Neurocomputing 232 (2017) 58–68

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Interactive evolutionary optimization of fuzzy cognitive maps

Karel Mls[a,*], Richard Cimler[c], Ján Vaščák[b], Michal Puheim[b]

[a] University of Hradec Králové, Hradec Králové, Czechia
[b] Technical University of Košice, Košice, Slovak Republic
[c] Center for Basic and Applied Research (CZAV), University of Hradec Králové, Czechia

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Fuzzy cognitive map
Interactive evolutionary optimization
Expert knowledge

ABSTRACT

Modeling dynamic systems with Fuzzy Cognitive Maps (FCMs) is characterized by the simplicity of the model representation and its execution. Furthermore, FCMs can easily incorporate human knowledge from the given domain. Despite the many advantages of FCMs, there are some drawbacks, too. The quality of knowledge obtained from the domain experts, and any differences and uncertainties in their opinions, has to be improved by different methods. We propose a new approach for handling incompleteness and natural uncertainty in expert evaluation of the connection matrix of a particular FCM. It is based on partial expert estimations and evolutionary algorithms in the role of an expert-driven optimization and outside of the FCM optimization (adaptation) research area known as Interactive Evolutionary Computing (IEC). In the present paper, a modification of IEC for the purposes of FCM optimization is presented, referred to as the IEO-FCM method, i.e., the Interactive Evolutionary Optimization of Fuzzy Cognitive Maps. Experimental results on two control problems suggest that the IEO-FCM method can improve the quality of an FCM even in situations without any measured data necessary for other known learning algorithms.

1. Introduction

There are two basic types of approaches applicable to the analysis of dynamic systems: *quantitative* and *qualitative*. Quantitative methods represent a systematic empirical investigation of the phenomena via statistical, mathematical, or computational techniques [1]. They are typically represented in the form of mathematical models, which ensures their general validity and broad use. Measurement plays a central role in this approach, where the main investigated questions are *what, where, when* or *how much*. However, relations, reasoning, and other intrinsic substances or mechanics, i.e., the questions *why* and *how*, are in the background of interest. These questions are tied to the qualitative methods, which are more difficult to be mathematically modeled. Typical examples of this approach are case studies, which are much more specific than general and thereby their use is limited. In other words, quantitative methods stay on the surface and qualitative methods try to go into depth. This division helps us understand the potential of a given method or what we can expect if we use it. For example, from the many existing techniques, a neural network can be classified as a pure quantitative method because its general structure and the definition of its elements (neurons and connections) as well as most of the learning methods is based on statistics without any regard to the meaning of the values being calculated. On the other hand, a Fuzzy Cognitive Map (FCM) is a method which is much more qualitative than quantitative because it defines the meanings of the map nodes, while their apparent equivalents, i.e., neurons, in neural networks are only simple processing units without any explicit meaning. Of course, this division is quite rough and it does not exclude the concurrent existence of attributes from both approaches in some techniques, but their overall character is determined by the prevailing properties: either qualitative or quantitative.

After a careful evaluation of expected potential of both neural networks and FCMs with regard to the previous statements, we have recognized FCM as more suitable method. Its expected potential is based on the measurement costs and the amount of specialized knowledge from the application domain required in order to develop a correct model (mathematical, case study, etc.). In principle, many complex nonlinear systems cannot be modeled quantitatively at all. Facing real world problems, quantitative modeling is difficult, costly, or even impossible in some cases. Qualitative approaches, however, are free from the above restrictions. Modeling dynamic systems with the use of FCMs (as a qualitative method) is characterized by the simplicity of both the model representation and its execution. Furthermore, FCMs can easily incorporate human knowledge and adapt to a given domain [2].

As FCMs also enable the quantification of the modeling to some

* Corresponding author.
E-mail addresses: karel.mls@uhk.cz (K. Mls), richard.cimler@uhk.cz (R. Cimler), jan.vascak@tuke.sk (J. Vaščák), michal.puheim@tuke.sk (M. Puheim).

http://dx.doi.org/10.1016/j.neucom.2016.10.068
Received 9 May 2016; Received in revised form 16 October 2016; Accepted 27 October 2016
Available online 21 December 2016
0925-2312/

extent, since in addition to qualitative semantic explanations they also provide computing engine similar to neural networks (which are considered as a quantitative method). Therefore they have found use in supervision, control, decision-making support, etc. [3]. Basically, FCMs are used either indirectly as auxiliary means for some special tasks tied to the modeling and prediction, or directly in the control and decision-making. One example of such an indirect application [4] deals with the use of FCMs for predicting time series based on historical data, which do not need to be complete. In [5], it is proposed that interval-valued time series are forecasted by fuzzy grey cognitive maps. The so-called *scenario planning* for energy deployment from wind power plants is solved using FCMs for prediction in [6], where the scenarios represent a set of possible future events, which help to understand a possible chain of causal events and thereby make an early decision about the energy deployment.

However, FCMs are advantageous mainly for direct application to decision-making and supervisory control. In [7], authors treat the design of decentralized management and control of an energy management system. In [8], a decision support system for the hospital admission procedure is proposed. Specifically for supervision, [9] dealing with chemical processes and [10] designing self-tuning PI controllers by FCMs can be mentioned. Robotics offers a variety of suitable tasks where FCMs can be used, such as goal detection [11] or navigation [12]. A detailed study of the potential uses of FCMs in robotics is presented in [13].

One of the most crucial problems of FCMs, as with fuzzy systems in general, is their design process. It is even more difficult than for conventional rule-based fuzzy systems because FCMs can be viewed as an extensions of the conventional ones. Thus the variability of the various designs is increased. Later, the process of FCM adaptation (or learning) will be discussed in more detail. Despite the heterogeneity of the methods of adaptation, all of them have one function in common: their suitability for comprehensible system analysis. Before the adaptation method proposes any changes to the FCM, it needs to do some kind of analysis of the observed system (regardless of whether is it used for control, prediction, etc.) employing some criteria. These criteria can be again quantitative or qualitative. However, all known FCM adaptation methods are based only on quantitative criteria, which are in many real cases insufficient. We need to take into consideration that humans participate in the operation of a vast number of various systems (using tacit expert knowledge) and therefore we should accept the fact that there are also some qualitative criteria. For instance, a car driven by a human depends not only on the traffic situation (described by the distances between the significant objects as well as their velocities), but also on an effort to drive comfortably and/or with a certain amount of enjoyment. This view suggests the need to search for an adaptation method that would also take into account qualitative criteria. The description of notions like *comfort* and *enjoyment* may be difficult (or even impossible) by answers to questions such as *what, where, when* or *how much*: it should be done rather by answering *why* or *how*. *Interactive Evolutionary Computing* (IEC) offers such a possibility. As far as we know, there is no prior publication that has used the concept of IEC for the design of FCMs. Thus, we implemented this idea by interconnecting the two concepts of IEC and FCM, which has resulted in a new optimization system, which we will refer to as *Interactive Evolutionary Optimization of Fuzzy Cognitive Maps* (IEO-FCM). We have further tested this system in two control applications.

The remainder of this paper is organized as follows. Section 2 delivers a brief introduction to the theoretical background of FCMs and their properties. Section 3 continues in introducing *interactive evolution* as a method for expert-driven optimization, with a comprehensive overview of the literature. The modification of IEC for the FCM design process is described in Section 4, which presents a novel contribution to FCM adaptation in the form of IEO-FCM. Two model examples are shown in Section 5, with numerical evaluations and some results, namely the design of an FCM for the supervisory control of a mixing tank system, and an FCM-based navigation system for tracking a car along a prescribed trajectory. Lastly, Section 6 presents some concluding remarks and outlooks for further research in this area.

2. Fuzzy cognitive maps

FCMs as formal models of complex dynamic systems can be described by a set of nodes (i.e., concepts) and relations (i.e., connections between concepts). Their nature enables both a schematic description of the basic structural relations of a given dynamic system and its modeling as well as the subsequent investigation of its properties. Originally, only values analogous to binary systems were used, i.e., 0, 1 and -1 for negative relations. However, such a definition was not suitable for the description and modeling of many technical systems. Some kind of extension of the definition was needed. Therefore, Kosko [14] used fuzzy set theory to enrich the representational power of CMs and created the conception of the FCM. Each concept and each relation is accompanied by a numerical value from the interval $[0; 1]$ and $[-1; 1]$, respectively, in this way expressing some expert opinions. The actual evaluation of all the concepts at the time t is called the *state* and is represented by a numerical vector. A broad overview of potential areas of the use of FCMs can be found in [3].

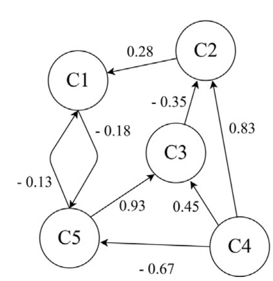
In numerous research papers, the FCM model has been presented in one of its two main ways of representation: either as a graph or as a connection matrix. Both representations give a static picture of the FCM, with its internal structure and the weights of its relations, see Fig. 1.

Representations of an FCM focused on its dynamics are less common. These representations show the change of concept activation values in time. Dynamic changes of the state vector and state space analysis are examples of two of them, see Fig. 2. In our approach, the state vector dynamics of the individuals of an FCM from a particular generation will be proposed as an easily obtained (phenotypic) visual representation of their quality within IEC.

2.1. Dynamic properties of fuzzy cognitive maps

The dynamic behavior of an FCM depends on its relational structure and on the inference mechanism defining the mode of transition from the state at time t to the next state at time $t+1$. There are three possible outcomes to this inference process. If the state vector settles on some stationary value after a finite number of time steps, then a so-called *fixed point* is reached. If the state vector periodically regains an equal state after a finite number of steps, a *limit cycle* is reached. In some cases, the state vector changes its position in the state space chaotically. The state space of the FCM is divided into independent regions (see Fig. 2b) with similar behavioral patterns, where all starting states result in a single attractor (i.e., a fixed point, a limit cycle, or a *chaotic attractor*).

The inference within an FCM, regardless of its representation, is based on the definition of the initial states of the n nodes with their state values $A_i(t)$ at time t=0 and the connection matrix E. If $A(t) = (A_1(t), \ldots, A_n(t))$ is the state vector at time t, then the new states



$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & -0.18 \\ 0.28 & 0 & 0 & 0 & 0 \\ 0 & -0.35 & 0 & 0 & 0 \\ 0 & 0.83 & 0.45 & 0 & -0.67 \\ -0.13 & 0 & 0.93 & 0 & 0 \end{bmatrix}$$

Fig. 1. An illustrative example of the same FCM as an oriented graph (left), and as a connection matrix (right)—the static perspective.

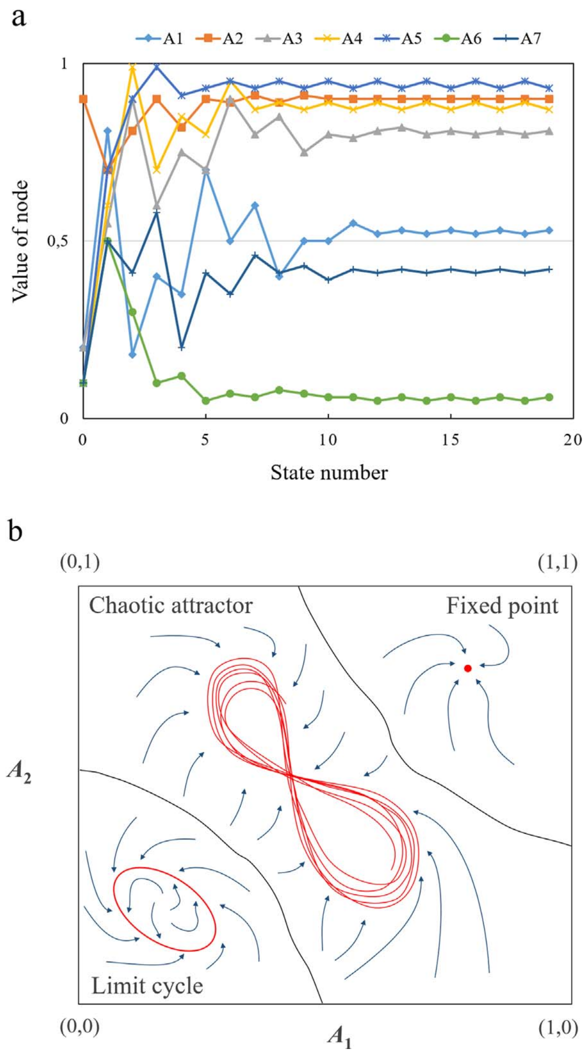


Fig. 2. An illustrative example of state vector changes in time (a), and the state space partition of a FCM (b)—the dynamic perspective.

at time $t+1$ are calculated in one of the following ways:

$$A_i(t+1) = L\left(\sum_{j=1}^{n} e_{ij}\cdot A_j(t)\right), \tag{1}$$

$$A_i(t+1) = L\left(A_i(t) + \sum_{j=1}^{n} e_{ij}\cdot A_j(t)\right), \tag{2}$$

where e_{ij} is the element of the matrix E which expresses the strength of a connection from the node j to the node i and L is the *limiting function* to keep the values of the nodes in an interval [0; 1].

While (1) assumes the FCM model originally introduced by Kosko [14] and considers only the immediate influences of the states for the computation of the new node activation values, Eq. (2) extends the model by computing an increment. In doing so, it establishes a memory factor for the nodes. Usually, self-connections of the nodes are not allowed, i.e., $e_{ii} = 0$, but in some papers, these are also used, with the aim of strengthening the memory factor. At present, the computation by (2) is preferred because of the better stability of the behavior of the FCM.

Such an approach is convenient for modeling a closed process, which is isolated from the measured data of the sensors. However, sensors play a key role in technical systems and hence a modification of the inference mechanism is inevitable for receiving measured data. For this reason, it is necessary to divide FCM nodes into at least two groups: *input* nodes and *output* (*decision*) nodes (eventually *intermediate* ones separated from the latter group if their values are not outputs of the FCM). The modification is quite simple. Instead of applying (2), the values of the input nodes will be substituted by transformed sensory data using membership functions, which are defined for each input node. An example of such an FCM is shown in Section 5.2.

In many papers, the attention is focused on a comparison of FCMs with other means of computational intelligence, especially with knowledge systems based on fuzzy production rules, finite-state automata and mainly Recurrent Neural Networks (RNNs). In [12], it is shown how FCMs can be rewritten into the form of a rule base. Theoretically, a rule base can also involve implication chains and closed loops, but in such cases it becomes complicated and the division into inputs and outputs becomes less clear. An FCM seems to be an extension of a rule base from this aspect. In contrast, FCMs are not 'obliged' to contain such implication chains or closed loops, and in such a case they may resemble a traditional Feedforward Neural Network.

However, the relation between FCMs and neural networks (NN) is much more complicated in general. It is often said that an FCM is a special case of an RNN (as an extension of a feedforward NN), but there are some significant differences. FCMs and RNNs have similar structures (topologies), but functionally, these are not fully equivalent. In [15], a comparison between the two is summarized. Their differences come from a pair of fundamental reasons. Firstly, there is their knowledge interpretability. The neurons in an NN are considered only as computational units. They do not represent any meaning (see the experimental examples in Section 5) and for this reason the obtained function describing such an NN does not reflect any explicit knowledge, i.e., its knowledge interpretability is limited. An NN can perfectly perform its task but we do not understand *why* and *how*. In other words, NNs are typical *quantitative* means. In contrast, in an FCM, each node has its meaning, i.e., its *linguistic concept*, and a clear relation to other nodes. The main reason why FCMs have become popular is their excellent knowledge interpretability for humans. However, if there are more complex connections between the nodes, they must be realized by some auxiliary nodes. As a consequence, FCMs are not universal approximators. This is the toll taken by their extension, compared to NNs. Secondly, there are some structural and inferential differences as well. For instance, an FCM is often regarded as an extension of a Hopfield RNN. However, this kind of RNN requires $e_{ij} = e_{ji}$ in order to converge to a stable solution. Further, the functions of basic types of NNs are not incremental, i.e., they can be described similarly to (1) but most FCMs operate by (2). Such differences in the inference process are examples of why we cannot directly use the learning algorithms of NNs and why we need to apply modifications.

Similarly to other fuzzy logic systems, the need for adaptation (or learning) of an FCM plays an important role in its practical utilization, since the design of an FCM is not a trivial task because of the complexity of the relations within the original modeled system. Therefore, we will provide a short overview of the adaptation methods used for FCMs in Section 2.2.

2.2. Adaptation of FCMs

The original approach to FCM design is based on human knowledge, which is in some way transformed into nodes and weighted connections. Because of the complex structure of an FCM, which may contain decision (implication) chains and closed loops leading to recurrences, its design is a nontrivial task even for experts. Therefore, the problem of its automatic or at least semi-automatic adjustment has become an important part of FCM research over the last 10–15 years. The main attention is put on adjusting the connections, where individual methods enable changing the weights, signs, or even the direction of the connections. Simpler methods require an initial set of existing connections (not yet weighted), and more advanced methods are able to create new connections during the adaptation process. However, the meanings of the nodes have to be defined manually and subsequently, there are methods such as [16] which adjust their transformation functions.

Learning (adaptation) methods of FCMs can be roughly divided into three basic groups [17]: *Hebbian-based, evolutionary-based* and *hybrid*. Hebbian-based learning assumes that if two close map nodes are activated simultaneously and repeatedly, then a connection between them should be strengthened. This principle is very loose and enables a variety of modifications, which have been investigated in many research papers. In short, we can state that only more advanced nonlinear forms of Hebbian-based learning can offer solutions of good quality. A short survey of such methods along with their mutual comparisons can be found in [18].

Hebbian-based learning can be categorized into a group of the so-called *unsupervised learning* methods, which try to find a function describing the knowledge structure of a given problem. Using this type of learning we can, for instance, observe the adaptation process when the system reaches some form of stability but there is no explicit error function, or more generally, no explicit reward function. Therefore, methods of supervised learning are required in many practical applications. Most of the evolutionary-based methods can be assigned to this group because of the existence of a *fitness function*. Fitness functions are usually based on the comparison of the FCM's response to the response of a real system [19]. Another approach is to use the error function directly. Evolutionary algorithms represent a wide variety of methods. An interesting example, which has come to the fore in the last years, is *particle swarm optimization* [20], but there are also other novel relative optimization approaches such as *artificial immune systems* [21] or *dynamic optimization* [22]. In addition to evolutionary optimization, there are also other supervised learning methods, such as modified *backpropagation of error* for RNNs [23], or *extreme learning* [24].

In other words, the task of Hebbian-based learning is the convergence of a system to a given state or prescribed region. The design expert takes the form of Hebbian-based learning. The task of evolutionary-based learning is to optimize the fitness value and the design expert takes the form of the fitness function and determines the final fitness value. Hybrid methods try to combine pros and cons of both or similar approaches into a multi-stage process [25,26].

Summarizing and concluding, all the mentioned methods are able to use only those criteria which can be described analytically. But there are many situations where such an approach cannot be used. The solution to this problem is the objective of the next Section 3.

3. Interactive evolutionary computing

Basically, there are two types of evaluation criteria that are incorporated into fitness functions in some way: *quantitative* and *qualitative*, or, in other words, *explicit* and *implicit* [27]. Conventional evolutionary systems use quantitative criteria, which can be easily represented as analytic functions. However, qualitative criteria are either uncertain (characterized by any type of uncertainty, e.g., probabilistic, fuzzy, etc.) or their formalization is strongly subjective. Examples of such criteria are the evaluation of taste, the beauty of images, the quality of sounds, etc., which are in general known as *aesthetic evaluation*. This kind of evaluation is very typical for humans and their senses and not only in the areas of psychology and perception. We can observe such an approach also in purely technical tasks like driving a car, the navigation of robots [28] and the control of their gait, in data mining, etc. The problem is that such criteria are hardly to be transformed into analytic functions. So there is the necessity to directly incorporate a human evaluator instead of a numerical fitness function into the optimization process, see Fig. 3, which is the core idea of IEC [29].

In other words, IEC is a kind of EC technology, where human(s) and EC cooperatively optimize a target system based on a mapping between the *feature parameter space* (genotype) and the *psychological space* (phenotype). In this sense, we can say that IEC is a technology embedding human preferences, intuitions, emotions, psychological aspects, or (to use a more general term) *kansei*, in the target system. Humans evaluate the proposed solutions, i.e., individuals, according to the distance between the target in their psychological spaces and the actual system outputs, and the EC searches for the global optimum in the feature parameter space according to the psychological distance [30]. This cooperation is provided by an interface, usually a graphical one, which offers to a human the possible solutions in a proper form. The user determines the fitness either as a simple (crisp) number (the traditional IEC approach) or as an interval or even in the form of a rough [31] or fuzzy set [32,33]. Thus the form and offered functionalities of such an interface play a crucial role in the success of the optimization process. It is not only a question of the graphical user-friendliness. This problem is discussed and reviewed in [34].

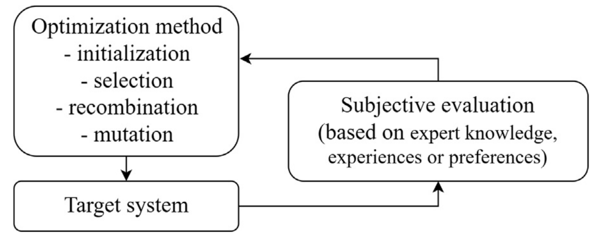


Fig. 3. Structure of an IEC based on subjective evaluation [29].

There can be found lots of applications where it is impossible to express human expectations or requirements in an exact form. Since its spread into practice in the middle of the 90s, IECs have found hundreds of applications in areas such as graphic and animation art, music, speech and image processing, medicine (e.g., hearing aid fitting), virtual reality simulation, data mining and database retrieval, control (including also robotics), and, of course, games and social systems (e.g., education), which were sketched in a still valid review done in [29].

During the last 20 years, some new directions of research, mainly with connection to other computational means, have been investigated and not only the advantages but also some critical drawbacks of IEC have been discovered. Current research tied to IEC can be summarized into the following often mutually interconnected and overlapping areas:

1. human fatigue,
2. modeling of human reasoning,
3. multi-objective (multi-criteria) decision making,
4. modified and hybrid approaches for IEC (e.g., using rough and fuzzy sets).

Human fatigue is the principal limitation of IEC. Usually, the power of EC is based on huge populations' being processed through thousands, sometimes tens of thousands, of generations. However, in this case it is impossible to apply such an approach to human evaluators, so a typical IEC population size is strictly limited by the number of individual images spatially displayed on a computer monitor as an interface and the number of IEC generations does not exceed 10–20. Of course, such a limitation does not guarantee a satisfactory exploration of the parameter target space. Therefore, one possible way is to construct a model of human evaluators or at least of some of their abilities. In [32], fuzzy sets are used to describe human perceptions, so a user need not be confined to using only crisp values during the evaluation process. A fuzzy rule-based system proposed in [35] gives a practical representation of a model of a human evaluator, and offers to the evaluator a pre-evaluated fitness value. So the evaluator can check his/her mental condition. A deep insight is presented in [30], where the problem of finding unknown variables, which define the human evaluation behavior, is discussed with the aim of making an equivalent model of a human evaluator. In [36], a collaborative human–computer strategy is proposed that enables the algorithm to adaptively learn from the human's responses. The system designed in [37] tries to quantify the evaluator's preferences using the so-called *affective model*, by constructing a relation between the evaluator's preferences and the affective space. Another way to reduce the human's fatigue is to divide the evaluation process either between several evaluators using the principle of *collective intelligence* [38], or into several successive stages for one evaluator [39].

Numerous applications require a multi-objective or multi-criteria evaluation approach to globally assess the quality of the proposed solutions. Of course, many of such criteria are contradictory and thereby their exact description in the form of an analytic function becomes very complicated if not impossible. Many methods of multi-objective optimization are based on the *a posteriori principle* and thereby lead to finding the so-called *Pareto front*. The optimization process is then concentrated on the most effective convergence to this front. There are several approaches to achieve such a convergence, e.g., for parameters in the form of intervals [40] or using fuzzy [41] and rough sets [31] for extracting evaluation (decision) if–then rules from a human's preferences. Of course, there are many other approaches to decision-making, like *affective computing* [37], *case-based reasoning* [42], *pattern search* [43], *metaheuristic search* [44], or the already mentioned collaborative human–computer strategy [36], etc. All these methods are combined with IEC optimization and constitute hybrid methods. An overview of them can be found in [45].

Concerning modified IEC systems, there can be mentioned various approaches, which for instance modify the fitness values to the form of linguistic values [41] and fuzzy numbers [33]. Besides, the idea of interactivity can be used in further types of evolutionary algorithms, too. As an example, the so-called *differential evolution* can be mentioned here [35,46].

4. IEO-FCM algorithm description

During the initial phase of a computational process, upper $\overline{W}$ and lower $\underline{W}$ limits of the input connection matrix characterizing the system are determined by the user or computed from individual evaluations obtained from a group of experts [47]. These values correspond to the allowed extreme values of the investigated system. During the process of creating a new generation of matrices, no value is allowed to exceed these limits. The values of the initial state vector $A(0)$ are also set during the initial system setup.

A relevant number of matrices is created due to the requirements of the IEC simulation. It is recommended to create up to 15 matrices in order to keep a simulation overview for the expert evaluation. The initial values of e_{ij}^0 in all initial matrices E^0 are random numbers between the upper and lower limits. New state vectors of particular matrices are then computed using the inference process in the form mentioned in Section 2 with an adjustable number of iterations. Based on the complexity of the system, the stabilization of the state vector values can take different numbers of steps.

The expert evaluator can see the dynamics of all depicted (maximally 15) state vectors $A(t)$ for a determined number of simulation steps $t = 0, 1, 2, \ldots, t_{max}$ in a given kth generation on a Graphical User Interface (GUI) in the form of 2D graphs. The time value t_{max} should be sufficient to enable the stabilization of the computational process. It is adjusted by the expert after doing some experiments and depends only on the dynamic properties of the given system. Based on the expert's experience and knowledge, the expert can choose and prioritize only those graphs that best correspond to the expert's opinions.

The next generation of matrices E^{k+1} $(k = 0, 1, \ldots, k_{max})$ is subsequently created based on the previous generation E^k evaluated by the expert. Matrices highlighted by the expert have a greater chance of being selected for the recombination phase and of participating in the creation of the new population. The new matrices could be created as the average of two chosen parent matrices or as a combination of the elements of chosen parent matrices. Lastly, as a solution, there may be the complete set of connection matrices $E^{k_{max}}$ or a subset of a chosen number given by the expert. The optimization process is finished if the evaluator is satisfied with the result or the number of generations reaches k_{max}. Fig. 4 shows the pseudocode of our IEC algorithm modified for FCM adaptation.

In order to provide a tool for numerical simulations with IEO-FCM as well as for experts' observations of the system development and choosing the most suitable scenarios, a program has been created in MATLAB 2015, both for programming the application and for GUI design as well. The tool can be used for any size of a given problem and is capable of evaluating different FCM dynamics and evolutionary operators.

5. Examples of FCM design using IEO-FCM method

The modified IEO-FCM method for the design of FCM connection matrices was applied to two systems, which differ from each other in some of their principal properties. These differences are easily seen from the depictions of their FCMs, see Fig. 6 and 10. The existence of closed loops indicates the appearance of inertia. Further, we can study the succession of influences from node to node, i.e., how much are the individual values of a system mutually influenced. The first system is a mixing tank, which can be used for preparing liquid mixtures with given concentrations or for controlling temperatures in the case of mixing cold and hot water. Basically, it is a very simple device but its control is complicated due to its inertia. The second system relates to the navigation problem. It is a tracking system, where a car tries to stay in the borders defining its route. It is a special kind of navigation, where the obstacles are in the form of route borders. Real navigation has to take into consideration own decision making providing a route between the obstacles as well as the physical properties of the car and its surroundings. In this case, the physics was omitted for reasons of simplicity, and therefore there are no typical closed loops or other complex connections. However, it is very interesting to see how a structurally simple FCM can solve relatively complex navigation problems in route searching.

5.1. Design of a supervisory control system for a liquid mixing tank

To test the proposed algorithm on a well documented case, the situation from [47] was taken. The technological system (Fig. 5) is

Inputs: $\overline{W}$, $\underline{W}$, vector $A(0)$, t_{max}, k_{max}.
Outputs: A given set (maximally 15) of connection matrices $E^{k_{max}}$.
Initialization: Create a given number (maximally 15) of random matrices E^0 with values e_{ij}^0 within intervals given by $\overline{W}$ and $\underline{W}$.

```
For each generation k (k = 1, ..., k_max)
    For each individual (i.e., each matrix from E^k)
        Calculate series of system state vectors A(t) for t_max steps.
        Store state vectors A(t) for this individual.
    end
    Display state vectors A(t) of all individuals in kth generation.
    Get the evaluation from the expert.
    Combine matrices in E^k based on their evaluation score
    and create a new generation of matrices E^{k+1}.
end
```

Fig. 4. Pseudocode of the IEO-FCM algorithm for FCM optimization.

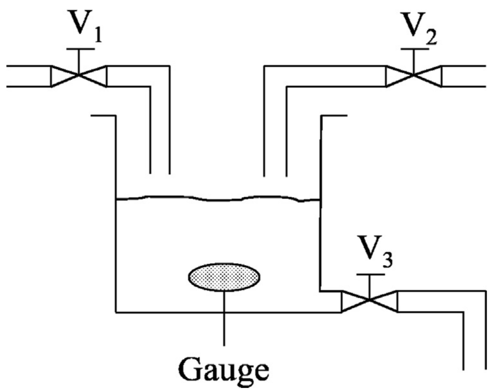


Fig. 5. Schematic depiction of the control system for a mixing tank [47].

composed of a tank, three valves, and a gauge. Two different liquids enter the tank and the goal of the process is to get a new substance by chemical reaction under the specified limit conditions. The control FCM proposed by a group of experts is shown in Fig. 6.

Here, Tank=E1, Valve 1=E2, Valve 2=E3, Valve 3=E4 and Gauge=E5. The corresponding crisp connection matrix E is

$$E = \begin{pmatrix} 0 & -0.4 & -0.25 & 0 & 0.3 \\ 0.36 & 0 & 0 & 0 & 0 \\ 0.45 & 0 & 0 & 0 & 0 \\ -0.9 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0.3 & 0 \end{pmatrix} \quad (3)$$

The initial state vector $A(0)$ characterizing the system at time t=0 was set to

$$A(0) = (0.1 \;\; 0.45 \;\; 0.39 \;\; 0.04 \;\; 0.01) \quad (4)$$

The values in the connection matrices have been perturbed in order to simulate uncertainty in the expert's or group of experts' evaluations. Perturbation means the maximal random difference from the original value. Two scenarios with different levels of estimated uncertainty in the expert evaluation of the connection matrix have been tested in the experiment:

1. non-zero values in the initial population of matrices E were perturbed ± 0.2,
2. non-zero values were perturbed randomly ± 1. Only the polarity was preserved.

Both systems generate stable patterns of behavior. Applying the perturbation ± 0.2 to the values of the elements of the connection matrix has caused only weak changes in the dynamics of the state vectors. Only the theoretically maximal uncertainty in the connection matrix (perturbation ± 1) led to more significant variations in the state space dynamics.

The expert's goal is to get a behavior of the evaluated FCMs identical to the optimal (expected) system development (depicted in Fig. 7) based on the values of A, without seeing the exact values of the connection matrix E. In the course of our experimentation, already in the fifth generation, the IEO-FCM algorithm based on the experts' choices generated very promising results—see, e.g., the following matrix, which is very close to the desired matrix:

$$E_{final} = \begin{pmatrix} 0 & -0.41 & -0.37 & 0.00 & 0.25 \\ 0.51 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.31 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.78 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.67 & 0.00 & 0.17 & 0.00 \end{pmatrix} \quad (5)$$

Fig. 8 illustrates the concrete GUI layout for this simulation. All 15 randomly generated connection matrices yield different state vector dynamics, which are presented to the user. Then any of the matrices may be prioritized by selecting the **Choose** button below the corresponding graph. The **Next generation** button starts the computational phase of the algorithm and a new set of outputs is displayed.

5.2. Design of an FCM-based controller for navigation purposes

In this task we want to show the design process of an FCM based on using IEC for the navigation of a car on an arbitrary trajectory, whose form is usually not known in advance. The goal is to design a controller in the form of an FCM, which would navigate a car along a prescribed trajectory trying to optimize some criteria, both quantitative and qualitative ones, as well. There are many approaches able to optimize quantitative criteria, such as, for instance, to minimize the time or energy consumption, but their designs could lead to behavior that is strange when compared to human behavior. Let us consider the situation of an autonomous car in real traffic. Drivers observe, besides many other indicators, the driving style of the surrounding cars so as to predict their behavior. If one of them were to behave strangely or suspiciously, this would lead to sometimes dangerous misunderstandings. For this reason, such a controller should own 'human thinking,' too. This is the reason to use IEC.

To prove the versatility of the proposed controller design, various forms of trajectories were proposed and tested, i.e., the variability of the curves, their number, and their tightness. The car can change its turning and speed. As with any driver, it can see only the immediate part of the trajectory. Thus the control is done in each sampling step. The controller will try to keep the car on the road as much as possible. As the trajectory also has its borders like a real road, the car can move inside these borders, depicted as gates, and further, we will call such a prescribed trajectory the "road." If the car moves outside these gates, this will be considered as a collision (wrong movement), see Fig. 9. So a

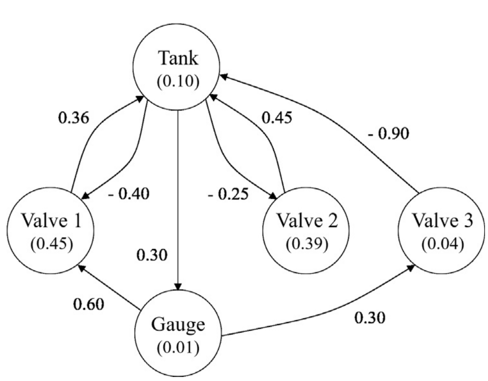


Fig. 6. FCM used for control of a mixing tank [47].

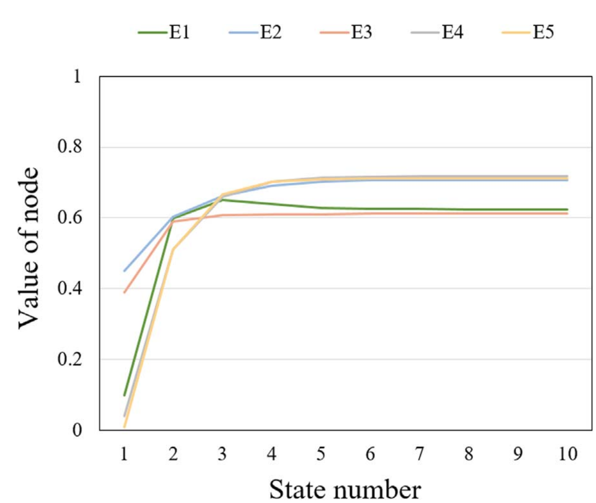


Fig. 7. Optimal system development, which is also the goal of the evolution.

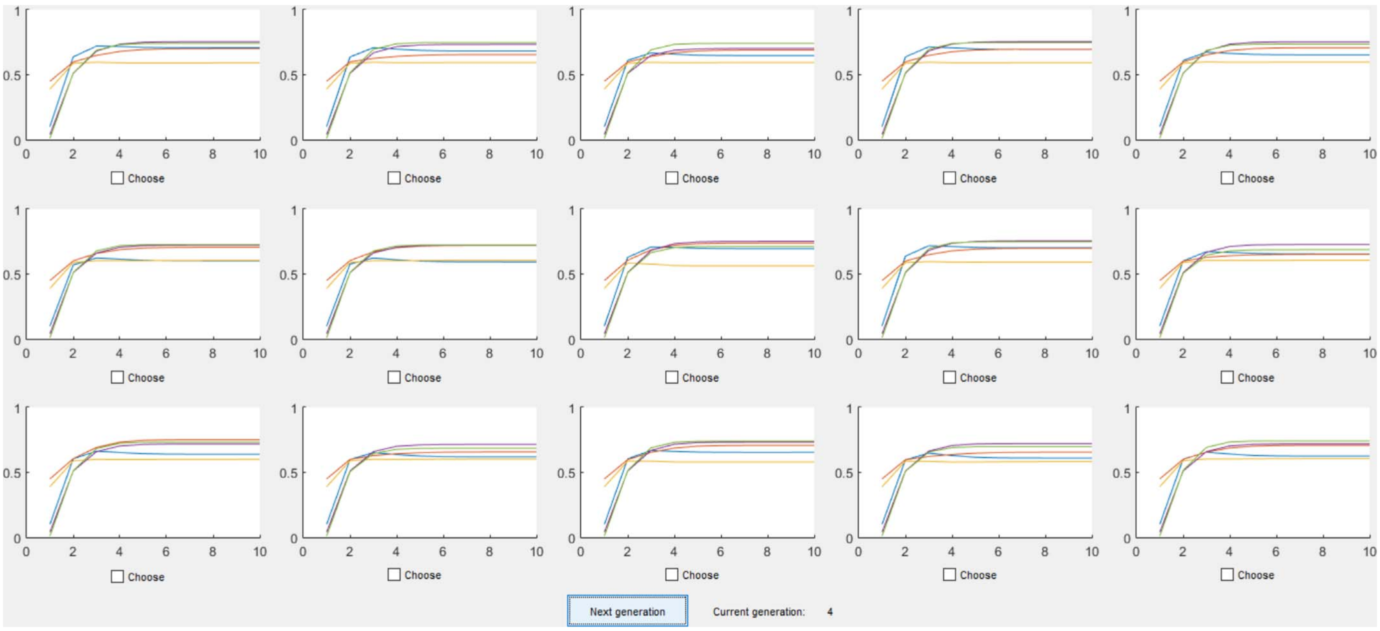


Fig. 8. Graphical user interface showing the state vector dynamics of the initial population of connection matrices generated by a ± 0.2 perturbation of the original matrix. The state number is on the X axis and the activation value of the node is on the Y axis.

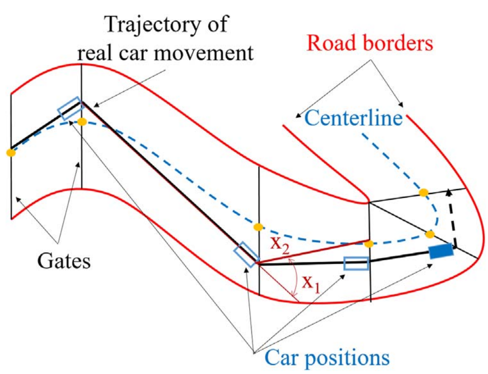


Fig. 9. Depiction of the sampling steps with the road centerline, borders, gates and real car trajectory with indicated inputs of the controller: x_1 is the angular deviation and x_2 is the magnitude of the distance.

car should move only inside the road and it can move in many ways, which we can evaluate using some criteria, see Table 1.

To keep the car in the road, we must know the distance of the car from the closer border (the shorter part of the gate) and the angle of the car's centerline relative to the gate, which are the inputs to the controller. Its outputs are the acceleration (if a negative value, then deceleration) as well as turning. The center of the next gate is the closest particular goal. Thus the car will move from one gate to the next until it reaches the last one, denoted as the final goal. For further considerations, we will need to know the distances and angular positions between the car and the particular goal, because these values will determine the control actions.

The movement between two adjacent gates is performed as follows:

1. The controller calculates the acceleration and turning angle for the car based on the current position of the car and the closest gate center as a particular goal.
2. The car starts its movement and the system checks its prescribed trajectory to the particular goal.
3. If the car intersects the gate, a new control action will be calculated, by repeating step 1.
4. In this way, the car will pass all intermediate goals until it reaches the final one.

The design of the controller in the form of an FCM requires that the user define the nodes of the map. The weighted connections between them will be designed using IEC. Based on the character of the task, its inputs and outputs and the fact that the control objective is the car with its control abilities, the structure of the FCM controller can be proposed in the form depicted in Fig. 10.

There are two inputs: the angular deviation between the car's centerline and the direction to the center of the next gate (the

Table 1
Comparison of experimental results obtained from control of car movement using FCM and ANFIS controllers in terms of criteria C1–C6.

Type	C1	C2	C3	C4	C5	C6
FCM 1	0.756	0.978	0.982	0.664	0.871	1
FCM 2	1	0.934	1	**1**	**0.601**	**0**
FCM 3	0.825	1	0.984	0.816	0.713	1
FCM 4	0.853	0.961	0.978	0.781	0.753	1
FCM 5	0.783	0.978	**0.974**	0.743	0.777	**0**
ANFIS	**0.553**	**0.417**	**0.974**	0.585	1	**0**

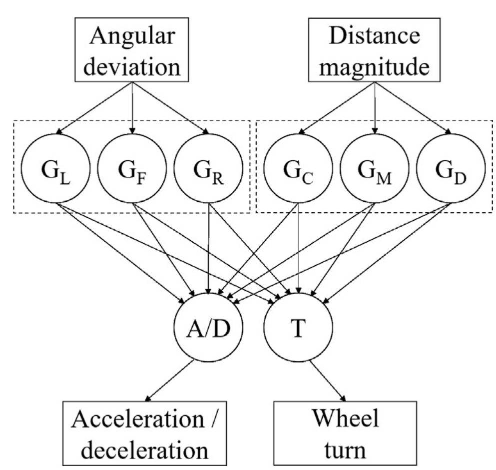


Fig. 10. Proposed structure of an FCM-based controller for navigating a car within the borders of a prescribed route.

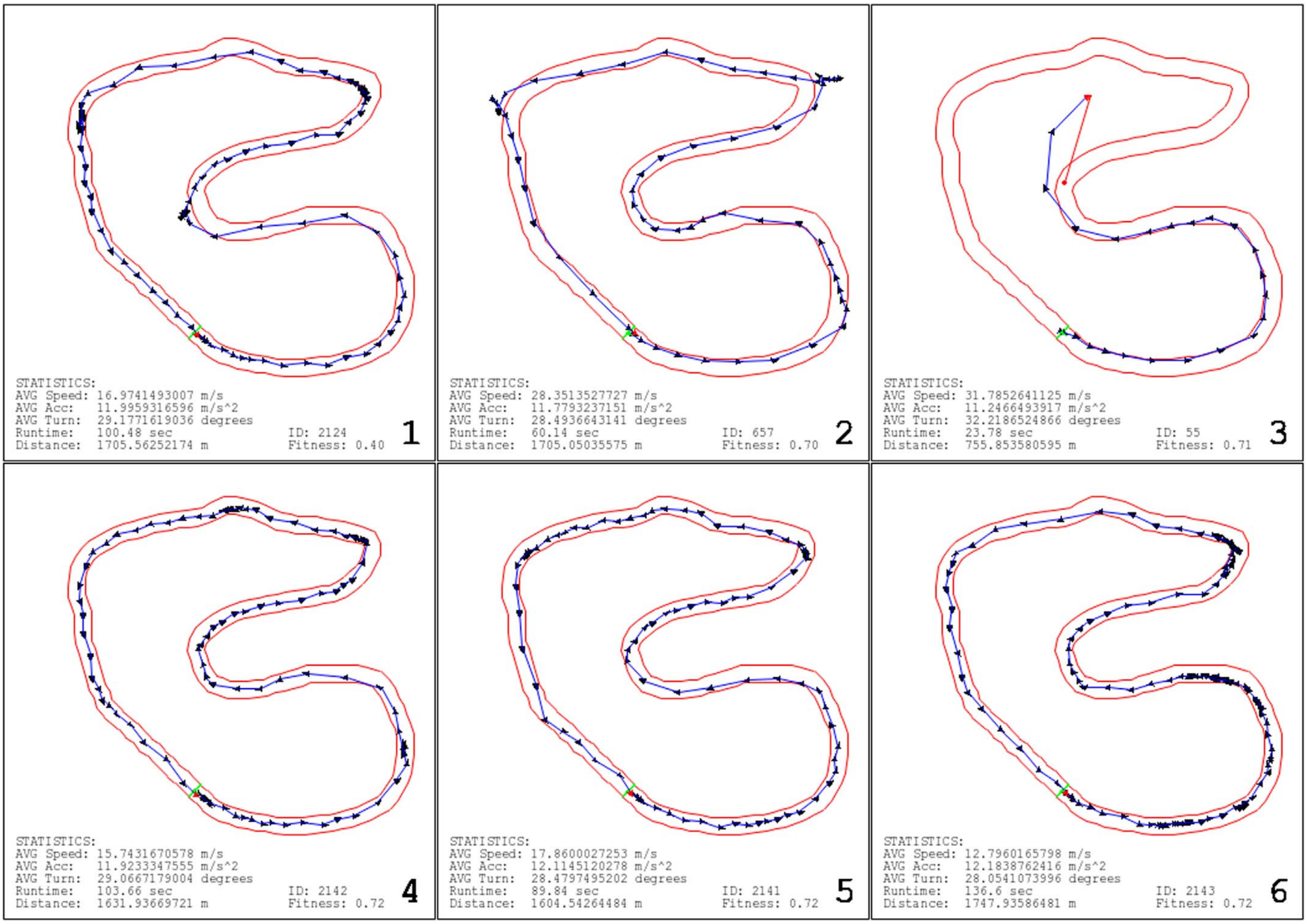


Fig. 11. Graphical user interface with the evaluation screen showing proposed car movement trajectories.

particular goal). This data can be easily obtained from the car's distance to the closer border and the car centerline angle related to the currently passed gate using goniometric calculations based on the cosine formula (the positions of each gate center and endpoints are known), see Fig. 9. The input values proceed to the nodes G_L, G_F, G_R, G_C, G_M and G_D (G as goal), i.e., L for *left*, F for *forward*, R for *right*, C for *close*, M for *medium* and D for *distant*. The first three nodes (G_L, G_F and G_R) are for membership calculations of the angular deviation to the individual directions (*left, forward* and *right*) between the car and the next particular goal. Analogously, the last three nodes (G_C, G_M and G_D) are for membership calculations of the magnitude of the distance of the car from the goal, i.e., *close, medium* and *distant*. From the appropriate combinations of weighted connections between these six nodes and the two output nodes, we will get the control values A/D and T, i.e., the acceleration or deceleration and turning angle, respectively. As our FCM is an open system, where the values from the sensors (the car's distance from the closer road border and the car centreline angle related to the currently passed gate) enter the FCM at each control step, then the inference process will be in the modified form as mentioned in Section 2. The definitions of the membership functions in the nodes G_L, G_F, G_R, G_C, G_M and G_D were adjusted manually preserving the condition of the so-called *fuzzy grid partitioning*.

The controller design process was based on quantitative as well as qualitative criteria. There were investigated these partially dependent quantitative criteria:

C1: relative average absolute angular turning,
C2: relative average absolute acceleration,
C3: relative trajectory length,
C4: relative average speed,
C5: relative road passing time,
C6: number of sheers (deviations from road borders),
C7: success or fail in passing the road (yes or no).

All these criteria have a cumulative character for the whole run. The criterion C1 represents the angles of the car centreline just passing a gate related to the center of the next gate, which are the turning angles—one of the FCM's outputs. These values are then averaged to one value. Of course, the absolute value of C1 depends on the form of the road, but we can compare the values of several solutions proposed for the same road. The smaller the value is, the more stable a solution is obtained. Similarly, the same holds for C2 (both accelerations and decelerations are taken into account), because steady speed changes require additional energy consumption and evoke suspicious driver's behavior, too. Therefore, C1 and C2 are mutually dependent criteria. Also mutually dependent are the criteria C3, C4 and C5, but unlike the previous pair, they are only partially dependent. The criterion C6 represents the number of collisions, which should be prevented in normal traffic. During the movement it can happen the car violates not only the criterion C6 but will not be able to come back to the road at all. In other words, such a solution represents a fail, because it will not pass the road in reality. The criterion C7 is the only one with a logical (nonnumeric) value, i.e., *yes* or *no*. The values of the relative criteria lie within the interval $[0; 1]$, and they are related to the larger value when comparing two solutions, see Table 1, where the relative criteria are related to the biggest value in a given column.

Qualitative criteria are evaluated using screens depicting some proposed solutions defined by connection matrices. Here, there are principally two possibilities of how to visualize these criteria. Either they are depicted individually as the time trends (diagrams) of related variables, as shown in the experiment with the mixing tank (see Fig. 8), or they are depicted in a cumulative manner, which is the case here. The user sees at once six experimental simulations on one screen and can visually evaluate the resulting trajectories, in order to select the best one according to his/her opinion. The selection is based on a subjective impression of which trajectory should be preferred, see Fig. 11, where always one solution is selected. The quantitative criteria presented on the screen play only an auxiliary role.

These trends represent the phenotype of a given solution, and together with quantitative criteria, the user selects according to his/her best opinion some prospective solutions which then proceed to the next generation.

The proposed controller design was compared to a neuro-fuzzy system ANFIS, whose parameters were adjusted using a self-organizing migrating algorithm (SOMA) as a special kind of EC [48]. In Table 1, the results of the five most successful FCM designs using IEC and fulfilling the criterion C7 are shown, where criteria C1–C6 are compared for the FCM and ANFIS controller.

As we can see in Table 1, the quality of results are quite comparable. From a population of 30 individuals, the first five ones reached a very good quality, although the ANFIS controller seems to be slightly better also from the visual viewpoint, see Fig. 12. For better visualization, the best results are boldfaced in the table. Thus we can see the FCM 2 controller is the best in terms of three criteria (C4, C5 and C6) and the ANFIS controller shows the best quality in four criteria (C1, C2, C3 and C6); these two controllers seem to be complementary regarding their quality. As the fitness value depends on the weights assigned to these criteria, we cannot unambiguously decide about their order from best to worst. Rather, we should classify these controllers as such of equal quality. However, if we consider the complexity of both algorithms (IEO-FCM and SOMA-ANFIS), then the simplicity is clearly on the side of IEO-FCM. It is also a demonstration of a human's reliability in design and decision processes in general if they are based on repetitive actions, such as, e.g., generating populations in our case.

6. Conclusions and future work

In this paper, a new optimization method for FCM design based on incomplete and uncertain expert knowledge was introduced. The method combines experts' knowledge concerning the modeled dynamic system with their subjective ideas about the expected behavior of the system. In fact, no additional data measured on the real system are necessary for this method of FCM design. The dynamics of the state vectors of several candidate FCMs are simultaneously presented to the expert on the GUI of the related application. This visual information enables the expert to evaluate the quality of the whole population at once. Without the need of analytically expressed fitness function, the expert uses the interface to prioritize the best FCM candidates in every population. The transition from one generation to the next is done automatically by the standard evolutionary operators (parent selection, recombination, mutation), and a new population is presented to the expert again.

Two numerical examples were presented. The first one was based on the classical technological supervisory system and its main objective was to test the application and its GUI. The second example deals with more recent research in mobile robotics. The trajectories of a particular autonomous car were evaluated not only according to the speed or time of movement but also according to their 'human-like' behavior in general.

Future research will be needed to better understand and optimally describe the evolutionary phase of the method, so as to increase the effectiveness of the whole process for more complex FCMs. For this reason, an overview of IEC research was presented in Section 3 to offer possible topics of future research. Perhaps the most interesting

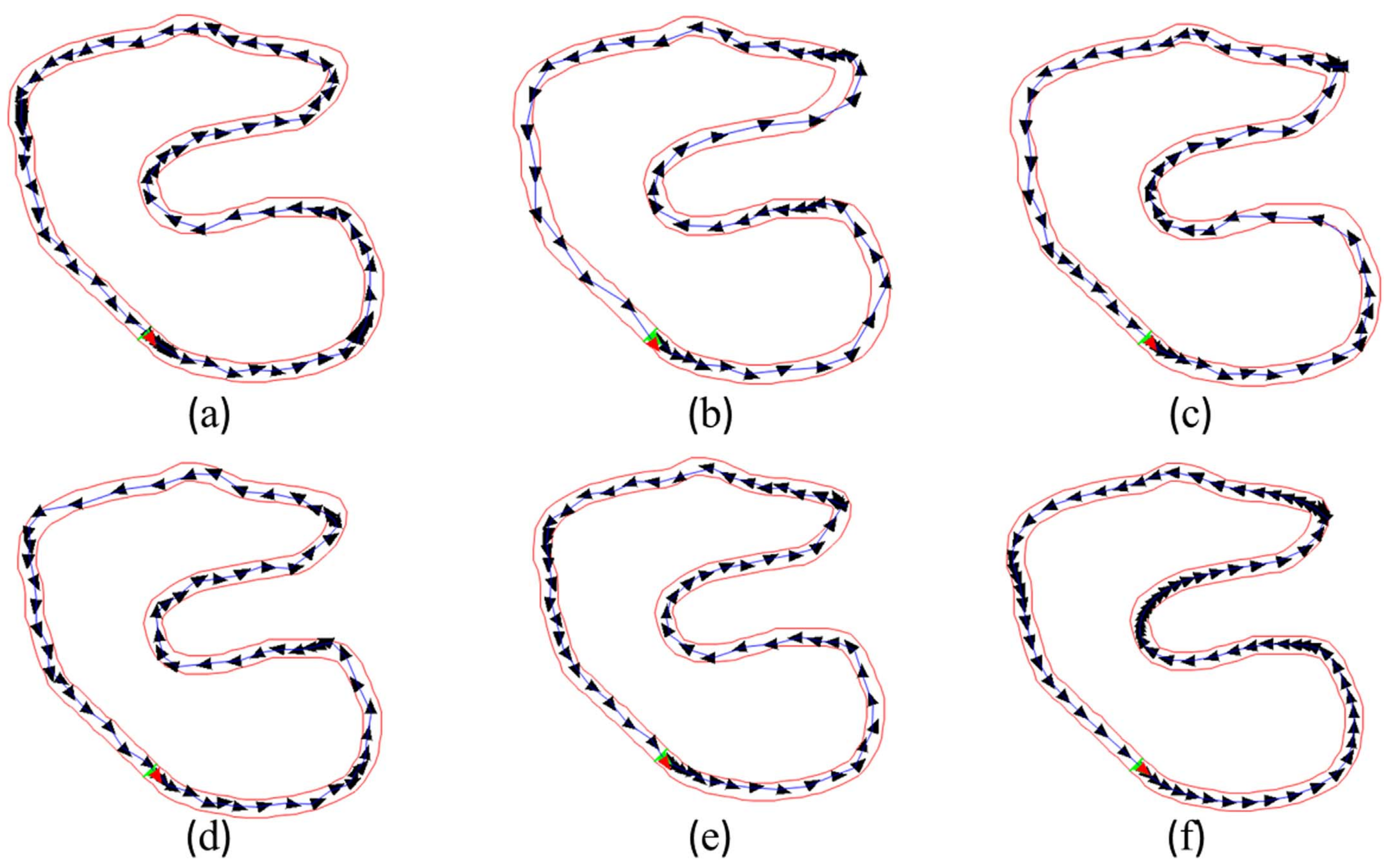


Fig. 12. Final movement trajectories for the following controller designs: (a): FCM 1. (b): FCM 2. (c): FCM 3. (d): FCM 4. (e): FCM 5. (f): ANFIS.

research potential is in constructing a model of a human expert, which could include the evaluation of qualitative criteria with the minimization of the human presence in the adaptation process.

Acknowledgments

The support of the Grant Agency of Excellence, University of Hradec Králové is greatly acknowledged. The work was also supported by the National Research and Development Project Grant 1/0773/16 2016–2019 "Cloud Based Artificial Intelligence for Intelligent Robotics."

References

[1] T. Davenport, J. Kim, Keeping Up with the Quants: Your Guide to Understanding and Using Analytics, Harvard Business Review Press, Boston, MA, 2013.
[2] W. Stach, L. Kurgan, W. Pedrycz, A survey of fuzzy cognitive map learning methods, Issues in Soft Computing: Theory and Applications, 2005, pp. 71–84.
[3] E.I. Papageorgiou, J.L. Salmeron, A review of fuzzy cognitive maps research during the last decade, IEEE Trans. Fuzzy Syst. 21 (2013) 66–79.
[4] W. Homenda, A. Jastrzebska, W. Pedrycz, Modeling time series with fuzzy cognitive maps, in: Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2014, pp. 2055–2062. ⟨http://dx.doi.org/10.1109/FUZZ-IEEE.2014.6891719⟩.
[5] W. Froelich, J.L. Salmeron, Evolutionary learning of fuzzy grey cognitive maps for the forecasting of multivariate, interval-valued time series, Int. J. Approx. Reason. 55 (2014) 1319–1335.
[6] M. Amer, T. Daim, A. Jetter, Application of fuzzy cognitive map for the development of scenarios: a case study of wind energy deployment, in: Research and Technology Management in the Electricity Industry: Methods, Tools and Case Studies, Springer, London, 2013, pp. 129–159.
[7] C.-S. Karavas, G. Kyriakarakos, K.G. Arvanitis, G. Papadakis, A multi-agent decentralized energy management system based on distributed intelligence for the design and control of autonomous polygeneration microgrids, Energy Convers. Manag. 103 (2015) 166–179.
[8] V.C. Georgopoulos, C.D. Stylios, Fuzzy cognitive map decision support system for successful triage to reduce unnecessary emergency room admissions for the elderly, in: R. Seising, E.M. Tabacchi (Eds.), Fuzziness and Medicine: Philosophical Reflections and Application Systems in Health Care: a Companion Volume to Sadegh-Zadeh's Handbook of Analytical Philosophy of Medicine, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 415–436. ⟨http://dx.doi.org/10.1007/978-3-642-36527-0_27⟩.
[9] M. Mendonça, B. Angelico, L. Arruda, F. Neves Jr, dynamic fuzzy cognitive map applied to chemical process supervision, Eng. Appl. Artif. Intell. 26 (2013) 1199–1210.
[10] E. Yesil, M.F. Dodurka, A. Sakalli, C. Ozturk, C. Guzay, Self-tuning pi controllers via fuzzy cognitive maps, in: H. Papadopoulos, A.S. Andreou, L. Iliadis, I. Maglogiannis (Eds.), Artificial Intelligence Applications and Innovations: 9th IFIP WG 12.5 International Conference, AIAI 2013, Paphos, Cyprus, September 30 to October 2, 2013, Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 567–576. ⟨http://dx.doi.org/10.1007/978-3-642-41142-7_57⟩.
[11] N. Ganganath, M. Walker, H. Leung, Fuzzy cognitive map based situation assessment framework for navigation goal detection, in: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 1444–1449.
[12] J. Vaščák, L. Madarász, Adaptation of fuzzy cognitive maps – a comparison study, Acta Polytech. Hung. 7 (2010) 109–122 (ISSN 1785-8860).
[13] J. Vaščák, N.H. Reyes, Use and perspectives of fuzzy cognitive maps in robotics, in: I.E. Papageorgiou (Ed.), Fuzzy Cognitive Maps for Applied Sciences and Engineering: From Fundamentals to Extensions and Learning Algorithms, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 253–266. ⟨http://dx.doi.org/10.1007/978-3-642-39739-4_15⟩
[14] B. Kosko, Fuzzy cognitive maps, Int. J. Man-Mach. Stud. 24 (1986) 65–75.
[15] M. Gregor, P.P. Groumpos, Tuning the position of a fuzzy cognitive map attractor using backpropagation through time, in: Proceedings of the 7th International Conference on Integrated Modeling and Analysis in Applied Control and Automation (IMAACA 2013), 2013, pp. 78–86.
[16] W. Froelich, E.I. Papageorgiou, Extended evolutionary learning of fuzzy cognitive maps for the prediction of multivariate time-series, in: Fuzzy Cognitive Maps for Applied Sciences and Engineering, Springer, 2014, pp. 121–131.
[17] E. Papageorgiou, Learning algorithms for fuzzy cognitive maps – a review study, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 42 (2012) 150–163.
[18] J. Vasčákˇ, Approaches in adaptation of fuzzy cognitive maps for navigation purposes, in: Proceedings of the IEEE 8th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 2010, pp. 31–36.
[19] W. Stach, L. Kurgan, W. Pedrycz, M. Reformat, Genetic learning of fuzzy cognitive maps, Fuzzy Sets Syst. 153 (2005) 371–401.
[20] N.J. Li, W.J. Wang, C.C.J. Hsu, Hybrid particle swarm optimization incorporating fuzzy reasoning and weighted particle, Neurocomputing 167 (2015) 488–501.
[21] A. Kannappan, E.I. Papageorgiou, A new classification scheme using artificial immune systems learning for fuzzy cognitive mapping, in: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ), 2013, pp. 1–8. ⟨http://dx.doi.org/10.1109/FUZZ-IEEE.2013.6622567⟩.
[22] J.L. Salmeron, W. Froelich, Dynamic optimization of fuzzy cognitive maps for time series forecasting, Knowl.-Based Syst. (2016).
[23] M. Gregor, P.P. Groumpos, Training fuzzy cognitive maps using gradient-based supervised learning, in: H. Papadopoulos, A. Andreou, L. Iliadis, I. Maglogiannis (Eds.), Artificial Intelligence Applications and Innovations, volume 412 of IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, 2013, pp. 547–556.
[24] J.C. Sun, N. Wang, M.J. Er, Y.C. Liu, Extreme learning control of surface vehicles with unknown dynamics and disturbances, Neurocomputing 167 (2015) 535–542.
[25] Z.C. Johanyák, O. Papp, A hybrid algorithm for parameter tuning in fuzzy model identification, Acta Polytech. Hung. (2012) 153–165.
[26] E.I. Papageorgiou, W. Froelich, Multi-step prediction of pulmonary infection with the use of evolutionary fuzzy cognitive maps, Neurocomputing 92 (2012) 28–35.
[27] M.-B. Radac, R.-E. Precup, Optimal behaviour prediction using a primitive-based data-driven model-free iterative learning control approach, Comput. Ind. 74 (2015) 95–109.
[28] F.A. Cheein, S. Blazic, M. Torres-Torriti, Computational approaches for improving the performance of path tracking controllers for mobile robots, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 6495–6500. ⟨http://dx.doi.org/10.1109/IROS.2015.7354305⟩.
[29] H. Takagi, Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation, in: Proceedings of the IEEE, vol. 89, 2001, pp. 1275–1296.
[30] H. Takagi, Interactive evolutionary computation for analyzing human awareness mechanisms, Appl. Comp. Intell. Soft Comput. 2012 (2012) 1–12.
[31] S. Greco, B. Matarazzo, R. Słowiński, Interactive evolutionary multiobjective optimization using dominance-based rough set approach, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2010, pp. 1–8. ⟨http://dx.doi.org/10.1109/CEC.2010.5585982⟩.
[32] J.B. Tolosa, S. Guadarrama, Collecting fuzzy perceptions from non-expert users, in: Proceedings of the 2010 IEEE International Conference on Fuzzy Systems (FUZZ), 2010, pp. 1–8. ⟨http://dx.doi.org/10.1109/FUZZY.2010.5584816⟩.
[33] D.W. Gong, J. Yuan, X.Y. Sun, Interactive genetic algorithms with individuals fuzzy fitness, Comput. Hum. Behav. 27 (2011) 1482–1492.
[34] H. Farooq, M.T. Siddique, A comparative study on user interfaces of interactive genetic algorithm, in: Proceedings of the 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014), Procedia Computer Science, vol. 32, 2014, pp. 45–52.
[35] Z. Krizsán, S. Kovács, Fuzzy model based differential iec for human-system interaction in the virca environment, in: Proceedings of the 3rd IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2012, pp. 169–173. ⟨http://dx.doi.org/10.1109/CogInfoCom.2012.6421974⟩.
[36] M. Babbar-Sebens, B.S. Minsker, Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design, Appl. Soft Comput. 12 (2012) 182–195.
[37] C. Su, H. Li, A novel interactive preferential evolutionary method for controller tuning in chemical processes, Chin. J. Chem. Eng. 23 (2015) 398–411.
[38] X. Sun, L. Yang, D. Gong, M. Li, Interactive genetic algorithm assisted with collective intelligence from group decision making, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2012, pp. 1–8. ⟨http://dx.doi.org/10.1109/CEC.2012.6252872⟩.
[39] R. Dou, C. Zong, G. Nan, Multi-stage interactive genetic algorithm for collaborative product customization, Knowl.-Based Syst. 92 (2016) 43–54.
[40] D. Gong, X. Ji, J. Sun, X. Sun, Interactive evolutionary algorithms with decision-maker's preferences for solving interval multi-objective optimization problems, Neurocomputing 137 (2014) 241–251.
[41] X. Shen, Y. Guo, Q. Chen, W. Hu, A multi-objective optimization evolutionary algorithm incorporating preference information based on fuzzy logic, Comp. Opt. Appl 46 (2010) 159–188.
[42] S. Ono, S. Nakayama, Fusion of interactive and non-interactive evolutionary computation for two-dimensional barcode decoration, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2010, pp. 1–8. ⟨http://dx.doi.org/10.1109/CEC.2010.5586287⟩.
[43] P. Vasant, A novel hybrid genetic algorithms and pattern search techniques for industrial production planning, Int. J. Model. Simul. Sci. Comput. 03 (2012) 250020-1–250020-19.
[44] H.J. Aljawawdeh, C.L. Simons, M. Odeh, Metaheuristic design pattern: Preference, in: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO Companion '15, ACM, New York, NY, USA, 2015, pp. 1257–1260. ⟨http://dx.doi.org/10.1145/2739482.2768498⟩.
[45] R.C. Purshouse, K. Deb, M.M. Mansor, S. Mostaghim, R. Wang A review of hybrid evolutionary multiple criteria decision making methods, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1147–1154. ⟨http://dx.doi.org/10.1109/CEC.2014.6900368⟩.
[46] Y. Pei, H. Takagi, Triple and quadruple comparison-based interactive differential evolution and differential evolution, in: Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms XII, FOGA XII '13, ACM, New York, NY, USA, 2013, pp. 173–182. ⟨http://dx.doi.org/10.1145/2460239.2460255⟩.
[47] C.D. Stylios, P.P. Groumpos, Fuzzy cognitive maps in modeling supervisory control systems, J. Intell. Fuzzy Syst. 8 (2000) 83–98.
[48] I. Zelinka, SOMA — self-organizing migrating algorithm, in: D. Davendra, I. Zelinka (Eds.), Self-Organizing Migrating Algorithm: Methodology and Implementation, volume 626 of Studies in Computational Intelligence, Springer-Verlag Berlin, 2016, pp. 3–49. ⟨http://dx.doi.org/10.1007/978-3-319-28161-2_1⟩.

Karel Mls received his Ph.D. degree from the Czech Technical University in Prague, Czech Republic. At present he is with the Department of Information Technologies, the University of Hradec Kralove. Dr. Mls focuses his research interests on fuzzy logic, computational intelligence, decision processes, decision support, and autonomous systems. He has participated in several research projects and is the author or co-author of more than 60 papers and book chapters.

Ján Vaščák received the Engineering and Ph.D. degrees from the Technical University of Košice, Slovakia. At present, he is an Assistant Professor with the Department of Cybernetics and Artificial Intelligence, the Technical University of Košice. He has been involved in several research projects dealing with the development of means of artificial intelligence, mainly in the area of control and decision making.

Richard Cimler received his Master's degree from the University of Hradec Králové where he is currently a Ph.D. student based in the Faculty of Informatics and Management. He is focused on agent based simulation, optimization, and decision making processes.

Michal Puheim received his Master's degree from the Technical University of Košice, Slovakia, where he is currently a Ph.D. student based in the Center for Intelligent Technologies. He is focused on the use of means of computational intelligence in the situational control of complex systems.