

Dynamic game difficulty balancing in real time using Evolutionary Fuzzy Cognitive Maps with automatic calibration

Lizeth Joseline Fuentes Pérez^{*}, Luciano Arnaldo Romero Calla[†],
 Anselmo Antunes Montenegro[‡], Luis Valente[§], Esteban Walter Gonzalez Clua[¶]
^{*†‡§¶} *Institute of Computing - Federal Fluminense University, Niterói - RJ*
 {^{*}lfuentes, [†]lromero, [‡]anselmo, [§]lvalente, [¶]esteban}@ic.uff.br

Abstract—Fuzzy Cognitive Maps (FCM) is a paradigm used to represent knowledge in a simple and concise way, expressing the grade of relation that exists between concepts and causal relationships. Due to its flexibility, FCM has been successfully applied in numerous applications in diverse research fields, such as, robotics, medical diagnosis, decision problems in information technology, games, and so forth. However, one critical drawback is the determination of the weights in the representation graph, which is generally done by an expert. The present paper proposes a semi-automated method for calibrating the weights in a solution for the problem of dynamic game difficulty balancing (DGB) using Evolutionary Fuzzy Cognitive Maps (E-FCM). The proposed algorithm adjusts the weights in real time, ensuring an equilibrium between the values generated according to the expert's contribution (based on a static analysis) and the changes produced in the values of the concepts by the calibration process during the simulation (a dynamic analysis).

Keywords-Automatic Calibration of Weights, Real-time Strategy, Dynamic Game Difficulty Balancing, Evolutionary Fuzzy Cognitive Maps

I. INTRODUCTION

Along the years, the survival of humanity has been challenged by many factors including, economical, socio-political, ecological and others. Because of that, we as humans have developed a logic reasoning as a part of a critical thinking that takes part in decision-making processes, which is able to analyze several factors and the relations among them.

Due to the increasing complexity of the problems, taking the right decisions can be, in most of the cases, difficult and time-consuming. However, with the technology advances we have been creating algorithms to simulate our reasoning process, which currently are still not perfect but can help us to understand and better control the high complexity associated to usual decision problems.

One example of these algorithms is the Fuzzy Cognitive Maps (FCM), proposed by Kosko [1], which is considered a hybrid methodology because combines characteristics of Artificial Neuronal Networks (ANN) and Fuzzy Logic [2]. FCM is a powerful tool to simulate complex systems, representing them in a graph, composed of concepts and

causal relations, which describe the way the concepts are related to each other. The causal relationships also measure the degree to which one concept affects another, which is represented by a weight.

Even though FCM have been applied in numerous research fields and applications, such as: Robotics, for example for navigation of robots [3], medical decision making [4], for example, radiotherapy [5], risk management [6], banking management [7], information technology [8], computer vision [9], controllers and supervisory agents of complex systems [10] and so forth; however, in the literature, there are few works devoted to the solution of one critical issue: the determination of the weights of the causal relationships, which is generally done by an expert.

The algorithms that focus on solving the fully automatic calibration, as far as we are concerned, can be categorized into three classes: the Hebbian learning based methods [11], Neuronal Networks Based methods [12], and the Genetic algorithms based methods [13]; we refer the interested reader to the work in [14], where some of these methods are compared.

The calibration of the weights in the FCM design step can be performed in two ways: using expert-based methods and performing fully automatic calibration. These two approaches have differences and characteristics that are briefly delineated below:

- In expert-based methods, the calibration depends on the experts' subjective judgment, which can introduce some errors in the simulation. Usually, the calibration is done by computing an average of the experts' judgments using a deductive reasoning. In this approach, as it is based on the expert's own beliefs, it is easy to devise a set of weights that is coherent with the static behavior expected for the system. On the other hand, it is not possible to guarantee that the system will behave as expected during the simulation (the dynamic behavior). This usually happens when the complexity of the problem grows. As the number of concepts and causal relationships increase, the experts cannot consider the problem as a whole. They are obliged to



Figure 1: *TimeOver Game*

group the concepts and causal relationships into parts making difficult any global analysis and prediction of how the system will evolve from the initial static setup.

- The fully automatic calibration algorithms require as an input, a priori knowledge of the system, which can be obtained from different sources, for example using data mining. From this initial input, an inductive learning process extracts a set of rules which are used to generate the weights in the FCM. Such methods are prone to produce stable dynamical behavior.

In a few words, expert-based approaches are good for coherent static weight definition and fully automatic algorithms are adequate for achieving a stable dynamic behavior. The main contribution of this work is a new method that has both good properties. It is based on a dynamic automatic calibration which combines expert knowledge with a real-time automatic weight adjustment.

We specifically applied our method to the solution of the dynamic game difficulty balancing problem that uses Evolutionary Fuzzy Cognitive Maps (E-FCM).

Evolutionary Fuzzy Cognitive Map (E-FCM) extends FCM proposed by [15], [16], where each state is evolving according to non-deterministic external causalities. Similar to the FCM, Evolutionary Fuzzy Cognitive Maps have been also applied to a variety of scientific areas, such as political decision making [17], interactive storytelling for the creation of virtual worlds [16], medical diagnosis [18], [19], computation pragmatics [20] and so forth.

In [21], the authors proposed a method to change the difficulty levels dynamically and in real-time, which is based on player interaction information, context variables, and Evolutionary Fuzzy Cognitive Maps. Player interactions involve actions where the player interacts directly with the game logic, for example, eating, jumping or collecting items.

The method proposed in [21], is different from others proposed in literature [22], [23], [24], because it presents

three important advantages: independence, adaptability, and scalability.

Independence, because the method is able to adjust the difficulty levels independently of explicit player feedback (e.g., surveys, reports, and related alternatives) inferring the necessary information directly from player actions in the game.

Adaptability, because it is possible to apply the method to any game, as E-FCM are flexible.

Scalability because, as the number of context variables and causal relationships increase, it is possible to consider more factors to evaluate player-related aspects, which may lead to better game experience and player engagement.

This paper is organized as follows. Section II presents an overview of Evolutionary Fuzzy Cognitive Maps (E-FCM). Section III describes the application of E-FCM for Dynamic Game Difficulty Balancing. Section IV describes the algorithm proposed for weights calibration in real-time. Section V presents experimental results and Section VI presents conclusions and future works.

II. EVOLUTIONARY FUZZY COGNITIVE MAPS

In this section, we present the components of the E-FCM model and how simulations in real-time can be performed using an updating step. The design of an E-FCM is basically the design of a FCM adding the evolving time.

A. E-FCM Design

An E-FCM is constructed with two main components: concepts and causal relationships.

- *Concepts* $C = (S, \Delta S, T, P_S)$, which represents important factors and characteristics of the real-time system. Each concept $c \in C$ is expressed as a tuple:

$$c = (s, \Delta s, et, p_s) \quad (1)$$

where, $s \in S$ denotes the value or level of the concept, $et \in T$ is the evolving time for the concept and p_s is the probability of self-mutation. $\Delta s \in \Delta S$ is the state value change for the concept, which will be updated in real-time simulations.

The evolving time et is important in a real-time system because it helps to control the change of the concept value s , at a given instant of time t .

For example: Given two concepts in vital functions of the human being system: *breathe* and *eat*, it is required that the value of *breathe* to be updated every five seconds, whereas the value of *eat* can be updated every six hours.

Different concepts might have different evolving times. Let C be a set with n concept variables. The set of its evolving times can be represented in vectorial form as

$$T = \begin{pmatrix} et_1 \\ et_2 \\ \vdots \\ et_i \\ \vdots \\ et_n \end{pmatrix} \quad (2)$$

The concepts in the system have their states updated in their respective evolving times. Besides, each concept alternates its internal state randomly in real time. Thus, to each concept we associate a very small mutation probability p_s . If the probability is high, the system can become very unstable.

- *Causal relationship* (r_{ij}), which represents the strength and probability of the causal effect from one concept c_i to another concept c_j . It is defined as a tuple:

$$r_{ij} = (w_{ij}, s, pm_{ij}) \quad (3)$$

where w_{ij} is the weight of the causal relationship and denotes the degree of influence of the concept c_i to the concept c_j . The weights are fuzzy values; they can normally be in the range of $[-1, 1]$ or $[0, 1]$.

The relationships between the concepts can be direct or inverse. This is expressed in the variable s , which can be positive or negative, describing respectively the direct or inverse relationships.

In the E-FCM model, concepts can be affected by causal relationships with probability pm .

Figure 2 illustrates how an E-FCM model describes causality. In the example, the concept c_1 affects the concept c_2 via a causal relation $c_1 \rightarrow c_2$, where c_1 is the causal variable and c_2 is the effect variable.

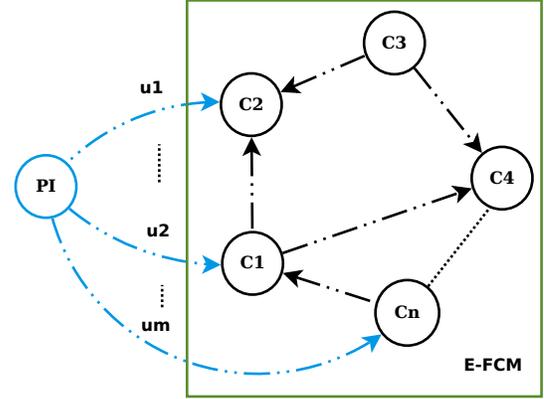


Figure 2: E-FCM modified to include u_i relationships, related to player interactions (PI). C_i represents context variables and dotted arrows represent the causal relationships.

In our examples, in each causal relationship, we consider the probabilities $pm = 1$, and the signs are implicitly defined in the weights $w_{ij} \in [-1, 1]$. Thus, the fuzzy causal relationships for a system with n variables can be directly represented as a $n \times n$ weight matrix W :

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{i1} & w_{ij} & \cdots & w_{in} \\ \vdots & \vdots & \cdots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{pmatrix} \quad (4)$$

B. Updating step for Simulations in Real Time

E-FCM works updating in each cycle all the context variables, states S , whenever a evolving time et of a concept c matches the current time value t . This update is a fundamental step in the theory of E-FCM and is called a running cycle [2].

The state value s_i of a concept is updated at time step $t + 1$ from the previous time step t by computing its state change according to equation (5):

$$\Delta s_i^{t+1} = k_1 \sum_{j=0, j \neq i}^n h(pm_{ij} - \lambda) \Delta s_j^t w_{ij} + k_2 \Delta s_i^t \quad (5)$$

where h is the Heaviside step function:

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

and λ is a probability threshold between 0 and 1. As we defined the probabilities $pm = 1$, equation (5) can be written in a simplified version:

$$\Delta s_i^{t+1} = k_1 \sum_{j=0, j \neq i}^n \Delta s_j^t w_{ij} + k_2 \Delta s_i^t \quad (6)$$

$$s_i^{t+1} = f(s_i^t + \Delta s_i^{t+1}) \quad (7)$$

where f is the activation function used to regulate the state value. It is important to adjust the concepts values in a given range. Some usual f functions are:

- Bipolar, for binary values 0 or 1.
- Logistics, for values in the range $[0, 1]$:

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (8)$$

where $\lambda \succ 0$.

- Hyperbolic, for values in the range $[-1, 1]$:

$$f(x) = \tanh(x). \quad (9)$$

The state value of the concept c_i at time t is represented as s_i . The state value change of the concept c_i at time t is Δs^t . The evolving time et is not included in this formula because it is used to verify in which running cycle s_i must be updated. Different concepts may have different evolving times. The k_1 and k_2 values are constants; k_2 represents the proportional contribution of the other concepts which affects to c_i and k_1 represents the influence of the previous value s_i .

III. USING E-FCM FOR DYNAMIC GAME DIFFICULTY BALANCING

In this section, we will explain briefly, the design of the E-FCM used to solve the DGB problem.

In [21], the authors introduced the player interactions as inputs that change the E-FCM values in real-time.

Note that with such introduction the E-FCM is able to simulate different behaviors along the gameplay. For example, the E-FCM provides to the player more items when his stamina is low. Conversely, when his stamina is high, the E-FCM increases the number and difficulty of the obstacles.

In this work, we use the Time Over game presented in [21], as a study case. This is a runner type game where the player is a boy who is escaping from a twister. In the game, he must avoid the obstacles and also collect some items such as food and water, which are provided to him to recover his energy. Figure 1 illustrates some screenshots of the TimeOver game.

One interesting feature of Time Over game is that the player's energy is not infinite. Hence, a *tiredness* factor is

considered, which will affect the player's energy and also his speed. Thus, the player is forced to collect items constantly to survive.

According to this game description, the game designer considers the following concepts as context variables:

- 1) **Stamina:** represents the player's energy, which increases as the player collects more items in the game.
- 2) **Speed:** represents the player's speed, which relates to stamina. Speed decreases over time to simulate the player character's tiredness.
- 3) **Obstacle type:** there are three types of obstacles: easy, default, and hard. These types represent how difficult the obstacles are.
- 4) **Obstacle period:** represents the period (time interval) that the game uses to insert obstacles in the game scene.
- 5) **Item type:** there are two types of collectible items in the game: water bottles and seeds. Both items increase player stamina, but water bottles provide more stamina than seeds.
- 6) **Item period:** represents the period (time interval) that the game uses to insert collectible items in the game scene.

In practice, players have different skill levels and may regard usual predetermined difficult levels as too easy or too hard, becoming frustrated or bored. For that reason, the game designer sets rules to keep the player on playing the game. Some examples of rules for this case are:

- If the player's energy increases, the speed also increases.
- If the player's speed is high, the number of food and water is reduced and the difficulty of the obstacles is increased.

Based on this rules, the game designer creates the causal relationships between concepts and decides whether the relation is direct or inverse. In this stage, it is important to define the degree of the causalities (the weights). The definition of the weights is a critical step because it controls the gameplay experience. Weights with high values along the gameplay can produce instability in the E-FCM because the changes are more violent. Weights with very low values can produce low interactivity along the gameplay, and it is not desired in real-time simulations.

According to [21], the new function that updates the concepts values, including the player interactions (u_i), is as follows:

$$\Delta s_i^{t+ti} = f \left(k_1 \sum_{j=0}^n \Delta s_j^t w_{ij} + k_2 (\Delta s_i^t + u_i) \right) \quad (10)$$

The player interactions u_i drastically alter the behavior of the E-FCM, so dynamically changes the game difficulty levels in real-time. However, this changes could produce abrupt changes in the game experience. Thus, it is necessary to achieve an equilibrium between the game designer expectation's, who knows the game logic, and the behavior of the E-FCM along the gameplay. The present paper proposes an algorithm to adjust the weights, using an initial configuration given by the game designer and controlling the ratio of changes in the E-FCM to improve the gameplay experience.

IV. CALIBRATING E-FCM WEIGHTS

So far, we detailed how E-FCM works, however, the game experience does not only depends on the correct work of the logic rules for DGB. Along the gameplay, it is also important that the game level changes are not abrupt because it generates instability in the game experience.

In the gameplay for example, if stamina decreases quickly, the other inverse concepts to stamina will increase so fast, that a lot of items will be generated in a short time.

In every instant of time, a running cycle is executed. Algorithm 1 summarizes the process of updating the concepts' values in a running cycle of E-FCM.

Algorithm 1 E-FCM running cycle in real-time

Require: $C = (S, T)$: concepts

Require: W : weight matrix

Require: t : current time

Require: ΔS^t : states value change at time t

```

1  $\Delta S^{t+1} \leftarrow \Delta S^t$ 
2 for all  $c_i \in C$  do
3   if  $t \bmod t_i = 0$  then
4      $\Delta s_i^{t+1} \leftarrow f \left( k_1 \sum_{j=1, i \neq j}^n \Delta s_j^t w_{ij} + k_2 (\Delta S_i^t + u_i) \right)$ 
5      $s_i^{t+1} \leftarrow s_i^t + \Delta s_i^{t+1}$ 
6   end if
7 end for
```

When the simulation is finished, the E-FCM produces functions (time series) for each concept, influenced by the player interactions. In general applications, the concepts of E-FCM tend to converge, according to the logic of E-FCM design. However, since the player interactions were introduced, the E-FCM concepts values are always changing along the gameplay. Because of that, it is necessary to control the changes in the values of the concepts avoiding abrupt changes along the simulations.

To cope with this issue, since it is impossible to predict the behavior of the functions as they strongly depend on player interactions, we consider two fundamental E-FCM elements that are known in each running cycle:

The weights w_{ij} and the state value change $\Delta s_i, \Delta s_j$, which indicates how the functions are changing at an instant of time.

In order to define a way to update the weights, so that the functions evolve in a smooth way, keeping the coherence with the causal relationships, we analyse the four basic combination of function behaviors and associated state changes. In each configuration we analyse the relation between function growth or decrease and the state change values and how the weights can be modified to control the function behavior.

- **Direct and increasing relation** $w_{i,j} > 0$ and $\Delta s_i > 0, \Delta s_j > 0$, when both functions are increasing. In this case, the weight is reduced proportionally to the concept changes, to moderate the independence degree between the functions associated to the concepts.
- **Direct and decreasing relation** $w_{i,j} > 0$ and $\Delta s_i < 0, \Delta s_j < 0$, when both functions are decreasing. In this case, the weight is also reduced proportionally to the concept changes, to reduce the independence degree between the functions of the concepts.
- **Indirect and increasing-decreasing relation** $w_{i,j} < 0$ and $\Delta s_i > 0, \Delta s_j < 0$, when the function generated by the causal concept c_i is increasing and the function generated by the effect concept c_j is decreasing. In this case, the two functions are moving away from each other, depending on the weight. Since c_i affects to c_j , it is necessary to reduce the repulsion force, increasing the weight in a factor proportional to the changes of the functions. The increase in the weight makes the causal concept have more influence in the effect concept making them have a closer behavior.
- **Indirect and decreasing-increasing relation** $w_{i,j} < 0$ and $\Delta s_i < 0, \Delta s_j > 0$, when the function generated by the causal concept c_i is decreasing and the function generated by the effect concept c_j is increasing. In this case, the two functions are approaching each other. Since c_i affects to c_j , it is necessary reduce the attraction force also the weight is increased in a proportional factor to the changes of the functions. As in the previous case, the increase in the weight has the same effect. Both variables will have greater correlation, as the causal variable will have more impact on the effect variable, diminishing the difference in the behavior of the functions, caused by the sign difference in the state changes Δs_i and Δs_j .

From the analysis, we update the weights in real-time according to equation (11):

$$w_{ij} = f(w_{ij}(1 - \Delta S_i^t \Delta S_j^t)) \quad (11)$$

where f is the regularization function which guarantees that the weight values are always in range $[-1, 1]$.

Finally, we developed a new algorithm, seen in Algorithm 2, wherein each running cycle the weights are adjusted, considering the changes of the concept levels. In the algorithm $\Delta s_i^t \Delta s_j^t$, represents the strength of the weight change, and if this is positive or negative. The adjustment is just proportional to the old weight according to $w_{ij} \leftarrow f(w_{ij}(1 - \Delta s_i^t \Delta s_j^t))$.

Algorithm 2 Update E-FCM real time with calibration

Require: $C = (S, T)$: concepts

Require: W : weight matrix

Require: t : current time

Require: ΔS^t : states value change at time t

```

1  $\Delta S^{t+1} \leftarrow \Delta S^t$ 
2 for all  $c_i \in C$  do
3   if  $t \bmod t_i = 0$  then
4      $w_{ij} \leftarrow f(w_{ij}(1 - \Delta s_i^t \Delta s_j^t))$ 
5      $\Delta s_i^{t+1} \leftarrow f\left(k_1 \sum_{j=1, i \neq j}^n \Delta s_j^t w_{ij} + k_2(\Delta s_i^t + u_i)\right)$ 
6      $s_i^{t+1} \leftarrow s_i^t + \Delta s_i^{t+1}$ 
7   end if
8 end for
    
```

Algorithm 2 summarizes the process of automatic calibration of weights in real-time.

V. EXPERIMENTS AND RESULTS

In order to experimentally validate our algorithm, we created the Social Drug E-FCM and also redesigned the E-FCM for the Time Over game.

A. Social Drug E-FCM

“Social Drug” E-FCM basically simulates the human emotions and its emotional states when a person experiments social approval (flattery) or social rejection (understate).

Figure 3, illustrates the E-FCM model for “Social Drug”, where the context variables are:

- c_1 : Animus.
- c_2 : Depression.
- c_3 : Sadness.
- c_4 : Courage.
- c_5 : Health.
- c_6 : Happiness.

The player interactions u_1 and u_2 , represent social approval and social rejection respectively.

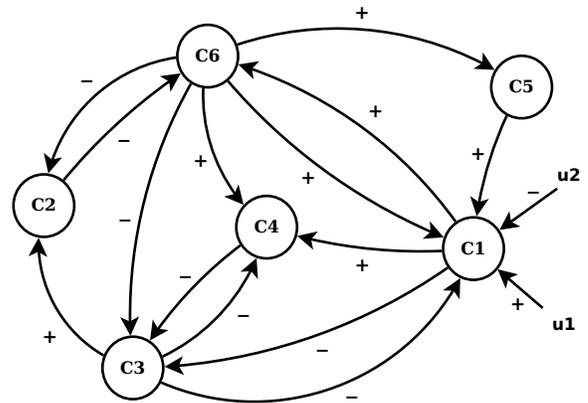


Figure 3: The E-FCM model for Social Drug.

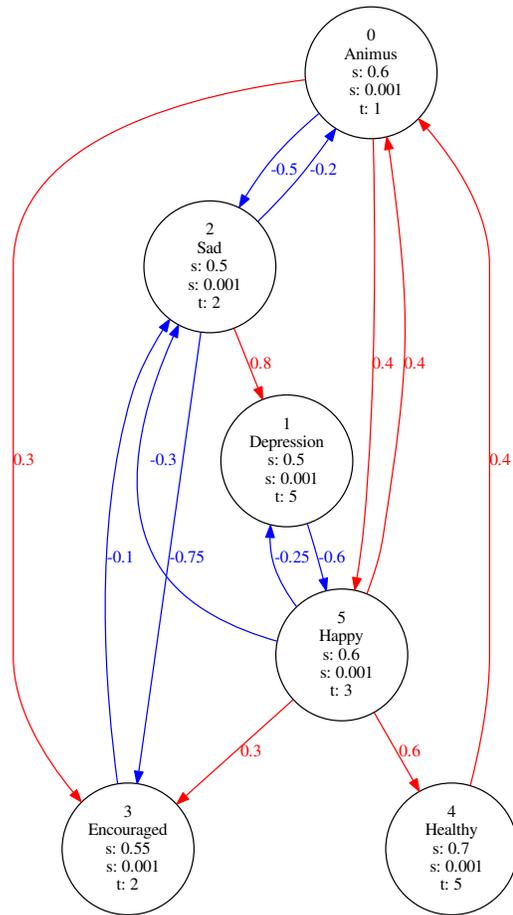


Figure 4: The E-FCM model for Social Drug.

The evolving time vector is $T = (1 \ 1 \ 2 \ 2 \ 5 \ 3)$ and the initial values for the context variables are

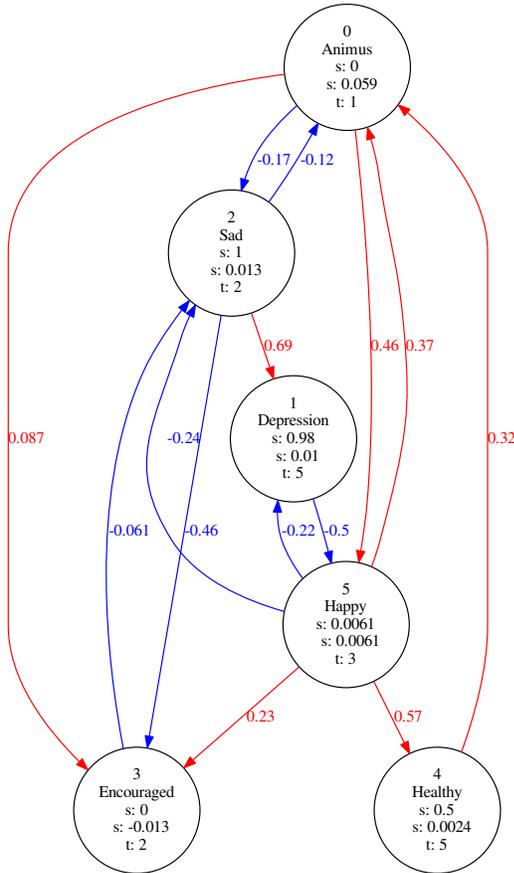


Figure 5: The E-FCM model for Social Drug.

$$S^0 = (0.6 \quad 0.5 \quad 0.5 \quad 0.55 \quad 0.7 \quad 0.6)$$

Figures 6, 7, illustrates the simulations of E-FCM Social Drug, where the interactions “Social Approval” and “Social Rejection” were simulated altering both values, emphasizing the factor “Social Rejection”. Because of this, along the simulation the concept “Animus” is decreasing in both calibrated and non-calibrated simulations.

The main difference between figures 6 and 7 is the behavior of the concept (state values) along the time. In Figure 6, we can observe that the concept values of *Animus*, *Courage*, *Health* and *Happiness* are decreasing faster than the concept values with calibrated weights. Moreover, the behavior at initial times is quite similar, which is an expected behavior because the calibration process is done along the simulations in real-time and also depends on the player interactions. The weights adjustment would be impossible to do in the game

design stage because is impossible to predict how the player will interact with the E-FCM along the simulations.

Therefore, for different gameplays, the algorithm will produce different weight matrices. In the case of the Social Drug model, we applied the same interactions in both simulations, the calibrated and non-calibrated, in order to better appreciate the effect of the calibration algorithm along the simulations.

Figure 6 shows the E-FCM behavior along the first 100 time instances, setting as initial configuration the E-FCM model illustrated in Figure 4; the convergence is reached close to $t = 100$ and the final configuration is shown in Figure 5.

The concepts’ values converge to 0 and 1, keeping the logic in the causal relations, but it should be noted that the *Health* concept converges slower than the other concepts.

Conversely, in Figure 7, which shows the simulation with weights calibration, not all concept’s values converge to 0 or 1 in time $t = 100$. The value of *Health* concept in this time is 0.5, and its curve did not drop to 0. It should be noted that the curve for the *Happiness* concept converges to 0, but it takes longer than in the non-calibrate graph whose value is already 0 in $t = 60$.

In the same way, *Depression* curve increases much slower, in $t = 100$ it still has not reached the value of 1.

Figure 7 shows the simulation of the E-FCM with calibrated weights. It is possible to see that the curves of the concepts change slower than those obtained in the simulation with the non calibrated E-FCM. This real-time calibration guarantees that changes in game difficulty are not abrupt, differently from those presented in [21].

Table I, shows the probabilistic weight output matrix W of the causal relationship for the Social Drug experiment, obtained after the weights were calibrated via the proposed algorithm.

Table I: Probabilistic weight matrix W of the causal relationship for Social Drugs

w_{ij}	c_1	c_2	c_3	c_4	c_5	c_6
c_1	0	0	-0.77	0.72	0	0.72
c_2	0	0	0	0	0	0.9899
c_3	-0.55	0.72	0	-0.88	0	0
c_4	0	0	-0.33	0	0	0
c_5	0.54	0	0	0	0	0
c_6	0.54	-0.33	-0.55	0.72	0.81	0

B. Time Over E-FCM

We also validate our model in a game, where the interactions are different from those in the Social Drug case. We run the E-FCM model with different player interactions, but

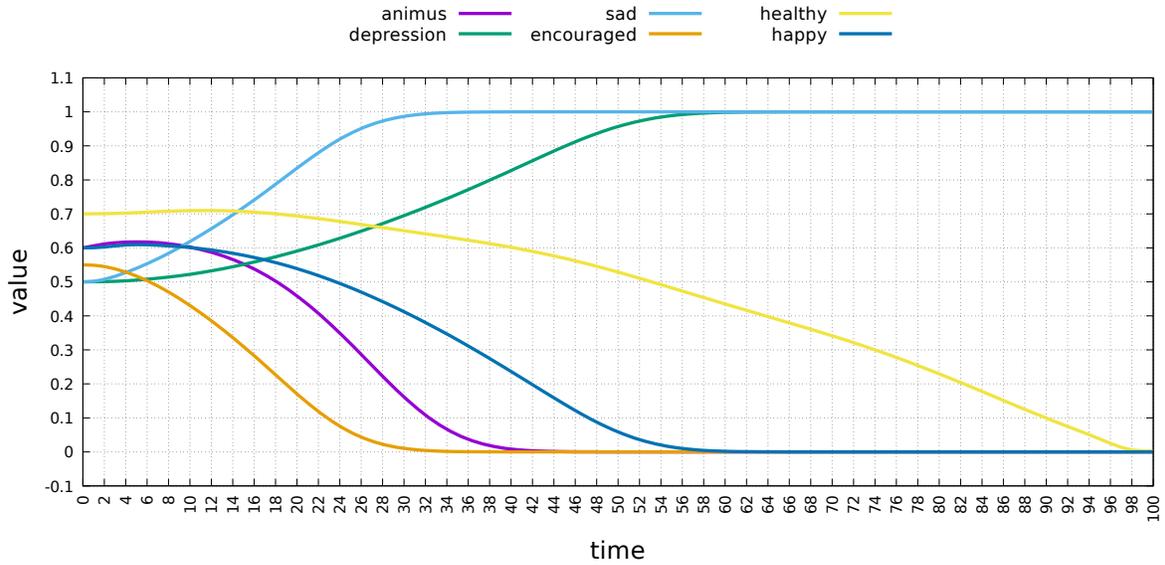


Figure 6: E-FCM simulation without calibrated weights in the Social Drug model

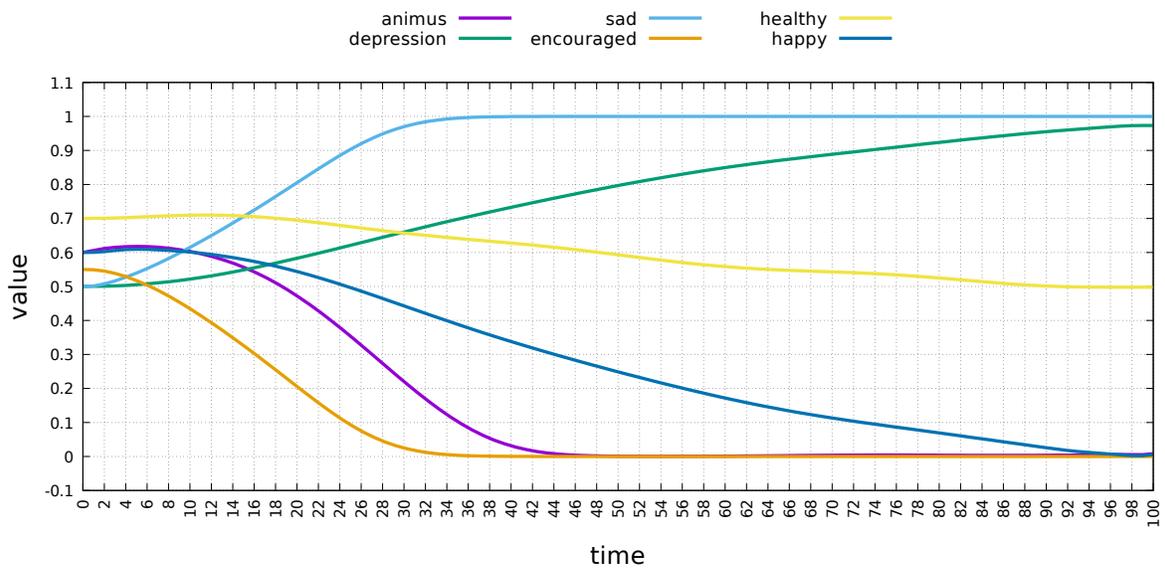


Figure 7: E-FCM simulation with calibrated weights in the Social Drug model

with the same player. Figure 8 shows the E-FCM model for TimeOver game.

Each context variable describes is a fuzzy value, normalized to the range $[0, 1]$. The mean of each variable value depends on specific game designs. For simplicity, we defined *obstacle type* as a mapping of the actual value of *obstacle type* to conceptual “easy”, “default”, and “hard” difficulty obstacle levels. The “easy” difficulty level maps to the range of $[0, 0.33]$, the “default” difficulty level maps to $[0.34, 0.66]$ and the “hard” level maps to the range of $[0.66, 1]$. The

Item type is a mapping of the actual value of *item type* to conceptual “water” and “seeds”. The water item appears when the corresponding value belongs to the range $[0, 0.5]$. The seed item appears when the corresponding values are in the range $[0.6, 1]$. We associate each context variable to the following concepts:

- c_1 : Stamina.
- c_2 : Speed.
- c_3 : Obstacle type.
- c_4 : Obstacle period.

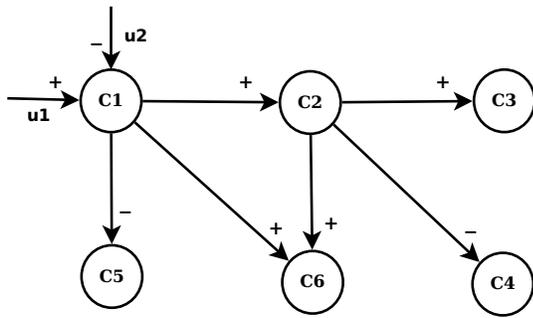


Figure 8: The E-FCM model for TimeOver game.

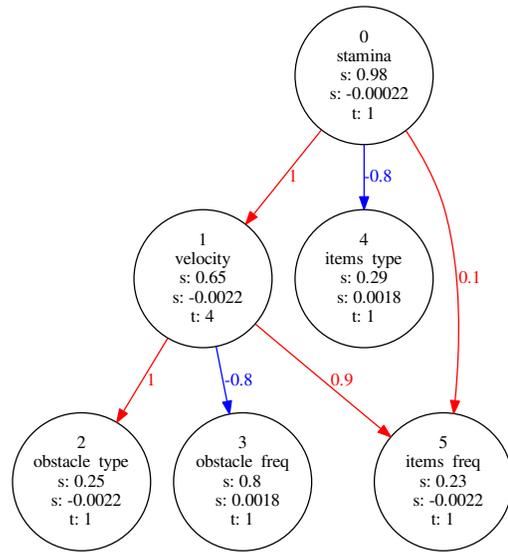


Figure 10: The E-FCM model result without calibration for TimeOver game.

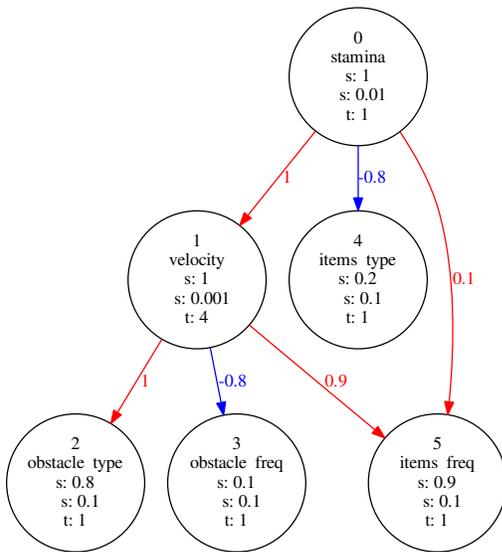


Figure 9: The initial E-FCM configuration for TimeOver game.

- c_5 : Item type.
- c_6 : Item period.

Figure 8 illustrates the final TimeOver’s E-FCM model, where $c_i, i \in [1,6]$ represents each context variable and signed arrows represent causal relationships between context variables. A positive sign (+), means positive causal relationship and negative sign (−) means negative relationship. Table II illustrates the probabilistic weight matrix W of causal relationships, which are determined either from an expert knowledge or learned from a knowledge base. In this paper we redesigned some of these values; as the model designed for this game is simple, the weights were provided

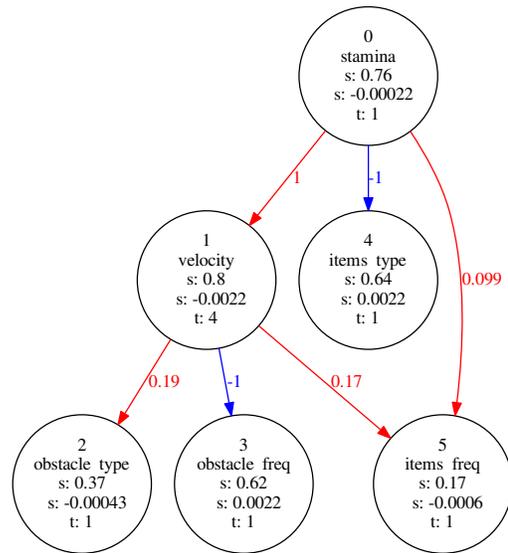


Figure 11: The E-FCM model result with calibration for TimeOver game.

by the game designer. The matrix P_m is a one’s matrix because we consider that the probability of a concept c_i affecting another concept c_j is equal to one.

The activation function selected for the experiments was the

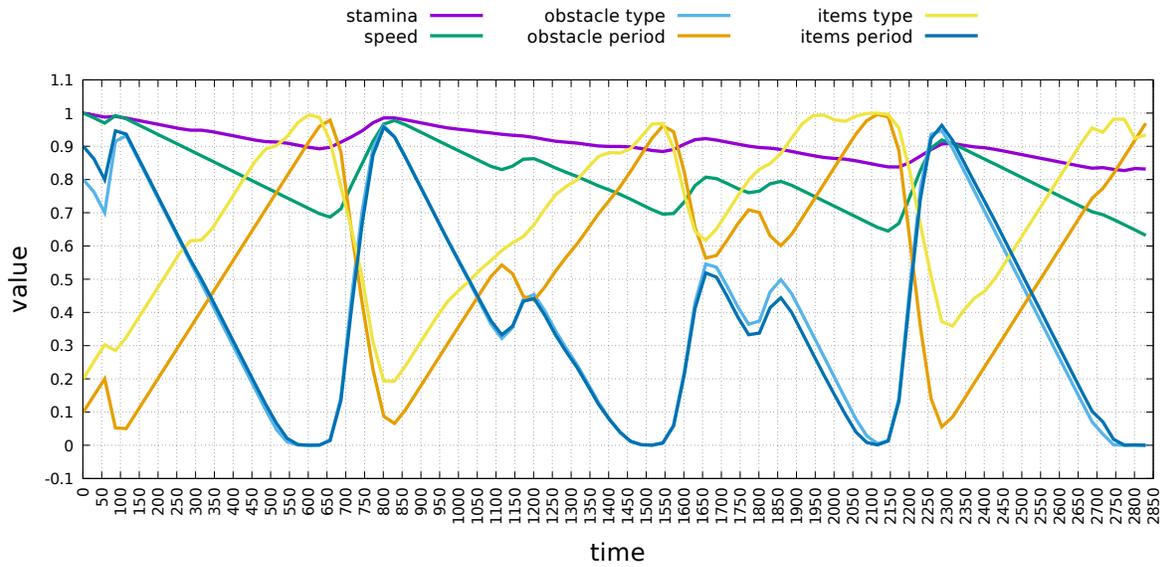


Figure 12: E-FCM simulation TimeOver game without calibration.

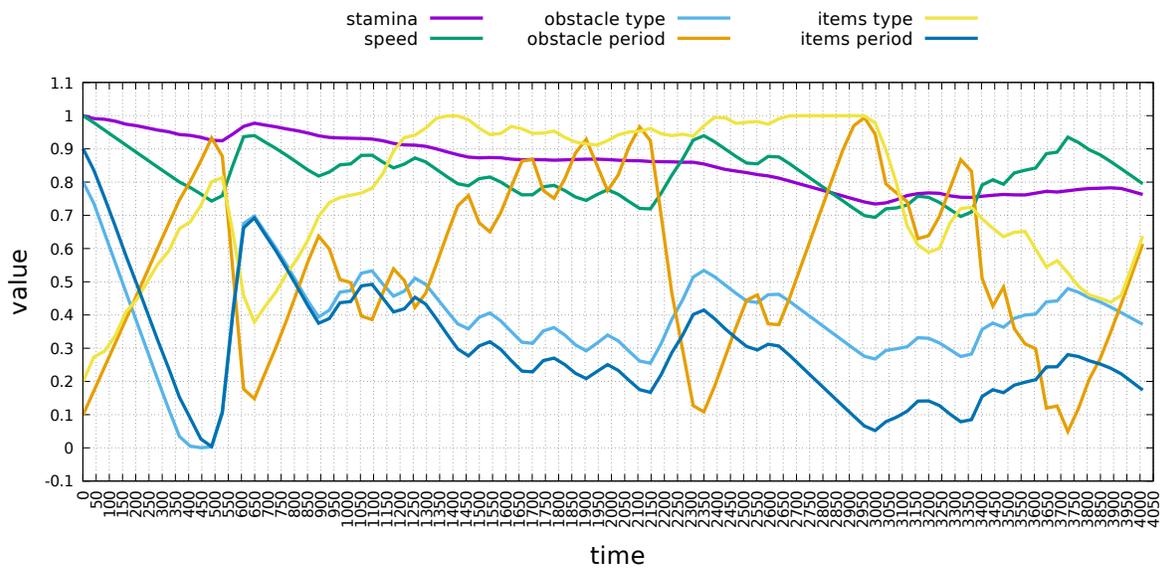


Figure 13: E-FCM simulation 1 in TimeOver game with calibration.

Table II: Probabilistic weight matrix W of the causal relationship for Time Over Game

w_{ij}	c_1	c_2	c_3	c_4	c_5	c_6
c_1	0	1	0	0	-0.8	0.1
c_2	0	0	1	-0.8	0	0.9
c_3	0	0	0	0	0	0
c_4	0	0	0	0	0	0
c_5	0	0	0	0	0	0
c_6	0	0	0	0	0	0

logistic function. This means that the concept values should be a number between 0 and 1, inclusive.

In order to model player interactions with the E-FCM, we added two arrows to the E-FCM model. The u_1 arrow, represents the stamina that the player earned by collecting items. The u_2 arrow represents stamina loss (due to player tiredness). The stamina value decreases constantly. We use the “game frame” as the time unit. In this regard, we consider that time evolves as the game frame sequence progresses. We update the six context variables each frame, according to the

evolving time vector T :

$$T = (1 \quad 4 \quad 1 \quad 1 \quad 1 \quad 1)$$

The values in T denote the time interval in which a variable is updated. For example, a value equal to 1 means that a variable is updated each frame. A value equal to 2 means that a variable is updated every two frames, and so on. Here we modified the speed evolving time value to 4 because we consider the speed changes in the player must be slower.

The initial values of the six context variables are:

$$S^0 = (0.01 \quad 0.001 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1)$$

Figure 9, illustrates the initial configuration of E-FCM for TimeOver game. Figures 11 and 13 illustrate the configuration results of real-time simulations that we designed and conducted to test the behavior of E-FCM model with non-calibrate and calibrate weights respectively.

Figures 12 and 13 illustrate the context variables in each game frame. Given the initial configuration in Figure 9, we expected that in the simulations, the concept values change as the player interact along the game. Each figure illustrates the context variables in each game frame, demonstrating that all simulations behaved as we expected.

Figures 12 and 13 are different, because the player interactions were performed in different ways. However, in both gameplays, the concept values changes according to the E-FCM model.

One remarkable difference between both gameplays, the calibrated and the non-calibrated, is the behavior of the functions (concepts values in each game frame). In Figure 13, the “obstacle type” and “items period” functions, becoming more independent of each other along the gameplay.

The functions of the concepts in Figure 13 present less abrupt changes, for example in frame 300, when the value of stamina decreases, the values of “speed”, “items period” and “obstacle type” also decreases but the change is smoother compared with the frame 2515 of the gameplay in Figure 12 with non-calibrate weights.

VI. CONCLUSIONS

Considering the results described in Section V, we observed that our algorithm enhances the gameplay experience, because it is able to regularize the changes produced by the player interactions, avoiding abrupt changes and considering only the principal elements of the causal relations such as: inverse or direct causal relationships (positive or negative sign) and the degrees of change (ΔS).

Therefore, we conclude that our algorithm generates softer simulations in real-time, considering the player interactions and also presents the following advantages:

Dynamic: The proposed algorithm takes advantage of the dynamic behavior of the E-FCM to generate the regularization of the functions in real-time.

Independence: The method considers mainly the own natural evolution the E-FCM. It does not require any external method for smoothing the functions.

Coherence: The smooth functions will always be coherent with the E-FCM design. In the case of our method, coherence is automatically obtained because the regularization of the functions relies solely on the data captured directly from the E-FCM simulation. It would be necessary more experiments to verify if other regularization methods, for example, those based on time-series smoothing or moving averages, would lead to convergence and present results coherent with the design of the E-FCM. This is something that we achieve without introducing external processes.

An interesting future work is a in depth study of the behavior of the E-FCM models in real-time, that is, to figure out how to predict its behavior in real-time simulations from the game design. Another interesting work is to do an analysis of the game experiences using E-FCM and E-FCM with calibration because it is possible to find out other factors that may help to improve the game experience based on this methodology.

ACKNOWLEDGEMENTS

We thank CAPES, CNPq, FAPERJ and NVIDIA for supporting the present research.

REFERENCES

- [1] B. Kosko, “Fuzzy cognitive maps,” *Int. J. Man-Mach. Stud.*, vol. 24, no. 1, pp. 65–75, Jan. 1986. [Online]. Available: [http://dx.doi.org/10.1016/S0020-7373\(86\)80040-2](http://dx.doi.org/10.1016/S0020-7373(86)80040-2)
- [2] P. P. Groumpos, *Fuzzy Cognitive Maps: Basic Theories and Their Application to Complex Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–22. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_1
- [3] D. Lorencik, J. Vascak, and M. Vircikova, *Adaptive Fuzzy Cognitive Maps Using Interactive Evolution: A Robust Solution for Navigation of Robots*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 703–711. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37374-9_67

- [4] C. D. Stylios, V. C. Georgopoulos, G. A. Malandraki, and S. Chouliara, "Fuzzy cognitive map architectures for medical decision support systems," *Applied Soft Computing*, vol. 8, no. 3, pp. 1243 – 1251, 2008, forging the Frontiers - - Soft Computing. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494607001226>
- [5] E. I. Papageorgiou, *A Novel Approach on Constructed Dynamic Fuzzy Cognitive Maps Using Fuzzified Decision Trees and Knowledge-Extraction Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 43–70. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_3
- [6] S. M. Hurtado, *Modeling of Operative Risk Using Fuzzy Expert Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 135–159. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_6
- [7] G. Xirogiannis, M. Glykas, and C. Staikouras, *Fuzzy Cognitive Maps in Banking Business Process Performance Measurement*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 161–200. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_7
- [8] J. L. Salmeron, *Fuzzy Cognitive Maps-Based IT Projects Risks Scenarios*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 201–215. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_8
- [9] G. Pajares, M. Guijarro, P. J. Herrera, J. J. Ruz, and J. M. de la Cruz, *Fuzzy Cognitive Maps Applied to Computer Vision Tasks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 259–289. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_11
- [10] A. Jose, *Dynamic Fuzzy Cognitive Maps for the Supervision of Multiagent Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 307–324. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_13
- [11] E. Papageorgiou, C. Stylios, and P. Groumos, "Active hebbian learning algorithm to train fuzzy cognitive maps," *International Journal of Approximate Reasoning*, vol. 37, no. 3, pp. 219 – 249, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0888613X04000349>
- [12] H. Song, C. Miao, Z. Shen, W. Roel, D. Maja, and C. Francky, "Design of fuzzy cognitive maps using neural networks for predicting chaotic time series," *Neural Networks*, vol. 23, no. 10, pp. 1264 – 1275, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608010001541>
- [13] E. I. Papageorgiou and P. P. Groumos, "A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps," *Applied Soft Computing*, vol. 5, no. 4, pp. 409 – 431, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494604001012>
- [14] W. Stach, L. Kurgan, and W. Pedrycz, *Expert-Based and Computational Methods for Developing Fuzzy Cognitive Maps*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 23–41. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03220-2_2
- [15] Y. Cai, C. Miao, A.-H. Tan, and Z. Shen, "Context modeling with evolutionary fuzzy cognitive map in interactive storytelling," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, June 2008, pp. 2320–2325.
- [16] Y. Cai, C. Miao, A.-H. Tan, Z. Shen, and B. Li, "Creating an immersive game world with evolutionary fuzzy cognitive maps," *Computer Graphics and Applications, IEEE*, vol. 30, no. 2, pp. 58–70, March 2010.
- [17] A. S. Andreou, N. H. Mateou, and G. Zombanakis, "Evolutionary fuzzy cognitive maps: A hybrid system for crisis management and political decision making," University Library of Munich, Germany, MPRA Paper, 2003. [Online]. Available: <http://EconPapers.repec.org/RePEc:pra:mpapa:51482>
- [18] W. Froelich, E. I. Papageorgiou, M. Samarinas, and K. Skriapas, "Application of evolutionary fuzzy cognitive maps to the long-term prediction of prostate cancer," *Applied Soft Computing*, vol. 12, no. 12, pp. 3810 – 3817, 2012, theoretical issues and advanced applications on Fuzzy Cognitive Maps. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1568494612000610>
- [19] E. Papageorgiou and W. Froelich, "Application of evolutionary fuzzy cognitive maps for prediction of pulmonary infections," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 1, pp. 143–149, Jan 2012.
- [20] D. E. Koulouriotis, I. E. Diakoulakis, D. M. Emiris, E. Antonidakis, and I. Kaliakatsos, "Efficiently modeling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps (invited paper)," *THE ABC OF COMPUTATIONAL PRAGMATICS 43*, vol. 1, pp. 41–65, 2003.
- [21] L. J. Fuentes Perez, L. R. Romero Calla, L. Valente, A. A. Montenegro, and E. W. G. Clua, "Dynamic game difficulty balancing in real time using evolutionary fuzzy cognitive maps," in *Computing Conference (SBGames), 2015 XIV Brazilian Symposium on Games and Digital Entertainment*, Nov 2015.
- [22] T. Tijs, D. Brokken, and W. IJsselsteijn, "Dynamic game balancing by recognizing affect," in *Fun and Games*, ser. Lecture Notes in Computer Science, P. Markopoulos, B. de Ruyter, W. IJsselsteijn, and D. Rowland, Eds. Springer Berlin Heidelberg, 2008, vol. 5294, pp. 88–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88322-7_9

- [23] R. Hunicke, “The case for dynamic difficulty adjustment in games,” in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ser. ACE '05. New York, NY, USA: ACM, 2005, pp. 429–433. [Online]. Available: <http://doi.acm.org/10.1145/1178477.1178573>
- [24] J. Olesen, G. Yannakakis, and J. Hallam, “Real-time challenge balance in an rts game using rtneat,” in *Computational Intelligence and Games, 2008. CIG '08. IEEE Symposium On*, Dec 2008, pp. 87–94.