training (Kosko 1997; Papageorgiou 2004b) or on evolutionary computation and swarm intelligence schemes (Khan et al. 2004; Koulouriotis et al. 2003; Papageorgiou et al. 2005; Parsopoulos et al. 2004b; Stach et al. 2005). In the latter case, properly defined objective functions that penalize weight settings corresponding to undesirable steady states, are used (Papageorgiou et al. 2004a, 2005; Parsopoulos et al. 2004a,b).

2.2 Particle swarm optimization

PSO is a distributed, stochastic optimization algorithm that exploits a population of individuals to explore the search space. It belongs to the class of *swarm intelligence* algorithms, which are inspired from the social dynamics and emergent behavior in socially organized colonies (Kennedy and Eberhart 2001). In PSO's context, the population is called a *swarm* and the individuals (i.e., the search agents) are called *particles*.

Each particle has three main features: an adaptable velocity, a memory where it stores the best position it has ever visited, and a mechanism of exchanging information with (some of) the rest of the particles. The swarm dynamic depends on the information exchange scheme, resulting in two variants of the algorithm. The *local* variant assumes that a particle shares information only with a small number of other particles that constitute its *neighborhood*. On the other hand, *global* PSO is a generalization of the local variant, where the whole swarm is considered as the neighborhood of each particle, i.e., it shares the globally best information.

Neighborhoods are usually defined in the space of particles' indices rather than in actual search space, based on a user-defined *neighborhood topology*. The reason for this selection is the alleviation of the heavy computational burden required for the computation of all distances among particles, as well as the promotion of diversity in the swarm. The most common neighborhood topology is the "ring topology" illustrated in Fig. 2 (left), where the particles, $x_1, \ldots, x_M$, are assumed to lie on a ring, having two immediate neighbors each. Thus, a neighborhood of radius r of the particle x_i, would consist of the particles,

$$\{x_{i-r}, x_{i-r+1}, \ldots, x_i, \ldots, x_{i+r-1}, x_{i+r}\},$$

assuming that x_1 is the particle that follows immediately after the last particle of the swarm, x_M. In the global variant of PSO, each particle shares information with the global best, and it is usually depicted as the "star" neighborhood illustrated in Fig. 2 (right), where only the best particle exchanges information with all other particles.

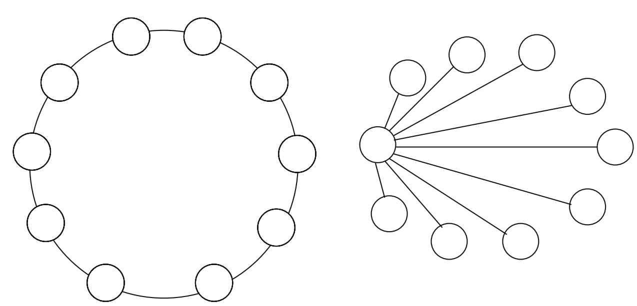

Fig. 2 The ring (*left*) and star (*right*) neighborhood topology of PSO

Let $\mathcal{S} \subset \mathbb{R}^n$ be the search space, $f : \mathcal{S} \to \mathbb{R}$ be the objective function, and $\mathbb{S}$ be a swarm consisting of M particles, $\mathbb{S} = \{x_1, \ldots, x_M\}$. The ith particle is an n-dimensional vector,

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^\top \in \mathcal{S}.$$

The velocity of this particle is also an n-dimensional vector,

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in})^\top.$$

The best previous position encountered by the ith particle in $\mathcal{S}$ is denoted by,

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

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, i.e.,

$$f\left(p_{g_i}(t)\right) \leqslant f\left(p_j(t)\right),$$

for all neighbors x_j of x_i, and t to be the iteration counter. Then, the swarm is manipulated by the equations (Clerc and Kennedy 2002):

$$v_{ij}(t+1) = \chi\left[v_{ij}(t) + c_1 r_1\left(p_{ij}(t) - x_{ij}(t)\right) + c_2 r_2\left(p_{g_i j}(t) - x_{ij}(t)\right)\right], \quad (2)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (3)$$

where $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, n$; χ is a parameter called *constriction factor*; c_1 and c_2 are two positive constants called *cognitive* and *social* parameter, respectively; and r_1, r_2, are random numbers drawn from a uniform distribution in [0, 1]. The constriction factor is a mechanism for controlling the magnitude of the velocities. Stability analysis of PSO provided different configurations of χ with respect to the parameters c_1 and c_2 (Clerc and Kennedy 2002; Trelea 2003).

In the global variant of PSO, all particles are attracted by the same overall best position, converging faster toward specific points. Thus, the global variant of PSO emphasizes exploitation over exploration. On the other hand, in the local variant, the information of the best position of each neighborhood is transmitted to the other particles of the swarm through their neighbors. Therefore, the attraction to specific best positions is weaker, avoiding the swarm from getting stucked in locally optimal solutions. Thus, the local variant of PSO emphasizes exploration over exploitation. The balance between these two properties is crucial for the performance of each global optimization algorithm (Parsopoulos and Vrahatis 2004, 2007; Törn and Žilinskas 1989).

2.3 Memetic algorithms

MAs are metaheuristic search algorithms used for global optimization tasks. Their name is derived from "meme" that was first introduced by Dawkins (1976) and represents a unit of cultural evolution that can exhibit refinement. MAs draw inspiration from natural adaptation models that combine individual evolution and learning within a lifetime. Thus, they consist of a global search component and a local search component. The first emulates evolution of the individual, while personal refinement is encouraged through the latter (Krasnogor and Smith 2005).

As already mentioned, the trade-off between exploration and exploitation is crucial for the algorithm's performance. MAs strive to balance this trade-off efficiently. For this purpose, evolutionary algorithms are usually employed, combined with local search methods. Early memetic schemes combined GAs with simulated annealing (Moscato 1989). Different names that were also used for such algorithms are Hybrid GAs, Genetic Local Searches, Lamarckian GAs, and Baldwin GAs. MAs have been used in a wide range of research areas, including operations research, single and multi-objective optimization, bioinformatics and machine learning (Krasnogor and Smith 2005). Also, they have been applied successfully in NP-hard combinatorial optimization problems, often achieving better results from evolutionary algorithms and other metaheuristics (Corne et al. 1999; Krasnogor 2002; Land 1998; Merz 1998). Recently, new schemes that combine PSO with stochastic local search were shown to be very efficient in solving function optimization problems (Petalas et al. 2007b). Also, preliminary results for FCMs learning were shown to be very promising (Petalas et al. 2005, 2007a).

MAs operation follows closely the operation of the employed evolutionary algorithm. However, after applying the evolutionary operators and evaluating the population, some individuals are selected as initial points for the local search. The obtained solutions usually replace these individuals in the population, if they improve their function values. An abstract description of an MA is provided below:

```
Begin
    Initialize Population S
    Repeat
        Apply evolutionary operators on S
        Evaluate S
        Select a subpopulation S_loc ⊂ S
        Apply local search on each element of S_loc
        Evaluate S_loc
        Update S
    Until stopping criterion is satisfied
    Return best solution
End
```

One of the most active research topics in MAs is the selection of the position(s) and frequency of local search application. Several schemes have been suggested in the literature. Hart (1994) describes the following basic schemes:

1. *Adaptive selection*. This scheme selects the position for application of local search according to the following strategies:
   (a) *Distribution-based strategy*: the goal is to favor the application of local search on solutions that differ adequately or lie in distant parts of the search space.
   (b) *Fitness-based strategy*: local search is applied on the best performing individuals, since their probability of lieing in the neighborhood of the global minimizer is higher.
2. *Non-adaptive selection*. According to this scheme, the individuals that will serve as the initial points for local search are selected randomly from the population, with a fixed user-defined probability.

The application of local search at every individual and iteration is rarely used, since it is considered computationally expensive. A very promising adaptive scheme for tackling this problem has been proposed in Lozano et al. (2004) for a GA-based MA. This scheme is fitness-based, with chromosomes generated by crossover and mutation being assigned probabilities based on the improvement they exhibit compared to the worst performing individual. The provided results on widely used benchmark functions, and comparisons with other memetic approaches, support the claim that such schemes can be proved very efficient (Lozano et al. 2004).

The employed local search methods play an important role in the applicability and efficiency of MAs. Gradient-based local search could be very efficient in well-defined problems with good mathematical characteristics, while stochastic local search is better choice for problems contaminated by noise and uncertainty, as in most real life applications. Different local search schemes have been investigated in the relative literature. Comprehensive reviews, along with techniques for dynamically selecting the local search algorithm can be found in Krasnogor and Smith (2005) and Ong and Keane (2004). In the following paragraph, the local search schemes used in the proposed approach are concisely described.

2.4 The employed local search methods

In the proposed memetic algorithm, we utilized the Hooke and Jeeves (1961) and the Solis and Wets (1981) search techniques. The main criterion for the selection of local search scheme was the preservation of the derivative-free nature of the memetic scheme, as well as its simplicity and easy adaptation on the problem at hand. The aforementioned techniques are characterized by simplicity, flexibility, and minor assumptions on the objective function (Hooke and Jeeves 1961; Solis and Wets 1981). Therefore, they appear proper for the FCMs learning task. In the following paragraphs we sketch their operation.

2.4.1 The Hooke and Jeeves algorithm

Hooke and Jeeves (HJ) is a deterministic pattern search algorithm (Hooke and Jeeves 1961; Rao 1992). It is a direct search algorithm in the sense that it uses only function evaluations and does not need function derivatives. Its main operation consists of a search along the coordinate axes of the search space, using a suitable step size, and it is described below [our implementation follows closely the one reported in (http://www.netlib.org/opt/hooke.c)].

Let x_1 be the initial point and $f_1 = f(x_1)$ be its function value. Also, let Δ be a user-defined initial step length. The starting point of the local search at each iteration of the algorithm is called the *base point*. Therefore, x_1 is the first base point. The first component of the base point is initially increased by Δ. If the produced point has lower function value, then it is assumed as the new *trial point* and the search continues with its next component. If there is no decrease, then the new trial point is produced by subtracting Δ from the first component of the base point. Again, if there is an improvement in the function value, the new point becomes the trial point. If there is no improvement also in this case, the search continues with the next component of the base point.

The aforementioned procedure performs a search on all coordinate directions of the base point. If this search results in a better point, then it becomes the new basis point, x_2, with function value f_2, and a pattern extension step takes place, otherwise Δ is reduced and a coordinate search starts again. The pattern extension step assumes a temporary base point defined as $x_{\text{temp}} = 2x_2 - x_1$. A coordinate search is performed on x_{temp}, and, if further improvement in the value of the base point x_2 is produced, then it becomes the new base point, x_3, and the search continues with the pattern extension step $2x_3 - x_2$. Otherwise, x_2 remains the base point, Δ is reduced, and the HJ algorithm conducts a coordinate search about x_2. The algorithm is terminated upon the achievement of user-defined criteria that are usually problem dependent.

2.4.2 The Solis and Wets algorithm

Solis and Wets have proposed several stochastic local search algorithms. In our approach, we used the algorithm reported as “Algorithm 1” in Solis and Wets (1981). This algorithm starts with an initial point, x_k, with $k = 0$. A random point, ξ, is produced in a region around x_k, following a multivariate normal distribution with center $x_k + b_k$ and covariance σ_k, where b_k is a bias factor that slants sampling toward directions where the most successes were recorded.

The sampled point becomes the new search point, x_{k+1}, if it improves its function value. On the other hand, if the function value of the sampled point is worse than that of x_k, then the new point, $2x_k - \xi$, is produced and evaluated. If neither this point improves x_k, then $x_{k+1} = x_k$, and a new sample is generated around it. In each case, b_k is modified properly.

At each iteration of the aforementioned procedure, the covariances of the distributions are modified based on the number of successes or failures in generating improving points. Thus, if new solutions are often improving the function value, then the search region is expanded, while, in the opposite case, it is contracted.

3 The proposed memetic learning scheme

The task of learning in FCMs is formulated as an optimization problem with respect to the FCMs' weights (Papageorgiou et al. 2005). In particular, the main goal is the detection of proper values of the FCM's weights in order to produce a desirable behavior of the system. The objective function of the optimization problem depends on the output concepts of the FCM. The global minimizers of the objective function are weight matrices that lead the FCM to a desired steady state, i.e., a state where the target output concepts lie in their bounding regions, while the weights fulfill restrictions related to their physical meaning, posed by the experts. Evolutionary algorithms proved to be very efficient in FCMs learning tasks. Learning methods that utilize PSO and DE have been introduced in Papageorgiou et al. (2005), Parsopoulos et al. (2004b) and Petalas et al. (2005, 2007a). GAs (Khan et al. 2004; Stach et al. 2005) and Evolution Strategies (ES) (Koulouriotis et al. 2001) have also been successfully used.

The memetic approach extends the established evolutionary learning algorithms, by combining the (evolutionary) global search component with a local search method, resulting in a new, memetic learning algorithm. The expected gain from such a hybrid scheme is a significant decrease in the number of function evaluations required for the detection of proper weight matrices, as well as an improvement in the quality of the obtained solutions.

In our approach, PSO was used as the global search component of the memetic algorithm. This choice was based primarily on its efficiency, simplicity and flexibility. A Memetic PSO (MPSO) with Random Walk with Directional Exploitation has been recently introduced with very promising results in a plethora of optimization problems (Petalas et al. 2007b). Convergence in probability has also been proved for this scheme (Petalas et al. 2007b). Preliminary results of MPSO with HJ and SW were very promising for FCMs learning tasks (Petalas et al. 2005, 2007a).

The memetic algorithm applies local search on selected best positions of the swarm at each iteration of the algorithm, in order to further improve their position. Previous works with standard PSO on FCMs, as well as MPSO on optimization problems Papageorgiou et al. (2005) and Petalas et al. (2007b) indicated that applying local search only on the best particle can boost the algorithm's performance, while imposing a mild computational burden. For these reasons we also adopted this approach. However, we used more sophisticated local search schemes, such as the HJ and SW described in the previous sections. The selection of such schemes was based on the necessity for derivative-free methods, since the objective functions in the studied FCMs learning problems are not always differentiable (Papageorgiou et al. 2005; Parsopoulos et al. 2004b).

A pseudocode of the generic MPSO approach follows below:

Input: n (dimension), χ, c_1, c_2, $x_{\min}$, $x_{\max}$ (bounds), f (objective function)
Set $t = 0$.
Initialize $x_i(t)$, $v_i(t) \in [x_{\min}, x_{\max}]$, $p_i(t) \leftarrow x_i(t)$, $i = 1, \dots, n$.
Evaluate $f(x_i(t))$.
Determine the indices g_i, $i = 1, \dots, n$.
While [stopping criterion is not satisfied] **Do**
    Update the velocities $v_i(t+1)$, $i = 1, \dots, n$, according to Eq. (2).
    Set $x_i(t+1) = x_i(t) + v_i(t+1)$, $i = 1, \dots, n$.
    Constrain each particle x_i in $[x_{\min}, x_{\max}]$.
    Evaluate $f(x_i(t+1))$, $i = 1, \dots, n$.
    If $f(x_i(t+1)) < f(p_i(t))$ **Then** $p_i(t+1) \leftarrow x_i(t+1)$
    Else $p_i(t+1) \leftarrow p_i(t)$.
    Update the indices g_i.
    When [local search is applied] **Do**
        Choose some $p_q(t+1)$, $q \in \{1, \dots, n\}$.
        Apply local search on $p_q(t+1)$ and obtain y.
        If $f(y) < f\left(p_q(t+1)\right)$ **Then** $p_q(t+1) \leftarrow y$.
    End When
    Set $t = t + 1$.
End While

The number of function evaluations allocated to the local search is heuristically determined. However, it can be crucial for the performance of the memetic algorithm, since it influences the outcome of the local search. For this purpose, we propose an adaptive scheme for modifying it at each iteration. More specifically, the algorithm is initialized with a budget of 20 function evaluations per application of the local search. Then, at each iteration, if the local search applied on the globally best position improves its value, the algorithm assumes that it may lie in a promising region of attraction, thereby decreasing the budget for the next iteration. On the other hand, if local search failed at improving the value of the best position, the budget is increased for the next application. The decrement/increment in our algorithm was assumed equal to 5.

The aforementioned adaptation scheme works similarly to the parameter adaptation scheme of the Solis and Wets algorithm (Solis and Wets 1981). Preliminary experiments using different strategies for the position and application of local search, indicated that the application of local search on

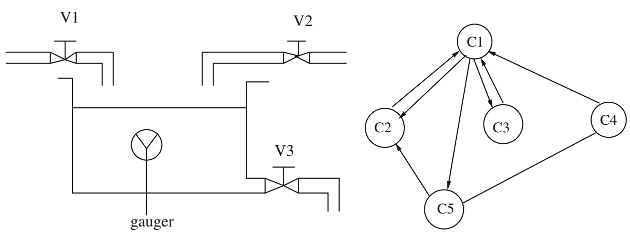

Fig. 3 Illustration of the tank-valves process control problem (*left*) and the corresponding FCM (*right*)

the best particle at each iteration is the most efficient scheme for FCMs learning, and for this purpose it was adopted in the proposed approach.

Regarding the implementation difficulty of the algorithm, it is clearly depending on the selected local search scheme. PSO is considered one of the simplest population-based algorithms, since it can be implemented in a few lines of code, as exhibited in Clerc (2006) and Parsopoulos and Vrahatis (2002b) for C and Matlab. Also, the aforementioned local search algorithms are very simple in implementation, rendering the proposed MPSO algorithm a very simple. However, if the user selects more sophisticated local search schemes, the implementation difficulty will be correspondingly increased.

4 Descriptions of the studied problems

The proposed method has been applied for FCM learning in four case studies, namely, two process control problems concerning the proper operation of industrial systems, a medical problem related to radiation therapy, and a problem about pollution in an ecological industrial park. In the following subsections, brief descriptions for the aforementioned problems are given.

4.1 The industrial tank-valves problem

A process control problem derived from chemical industry is used as our first case study (Stylios and Groumpos 1998). Figure 3 (left) illustrates the process control problem, which consists of a tank and three valves that control the amount of liquid in the tank. Two different liquids are poured and mixed into the tank through valve V_1 and valve V_2. During mixing, a chemical reaction takes place in the tank, and a new liquid is produced. Valve V_3 empties the tank as soon as the amount of the produced liquid reaches a specific level. The specific gravity of the produced liquid is measured by a sensor placed inside the tank. 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 constructed FCM for this problem is depicted in Fig. 3 (right) and consists of five concepts,

C_1: amount of the liquid in the tank,
C_2: state of valve V_1,
C_3: state of valve V_2,
C_4: state of valve V_3,
C_5: specific gravity of the produced liquid in the tank.

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 given in Table 1.

The output concepts of the FCM are C_1 and C_5, and their values, A_1 and A_5, respectively, lie within lower and upper bounds determined by the experts:

$$\left[A_1^{\mathrm{lb}}, A_1^{\mathrm{ub}}\right] = [0.68, 0.70], \quad \left[A_5^{\mathrm{lb}}, A_5^{\mathrm{ub}}\right] = [0.78, 0.85].$$

The objective function defined in Papageorgiou et al. (2005) was adopted, i.e.,

$$f(w) = H\left(A_1^{\mathrm{lb}} - A_1\right) + H\left(A_5^{\mathrm{lb}} - A_5\right) + H\left(A_1 - A_1^{\mathrm{ub}}\right) + H\left(A_5 - A_5^{\mathrm{ub}}\right), \quad (4)$$

where H is the function,

$$H(x) = \begin{cases} 0, & x \leqslant 0, \\ x, & x > 0, \end{cases}$$

and A_1, A_5, are the output concepts' values for the weight matrix w.

4.2 The radiation therapy problem

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

Table 1 Ranges of weights for the tank-valves problem

$-0.50 \leqslant w_{12} \leqslant -0.30$	$-0.40 \leqslant w_{13} \leqslant -0.20$	$-0.40 \leqslant w_{13} \leqslant -0.20$
$0.10 \leqslant w_{15} \leqslant 1.00$	$0.30 \leqslant w_{21} \leqslant 0.40$	$0.40 \leqslant w_{31} \leqslant 0.50$
$-1.00 \leqslant w_{41} \leqslant -0.80$	$0.10 \leqslant w_{52} \leqslant 1.00$	$0.10 \leqslant w_{54} \leqslant 1.00$

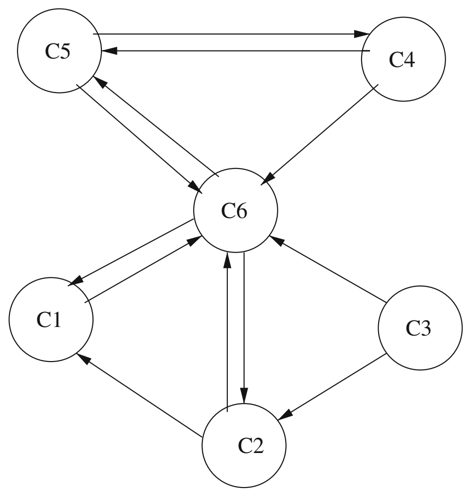

Fig. 4 The supervisor FCM for the radiation therapy problem

is needed before the final treatment execution (Parsopoulos et al. 2004b).

The radiation therapy process can be modeled and analyzed through a supervisor-FCM, constructed by experts (Parsopoulos et al. 2004b). The FCM consists of the following concepts:

C_1: tumor localization,
C_2: dose prescribed from the treatment planning,
C_3: machine factors,
C_4: human factors,
C_5: patient positioning and immobilization,
C_6: final dose received by the target volume,

and it is illustrated in Fig. 4.

The objective of the supervisor-FCM is the maximization of the final dose, C_6, received by the target volume (tumor), and the dose C_2 derived from the treatment planning within the target region. These objectives are defined by the related AAPM and ICRP protocols (Khan 1994; Wells and Niederer 1998; Willoughby et al. 1996) for the determination of accepted dose levels for each organ and part of the human body. The ranges of the weights for the supervisor-FCM model are given in Table 2.

The objective function defined in Parsopoulos et al. (2004b) was used,

$$f(w) = -(A_2 + A_6), \tag{5}$$

where A_2 and A_6 are the values of the output concepts C_2 and C_6, respectively, that correspond to the weight matrix w. The minus signs are used to transform the maximization problem to a minimization one.

4.3 The heat exchanger problem

The third case study is the heat exchanger FCM model presented analytically in Stylios and Groumpos (2004). Heat exchanger is a standard part in the chemical and process industry. The system is depicted in Fig. 5 (left) and it consists of a tubular steam/water heat exchanger (denoted as W_1) and a cross-flow water/air exchanger (denoted as W_2). The water in the circuit is heated by means of W_1, while it is cooled in the cross-flow water/air heat exchanger W_2 by using cold air from the environment, which has a temperature T_{ai}. The environmental temperature cannot be manipulated since it depends on the weather and season of the year. After passing the heat exchanger and the fan, the air is blown out back to the environment. The water temperature, T_{wo}, is controlled by manipulating the fan speed, S_f. The control variable T_{wo} depends on the manipulated variable S_f and the measurable disturbances: inlet water temperature, T_{wi}, air temperature, T_{ai}, and water flow rate, F_w. In most plants, the water flow rate is usually regulated by a PI-controlled pneumatic valve and it influences heavily the behavior of the heat exchanger W_2. The design of a temperature controller for T_{wo} that can handle widely varied flow rates is considered a major challenge (Stylios and Groumpos 2004).

The corresponding FCM has five concepts:

C_1: fan speed, S_f,
C_2: water flow rate, F_w,
C_3: water inlet temperature, T_{wi},
C_4: air inlet temperature, T_{ai},
C_5: water outlet temperature, T_{wo}, (output concept of the model).

The ranges of the weights for the FCM model are reported in Table 3. The output concept C_5 was bounded within $\left[A_5^{\text{lb}}, A_5^{\text{ub}}\right] = [0.2, 0.4]$. The objective function defined in Sect. 4.1 was employed also in this problem, though for a single output concept, i.e.,

$$f(w) = H\left(A_5^{\text{lb}} - A_5\right) + H\left(A_5 - A_5^{\text{ub}}\right). \tag{6}$$

4.4 The ecological industrial park problem

The last test problem concerns an FCM model that is used for studying the impact of developing an eco-industrial park

Table 2 Ranges of weights for the radiation therapy problem

$0.30 \leqslant w_{16} \leqslant 0.50$	$0.20 \leqslant w_{21} \leqslant 0.40$	$0.50 \leqslant w_{26} \leqslant 0.70$
$-0.40 \leqslant w_{32} \leqslant -0.20$	$-0.40 \leqslant w_{36} \leqslant -0.10$	$-0.50 \leqslant w_{45} \leqslant -0.20$
$-0.50 \leqslant w_{46} \leqslant -0.20$	$-0.60 \leqslant w_{54} \leqslant -0.10$	$0.50 \leqslant w_{56} \leqslant 0.80$
$0.20 \leqslant w_{61} \leqslant 0.40$	$0.60 \leqslant w_{62} \leqslant 0.90$	$0.50 \leqslant w_{65} \leqslant 0.90$

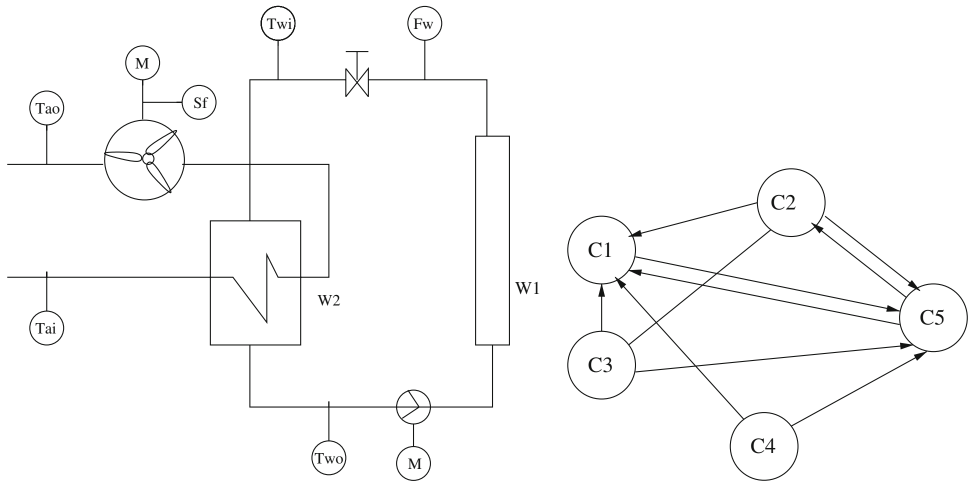

Fig. 5 The heat exchanger system (*left*) and the corresponding FCM (*right*)

Table 3 Ranges of weights for the heat exchanger problem

$-0.80 \leqslant w_{15} \leqslant -0.50$	$0.50 \leqslant w_{21} \leqslant 0.70$	$0.00 \leqslant w_{25} \leqslant 0.50$
$0.50 \leqslant w_{31} \leqslant 0.75$	$-0.75 \leqslant w_{32} \leqslant -0.50$	$0.00 \leqslant w_{35} \leqslant 0.50$
$-0.30 \leqslant w_{41} \leqslant 0.00$	$-0.75 \leqslant w_{45} \leqslant -0.50$	$0.50 \leqslant w_{51} \leqslant 0.80$
$-0.80 \leqslant w_{52} \leqslant -0.50$		

(EIP), presented in detail in Fons et al. (2004). EIPs are characterized by a network of synergistic resource linkages among facilities within a geographical area. They are designed such that industrial areas are developed mimicking a natural ecosystem. Such systems are self-contained, self-sustaining, and they produce zero waste through complex interactions of food chains. The studied FCM concerns the Lloydminster EIP at the Western Canadian plains and it is depicted in Fig. 6.

The concepts of the FCM model are the following:

C_1: pollution,
C_2: water disposal,
C_3: unutilized byproducts/wastes,
C_4: demand,
C_5: byproducts/wastes provided by existing facilities,
C_6: secondary facilities,
C_7: availability,
C_8: vehicles,
C_9: employment,
C_{10}: property cost,
C_{11}: owned housing,
C_{12}: roads,
C_{13}: population,
C_{14}: available land,
C_{15}: rental cost,
C_{16}: schools and recreation facilities,
C_{17}: service facilities,
C_{18}: byproducts/wastes provided by co-locating facilities.

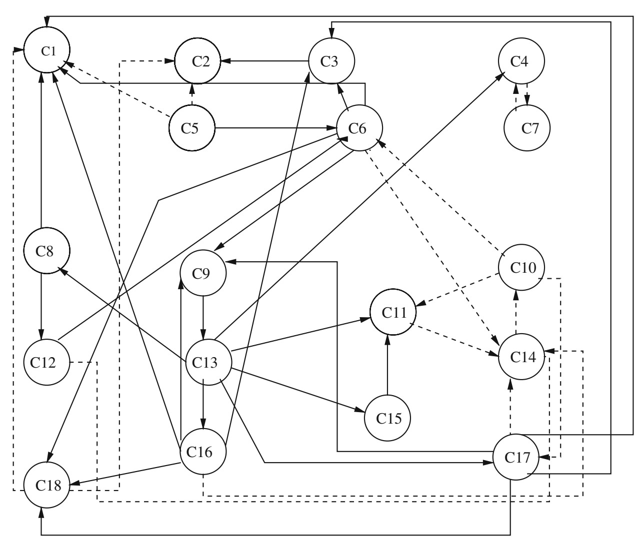

Fig. 6 The FCM for the ecological industrial park problem. *Dashed lines* denote negative causality, while *solid lines* denote positive causal relationship

In our study, we considered pollution as the output concept and our aim was to detect weights that retain it within the bounds $\left[A_1^{\text{lb}}, A_1^{\text{ub}}\right] = [0.1, 0.2]$. For this purpose, the same type of objective function as in Sect. 4.1 was employed, i.e.,

$$f(w) = H\left(A_1^{\text{lb}} - A_1\right) + H\left(A_1 - A_1^{\text{ub}}\right). \qquad (7)$$

The 40 initial weights lie within the ranges reported in Table 4.

Table 4 Ranges of weights for the eco-industrial park problem

Range	Weights
[0.0, 0.25]	$w_{56}, w_{61}, w_{81}, w_{12,6}, w_{17,1}, w_{16,1}$
[0.0, 0.5]	$w_{63}, w_{6,18}, w_{13,8}, w_{13,11}, w_{13,15}, w_{13,17}, w_{15,11}, w_{16,3}, w_{16,18}, w_{17,3}$
[0.5, 1.0]	$w_{32}, w_{69}, w_{9,13}, w_{13,4}, w_{16,9}, w_{17,9}$
[−0.5, 0.0]	$w_{47}, w_{52}, w_{6,14}, w_{74}, w_{10,6}, w_{10,17}, w_{11,14}, w_{14,10}, w_{16,14}$
[−1.0, −0.5]	$w_{51}, w_{10,11}, w_{12,14}, w_{17,14}, w_{18,1}, w_{18,2}$
[0.25, 0.55]	$w_{17,18}$
[0.25, 0.50]	$w_{8,12}$
[0.50, 0.75]	$w_{13,16}$

5 Experimental results

For each one of the four test problems (denoted as TP1–TP4, respectively) described in the previous sections, we applied the global and local variant of MPSO with HJ and SW local search. We denote these variants as PSO_g^{HJ}, PSO_g^{SW}, for the global case, and $\text{PSO}_\ell^{\text{HJ}}$, $\text{PSO}_\ell^{\text{SW}}$, for the local case, respectively. For each variant, the two schemes described at the end of Sect. 3 for the maximum number of function evaluations assigned to each local search, were considered. More specifically, in the first (standard) scheme, a fixed maximum number, $\text{LS}_{\max}^{\text{st}}$, of function evaluations is assigned to each local search. The value $\text{LS}_{\max}^{\text{st}} = 20$ was used for TP1 and TP2, while $\text{LS}_{\max}^{\text{st}} = 60$ was used for TP3 and TP4, due to their increased difficulty. In the second (adaptive) scheme, an initial value $\text{LS}_{\text{ini}}^{\text{ad}} = 20$ was used, and it was dynamically modified within the range [5, 100], using increments or decrements equal to $\Delta_{\text{LS}} = 5$, as described in Sect. 3. Regarding the PSO component of MPSO, the default set of parameters due to Clerc and Kennedy (2002), $\chi = 0.729$, $c_1, c_2 = 2.05$, was used. For the local variants, the ring neighborhood topology of PSO with radius $r = 1$ was used.

MPSO was compared with three different metaheuristics that have been proposed for FCMs learning in the relative literature, namely, the global and local variant of the standard PSO, denoted as PSO_g and PSO_ℓ, respectively, as well a DE-based and a GA-based approach. For PSO, the same parameters with MPSO were used. For the DE approach, we considered the six basic operators (Plagianakos and Vrahatis 2002; Storn and Price 1997) with parameters $F = 0.6$ and $CR = 0.8$. This setting was selected among others due to its enhanced efficiency. Regarding the GAs, the approach reported in Herrera et al. (1998) and Stach et al. (2005) was closely followed, with simple crossover, non-uniform mutation and roulette wheel selection. The crossover rate was equal to $C_r = 0.6$, while the mutation rate was equal to $M_r = 0.5$. The non-uniform mutation is described in Michalewicz (1999) and it correlates the magnitude of the mutation for each component of an individual, with the number of iterations. As a result, this property favors better exploration in the initial iterations of the GA, and local fine-tuning in the later stages. This setting was selected among others, due to its nice performance (Herrera et al. 1998). All algorithms and their parameters are summarized in Table 5.

The desired accuracy in all cases was equal to 10^{-8}. For each test problem, 100 experiments were performed per algorithm and for different values of the swarm size, equal to 20, 40, 60 and 80, in order to investigate the performance scaling

Table 5 Summary of the employed algorithms

Algorithm	Notation	Description	Parameters
Memetic PSO	PSO_g^{HJ}	Global variant of Memetic PSO with HJ local search	$\chi = 0.729$ $c_1 = c_2 = 2.05$
	$\text{PSO}_\ell^{\text{HJ}}$	Local variant of Memetic PSO with HJ local search	$\text{LS}_{\max}^{\text{st}} = 20, 60$ $\text{LS}_{\text{ini}}^{\text{ad}} = 20$
	PSO_g^{SW}	Global variant of Memetic PSO with SW local search	$5 \leqslant \text{LS}^{\text{ad}} \leqslant 100$ $\Delta_{\text{LS}} = 5$
	$\text{PSO}_\ell^{\text{SW}}$	Local variant of Memetic PSO with SW local search	
Standard PSO	PSO_g PSO_ℓ	Global variant of Standard PSO Local variant of Standard PSO	$\chi = 0.729$ $c_1 = c_2 = 2.05$
DE	DE_1	rand/1/bin	
	DE_2	modified rand/1/bin	$F = 0.6$
	DE_3	best/1/bin	$CR = 0.8$
	DE_4	rand-to-best/1/bin	
	DE_5	best/2/bin	
	DE_6	rand/2/bin	
GA	GA	Simple crossover, non-uniform mutation with roulette wheel selection	$C_r = 0.6$ $M_r = 0.5$

Table 6 Results of MPSO and PSO for TP1 (industrial tank-valves problem)

SS		PSO_g	PSO_g^{HJ}	PSO_g^{SW}	PSO_ℓ	PSO_ℓ^{HJ}	PSO_ℓ^{SW}
20	Suc	77	100/100	100/100	97	100/100	100/100
	Mean	153.5	**60.5/62.2**	117.7/141.5	220.4	62.1/64.9	141.0/134.1
	SD	48.5	**20.0/30.6**	78.7/110.3	70.0	25.0/37.5	84.6/ 98.3
	Min	80	47/47	58/57	100	47/47	58/56
	Max	340	142/ 174	537/681	460	188/290	417/548
40	Suc	81	100/100	100/100	100	100/100	100/100
	Mean	273.1	**99.1**/105.7	199.5/208.2	368.0	101.1/**103.0**	198.3/210.0
	SD	120.5	**22.9**/41.7	155.2/ 132.2	91.5	30.3/**35.3**	100.0/135.0
	Min	160	87/87	97/97	200	87/87	97/98
	Max	1080	212/ 273	998/658	680	223/ 273	518/760
60	Suc	91	100/100	100/100	100	100/100	100/100
	Mean	385.7	**143.9/138.8**	247.0/250.6	498.0	145.0/154.9	274.0/252.0
	SD	162.8	**34.9/26.9**	136.3/171.4	124.9	41.9/76.2	136.0/143.6
	Min	240	127/127	138/137	300	126/126	138/137
	Max	1500	302/287	939/1252	900	387/659	620/766
80	Suc	91	100/100	100/100	100	100/100	100/100
	Mean	489.7	182.9/**179.0**	329.9/300.3	599.2	**175.6**/188.9	334.9/342.2
	SD	144.0	42.0/**31.8**	209.3/144.0	145.4	**21.6**/74.0	171.1/187.1
	Min	320	166/165	178/176	320	165/167	178/178
	Max	1120	372/367	1577/894	960	282/667	978/884

Table 7 Results of DE and GAs for TP1 (industrial tank-valves problem)

PS		DE_1	DE_2	DE_3	DE_4	DE_5	DE_6	GA
20	Suc	100	96	99	94	100	100	100
	Mean	396.0	194.6	205.9	408.9	**173.0**	373.4	428.8
	SD	99.6	**56.3**	103.6	254.7	89.1	104.2	97.2
	Min	240	80	80	140	40	120	260
	Max	820	360	840	1500	840	660	751
40	Suc	100	100	100	100	100	100	100
	Mean	657.6	319.2	256.0	482.0	**249.6**	663.6	704.8
	SD	155.5	**75.2**	96.5	157.1	84.0	204.9	113.2
	Min	280	120	120	240	80	160	474
	Max	1000	480	920	1320	480	1120	929
60	Suc	100	100	100	100	100	100	100
	Mean	900.6	436.8	322.2	588.6	**313.2**	810.6	933.9
	SD	201.4	108.6	**89.7**	172.6	92.9	240.7	154.1
	Min	360	120	180	300	120	240	609
	Max	1380	660	600	1200	660	1260	1394
80	Suc	100	100	100	100	100	100	100
	Mean	1116.8	539.2	381.6	696.8	**352.0**	1043.2	1157.3
	SD	248.3	124.9	**92.3**	214.3	101.1	299.5	176.9
	Min	560	240	160	320	240	320	826
	Max	1600	880	640	1520	800	1760	1742

under different swarm sizes. Each algorithm was applied for a maximum of 120,000 function evaluations.

All results are reported in Tables 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17. The first column of each table denotes the swarm (or population) size, denoted as SS (or PS, respectively). In the rest of the table, and for each algorithm, we report the number of successful experiments (i.e., where the desired accuracy was achieved), as well as its mean, standard

Table 8 Results of Overall Best Algorithms for TP1 (industrial tank-valves problem)

PS		Suc	Mean	SD	Min	Max
20	PSO_ℓ	97	220.4	70.0	100	460
	PSO_g^{HJ}	100	**60.5**	20.0	47	142
	DE_5	100	173.0	89.1	40	840
	GA	100	428.8	97.2	260	751
40	PSO_ℓ	100	368.0	91.5	200	680
	PSO_g^{HJ}	100	**99.1**	22.9	87	212
	DE_5	100	249.6	84.0	80	480
	GA	100	704.8	113.2	474	929
60	PSO_ℓ	100	498	124.9	300	900
	PSO_g^{HJ}	100	**138.8**	26.9	127	287
	DE_5	100	313.2	92.9	120	660
	GA	100	933.9	154.1	609	1394
80	PSO_ℓ	100	599.2	145.4	320	960
	PSO_ℓ^{HJ}	100	**175.6**	21.6	165	282
	DE_5	100	352	101.1	240	800
	GA	100	1157.3	176.9	826	1742

deviation, minimum and maximum number of required function evaluations for the successful experiments, including the function evaluations required by the local search. Unsuccessful experiments were excluded since the expected behavior of each algorithm was under investigation.

For each memetic algorithm, the statistics of the standard and the adaptive scheme are given in the same column of the table, divided by a slash. For example, in Table 6, the mean number of function evaluations for the global MPSO with HJ is given in the column under "PSO_g^{HJ}" and it is equal to 60.5 for the standard memetic scheme and 62.2 for the corresponding adaptive memetic scheme. Also, the lowest mean number and standard deviation that correspond to a completely successful algorithm (i.e., it did not fail in any experiment), are boldfaced in the table both for the standard and the adaptive memetic schemes. The same holds also for the tables with the DE and GAs results. In order to facilitate comparisons between the algorithms, Tables 8, 11, 14 and 17 contain the statistical information of the best algorithm per approach, i.e., best MPSO, best PSO, best DE, and the GA, for each test problem and population size. Again, the overall best approach is boldfaced in these tables.

Furthermore, statistical tests were conducted to justify the significance of the performance differences among the algorithms. For this purpose, the non-parametric Wilcoxon rank sum test was used at significance level 99%, for comparing the best performing MPSO approach with the best performing PSO, DE, as well as the GA, one by one, per problem and swarm size. Again, the best performing algorithm was defined as the algorithm that had 100 successful experiments and the lowest mean number of function evaluations. For example, in TP1 and for swarm size 20, the best performing MPSO approach is PSO_g^{HJ} with 100 successful

Table 9 Results of MPSO and PSO for TP2 (radiation therapy problem)

SS		PSO_g	PSO_g^{HJ}	PSO_g^{SW}	PSO_ℓ	PSO_ℓ^{HJ}	PSO_ℓ^{SW}
20	Suc	17	100/100	100/100	44	100/100	100/100
	Mean	282.4	**99.4**/116.3	1860.3/1400.1	563.6	107.5/**102.4**	1051.6/1072.1
	SD	75.5	**54.1**/73.9	2733.4/1692.7	184.1	87.0/**64.2**	1684.8/976.9
	Min	180	52/58	97/92	240	56/57	58/59
	Max	420	396/340	13898/8056	1100	486/340	9736/5432
40	Suc	24	100/100	100/100	91	100/100	100/100
	Mean	506.7	**173.8/177.8**	2448.5/1235.3	1005.7	173.9/184.9	887.1/1085.5
	SD	266.6	**108.2/89.9**	3619.5/1334.1	325.1	127.2/109.7	705.4/827.2
	Min	280	96/96	100/100	400	94/96	159/100
	Max	1280	534/445	20798/5592	2040	853/582	5858/4245
60	Suc	34	100/100	100/100	94	100/100	100/100
	Mean	742.9	232.0/**244.2**	2579.9/1531.6	1386.4	**203.9**/254.0	1220.4/1250.9
	SD	318.2	111.8/**133.7**	4850.5/1584.8	401.6	**109.7**/142.4	670.6/692.6
	Min	360	136/136	139/139	600	136/136	140/140
	Max	1500	631/615	28138/10240	2940	899/711	4859/3714
80	Suc	30	100/100	100/100	100	100/100	100/100
	Mean	920.0	**332.9/308.4**	3428.7/2087.5	1725.6	367.2/352.1	1249.7/1451.5
	SD	518.8	**179.4/137.8**	6442.6/2437.1	532.6	361.9/244.8	628.1/919.0
	Min	480	176/176	179/179	800	176/174	279/180
	Max	2720	1012/675	48178/13446	4560	2047/1381	2879/4952

Table 10 Results of DE and GAs for TP2 (radiation therapy problem)

PS		DE_1	DE_2	DE_3	DE_4	DE_5	DE_6	GA
20	Suc	99	77	62	8	100	100	100
	Mean	1636.4	340.3	1863.5	1775.0	**712.2**	2109.2	2821.4
	SD	216.6	91.7	3438.4	673.3	**159.4**	324.3	1252.2
	Min	1000	180	340	1240	300	1100	1517
	Max	2140	820	18460	3340	1400	3160	6341
40	Suc	100	97	98	92	100	100	100
	Mean	3211.2	576.1	1318.4	3421.7	**1180.0**	4848.4	3460.1
	SD	365.9	120.6	942.1	3315.3	**221.6**	550.8	388.9
	Min	2280	240	440	1560	480	3400	2794
	Max	3960	1080	7600	27120	2120	6320	4381
60	Suc	100	100	100	100	100	100	100
	Mean	4971.0	**795.0**	1164.0	3414.6	1568.4	7209.0	4501.5
	SD	520.4	**136.0**	359.3	1801.6	274.0	831.8	489.3
	Min	3300	420	600	2040	660	4620	3365
	Max	6000	1140	3000	17040	2280	8820	6347
80	Suc	100	99	100	100	100	100	100
	Mean	6623.2	1006.1	**1548.0**	3713.6	1804.8	9669.6	5421.7
	SD	661.9	155.5	771.0	786.6	**388.7**	1028.5	537.7
	Min	4560	640	720	2720	880	5120	3634
	Max	8240	1440	5920	7920	2560	11520	6779

Table 11 Results of Overall Best Algorithms for TP2 (radiation therapy problem)

PS		Suc	Mean	SD	Min	Max
20	PSO_ℓ	44	563.6	184.1	240	1100
	PSO_g^{HJ}	100	**99.4**	54.1	52	396
	DE_5	100	712.2	159.4	300	1400
	GA	100	2821.4	1252.2	1517	6341
40	PSO_ℓ	91	1005.7	325.1	400	2040
	PSO_g^{HJ}	100	**173.8**	108.2	96	534
	DE_5	100	1180.0	221.6	480	2120
	GA	100	3460.1	388.9	2794	4381
60	PSO_ℓ	94	1386.4	401.6	600	2940
	PSO_ℓ^{HJ}	100	**203.9**	109.7	136	899
	DE_2	100	795.0	136.0	420	1140
	GA	100	4501.5	489.3	3365	6347
80	PSO_ℓ	100	1725.6	532.6	800	4560
	PSO_ℓ^{HJ}	100	**308.4**	137.8	176	675
	DE_3	100	1548.0	771.0	720	5920
	GA	100	5421.7	537.7	3634	6779

experiments and mean function evaluations equal to 60.5. Thus, this approach is tested statistically through the Wilcoxon rank sum test with PSO_ℓ, which is the best performing PSO for the same problem and swarm size, as well as with the best performing DE approach, DE_5, and the GA, independently. The results are reported in Table 18, with the "+" sign denoting significantly different results between the memetic approach specified in the row and the corresponding algorithm at each column. Also, at each row of Table 18, the MPSO approach that is compared with the rest of the algorithms is reported, with "PSO_g^{HJ} ad." denoting the adaptive variant of PSO_g^{HJ}.

In the 8-dimensional industrial control problem (TP1), MPSO outperformed clearly all standard PSO, DE and GA methods as reported in Tables 6, 7 and 8. All memetic schemes were completely successful, requiring the lowest amount of function evaluations. The global variants were marginally better than their local counterparts, with HJ-based schemes requiring significantly lower amounts of function evaluations than the SW-based schemes. The adaptive scheme worked favorably only for the HJ-based approaches and for high swarm sizes, while, for the rest cases, it required a slightly higher amount of function evaluations to achieve the desirable accuracy.

Thus, for swarm sizes equal to 20 and 40, PSO_g^{HJ} was the best algorithm, while for 60 and 80, the adaptive PSO_g^{HJ} was most efficient, requiring just a small fraction, around 30%, of the function evaluations required by the established PSO learning schemes. In some cases, the standard PSO learning schemes, PSO_g and PSO_ℓ, suffered search stagnation, especially when small swarm sizes were used, due to the high desirable accuracy. Regarding DE, the DE_5 strategy was the most

Table 12 Results of MPSO and PSO for TP3 (heat exchange problem)

SS		PSO_g	PSO_g^{HJ}	PSO_g^{SW}	PSO_ℓ	PSO_ℓ^{HJ}	PSO_ℓ^{SW}
20	Suc	7	100/100	100/100	72	100/100	100/100
	Mean	177.1	**65.5**/147.5	969.9/988.8	385.6	66.3/**137.2**	816.1/827.6
	SD	68.7	10.7/100.4	931.3/812.6	148.3	**10.6/103.0**	660.7/972.3
	Min	100	57/57	98/57	100	57/57	98/60
	Max	300	95/336	5533/3840	940	95/474	4335/8097
40	Suc	17	100/100	100/100	87	100/100	100/100
	Mean	395.3	105.0/173.5	1052.7/1394.3	689.2	**102.3/171.3**	1004.3/847.9
	SD	131.8	9.9/**125.9**	1012.5/1263.5	255.6	**9.1**/133.8	673.9/581.2
	Min	160	97/97	136/96	200	97/97	136/97
	Max	640	120/403	5538/8458	1520	137/620	3338/2784
60	Suc	22	100/100	100/100	97	100/100	100/100
	Mean	523.6	142.9/214.8	1265.5/1526.4	968.0	**142.0/211.3**	1021.4/956.5
	SD	231.4	**8.8/145.9**	1154.0/1360.1	347.9	8.8/151.4	644.0/524.8
	Min	240	137/137	174/138	240	137/137	176/140
	Max	1260	160/542	5694/6548	1860	160/755	3418/3203
80	Suc	21	100/100	100/100	100	100/100	100/100
	Mean	796.2	181.7/**254.7**	1648.0/1629.6	1230.4	**181.6**/257.3	1152.2/1226.8
	SD	360.0	8.7/**173.9**	1495.9/1350.7	467.6	**8.3**/187.2	641.6/771.8
	Min	400	177/175	219/179	480	177/177	217/178
	Max	1760	216/682	8339/6953	3040	200/866	2599/3966

Table 13 Results of DE and GAs for TP3 (heat exchange problem)

PS		DE_1	DE_2	DE_3	DE_4	DE_5	DE_6	GA
20	Suc	100	71	79	53	100	99	100
	Mean	643.8	269.0	800.8	909.8	**341.4**	661.6	796.6
	SD	182.2	79.6	1733.6	629.5	226.5	152.7	**150.5**
	Min	240	80	100	320	120	180	481
	Max	1820	500	13880	3420	2160	1040	1250
40	Suc	100	96	100	98	100	100	100
	Mean	1182.4	462.9	679.6	1918.4	**446.0**	1278.0	1196.9
	SD	189.3	94.8	589.7	4011.3	**139.4**	253.3	189.8
	Min	760	240	200	560	160	760	666
	Max	1520	640	3680	37800	920	1880	1659
60	Suc	100	100	100	100	100	100	100
	Mean	1701.0	654.0	714.0	1487.4	**561.6**	1770.0	1631.4
	SD	295.2	**138.2**	412.0	923.7	171.5	392.0	198.4
	Min	960	300	300	480	240	780	1210
	Max	2460	960	3240	7200	1020	2520	2089
80	Suc	100	100	100	100	100	100	100
	Mean	2132.8	822.4	754.4	1627.2	**664.8**	2283.2	1991.3
	SD	393.4	**182.3**	337.0	653.9	189.8	507.8	270.8
	Min	960	480	240	800	240	640	1097
	Max	3040	1200	2560	5520	1200	3360	2702

efficient for all population sizes, although DE_2 and DE_3 were better in terms of robustness, having slightly smaller standard deviations. The GA was successful in all experiments, although at the cost of excessive number of function evaluations. As expected, the statistical hypothesis tests reported in Table 18, confirm the significance of the performance difference among these algorithms and the best performing MPSO.

Table 14 Results of Overall Best Algorithms for TP3 (heat exchange problem)

PS		Suc	Mean	SD	Min	Max
20	PSO_ℓ	72	385.6	148.3	100	940
	PSO_g^{HJ}	100	**65.5**	10.7	57	95
	DE_5	100	341.4	226.5	120	2160
	GA	100	796.6	150.5	481	1250
40	PSO_ℓ	87	689.2	255.6	200	1520
	PSO_ℓ^{HJ}	100	**102.3**	9.1	97	137
	DE_5	100	446.0	139.4	160	920
	GA	100	1196.9	189.8	666	1659
60	PSO_ℓ	97	968.0	347.9	240	1860
	PSO_ℓ^{HJ}	100	**142.0**	8.8	137	160
	DE_5	100	561.6	171.5	240	1020
	GA	100	1631.4	198.4	1210	2089
80	PSO_ℓ	100	1230.4	467.6	480	3040
	PSO_ℓ^{HJ}	100	**181.6**	8.3	177	200
	DE_5	100	664.8	189.8	240	1200
	GA	100	1991.3	270.8	1097	2702

In the 12-dimensional radiation therapy problem (TP2), the algorithms exhibited similar behavior with TP1, as reported in Tables 9, 10 and 11. Overall, PSO_g^{HJ} was the best scheme, with its adaptive counterpart performing better in cases of high swarm sizes. The local variant, PSO_ℓ^{HJ}, was better than the rest memetic approaches, only for the case of swarm size 60.

In contrast to TP1, the SW-based schemes were not also better than the rest of the algorithms, indicating that the selection of the local search algorithm can be crucial for the performance of memetic algorithms in FCMs learning problems. However, we must notice the increased efficiency of the adaptive SW-based schemes, compared to their standard counterparts, which implies the ability of the adaptive scheme to decrease the required number of function evaluations when an excessive number is needed from the standard scheme, for achieving the desirable accuracy. The remarkable difference in the performance between the two local search methods can be attributed to their different nature, since HJ is deterministic, while SW is a stochastic algorithm.

Regarding the rest of the algorithms, DE_5 was better for small swarm sizes, while DE_2 was the best scheme for higher swarm sizes. The standard PSO was able to achieve the required accuracy only for the maximum number of particles, while GA was always successful, at the cost of increased number of function evaluations, that were 20–40 times higher than the best memetic approach. The statistical significance reported in Table 18 confirm these differences.

The 10-dimensional heat exchange problem (TP3) proved to be harder than the previous problems, as reported in Tables 12, 13 and 14. The standard PSO_g learning algorithm was not able to compute the solution with the required accuracy, especially with small swarm sizes. The superiority

Table 15 Results of MPSO and PSO for TP4 (eco-industrial park problem)

SS		PSO_g	PSO_g^{HJ}	PSO_g^{SW}	PSO_ℓ	PSO_ℓ^{HJ}	PSO_ℓ^{SW}
20	Suc	20	100/100	100/100	86	100/100	100/100
	Mean	196.0	**285.3**/776.6	5409.0/8502.5	401.6	359.5/**740.1**	3161.4/2699.9
	SD	88.6	**85.9**/506.6	7418.1/11048.7	138.4	158.1/**305.6**	9266.5/5532.7
	Min	100	151/126	99/94	120	212/243	100/60
	Max	480	768/2252	36415/68384	780	1166/2202	78015/32093
40	Suc	31	100/100	100/100	99	100/100	100/100
	Mean	361.3	**355.6/852.3**	8654.5/8243.2	753.9	462.6/859.0	1447.4/1706.5
	SD	160.2	**73.4**/481.4	11301.3/11136.9	247.7	178.1/**232.0**	871.3/4923.6
	Min	200	195/261	139/99	400	240/320	138/99
	Max	840	546/2974	73138/80297	1680	1280/1388	5137/38753
60	Suc	43	100/100	100/100	100	100/100	100/100
	Mean	477.2	**412.2/1029.9**	9898.9/9860.6	1026.6	566.5/1077.2	1413.5/1048.2
	SD	152.1	**83.5**/508.7	15841.1/12749.8	262.6	198.2/**267.9**	722.2/575.6
	Min	240	247/286	180/212	480	287/450	180/213
	Max	840	747/3092	102057/67430	1620	1372/1929	3778/2756
80	Suc	36	100/100	99/100	100	100/100	100/100
	Mean	657.8	**473.7/1031.3**	8056.5/11339.9	1295.2	676.8/1245.6	1718.9/1206.0
	SD	210.5	**77.4**/489.7	14051.4/15277.9	307.2	221.8/**437.7**	844.4/527.7
	Min	320	323/367	215/180	640	452/377	219/179
	Max	1600	742/3315	67136/60838	2000	1467/3233	4277/3271

Table 16 Results of DE and GAs for TP4 (eco-industrial park problem)

PS		DE_1	DE_2	DE_3	DE_4	DE_5	DE_6	GA
20	Suc	100	61	59	16	99	100	100
	Mean	**1099.4**	347.2	2109.8	1733.8	605.9	1252.0	1513.0
	SD	**221.2**	85.8	3380.1	669.8	226.4	234.1	288.2
	Min	640	140	340	940	240	800	923
	Max	2000	760	16360	3060	1340	2400	2298
40	Suc	100	95	94	96	100	100	100
	Mean	2019.6	630.3	1545.1	2985.0	**839.2**	2529.2	2126.1
	SD	228.4	145.5	2304.6	2797.5	**215.2**	380.8	337.9
	Min	1240	360	400	1040	320	1600	1304
	Max	2480	1040	20680	23760	1600	3440	3226
60	Suc	100	96	99	100	100	100	100
	Mean	2929.2	890.6	1480.0	3474.6	**1049.4**	3904.2	2710.7
	SD	312.8	165.7	1377.7	4473.9	**223.1**	524.6	294.3
	Min	1800	420	480	1140	480	2340	1956
	Max	3540	1200	10980	38400	1920	5040	3496
80	Suc	100	99	99	100	100	100	100
	Mean	3864.8	1049.7	1695.4	3516.0	**1284.0**	5297.6	3356.1
	SD	480.0	184.7	1448.3	2691.7	**286.2**	578.5	389.0
	Min	2080	640	560	1680	560	3760	2594
	Max	4800	1520	10800	20400	2080	6720	4273

Table 17 Results of Overall Best Algorithms for TP4 (eco-industrial park problem)

PS		Suc	Mean	SD	Min	Max
20	PSO_ℓ	86	401.6	138.4	120	780
	PSO_g^{HJ}	100	**285.3**	85.9	151	768
	DE_1	100	1099.4	221.2	640	2000
	GA	100	1513.0	288.2	923	2298
40	PSO_ℓ	99	753.9	247.7	400	1680
	PSO_g^{HJ}	100	**355.6**	73.4	195	546
	DE_5	100	839.2	215.2	320	1600
	GA	100	2126.1	337.9	1304	3226
60	PSO_ℓ	100	1026.6	262.6	480	1620
	PSO_g^{HJ}	100	**412.2**	83.5	247	747
	DE_5	100	1049.4	223.1	480	1920
	GA	100	2710.7	294.3	1956	3496
80	PSO_ℓ	100	1295.2	307.2	640	2000
	PSO_g^{HJ}	100	**473.4**	77.4	323	742
	DE_5	100	1248.0	286.2	560	2080
	GA	100	3356.1	389.0	2594	4273

of the PSO_g^{HJ} scheme was equalized in this problem by its local counterpart, due to the requirement for more intense search. The performance of the HJ-based schemes proved to be significantly different than the best performing of the rest algorithms, as reported in Table 18. We must notice that all HJ-based schemes had very small standard deviations, compared to the rest of the algorithms, with PSO_g^{HJ} being more robust. Again, the global SW-based scheme proved to be less efficient, especially for small swarm sizes, although its local variant had remarkably better results. However, we must notice that it never failed to achieve the requested accuracy. Also, the minimum number of function evaluations required by PSO_g^{SW} was competitive to that of the most efficient HJ-based schemes. Regarding the rest of the algorithms, DE_5 was the best DE approach, while GAs exhibited similar performance to the previous test problems.

In the 40-dimensional eco-industrial park problem (TP4), PSO_g^{HJ} was again the most efficient scheme, requiring almost half the function evaluations required by the standard PSO_ℓ, as reported in Tables 15, 16 and 17. This claim is supported also by the statistical test reported Table 18. The adaptive scheme proved to be valuable both for the local and global SW-based schemes, increasing noticeably its efficiency compared to its standard counterpart. However, the performance of its global variant was poor, compared with PSO_ℓ with large swarm sizes, as well as with some DE strategies and the GAs.

Overall, the HJ-based memetic schemes were shown to be the best algorithms for FCMs learning in the studied test problems. The SW-based schemes were also very effective, achieving the required accuracy in every case, although at a higher computational cost. The proposed adaptive scheme

Table 18 Statistical significance tests among the best performing algorithms per problem and swarm size

Test Problem	SS	MPSO	PSO	DE	GA
1	20	PSO_g^{HJ}	+	+	+
	40	PSO_g^{HJ}	+	+	+
	60	PSO_g^{HJ} ad.	+	+	+
	80	PSO_g^{HJ} ad.	+	+	+
2	20	PSO_g^{HJ}	+	+	+
	40	PSO_g^{HJ}	+	+	+
	60	PSO_ℓ^{HJ}	+	+	+
	80	PSO_g^{HJ} ad.	+	+	+
3	20	PSO_g^{HJ}	+	+	+
	40	PSO_ℓ^{HJ}	+	+	+
	60	PSO_ℓ^{HJ}	+	+	+
	80	PSO_ℓ^{HJ}	+	+	+
4	20	PSO_g^{HJ}	+	+	+
	40	PSO_g^{HJ}	+	+	+
	60	PSO_g^{HJ}	+	+	+
	80	PSO_g^{HJ}	+	+	+

for the number of function evaluations allocated to the local search, was shown to be very useful in high swarm sizes, reducing the total computational burden of the algorithm. The global memetic variants were more promising than the local variants, perhaps due to the specific nature of the problems. The same was observed also for the DE approaches, where the best performing strategy, DE_5, uses the best individual. Comparisons with established learning algorithms based on the standard PSO, DE, as well as GAs, confirmed the superiority of the proposed memetic approaches in FCMs learning, and statistical tests justified the significance of the results.

6 Conclusions

Two different memetic schemes for FCMs learning were studied extensively on four real-life test cases. The first scheme uses the deterministic local search algorithm of Hooke and Jeeves, while the second employs the stochastic method of Solis and Wets. We also proposed a technique for the adaptation of the number of function evaluations performed for each local search application, based on a success/failure scheme similar to the parameter adaptation scheme of the Solis and Wets algorithm. Variants of the aforementioned methods that utilize local and global PSO as their global search component were applied on four real life applications, and compared with standard PSO, DE, and GA-based learning approaches.

Overall, the HJ-based methods proved to be most efficient. This can be attributed to their deterministic nature that promotes refined local search. Also, the adaptive scheme proved to be very useful, especially for the less efficient SW-based variants, mostly in cases with large swarm sizes, while it exhibited competitive results even for the HJ cases. Therefore, it can be considered as a promising alternative that can enhance the performance of mediocre memetic approaches. None of the rest algorithms was able to outperform the HJ-based scheme, for all test problems and different swarm sizes.

The improvement that hybrid algorithms can exhibit in their performance compared to their standard components is always problem-dependent. Therefore, we do not expect that a general MA, with a specific setting, will be the best choice for all optimization problems. However, for the FCMs learning tasks, our results support the claim that the proposed MAs can be very efficient especially in the cases where high accuracy is desirable, requiring minor implementation efforts. Future research will include investigations on different local search algorithms, as well as a more enhanced technique for the online selection of the position of application and frequency of the local search, along with the budget of function evaluations allocated to it.

References

Clerc M (2006) Particle Swarm Optimization. ISTE Ltd

Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73

Corne D, Dorigo M, Glover F (1999) New ideas in optimization. McGraw-Hill, London

Craiger JP, Goodman DF, Weiss RJ, Butler A (1996) Modeling organizational behavior with fuzzy cognitive maps. J Comput Intell Organ 1:120–123

Dawkins R (1976) The selfish gene. Oxford University Press, New York

Fan SKS, Liang YC, Zahara E (2004) Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. Eng Optim 36(4):401–418

Fons S, Achari G, Ross T (2004) A fuzzy cognitive mapping analysis of the impacts of an eco-industrial park. J Intell Fuzzy Syst 15(2):75–88

Georgopoulos V, Malandraki G, Stylios C (2003) A fuzzy cognitive map approach to differential diagnosis of specific language impairment. Artif Intell Med 29(3):261–278

Groumpos PP, Stylios CD (2000) Modelling supervisory control systems using fuzzy cognitive maps. Chaos Solit Fractals 11:329–336

Hart WE (1994) Adaptive global optimization with local search. PhD thesis, University of California, San Diego, USA

Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: operators and tools for the behavioral analysis. Artif Intell Rev 12(4):265–319

Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. J ACM 8:212–229

Juang CF (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern B Cybern 34(2):997–1006

Juang CF, Hsu CH (2005) Temperature control by chip-implemented adaptive recurrent fuzzy controller designed by evolutionary algorithm. IEEE Trans Circuits Syst 25(11):2376–2384

Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann Publishers, San Francisco

Khan F (1994) The physics of radiation therapy. Williams & Wilkins, Baltimore

Khan MS, Khor S, Chong A (2004) Fuzzy cognitive maps with genetic algorithm for goal-oriented decision support. Int J Uncertain Fuzziness Knowledge-based Syst 12:31–42

Kosko B (1986) Fuzzy cognitive maps. Int J Man Mach Stud 24:65–75

Kosko B (1997) Fuzzy engineering. Prentice Hall, New York

Koulouriotis DE, Diakoulakis IE, Emiris DM (2001) Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior. In: Proceedings of 2001 IEEE congress Evol Comp, Seoul, Korea

Koulouriotis DE, Diakoulakis IE, Emiris DM, Antonidakis EN, Kaliakatsos IA (2003) Efficiently modeling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps. Int J Comput Cogn 1(2):41–65

Krasnogor N (2002) Studies on the theory and design space of memetic algorithms. PhD thesis, University of the West of England, Bristol, UK

Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy and design issues. IEEE Trans Evol Comput 9(5):474–488

Land MWS (1998) Evolutionary algorithms with local search for combinatorical optimization. PhD thesis, University of California, San Diego, USA

Lozano M, Herrera F, Krasnogor N, Molina D (2004) Real-coded memetic algorithms with crossover hill-climbing. Evol Comput 12(3):273–302

Merz P (1998) Memetic algorithms for combinatorial optimization: fitness landscapes and effective search strategies. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany

Michalewicz Z (1999) Genetic algorithms + data structures = evolution programs. Springer, Berlin

Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report C3P Report 826, Caltech Concurrent Computation Program, California, USA

Moscato P (1999) Memetic algorithms: a short introduction. In: Corne D, Dorigo M, Glover F (eds) New ideas in optimization. McGraw-Hill, London pp 219–235

Ong YS, Keane AJ (2004) Meta-lamarkian learning in memetic algorithms. IEEE Trans Evol Comput 8(2):99–109

Papageorgiou EI, Parsopoulos KE, Groumpos PP, Vrahatis MN (2004a) Fuzzy cognitive maps learning through swarm intelligence. In: Rutkowski L, Siekmann J, Tadeusiewicz R, Zadeh LA (eds) Lecture Notes in Computer Science (LNAI), vol 3070. Springer, Heidelberg, pp 344–349

Papageorgiou EI, Stylios CD, Groumpos PP (2004b) Active hebbian learning algorithm to train FCMs. Int J Approx Reason 37(3):219–249

Papageorgiou EI, Parsopoulos KE, Stylios CD, Groumpos PP, Vrahatis MN (2005) Fuzzy cognitive maps learning using particle swarm optimization. J Intell Inf Syst 25(1):95–121

Parsopoulos KE, Vrahatis MN (2002a) Initializing the particle swarm optimizer using the nonlinear simplex method. In: Grmela A, Mastorakis NE (eds) Advances in intelligent systems, fuzzy systems, evolutionary computation. WSEAS Press, pp 216–221

Parsopoulos KE, Vrahatis MN (2002b) Recent approaches to global optimization problems through particle swarm optimization. Nat Comput 1(2–3):235–306

Parsopoulos KE, Vrahatis MN (2004) On the computation of all global minimizers through particle swarm optimization. IEEE Trans Evol Comput 8(3):211–224

Parsopoulos KE, Vrahatis MN (2007) Parameter selection and adaptation in unified particle swarm optimization. Math Comput Modell 46(1–2):198–213

Parsopoulos KE, Papageorgiou EI, Groumpos PP, Vrahatis MN (2004a) Recent advances in fuzzy cognitive maps learning using evolutionary computation techniques. In: Proceedings of the 1st international conference “From Scientific Computing to Computational Engineering” (IC-SCCE 2004), Athens, Greece

Parsopoulos KE, Papageorgiou EI, Groumpos PP, Vrahatis MN (2004b) Evolutionary computation techniques for optimizing fuzzy cognitive maps in radiation therapy systems. In: Lecture Notes in Computer Science (LNCS), vol 3102. Springer, Heidelberg, pp 402–413

Petalas YG, Papageorgiou EI, Parsopoulos KE, Groumpos PP, Vrahatis MN (2005) Fuzzy cognitive maps learning using memetic algorithms. In: Proceedings of the international conference of “Computational Methods in Sciences and Engineering” (ICCMSE 2005), pp 1420–1423

Petalas YG, Parsopoulos KE, Papageorgiou EI, Groumpos PP, Vrahatis MN (2007a) Enhanced learning in fuzzy simulation models using memetic particle swarm optimization. In: IEEE 2007 Swarm Intelligence Symposium (SIS’07), Hawaii, USA, pp 16–22

Petalas YG, Parsopoulos KE, Vrahatis MN (2007b) Memetic particle swarm optimization. Ann Oper Res 156(1):99–127

Plagianakos VP, Vrahatis MN (2002) Parallel evolutionary training algorithms for “hardware-friendly” neural networks. Nat Comput 1(2–3):307–322

Rao SS (1992) Optimization: theory and applications. Wiley Eastern, London

Solis F, Wets R (1981) Minimization by random search techniques. Math Oper Res 6:19–30

Stach W, Kurgan L, Pedrycz W (2005) Genetic learning of fuzzy cognitive maps. Fuzzy Sets Syst 153(3):371–401

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

Styblinski MA, Meyer BD (1988) Fuzzy cognitive maps, signal flow graphs, and qualitative circuit analysis. In: Proceedings of 2nd IEEE international conference in neural networks, San Diego, USA, pp 549–556

Stylios CD, Groumpos PP (1998) The challenge of modelling supervisory systems using fuzzy cognitive maps. J Intell Manufact 9:339–345

Stylios CD, Groumpos PP (2000) Fuzzy cognitive maps in modeling supervisory control systems. J Intell Fuzzy Syst 8(2):83–98

Stylios CD, Groumpos PP (2004) Modeling complex systems using fuzzy cognitive maps. IEEE Trans Syst Man Cybern 34:159–165

Stylios CD, Georgopoulos V, Groumpos PP (1999) Fuzzy cognitive map approach to process control systems. J Adv Comp Intell 3(5):409–417

Taber R (1991) Knowledge processing with fuzzy cognitive maps. Expert Syst Appl 2:83–87

Taber R (1994) Fuzzy cognitive maps model social systems. AI Experts 9:8–23

Törn A, Žilinskas A (1989) Global optimization. Springer, Berlin

Trelea IC (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 85:317–325

Wells D, Niederer J (1998) A medical expert system approach using artificial neural networks for standardized treatment planning. Int J Radiat Oncol Biol Phys 41(1):173–182

Willoughby T, Starkschall G, Janjan N, Rosen I (1996) Evaluation and scoring of radiotherapy treatment plans using an artificial neural network. Int J Radiat Oncol Biol Phys 34(4):923–930