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

Lizeth Joseline Fuentes Pérez*, Luciano Arnaldo Romero Calla*, Luis Valente*, Anselmo Antunes Montenegro*, and Esteban Walter Gonzalez Clua*

*Institute of Computing - Federal Fluminense University, Niterói - RJ

Abstract

Players may cease from playing a chosen game sooner than expected for many reasons. One of the most important is related to the way game designers and developers calibrate game challenge levels. In practice, players have different skill levels and may find usual predetermined difficult levels as too easy or too hard, becoming frustrated or bored. The result may be decreased motivation to keep on playing the game, which means reduced engagement. An approach to mitigate this issue is dynamic game difficulty balancing (DGB), which is a process that adjusts gameplay parameters in real-time according to the current player skill level. In this paper we propose a real-time solution to DGB using Evolutionary Fuzzy Cognitive Maps, for dynamically balancing a game difficulty, helping to provide a well balanced level of challenge to the player. Evolutionary Fuzzy Cognitive Maps are based on concepts that represent context game variables and are related by fuzzy and probabilistic causal relationships that can be updated in real-time. We discuss several simulation experiments that use our solution in a runner type game to create more engaging and dynamic game experiences.

Keywords: Real-time Strategy, Dynamic Game Difficulty Balancing, Evolutionary Fuzzy Cognitive Maps

Author’s Contact:

{lujens, lromero, lvalente, anselmo, esteban}@ic.uff.br

1 Introduction

Gameplay in digital games comprises several elements such as actions and challenges that players must undertake to complete game activities. A game designer may adjust the game mechanics to make challenges easier or harder to solve, providing predetermined difficulty levels, such as “easy”, “normal”, and “hard”. However, these adjustments are static and may be created based on an arbitrary baseline, which is not suitable for all users.

In practice, players have different skill and experience levels and may find predetermined difficult levels as “too easy” or “too hard”, becoming frustrated or bored. The result may be decreased motivation to keep on playing the game, which means reduced engagement.

A solution to cope with these issues is to dynamically adjust the game difficulty levels according to the current playing context, which includes monitoring player actions, errors, and performance in the game. The literature refers to solutions based on this idea as “dynamic game difficulty balancing (DGB)” and “dynamic difficulty adjustment (DDA)”. There are several works that approach DGB and related issues. For example Tijs and co-authors [Tijs et al. 2008] proposed to balance difficulty levels using the player’s emotional state. However, the work by Tijs and co-authors [Tijs et al. 2008] presents some drawbacks. Firstly, their approach needs to ask the player about his/her emotional state during the game. Secondly, their approach lacks a properly functional decision making system. In another related work, Hunnicke [Hunicke 2005] examined how dynamic difficulty adjustment affected player progress while conducting experiments that manipulated supply and demand of various items in the game. Vasconcelos de Medeiros [Vasconcelos De Medeiros and Vasconcelos De Medeiros 2014] proposed a static level balancing (created during game design), based on the feedback of real gaming experiences. This approach is interesting because the difficulty level is modeled using real-world data (instead of using a random and arbitrary baseline). However, this solution is not dynamic and the difficulty levels remain the same during the entire game.

Olesen [Olesen et al. 2008], proposes the use of Real Time Neuro-evolution of Augmenting Topologies (rNEAT) for dynamic-challenge balancing. In their approach they perform off-line experiments in order to investigate challenge factors such as aggressiveness and number of warriors using NEAT. The game used in this approach is Globulation 2 (G2). Also, they designed a challenge rating formula composed by six factors; then they used rNEAT to minimize the difference of challenge rating between the AI agent and its opponent in real-time. Their method was successfully applied only to RTS games that share features with G2. Hence, the application of this approach to different games is not trivial, differently from the method we propose in this work.

In this paper, we propose 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 comprises important actions in a game, such as “jumping”, “eating”, and “running”, defined in the game design stage. Context variables are related to game state and Salen and Zimmerman [Salen and Zimmerman 2003] define “game state as the current condition of the game at any given moment”. Consider as an example a Soccer Game. In its game state elements we could find the following context variables: the half time being played, the remaining time, team information, current score and current weather conditions.
Evolutionary Fuzzy Cognitive Map (E-FCM), is a modeling tool, proposed by [Cai et al. 2008], [Cai et al. 2010], based on Fuzzy Cognitive Maps, with the difference that in E-FCM each state is evolving based on non-deterministic external causalities in real-time. Our approach creates an E-FCM based on game context variables, which is later modified to include player interactions, such as jump, eat and run; which depend of the game design. The E-FCM updates all context variables in real-time depending on player interactions, which changes the game difficulty levels while a game session is happening. We use E-FCMs because they are a powerful tools to assist in reasoning and decision making processes. The literature provides examples of using E-FCMs in several different areas, such as political crisis management and political decision making [Andreou et al. 2003], interactive storytelling [Cai et al. 2010], long-term prediction of prostate cancer [Froelich et al. 2012], pulmonary infection prediction [Papageorgiou and Froelich 2012], and computation pragmatics [Koulouriotis et al. 2003], to model and control complex dynamic systems efficiently. Also, Mateou [Mateou et al. 2006] proposed an extension called Evolutionary Multilayered Fuzzy Cognitive Maps to handle large-scale, complex, real-world problems.

This paper is organized as follows. Section 2 presents a brief overview of Evolutionary Fuzzy Cognitive Maps (E-FCM). Section 3 describes the methodology and the proposed method. Section 4 presents experimental results and Section 5 presents conclusions and future works.

2 Evolutionary Fuzzy Cognitive Map

Modeling a dynamic system can be hard in a computational sense. In addition, formulating a mathematical model may be difficult, costly and in some cases even impossible. These approaches offer the advantage of quantified results but suffer several drawbacks, such as the requirement to have specialized knowledge outside the domain of interest. Fuzzy Cognitive Maps (FCM) are a qualitative alternative approach to dynamic systems, where the gross behavior of a system can be observed quickly and without the services of an operation research expert [Andreou et al. 2003]. In the Evolutionary Fuzzy Cognitive Maps each state is evolving based on non-deterministic external causalities in real-time.

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

- **Concept (C)**, which represents a variable of interest in a real-time system and is expressed as a tuple:

\[
C = (S, T, P) \tag{1}
\]

where, \(S\) denotes the state value of the concept, \(T\) is the evolving time for the concept, representing a multiple of a fixed time slice \(t_0\) and \(P\) is the probability of self-mutation.

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

\[
R = (W, s, P) \tag{2}
\]

where \(W\) is the weight matrix of the causal relationship, \(w_{ij} \in [0, 1]\), \(s\) denotes whether the causal relationship is either positive (+) or negative (−). \(P\) is the probability that the causal concept affects the result concept \(C\).

Fuzzy causal relationships for a system with \(n\) variables can be represented as a \(n \times n\) weight matrix \(W\):

\[
W = \begin{pmatrix}
w_{11} & w_{12} & \cdots & w_{1n} \\
w_{21} & w_{22} & \cdots & w_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
w_{n1} & w_{n2} & \cdots & w_{nn}
\end{pmatrix} \tag{3}
\]

For a system with \(n\) variables, the mutual causal probability can be represented as a \(n \times n\) matrix \(P\):

\[
P_m = \begin{pmatrix}
p_{11} & p_{12} & \cdots & p_{1n} \\
p_{21} & p_{22} & \cdots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \cdots & p_{nn}
\end{pmatrix} \tag{4}
\]

Different concepts might have different evolving times. For a system with \(n\) variables, it can be represented as a vector \(T\):

\[
T = \begin{pmatrix}
t_1 \\
t_2 \\
\vdots \\
t_n
\end{pmatrix} \tag{5}
\]

Besides the causal effects from others concepts, each concept will also alternate its internal state randomly in real time. Each concept is modeled with very small mutation probability. If the probability is high, the system would become very unstable. For a system with \(n\) variables it can be represented as a vector \(P\):

\[
P = \begin{pmatrix}
p_{1} \\
p_{2} \\
\vdots \\
p_{n}
\end{pmatrix} \tag{6}
\]

The concepts in the system update their states in their respective evolving time. The state value of concept \(C_i\) is updated according to the following equations:

\[
\Delta S_i^{+T} = f \left( k_1 \sum_{j=0}^{n} \Delta S_j^{w_{ij}} + k_2 \Delta S_i^T \right) \tag{7}
\]

\[
S_i^{+T} = S_i^T + \Delta S_i^{+T} \tag{8}
\]

where \(f\) is the activation function to regulate the state value (e.g., bipolar, tri-polar, and logistic). \(S_i^T\) is the state value of concept \(C_i\) at time \(t\). \(\Delta S_i^{w_{ij}}\) is the state value change of concept \(C_i\) at time \(t\). \(T\) is the evolving time of concept \(C_i\) to update its value. Different concepts may have different evolving times. The \(k_1\) and \(k_2\) values are two weight constants. The summation \(\Delta S_i^{w_{ij}}\) is subjected to conditional probability \(P\), and \(\Delta S_i^{w_{ij}}\) is subjected to self-mutation probability \(P_i\).
3 Methodology

The proposed method is based on two important aspects: context variables and player interactions.

**Context variables** are background real-time variables that model the game context. Examples of context variables include player status and resources such as stamina, speed, and number of collected items. The game context is dynamic as the context variables change over time while a game session is happening.

**Player interactions** are general player actions in the game, such as “jump”, “run”, and “collect item”. Player interactions may change context variables. As our method uses context variables to balance game difficulty, player interactions may lead to changes in difficulty levels.

![E-FCM](image)

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

We created a solution based on a E-FCM model and using the player interactions and context variables to balance game difficulty in real-time. We use context variables as concepts in the E-FCM, identifying the causal relationships between concepts according to the current game state. Our E-FCM model includes $u_i$ relationships to represent causal relationships between player interactions and some context variables that are related to player skills. Figure 2 illustrates the E-FCM model and $u_i$ relationships.

Equation 9 represents the original equation 7 modified to include the $u_i$ relationship.

$$\Delta^T = f \left( \sum_{j=0}^{n} \Delta^j w_{ij} + k_2 (\Delta^j + u_i) \right)$$

4 Experiments and results

In order to experimentally validate our model, we developed the TimeOver game. TimeOver is a runner type game where a young man escapes from a twister to save himself. Figure 1 illustrates some screenshots of TimeOver game. In a preliminary version, the game had only two context variables: score and speed. The game context is dynamic as the context variables change over time while a game session is happening.

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 bottle 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.

Each context variable is a fuzzy value, normalized to the range of $[0,1]$. The mean of each variable value depends on specific game designs. For simplicity we defined obstacle type as mapping the actual value of obstacle type to conceptual “easy”, “default”, and “hard” difficulty obstacle levels. The “easy” difficulty level maps to the range of $[0.0,0.33]$, the “default” difficulty level maps to $[0.34,0.66]$ and the “hard” level maps to range $[0.66,1]$. The item type as mapping the actual value of item type to conceptual “water” and “seeds”. The water item appears to the range $[0.0,0.5]$. The seed item appears to 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.
- $C_5$: Item type.
- $C_6$: Item period.

Figure 3 illustrates the final TimeOver’s E-FCM model. $C_i, i \in [1,6]$ represents each context variable, signed arrows represent causal relationships between context variables. A positive sign (+), means positive causal relationship and negative sign (−) means negative relationship. Table 1 illustrates the probabilistic weight matrix $W$ of causal relationships, which are determined either from an expert knowledge or learnt from a knowledge base; as the model designed for this game is simple, the weights were provided by the game designer. The matrix $P_{in}$ is a ones matrix because we consider the probability that a concept $C_i$ affecting another concept $C_j$ is one.

![E-FCM](image)

**Figure 3:** The E-FCM model for TimeOver game.

**Table 1:** Probabilistic weight matrix $W$ of the causal relationship

<table>
<thead>
<tr>
<th>$W_{ij}$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0.1</td>
</tr>
<tr>
<td>$C_2$</td>
<td>-0.1</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0.06</td>
</tr>
<tr>
<td>$C_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The activation function selected for the experiments was the logistic function, because of the soft thresholding. This means that the result of logistic regression can be interpreted as the probability of...
observing certain response and probability 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 $T$:

$$T = (1 \ 1 \ 1 \ 1 \ 1 \ 1)$$

The values in $T$ denote the time interval in which a variable is updated. For example, a value of 1 means that a variable is updated every two frames, and so on. For TimeOver game, due to its simplicity, we assign the value of one to all context variables in $T$. In other contexts, when it is required that different context variables are updated asynchronously, each context variable must have its respective evolving time. For example, to model the factor of rain in an ecosystem, the evolving time of the rain could be 10 if we want to specify that the rain factor is updated every 10 to frame.

The initial values of the six context variables are:

$$S_0 = (1 \ 1 \ 0.5 \ 0.1 \ 0.4 \ 0.09)$$

Figures 4, 5, 6, 7, and 8 illustrate the results of real-time simulations that we designed and conducted to test the behavior of our E-FCM model. Each figure illustrates the context variables in each game frame. Given the initial configuration $S_0$, we expected that in the simulations, the context variables 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.

Figure 4, illustrates that the values of speed, stamina and obstacle type increase and decrease directly proportional, since they obey the causal relationships of the E-FCM model; it also illustrates that the obstacle period is inversely proportional to the speed values, because as the player increase the speed value, the network tends to decrease the obstacle period, adjusting the game difficulty; in contrast, if the player is losing speed, the obstacle period is increased,
leading to a few quantity of obstacles along the gameplay.

The player actions of eating more or fewer items are reflected in the increase and decrease of the stamina value. The items period is proportional to the stamina, but its curve is softer since there is less stamina and the items period is shorter, ensuring that the player will have items to eat, in order to increase his stamina value and, therefore, increase his speed value. The item type is inversely proportional related to the stamina value because of the impact of items when the value of stamina is low: it must be higher so that the stamina value can be increased. Due to these changes, which affect directly to the actions of eating or not the items, the context variables tend to present peaks. Figure 4 illustrates all these changes in real time during 1900 frames.

Figure 5 illustrates a different situation. When stamina increases in frame 200, the values of speed, obstacle type, and item period increase. In this situation, the difficulty level is harder and there are less items to collect, although the player was able to increase his/her stamina and speed. Figure 5 also illustrates that the value of obstacle period decreases, which contributes to make the game harder to play as there will be more obstacles in the game scene.

In the same way, in Figures 6, 7 and 8; the context variable values are updated with the same proportions. The difference in all tests is the player actions along the gameplay, which influences directly on the context variables values illustrated in the figures. Besides that, the player doesn’t play the same game, because the game difficulty is managed dynamically by the network (obstacles and items don’t appear always at the same time, neither are of the same type).

For example, Figure 8 illustrates that when the value of stamina decreases considerably in frame 800, the values of speed, obstacle type, and item period also decrease because stamina directly affects speed. Figure 8 also illustrates that obstacle type with low values correspond to “easy” obstacles and if item period decreases, the number of items created in the game increases. In contrast, if obstacle period increases, the number of obstacles in the game decreases, which may help the player to have better performance in the game.

Figure 8, at frame 800, the speed decreases but then increases, which means that the player started to eat more items recovering his stamina and speed. It is also possible to notice that the obstacles period is increased and the items period decreases, adjusting the game difficulty. However, as shown at frame 2100, the speed value is very low, and it means that the player died because was reached
The proposed method is able to adjust the difficulty and is adaptable to the player needs in real time, improving the gameplay experience. For example in the tests, if the player stamina level is increasing, the speed value is always increasing, the obstacles period decreases and the item period increases, the obstacles and item types also are affected.

Our method also presented three important advantages over other approaches: independence, adaptability, and scalability:

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

- **Adaptability:** As the method is based on E-FCM, it is possible to apply our method to any game;

- **Scalability:** 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. Note that although the number of context variables increases, the number of causal relationships does not increase as fast. The weight matrix is sparse because not all context variables are related each other, especially when there are a lot of context variables.

An interesting future work is to calibrate automatically the weights of the E-FCM model using techniques of interactive evolutionary computing, because the behavior of the E-FCM depends on the design of the network and the matrix of weights is assigned manually. Another future work is adding new concepts to the E-FCM in real-time. This seems promising because, the complexity of the game could be increased in real-time, while the game is learning new features itself, and it allows more dynamicity; also would be interesting to explore automatic methods for learning new causal relationships, in order to facilitate the effort in the model construction of E-FCM.

## 5 Conclusions

Considering the results described in Section 4, we observed that modifying the E-FCM (with player interaction and context variables) produced the desired outcome, as the player plays the game, our method was able to adjust the difficulty levels dynamically using the context variables and player interaction as information sources. Therefore, we conclude that the proposed method is efficient and is adaptable to the player needs in real time, improving the gameplay experience. For example in the tests, if the player stamina level is increasing, the speed value is always increasing, the obstacles period decreases and the item period increases, the obstacles and item types also are affected.

Our method was able to adjust the difficulty and is applicable to the player needs in real time, improving the gameplay experience. For example in the tests, if the player stamina level is increasing, the speed value is always increasing, the obstacles period decreases and the item period increases, the obstacles and item types also are affected.

Our method also presented three important advantages over other approaches: independence, adaptability, and scalability:

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

- **Adaptability:** As the method is based on E-FCM, it is possible to apply our method to any game;

- **Scalability:** 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. Note that although the number of context variables increases, the number of causal relationships does not increase as fast. The weight matrix is sparse because not all context variables are related each other, especially when there are a lot of context variables.

An interesting future work is to calibrate automatically the weights of the E-FCM model using techniques of interactive evolutionary computing, because the behavior of the E-FCM depends on the design of the network and the matrix of weights is assigned manually. Another future work is adding new concepts to the E-FCM in real-time. This seems promising because, the complexity of the game could be increased in real-time, while the game is learning new features itself, and it allows more dynamicity; also would be interesting to explore automatic methods for learning new causal relationships, in order to facilitate the effort in the model construction of E-FCM.

## Acknowledgements

We would like to thank the others developers members of the TimeOver team, José Luis Valencia Gutiérrez, Rommel Anatoli Quintanilla Cruz and Josué Benjamín Arambide Quispe; and the Professional School of Computer Science from National University of Saint Augustine, where TimeOver game was developed as a project in the undergraduate program. We thank CAPES, CNPq, FAPERJ and NVIDIA for supporting the present research.

## References


