Creature Learning to Cross a CA Simulated Road

Anna T. Lawniczak[1,2], Jason B. Ernst[3], and Bruno N. Di Stefano[2,4]

[1] Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, Canada
alawnicz@uoguelph.ca
[2] The Fields Institute for Research in Mathematical Sciences, Toronto, Ontario, Canada
[3] School of Computer Science, University of Guelph, Guelph, Ontario, Canada
jernst@uoguelph.ca
[4] Nuptek Systems Ltd, Toronto, Ontario, Canada
bruno.distefano@nupteksystems.com, b.distefano@ieee.org

Abstract. Agent-based models approximate the behaviour of simple natural and man-made systems. We present a simple cognitive agent capable of evaluating if a strategy has been applied successfully and capable of applying this strategy again with small changes to a similar but new situation. We describe some experimental results, present our conclusions, and outlines future work.

Keywords: Agents, Cognitive Agents, Learning, Fuzzy Logic, Fuzzy Learning, Cellular Automata, Rule 184, Nagel-Schreckenberg model.

1 Introduction

An agent is an abstraction of an autonomous entity capable of interacting with its environment and other agents, [1], [2], [3], [4], [5]. The actual embodiment of the agent, e.g. software program in a simulated reality (e.g., virtual reality) or a stand-alone hardware, depends on the problem for which the agent has been designed. Even when the agent results in something that is actually built in a tangible form, almost always a software instantiation exists to model and simulate it before it is actually built. This makes mistakes and misunderstandings explicit, thus avoiding costly hardware modifications. In a given simulated reality, agents perform "Reflexive Acts", i.e. perceiving reality (the environment and the other agents) and responding to the actions (i.e., the dynamics) of other entities. The ability of agents to act autonomously is limited to the predefined environment and to the predefined situations to which the agent is expected to respond, because agents can only act in a situation compatible with the way they are designed. In fact the behaviour of the agent is provided by means of a finite state machine or a set of finite state machines. A problem with all finite state machines is that their design, verification, validation, coding, and testing becomes progressively harder when trying to prepare the finite state machine for all possible scenarios beyond a small number. "Cognitive agents" partially solve this problem by performing "Cognitive Acts" (i.e., a sequence consisting of all of the following acts: Perceiving, Reasoning, Judging, Responding, and Learning). The functionality and performance of cognitive agents requires replacing the finite state

G.C. Sirakoulis and S. Bandini (Eds.): ACRI 2012, LNCS 7495, pp. 425–433, 2012.

machine typical of "non cognitive" agents, i.e. "reactive agents", with more complex functional blocks, built using computational intelligence methodology, i.e.: fuzzy logic, neural networks, evolutionary computation, and various types of bio-mimicry [6], [7]. In reality, also cognitive agents are implemented by means of software or a mix of hardware and software and are still far from the performance of animals and humans.

In this paper we explore what may be considered some of the simplest possible cognitive functions of a primitive cognitive agent.

The paper is structured as follows: section 2 describes the universe of the problem domain that we are studying (i.e., the environment, the population of agents, and the experiment); section 2.1 describes and discusses the learning algorithms implemented for our agents; section 2.2 describes our model and its software implementation; section 3 contains some experimental results; section 4 presents our conclusions and outlines future work.

2 The Environment, the Population of Agents, and the Experiment

For our research we assume that an agent is "*an autonomous entity capable of interacting with its environment and other agents*", [1], [2], [3], [4], [5]. For simplicity and for ease of visualization as in [8], we assume that the agent is a creature with a strong instinct to forage for food. The environment is a piece of land with a long stretch of highway characterized by unidirectional vehicular traffic, without any intersection [8]. In some scenarios the highway is a single lane highway, while in other scenarios it can be a two or three lane highway. While in this paper we do not report on bidirectional traffic, bidirectional vehicular traffic is conceivable in our model. We assume that each creature must cross the highway in order to reach food. However, given the presence of vehicular traffic, crossing the highway may or may not be successful for the creature. If successful, the creature simply crosses, reaches the food and never crosses again. Otherwise if not successful, the creature is struck by a vehicle and dies. Crossing may happen at any point of the highway.

We assume that the creatures are a population of agents who can observe the outcome of the previous attempt to cross by other creatures of the same species [8]. Based on this observation each creature may decide to postpone crossing if the situation resembles one that has resulted in the death of another creature that has previously crossed. However, for realism's sake we assume that the creature cannot derive quantitative information from its observations. Thus, the creature can perceive only "fuzzy" categories for speeds such as "fast" "medium" and "slow", and proximities such as "close distance", "medium distance", and "far distance".

The experiment consists in studying how the creature can "naturally" learn to avoid being struck by vehicles after having observed a sufficient number of other creatures attempting the same or similar crossings. The creature is capable of applying several learning algorithms. We model the motion of the vehicles according to the Nagel-Schreckenberg model, [9], which can be seen as an extension of ECA (Elementary

CA) Rule 184. This rule accurately describes the motion of a vehicle at constant speed of one cell per time step and null acceleration. This is unrealistic, but is a good starting point to apply extensions to Rule 184 as it may be needed. It is important to notice that Rule 184 is deterministic and cannot simulate real traffic with accidents. The Nagel-Schreckenberg model solves the problem adding stochastic behaviour, a larger size neighbourhood that can be used to implement variable speed and non null acceleration. Nagel and Schreckenberg extend the neighbourhood from one cell (as in ECA Rule 184) to five cells. They introduce six discrete velocities. The model consists of four steps that are applied simultaneously to all cars:

- Acceleration
- Safety Distance Adjustment
- Randomization
- Change of Position

For our investigation the implementation of the Nagel-Schreckenberg model requires to modify the Cellular Automata (CA) paradigm and to make the evolution of the CA not only dependent on the state of the neighbourhood but also on the current velocity of each vehicle. This implies that each cell is characterized not only by presence or absence of a vehicle but also by a pointer to a data structure containing the current velocity of the vehicle. The motion of the creature is modeled similarly to the motion of the vehicle, with a CA-like approach. However, the creature decides if to move or not to move not only based on positional criteria, but also on reasoning dependent on what the creature has learned observing the prior experience of other creatures.

2.1 The Learning Algorithm

A lot of interesting research on learning algorithms has been conducted [9], [11]. Many algorithms have been developed for various situations [6], [7]. However, at this stage our interest is to start with the simplest possible algorithm, the one requiring the least complex brain [8]. We assume that our creature is not capable of detailed quantitative reasoning. We apply two main learning algorithms: a simple "naïve" algorithm and a fuzzy logic based algorithm.

We First Describe the Simple "Naïve" Algorithm

We assume that the creature is capable of matching simple patterns. If a set of values of distance, velocity, and crossing point resulted in success (or failure) for other creatures, the creature attempting to cross the highway will expect that a similar set of values will result in a similar outcome, i.e. success or failure as it may apply. We assume that the creature will always repeat the action that has previously resulted in success. If the set of distance, velocity and crossing point does not correspond to a known outcome, i.e. if this situation has never occurred earlier, the creature will assume that crossing is possible. At each cell where the crossing may take place the creature builds a "mental" table with all possible outcomes for all possible combinations of vehicle distance and vehicle velocity. When behaving according to the "naïve" algorithm, the creature has an optimistic approach. What is not known to have failed earlier, it is assumed to be successful. A “0” in the table means that either that situation has never occurred earlier or that it resulted in success. If crossing results in a creature being struck all other creatures will “write” a “1” for the specific (distance, velocity) combination, while if the crossing is successful a “0” will be left for the specific (distance, velocity) combination. In other words, the mental table in the beginning is populated with 0s in the assumption that all possible distance velocity combinations allow crossing.

“Naïve” Algorithm with “Fear” and “Desire”

The mental table described so far is the knowledgebase of the creature. The “naïve” algorithm is totally based on this knowledge-base. However, “fear” and “desire” may alter the behavior of the creature. When a creature is created, its fear and desire are both random between 0 and 1. The algorithm can use this information so that creatures which are fearless and have a strong motivation such as availability of food on the other side of the road are more likely to cross in risky situations, as opposed to fearful creatures that even under safe conditions may not cross the highway. However, it is possible to use these two parameters to make all creatures behave as fearless/fearful. In short “fear” and “desire” act as modifiers of the decision made by the creature if to cross or not to cross.

Fuzzy Logic Based Algorithm

Two types of fuzzy logic inference algorithms are available to the creature, one using three membership functions and the other using five membership functions to evaluate the distance and velocity of the vehicle to decide if to cross or not to cross. Currently, the membership functions are triangle shaped, but more complex shapes are possible and will be studied in the future.

2.2 The Model and Its Software Implementation

The simulation software implementation is based on a configuration file containing in plain ascii text all parameters of the specific experiment.

The highway traffic is modeled adopting the Nagel – Schreckenberg model, [9]. As customary in the traffic modeling literature, we model the one lane highway as a large number of adjacent cells, with each cell representing a segment of highway of 7.5m in length [12]. Such representation has been chosen because it corresponds on average to the space occupied by the typical car plus the distance to the preceding car in a situation of dense traffic jam of cars of more or less homogeneous length (i.e., trucks and busses are excluded). The simulation also supports multiple lanes.

The program is based on two loops, an “external” time loop and an “internal” space loop [12]. The time loop simulates the passing of time, assigning a number of seconds to each time step. The space loop “scans” the representation of the physical highways, where distances are converted into cells, for every cell checking if the cell is occupied by a car or not. If it is not occupied by a car the next cell is examined, while if the cell is occupied by a car the “rule of transition” (also called “rule of motion”) is applied. At each time step in the simulation, for each lane, a new car may be generated with a probability specified in the configuration file as car creation probability. If there is already a car in the first lane because it hasn't sped up enough, or traffic is congested, it is added to a queue of cars waiting to enter the highway. The entrance point is always cell zero of each lane. Cars move according to the Nagel – Schreckenberg model. That is to say, they accelerate by one until they reach their maximum speed which is specified in the configuration file. With the assumption that the cells are 7.5m, a maximum speed of 10 corresponds to 99 km/h. If a car encounters another car in front of it, it slows down to match the speed to avoid a collision. The simulation also supports the idea of random deceleration of cars (as specified by the Nagel-Schrekenberg model, [9]), and can be turned on or off by setting RANDOM_DECEL to TRUE in the configuration file. So far we have not experimented with this, as we commence our investigation focusing on understanding simple experiments. It is our intention to explore more complex situations in the future.

The creatures are implemented in a fashion similar to the cars. They also use a queue so that if a creature has not yet crossed, the new creatures line up behind it. The creatures are generated with a creation probability at each time step, and at each cross point. Cross points are specified in the configuration file and can be repeated as many times as there are places for the creature to cross. When a creature crosses, it does so one lane at a time. In the presented study we consider only one fixed crossing point and one lane highway. In a single time step, the creature looks at the environment (where cars are, and with what speed they are travelling) then decides to move. If it decides to move, it moves onto the highway, then the cars move. If a car moves into the cell the creature is occupying or to a cell with coordinate number higher than the creature's cell number, then the creature is hit. In order for a creature to decide whether to move, it must consult the global "knowledgebase" derived from the creatures' past experiences. This "knowledgebase" is a table of states and results. The table has "fuzzy" categories for speed (e.g., "fast", "medium", and "slow") and proximities (e.g., "close", "med", and "far"). The values are all set in the configuration file for "fast", "medium", "slow", "close" and "far". When a creature attempts to cross under one of these conditions, if it is hit, a negative result is recorded. If the creature successfully crosses, a positive result is recorded. In a multi-lane highway, the result is propagated to all of lanes – condition pairs the creature encountered on its trip across the lanes.

3 Experimental Results

The software developed to simulate our model allows, among other things, to log the results of the simulation. At this preliminary stage of our research, we have produced 500 simulation plots, each consisting of 1009 time steps. In what follows, as an example, we show two sets of plots for different simulation parameters.

Figure 1 shows the "crossing success rate" for different values of vehicle creation probability, which is equivalent to the traffic density on the road. In all cases the crossing algorithm is the "naïve" algorithm. The stretch of road under examination (i.e., being simulated) is 1009 cells long (equivalent to 7567.5 m). The road consists of 1 lane only. The duration of the simulation is 1009 time steps (only 50 displayed).

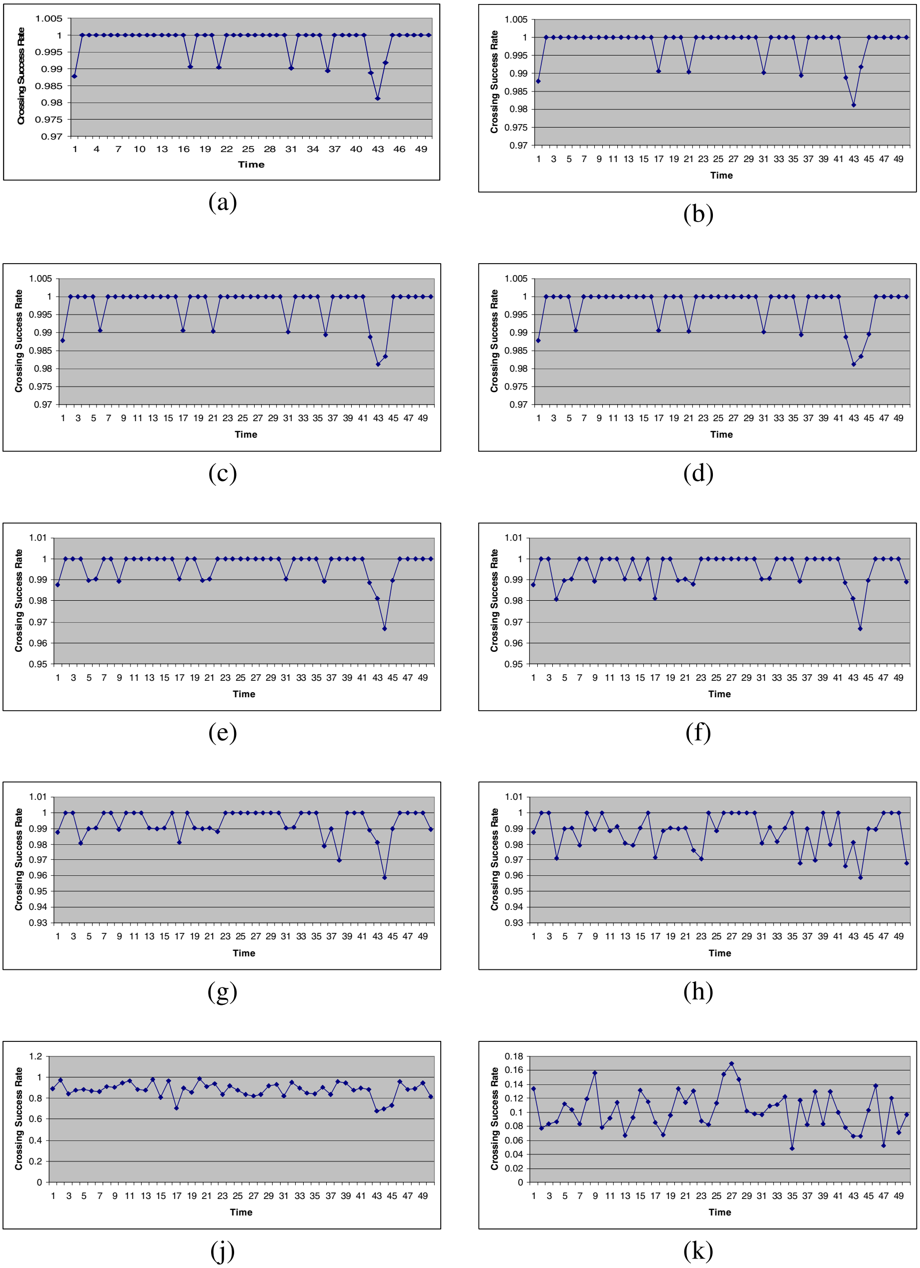
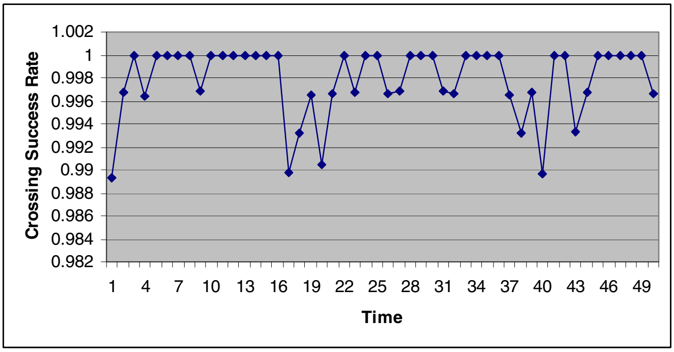
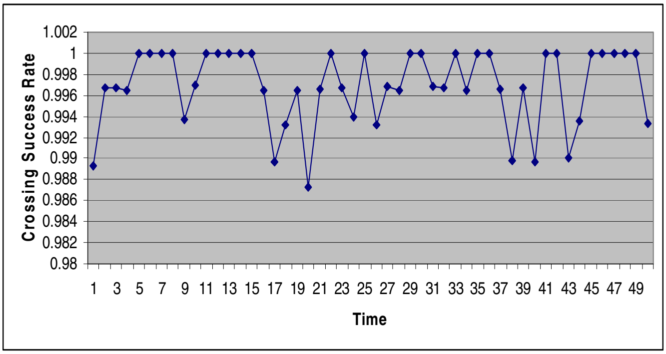


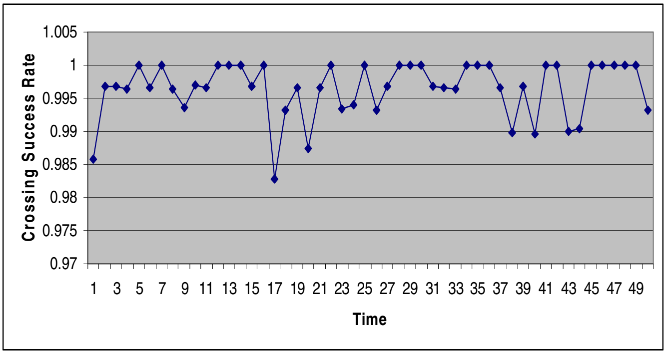
Fig. 1. “naïve” Algorithm; 1009 cells (equivalent to 7567.5 m); 1 lane; 1009 time steps (only 50 displayed); creature creation probability 0.1 – Vehicle creation probability: (a) 0.1, (b) 0.2, (c) 0.3, (d) 0.4, (e) 0.5, (f) 0.6, (g) 0.7, (h) 0.8, (j) 0.9, (k) 1.0



(a)



(b)



(c)

Fig. 2. "naïve" Algorithm; 1009 cells (equivalent to 7567.5 m); 1 lane; 1009 time steps (only 50 displayed); creature creation probability 0.3 – Vehicle creation probability: (a) 0.1, (b) 0.2, (c) 0.3

The creature creation probability is 0.1, while the vehicle creation probability ranges from 0.1, to 1.0 (i.e., case when at each time step a vehicle is always created. The plots show that with denser traffic the "naïve" algorithm is not sufficient to allow for a safe crossing. Figure 2 shows how with creature creation probability equal to 0.3, the crossing success rate deteriorates already for values of vehicle creation probability: 0.1, 0.2, and 0.3.

4 Conclusions and Future Work

Agent-based models approximate the behaviour of simple natural and man-made systems. We present a simple cognitive agent capable of evaluating if a strategy has been applied successfully and capable of applying this strategy again with small changes to a similar but new situation.

Preliminary results shown in section 3 indicate that simple pattern matching as in the "naïve" algorithm is not sufficient to allow safe crossing. We suspect that "jaywalkers", as pedestrians crossing illegally between traffic lights are called in North America, have much better cognitive equipment than the little creatures of our experiment. We are planning to continue our study first by examining the outcome of the adoption of the two fuzzy inference algorithms programmed in our simulator. Later, if needed, we will apply other computational intelligence techniques such as genetic algorithm,

Acknowledgments. A. T. Lawniczak acknowledges partial financial support from the Natural Science and Engineering Research Council (NSERC) of Canada. B. N. Di Stefano acknowledges full financial support from Nuptek Systems Ltd., J. Ernst acknowledges full financial support from a SHARCNET Research Fellowship provided by A. T. Lawniczak. B. N. Di Stefano and A. T. Lawniczak acknowledge the hospitality of The Fields Institute for Research in Mathematical Sciences during the preparation of this manuscript. They also acknowledge the contribution of Prof. K. N. Plataniotis who invited them to a "Cognitive Agents Roundtable" at IPSI ("Identity, Privacy and Security Institute", at the University of Toronto), on March 13, 2009. Lawniczak and Di Stefano were inspired by the problems discussed at this roundtable and decided to search a solution for some of these problems. B.N.Di Stefano acknowledges the inspiration derived from the open discussion following his delivery of the tutorial "Simulation of cognitive multi-agents", on Friday June 18th 2010, during SOLSTICE 2010 (2nd Summer Solstice International Conference on Discrete Models of Complex Systems), LORIA laboratory (http://www.loria.fr/le-loria-1-en), Nancy, France.

References

1. Ferber, J.: Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison Wesley, London (1999)
2. Di Stefano, B.N., Lawniczak, A.T.: Cognitive agents: Functionality & performance requirements and a proposed software architecture. In: 2009 IEEE Toronto International Conference on Proc. of Science and Technology for Humanity (TIC-STH), pp. 509–514 (2009)

3. Lawniczak, A.T., Di Stefano, B.N.: Computational intelligence based architecture for cognitive agents. In: ICCS 2010, Procedia Computer Science, Amsterdam, Holland, May 31 - June 2, vol. 1(1), pp. 2221–2229. Elsevier (2010)
4. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, Ltd., Chichester (2009)
5. Uhrmacher, A.M., Weyns, D.: Multi-Agent Systems Simulation and Applications. CRC Press, Taylor & Francis Group, Boca Raton (2009)
6. Kasabov, N.K.: Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering. The MIT Press, Cambridge (1988)
7. Tettamanzi, A., Tomassini, M.: Soft Computing, Integrating Evolutionary, Neural, and Fuzzy Systems. Springer, Heidelberg (2001)
8. Di Stefano, B.N., Lawniczak, A.T.: Modeling a Simple Adaptive Cognitive Agent. Acta Physica Polonica, B Proceedings Supplement 5(1), 21–29 (2012)
9. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Physique I 2, 2221–2229 (1992)
10. Marsland, S.: Machine Learning: An Algorithm Perspective. Chapman and Hall/CRC (2009)
11. Alpaydin, E.: Introduction to Machine Learning, 2nd edn. The MIT Press, Cambridge (2009)
12. Lawniczak, A.T., Di Stefano, B.N.: Development of CA model of highway traffic. In: Adamatzky, A., Alonso-Sanz, R., Lawniczak, A., Martinez, G.J., Morita, K., Worsch, T. (eds.) Automata-2008, Theory and Applications of Cellular Automata. Luniver Press, U.K. (2008)