ANFIS-Based NPC Population Control in Video Games

Tomasz BOINSKI

Gdansk University of Technology, Gdansk, Poland, tomboins@pg.edu.pl
ORCID: 0000-0001-5928-5782

Abstract

Modern computer games aim at providing rich, vivid worlds. The aim is to encourage the player to explore and interact with the in-game world. To describe the complex relations between in-game NPCs and their surrounding fuzzy logic is used. The paper presents ANFIS based population control in the video game. We present an approach allowing stabilizing the number of NPCs in-game by providing a certain amount of food to the environment. The aim of the solution is to emulate a lifelike but stable environment. The system is tasked to react to the player's actions. The paper describes the solution and provides its evaluation.

Keywords: ANFIS, Fuzzy logic, Video games.

Introduction

In modern computer games, the worlds tend to be richer and richer. In some cases, the vividness of the world should encourage the player to explore and interact with the world. To make the game world more real developers tend to implement dependencies between different types of animals, food availability, player actions, etc. as the world itself is often changing based on those factors. For example, the population of the NPC can change, the amount of food or animals can vary depending on the player acting aggressively or not. Furthermore changing world have to take into account the performance and stability of the game application itself. Too many characters, whether they are inhabitants of the game world or general fauna can hinder the game performance and make the world look unnatural.

We aim at balancing the world in a game called Mrowisko (Anthill) developed at the Faculty of Electronics, Telecommunication and Informatics, Gdansk University of Technology. The game is available online at https://kask.eti.pg.gda.pl/mrowisko/. The game simulates an environment where the ant colony performs daily tasks like food gathering, anthill cleanup from the debris, etc. We aim at keeping the ant population at a certain level so that the anthill and its surrounding will not look empty or overcrowded. The assumption here is that the ants reproduce if there is enough food available in the anthill. The player can however consume the food him or herself or bring food from outside the anthill. Furthermore, the ants can be killed either by the player or the environment thus reducing the current food requirement for the ants to reproduce. All the dependencies are described using fuzzy logic.

In this paper, we propose an ANFIS based approach to balance the number of NPCs (other ants) in the game. In Section 2 to the usage of fuzzy logic in video games are presented. Section 3 the Anthill game is briefly presented. The ant life-cycle and the food generation are also presented (Section 3.1. Next, the network and training approach is presented (Section 3.2) and the evaluation of the proposed solution is given (Section 3.3). Finally, a summary is given.

Fuzzy Logic in Video Games

Fuzzy logic was introduced at the beginning of the 20th century, however, the current definition of fuzzy logic and fuzzy sets were established by Lot Aliasker Zadeh in 1965 (Zadeh 1965). The basic concepts in fuzzy logic are graduation and granulation. In general, everything is or is allowed to be graduated, thus being a subject of degree. Similarly, everything is or is allowed to be granulated, thus having not a concrete value but some category attribution based on e.g. similarity or proximity of the possible values. The animal does not have a hunger value of X, it is either hungry or not. It also can be part hungry, part full. How the given animal belongs to the hungry and full sets is determined by a membership function which value can be gradated.

Computer games developers tend to use rather simple methods during game implementation like decision trees. They main advantage is ease of implementation and, what sometimes is more important, testing (Pirovano 2012). Due to relative ease of

Cite this Article as: Tomasz BOINSKI "ANFIS-Based NPC Population Control in Video Games" Proceedings of the 37th International Business Information Management Association (IBIMA), ISBN: 978-0-9998551-6-4, 1-2 April 2021, Cordoba, Spain

implementation and testing, fuzzy logic proved to be usable in video games, where authors require more complex and versatile solution, however do not want to risk unpredictability of e.g. evolutionary algorithms (Kose 2012; Pirovano 2012). Due to the lack of strict, binary state assignments, the behavior of computer-simulated game world inhabitants can be more natural. The NPC (Non-Player Character) can thus take actions based on multiple changing conditions without the need for complex rules. Due to the ability to calculate membership functions for multiple fuzzy sets, we can introduce multi-modal behavior simulation. As an effect, a huge number of games utilize this approach (Calderon-Vilca, Chavez, and Guimarey 2020) for behavior control of NPS's. Fuzzy logic is often used for decision making (McCuskey 2006). The fuzzy states allow implementation of more gradual state changes when require. This allows for the implemented agent to behave more naturally, without sudden decision changes (Millington 2006). It also allow an input-output relationship implementation based on the expert's knowledge. With fuzzy logic experts reasoning process can be reproduced without usage of sometimes complex mathematical model that might be difficult to implement or test (Wang et al. 2006). The nature of fuzzy logic, where rules can be resolved in any order, allows also easy simplification or complication of the model. Programmers can easily add or remove rules based on the e.g. hardware or time constraints put upon the model (Musilek et al. 2004).

Fuzzy logic can be used for the implementation of adaptive neuro-fuzzy inference system (ANFIS) (Joelianto and Rahmat 2008). The approach was first introduced by Jyh-Shing Roger Jang (J.-S. R. Jang et al. 1991; J. .-. R. Jang 1993). This approach combines fuzzy logic with neural networks, thus allowing the proper setup of the parameters of the fuzzy inference system (Al-Hmouz et al. 2011). This way we are both able to describe nonlinear or unclear dependencies between conditions in the implemented world and can adapt the description to the changes introduced e.g. by the player.

This approach proved to be useful in designing computer games, mainly in the dynamic adaptation of the game world to player actions or his or her performance. ANFIS approach allows thus modification of the world resources to maintain a status quo or dynamic adjustment of the game difficulty to better suit the player capabilities (Araujo, Gonzalez, and Mendez 2019). It can also be used to change the NPC behavior based on the player performance (Desouky and Schwartz 2009).

World Population Control in Anthill Game

The original aim of the Anthill game was to check human behavior in an isolated environment (Kaliszewicz 2018). In the game the player could roam the world performing different tasks, exploring or killing other ants (NPCs). The player and had to eat or it would die from hunger. This aspect was implemented as the player's hunger level, the lower the level the more hungry the player's ant is. There were other ants, further referenced as NPCs, who also need to eat. This aspect is implemented in a form of a colony's global hunger level. Once again the lower the global hunger is the more food is required for the colony. To maintain the anthill the player, in between exploration and performing quests, should gather food and store it in the anthill. In its basic form, the game had two types of NPC's - stationary quest-related ants and primitive "hauler" ants, that move around the hive over predefined paths. The number of hauler NPCs was constant and they did not perform any real actions. They could be however killed by the player. As they did not reproduce this can lead to an empty anthill if the player decides to kill all such NPCs.

To make the world more life-like and provide means for anthill regeneration even during hostile player actions we decided to extend the hauler-ants behavior. First of all the ants do not follow predefined paths around the anthill but are tasked with food gathering. The amount of food required is directly related to the current number of ants in the anthill. If the hunger level is high the ants will also reproduce increasing the number of hauler-ants. This will increase the speed of resource consumption and the speed at which the ants are able to gather food. If the hunger level is low the ants will start to die. In this case, however, the global food requirement to sustain the anthill is reduced. The speed of food replenishment for the colony is also reduced. All the actions and relations between the amount of food in the anthill and the number of ants were described using fuzzy logic. The aim is to use ANFIS to keep the number of ants at a constant level (equal to 50). The system increases the amount of food available in the environment when the number of ants is low (which can be caused by the player killing ants, disrupting food supplies or just eating the food available in the anthill) and decreases when the number of ants is too high or there is too much food (because the player helps with food gathering).

Food Generation and Ant Life Cycle

The ants are required to bring food to the anthill. To increase the amount of food the ants have to travel to the pick-up point, where food is generated, and then return to the certain drop points where food is stored. The available paths can be seen in Fig. 1. If there is food available the ant takes it, reduces its level by one, and returns to the anthill. When the ant reaches proper drop point the global hunger is increased by 1 with each ant successfully bringing food to the drop point and is decreased by *number_of_ants* * 0,3429 * 0,75. The value 0,3429 is the average number of food brought to the anthill during the 60 seconds time interval. The 0,75 modifier ensures that the colony will grow if there is no food shortage and the player does not kill the ants or disrupts the process otherwise (e.g. by killing ants that carry food or destroy the food carried).

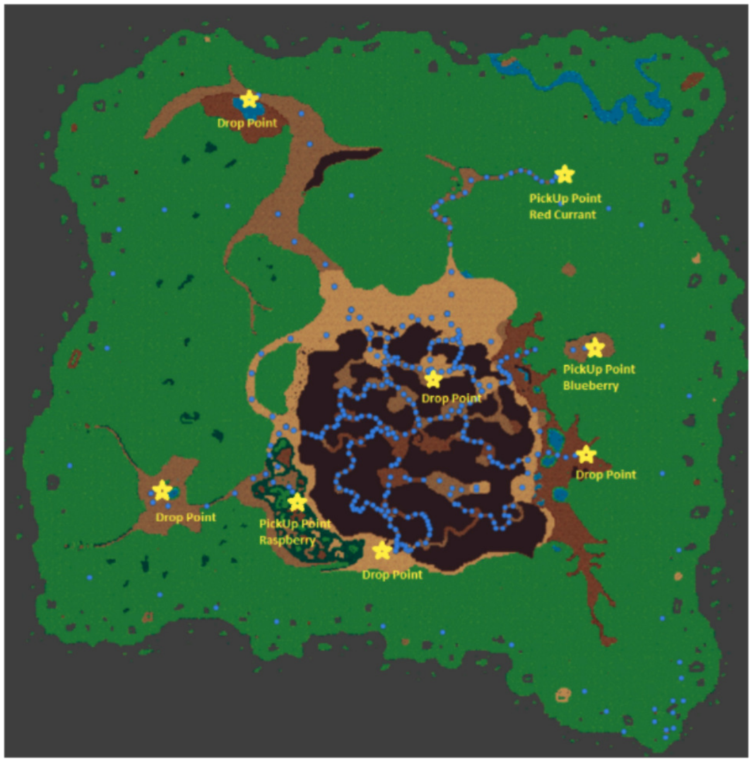

Fig 1. Available paths, pick-up and drop points.

The amount of food in a given pick-up point is presented to the player using one of the 4 representations of 4 fuzzy sets: empty, small, medium, and big (Fig. 2). The representations are assigned by a membership function based on the current level of food available as can be seen in Fig. 3. The number of ants is increased or decreased as follows:

- if the hunger level is at 75% or more new ant is generated,
- if the hunger level is between 25% and 75% the number of ants remain constant,
- if the hunger level is below 25% one ant is killed.

Fig. 2: Graphical representation of the food availability.

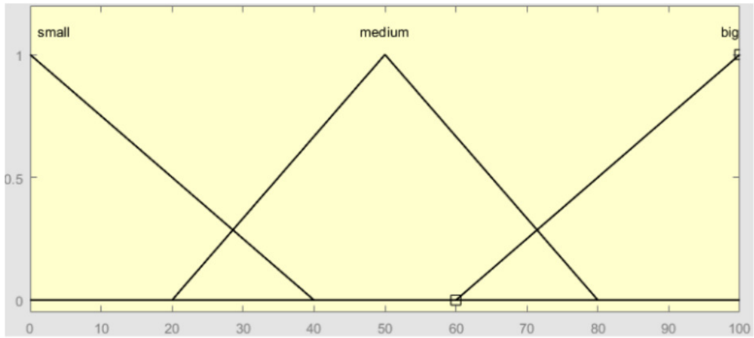

Fig. 3: Fuzzy representations of small, medium and large sets.

Network Training

The training data was prepared manually during play sessions. For possible interactions with the world, we observed how it will impact the amount of food provided and the ability of the anthill to gather it. We also observed how long it takes the ants to gather food on average, what routes are possible etc. Based on the observations we prepared the training set describing the relation between the availability of the food and the number of ants in the colony.

As the membership function, we decided to use the Gaussian function. It requires two mean values and a standard deviation. Our tests show that for each input variable the best results are achieved with four membership functions, each differing by the mean values and the standard deviation.

We used the hybrid approach combining the gradient descent (Werbos and John 1974; Andrychowicz et al. 2016) and lest squares (Miller 2006) approaches. We assumed that the error rate has to be no higher than 0.00001. If it's higher we assume that there is no convergence and we need to do the backward pass. The training was done using 50 epochs or before the convergence is met. Training the network for more than 50 epochs in all cases increased the error value.

The Evaluation

We tested our solution with different scenarios. During all tests, the decisions by the system were calculated every 60 seconds. We tested the solution with 3 scenarios: the killer, the gatherer, and the explorer to see how the system will react to different player behaviors. In all cases, the game started with a population equal to 50 ants, the player's ant hunger equal to 64, and the global hunger level equal to 70.

During the killer scenario play-through, we tested the ability to recover when the number of ants dropped. During this test, we acted as a hostile player that killed the ants. When the colony population dropped to 25 the system required 20 minutes of gameplay to return to the desired population. When we killed more ants, leaving only 10 the system needed 30 minutes to provide enough food for the colony to recreate. An interesting situation happened when the number of hauler ants dropped to 0. In this case, the system reacted correctly by providing the maximum amount of food. However, there were no workers to bring it to the anthill. In this case, the player had to intervene and manually bring food to the anthill. When there was enough food the ants started to spawn and the colony once again returned to the desired state.

During the gatherer, the player was tasked with bringing food to the anthill and doing all the quests that increase the global hunger of the colony. In this case, the system has limited control over how much food is brought into the colony. The NPC-gathered food was not generated, however, the player still could bring other consumables to the anthill and performed certain quests. After 20 minutes of playtime, we managed to increase the colony population to 100 ants. After this time we stopped bringing food. The global hunger level quickly dropped as there was no food available and the ants needed to eat. After 20 minutes the population dropped to 31 and at this time started to rebuild itself. Once again the system stabilized at a population equal to 50 ants. Based on this test we can see that the solution could be extended so that the quests available that rely on bringing food to the colony also should be connected to the current hunger and population levels.

During the third test, the player was tasked with exploring the environment and doing a quest that did not impact the global hunger level. During the initial state of the game the population rose to 80 ants however at this point the system limited the amount of food available and the population dropped to 45. Next, it went back to 50 and stayed at this level for at least an hour.

The initial rise in the population was dependent on the initial amount of food available at the beginning of the game. The ants also brought more food than they consumed. After 40 minutes the situation settled and the colony was stable.

Summary and Future Work

The proposed solution allows population control in video games. The aim was to be able to react to player actions and keep the world vivid and lifelike. By using ANFIS we were able to achieve this goal. The system reacts accordingly to the player actions (or lack of it) keeping the game NPCs population in check.

The proposed solution can be used not only to control the look and feel of the game but also can be used to manage in-game resources, especially the ones that naturally restore over time. It can also be used to represent the state of e.g. enemy colony dependently on the current actions of the player.

In the future we plan on extending the solution to better simulate the game world, e.g. there should be fewer quests to bring food to the anthill if the population is high, so that it will not grow too much as we could see in the gatherer test.

Acknowledgements

I would like to thank my students, Joanna Litwin, Dawid, Łuczak and Stanislaw Kosiński for their invaluable help in the implementation of the presented solution.

References

- Al-Hmouz, Ahmed et al. (2011). "Modeling and simulation of an adaptive neuro-fuzzy inference system (ANFIS) for mobile learning". In: IEEE Transactions on Learning Technologies 5.3, pp. 226-237.
- Andrychowicz, Marcin et al. (2016). "Learning to learn by gradient descent by gradient descent". In: arXiv preprint arXiv:1606.04474.
- Araujo, Vladimir, Alejandra Gonzalez, and Diego Mendez (2019). "Dynamic Difficulty Adjustment for a Memory Game". In: Technology Trends. Ed. by Miguel Botto-Tobar et al. Cham: Springer International Publishing, pp. 605-616. isbn: 978-3-030-05532-5.
- Calderon-Vilca, H., N. M. Chavez, and J. M. R. Guimarey (2020). "Recommendation of Videogames with Fuzzy Logic". In: 2020 27th Conference of Open Innovations Association (FRUCT), pp. 27-37. doi: 10.23919/FRUCT49677.2020.9211082.
- Desouky, S. F. and H. M. Schwartz (2009). "Hybrid intelligent systems applied to the pursuit-evasion game". In: 2009 IEEE International Conference on Systems, Man and Cybernetics, pp. 2603-2608. doi: 10.1109/ICSMC.2009.5346113.
- Jang, J. -. R. (1993). "ANFIS: adaptive-network-based fuzzy inference system". In: IEEE Transactions on Systems, Man, and Cybernetics 23.3, pp. 665-685. doi: 10.1109/21.256541.
- Jang, Jyh-Shing Roger et al. (1991). "Fuzzy modeling using generalized neural networks and kalman filter algorithm." In: AAAI. Vol. 91, pp. 762-767.
- Joelianto, Endra and Basuki Rahmat (Jan. 2008). "Adaptive neuro fuzzy inference system (ANFIS) with error backpropagation algorithm using mapping function." In: International Journal of Artificial Intelligence [electronic only]
- Kaliszewicz, Anna (2018). "Analysis of human behavior using open-world games. (in Polish)". MA thesis. Poland: Gdansk University of Technology.
- Kose, Utku (2012). "Developing a fuzzy logic based game system." In: Computer Technology and Application 3.7.
- McCuskey, Mason (2000). "Fuzzy logic for video games." In: GameProgramming Gems 1.
- Millington, Ian (2006). "Artificial Intelligence for Games." In: The Morgan Kaufmann Series in Interactive 3D Technology. Morgan Kaufmann Publishers Inc., San Francisco, CA,USA.
- Miller, Steven J (2006). "The method of least squares." In: Mathematics Department Brown University 8, pp. 1-7.
- Musilek, Petr,Li, Yifan and Wyard-Scott, Loren (2004). "Fuzzy logic in agent-based game design." In: Fuzzy Information, 2004. Processing NAFIPS '04. IEEE Annual Meeting of the.
- Pirovano, Michele (2012). "The use of fuzzy logic for artificial intelligence in games." In: University of Milano, Milano.
- Wang, Jiljang and Wong, Gabriyel (2006). "A fuzzy-control approach to managing scene complexity." In: Game Programming Gems 6.

- Werbos, Paul and Paul John (Jan. 1974). “Beyond regression : new tools for prediction and analysis in the behavioral sciences.”
- Zadeh, Lot Aliasker (1965). “Fuzzy sets, Information and Control, vol. 8.” In: Google Scholar Google Scholar Digital Library Digital Library, pp. 338-353.