



Procedia Computer Science

Volume 80, 2016, Pages 542–553

ICCS 2016. The International Conference on Computational Science



Success Rate of Creatures Crossing a Highway as a Function of Model Parameters

Anna T. Lawniczak^{1*}, Leslie Ly², and Fei Yu³^{1,2,3}University of Guelph, Guelph, Ontario, Canada¹alawnicz@uoguelph.ca, ²lyl@alumni.uoguelph.ca, ³fyu03@uoguelph.ca

Abstract

In modeling swarms of autonomous robots, individual robots may be identified as cognitive agents. We describe a model of population of simple cognitive agents, naïve creatures, learning to safely cross a cellular automaton based highway. These creatures have the ability to learn from each other by evaluating if creatures in the past were successful in crossing the highway for their current situation. The creatures use “observational social learning” mechanism in their decision to cross the highway or not. The model parameters heavily influence the learning outcomes examined through the collected simulation metrics. We study how these parameters, in particular the knowledge base, influence the creatures’ success rate of crossing the highway.

Keywords: Autonomous robots, agents, cognitive agents, learning, cellular automata, computational intelligence, data visualization

1 Introduction

For the development of swarms of autonomous robots, which may require them to learn how to accomplish some tasks in unknown dynamically changing environments, it is important to study the process of learning through observation and repetition, [1]- [3]. Since autonomous robots may be identified with cognitive agents, [4]- [7], this type of study may be carried out through modeling and simulation, which allows to test various ideas before building actual robots. Thus, avoiding or minimizing any mistakes of major consequences.

In this work, we investigate the performance of a simple learning algorithm based on an “observational social learning” mechanism, [1], [8]- [10] where each cognitive agent learns by observing the outcomes from the actions of cognitive agents that have already attempted to carry out a task and imitating the successful ones. The principles of the “observational social learning” mechanism have been applied in the context of multi-agent learning and to develop new optimization algorithms [11]- [14], among others for the purpose of finding more effective optimizers than swarm

* Corresponding author. The authors acknowledge useful discussions with B. Di Stefano, H. Wu and S. Xie. A. T. L. acknowledges partial financial support from the NSERC of Canada.

intelligence algorithms. With the goal in mind of possible hardware implementations, our interest is in identifying the simplest possible architecture still capable of producing meaningful results for the desired task, thus “*our emphasis is on minimal entities, both in terms of storage and in terms of logical primitives (e.g., conjunction, disjunction, and negation)*”. This is the benefit of the proposed and explored learning algorithm in [15]- [20], which we investigate further in this paper.

We consider the model of naïve creatures learning to cross a highway introduced and described in [15]- [20]. In this model, creatures use a simple decision making formula and build their knowledge base (KnB) by observing the performance of the creatures that attempted to cross the highway earlier. We investigate the effects of creatures’ KnB accumulation through learning and its interaction with other model parameters on creatures’ success rate of crossing a highway. In particular, we investigate the effects of the transferring of the KnB after building it through many populations of creatures within the same environment (i.e., the highway with the same vehicle traffic density) as opposed to building it through only a single population of creatures within the same environment.

The paper is organized as follows: Section II describes the model; Section III describes simulation setup and resulting data; Section IV discusses the effects of creatures’ KnB accumulation and its interaction with other model parameters on creatures’ success rate of crossing a highway, and Section V reports our conclusions and outlines future work.

2 Model of Creatures with Fear and Desire Learning to Cross a Highway

We review here only the main features of the model, which has been developed and described in [15]- [20], we refer the reader for details there. We assume here that (1) the environment is a single lane unidirectional highway without any intersection; (2) a creature is “an autonomous entity, capable of interacting with its environment and other agents”, which has a strong instinct to survive; (3) all creatures are born on one side of the highway and want to cross it without being killed by the oncoming vehicles.

We assume that each creature is capable of: (1) matching simple patterns; (2) evaluating distances in an approximate way; (3) evaluating the velocity of moving vehicles in an approximate way; (4) assigning a discrete number to an approximate class; (5) witnessing what had happened to the creatures that previously crossed the highway at this crossing point (with exclusion of the first creature); (6) evaluating what they witnessed in (5), i.e. if the crossing was successful or not; (7) imitating the creatures which crossed successfully; (8) deciding not to cross and wait for better conditions or to look for a different crossing point when unsuccessful crossings outnumber the successful ones. All of these allow each crossing point (CP) to build one knowledge base (KnB) during the experiment that is available to all creatures at that CP.

We model the highway traffic by adopting the Nagel-Schreckenberg cellular automaton model, see for details [21]- [24]. The model consists of four steps that are applied simultaneously to all cars: acceleration, safety distance adjustment, randomization, and change of position. The cars are generated at “starting cells” randomly with car creation probability (Car Prob.) and are assigned a random speed between zero and the maximum allowed speed for cars, which is set in the configuration file. As customary in the traffic modeling literature, each cell represents a segment of a highway of 7.5m in length, [21].

In our study at each time step, creatures are generated only at the CP set at the initialization step and are placed into the queue at this CP. Each generated creature falls with equal probability (0.25) into one of the four categories: (1) no fear nor desire; (2) only fear; (3) only desire; (4) both fear and desire. The creatures’ attributes play a role in their decision making process on whether or not to cross

the highway through the values of fear (aversion to risk taking) and desire (propensity to risk taking) that creature may experience.

The creatures attempt to cross the highway having a limited horizon of vision and perceiving only fuzzy levels of distance (e.g., “close”, “medium”, “far”) of cars within this horizon and their speeds (e.g., “slow”, “medium”, “fast”, “very fast”). The ranges for these qualitative categories are set in the simulator’s configuration file. The creatures may build up in the queue as a result of not crossing at each time step. If the simulation setup permits, after deciding not to cross the highway, a creature may move randomly along the highway horizontally in either traffic direction to a new CP or it may stay at the same CP with equal probability of 1/3. The number of horizontal cells a creature may move in one-time step is 1 and the maximum distance the creature may deviate from its original CP in both directions is 5. If the creature at the top of a queue leaves the queue, the creature that was behind moves to the top of the queue. When a creature crosses the highway at a given CP, information is recorded into the knowledge base (KnB) of all the creatures at this CP. The KnB table’s rows and columns index the different combinations of the qualitative distance (such as “close”, “medium”, “far”) and velocity (such as “slow”, “medium”, “fast”, “very fast”) categories, as perceived by the creatures. Each KnB table entry is calculated by: the number of the “*successful crossings*” minus the number of the “*unsuccessful crossings*” for that particular (distance, velocity) pair up to this time.

The KnB table is initialized as “tabula rasa”, i.e. with all its entries set to 0, allowing creatures to cross the highway regardless of the observed (distance, velocity) levels until the first successful crossing of a creature, or five consecutive unsuccessful crossings, whichever comes first. If a creature successfully crosses the highway, the perceived (distance, velocity) score in the KnB table is increased by one point. If the creature was killed, it is decreased by one point.

After the initialization of simulation, each creature at the top of the queue consults the KnB table to decide if it is safe or not to cross. Its decision is based on the implemented intelligence/decision making algorithm, which for a given (distance, velocity) pair combines the “*success ratio*” of crossing the highway for this (distance, velocity) pair with the creature’s fear and/or desire values, as follows.

For each (distance, velocity) pair at each time step, the numerator in the *success ratio* is the current value from the KnB table entry. Recall that it is the number of “*successful crossing*” minus the number of “*unsuccessful crossings*” for this (distance, velocity) pair up to this time. The denominator is the total number of creatures that have crossed successfully the highway regardless of the (distance, velocity) combination up to this time; i.e. it is the number describing the creatures’ entire population success up to this time. If for some (distance, velocity) configuration at the simulation start, all creatures are killed, ratio becomes “-5/0”. In this case, we set the success ratio to zero since “division by zero” is undefined.

A randomly generated creature will base its decision on the formula: (1) “*success ratio + value of desire – value of fear*”, if it has both fear and desire; (2) “*success ratio – value of fear*”, if it has only fear; (3) “*success ratio + value of desire*”, if it has only desire; (4) “*success ratio*”, if it has no fear and no desire. If a creature and a given (distance, velocity) combination yield from the formula, a value that is less than zero, then the creature will not attempt to cross the highway under this condition and it will wait for a configuration for which the value of the formula is non-negative or it may decide to move to another crossing point.

The main simulation loop of the model consists of: (1) generation of cars at each lane of the highway using the Car Prob.; (2) generation of creatures at each CP with their attributes; (3) update the car speeds according the Nagel-Schreckenberg model; (4) movement of the creatures from the CP queues into the highway (if the decision algorithm indicates this should occur); (5) update of locations of the cars on the highway. This includes passing other cars in the case of multi-lane highway. It also includes the logic to check if any creature has been killed; (6) advancement of the current time step. After the simulation has been completed, the results are written to output files using an output function.

3 Simulation Setup and Simulation Data

A single run of the simulation lasts for a number of time steps set up at the initialization step. Various output files are generated, e.g. (1) file containing the total number of creatures that have successfully crossed the highway, the total number of creatures that were killed while crossing, and the number of queued creatures at the end of the simulation; (2) file containing time dependent data, i.e. at the end of each discrete time step, the number of creatures that have successfully crossed the highway up to this time, the number of creatures that were killed while crossing up to this time, and the number of queued creatures at this time; (3) file containing the state of the KnB table at each time step. A large amount of data files is generated when the simulation is looped many times both at a particular configuration of the adjustable simulation parameters/factors (to create repetitions) and also at different configurations of the parameters (in order for comparison).

We consider the data generated from the simulation looped many times. The parameters that remain constant are: one lane highway with a length of 120 cells (900 meters long), 1511 time steps, 30 repetitions, random deceleration equal to 0 (there are no erratic drivers), and a 3 by 4 KnB table with an extra entry. The KnB table has 3 groupings of distance and 4 groupings of speed. The creatures in this case specifically perceive: (1) “close” for a vehicle being 0 to 3 cells away, “medium” if it is 4 or 5 cells away, “far” if it is 6 or 7 cells away and “out of range” if it is 8 or more cells away, regardless of the velocity of the vehicle, and this is encoded in the extra entry; (2) “slow” when the perceived velocity of a vehicle is 0 to 3 cells per time step, “medium” when it is 4 or 5 cells per time step, and “fast” when it is 6 or 7 cells per time step and “very fast” when it is 8 to 11 cells per time step. A vehicle’s max speed is at 11 cells per time step.

There are 6 parameters that vary through the main simulation loop. These parameters are: (1) car creation probability (Car Prob.); (2) crossing point (CP); (3) value of fear, (4) value of desire; (5) the KnB transfer direction (KnB Transf.), and (6) horizontal creature movement (Horiz. Cre.).

The Car Prob. determines the density of the vehicle traffic and it varies between the values: 0.1, 0.3, 0.5, 0.7, and 0.9. A vehicle is generated at the start of the highway at each time step with a given Car. Prob.

The CP determines the location at which the creature will cross the highway and it varies between the values: 40, 60, and 80 (the cell number of the highway). The distance from where the vehicles are generated is important because it will affect the nature of the vehicle traffic. For example, there will likely be more vehicles travelling at max speed and in a more homogenous manner near CP 80 than at CP 40.

The value of fear and value of desire parameters both vary between the values: 0, 0.25, 0.5, 0.75, and 1. Being a part of the decision formula, these values influence the creatures’ decision making process of whether or not to cross.

The KnB Transf. parameter varies from: “none”, “forward”, and “backward”, and determines whether or not the KnB table at the end of one run of the simulation is transferred to the beginning of the simulation at a different traffic environment (i.e., with a different value of Car Prob.). When KnB Transf. is set to “none”, the KnB table is not transferred from an environment with one car traffic density to an environment with another car traffic density. When KnB Transf. is set to “forward”, the KnB table is transferred from a less dense traffic environment to the one with immediately denser traffic. In this case, the simulations start in the environment with Car Prob. 0.1 and with KnB table containing all entries of 0. Thus, the data with Car Prob. equal to 0.9 starts with the KnB accumulated over the other four less dense traffic environments. When KnB Transf. is set to “backward”, the KnB table from a denser traffic environment carry over to the one with immediately less dense traffic. In this case, the simulations start in the environment with Car Prob. 0.9 with KnB table containing all entries of 0. Thus, in the “backward” case, the KnB table in the environment with Car Prob. equal to 0.1 is accumulated from the other four traffic environments with higher car density. This transfer of KnB in either direction happens after building the KnB by a single population of creatures within a

given environment (Framework (I)). See the left side of Figure 1 for an illustration of the difference between the KnB Transf. “none” and the KnB Transf. “forward” cases.

Under a different scenario (Framework (II)), the transferring of the KnB in either direction happens after building the KnB by several populations of creatures within a given car traffic density environment. Thus in Framework (II), the KnB tables are always transferred from the current repetition to the next repetition within any particular configuration of the parameters’ values. In this framework, even when KnB Transf. is “none”, the KnB is still transferred at the end of a simulation from one repetition to the next one within each environment with the same car traffic density. In Framework (II), when KnB Transf. is “forward” (“backward”), the KnB is additionally transferred at the end of all the repetitions in the environment with lower (higher) Car Prob. to the one with an immediately higher (immediately lower) Car Prob. The distinction between Framework (I) and Framework (II) is important as the amount of learning under each Framework is different and affects the creatures’ success in crossing the highway. In Framework (II), the KnB tables become much more developed as there is much more transferring of the KnB tables occurring. See Figure 1 for an illustration of the differences between the two frameworks.

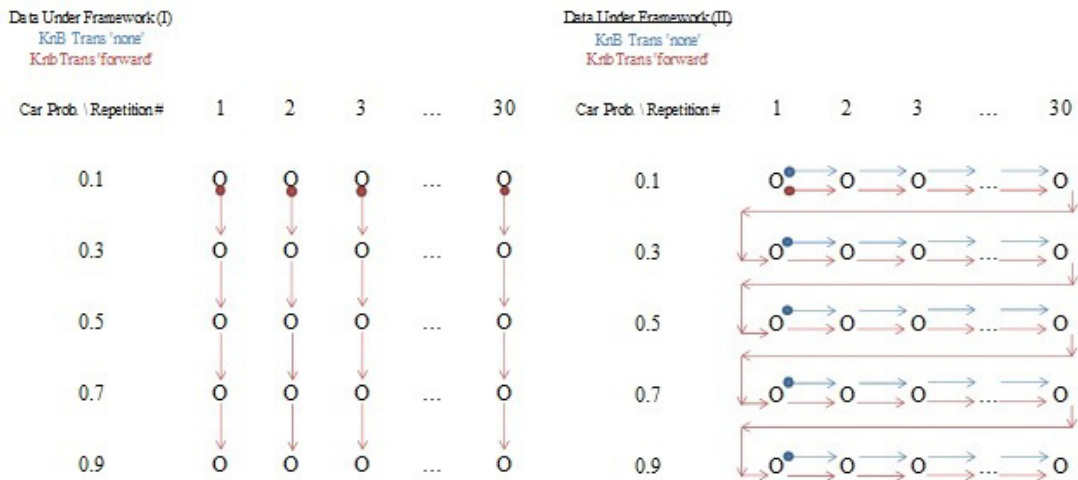


Figure 1: Illustration of the differences between the two frameworks of the data, i.e. Framework (I) and Framework (II), when KnB Transf. is either “none” or “forward”. The left side illustrates the data under Framework (I), while the right side illustrates the data under Framework (II). The unfilled circles are a representation of a simulation specified by its location determining the value of Car Prob. and the repetition number. The blue and red filled circles mark simulations that start with a blank knowledge base table. The blue and red arrows display the flow in the transfer of the knowledge base. Blue items represent the scenario when KnB Transf. is “none” while red items represent the scenario when KnB Transf. is “forward”.

The Horiz. Cre. varies from 0 and 1 and determines whether or not the creatures are allowed to move horizontally in either traffic direction to a different crossing point, if they decide to not cross the highway. The creatures are only allowed to move to a different crossing point if Horiz. Cre. equals 1.

Every combination of the 6 parameters/factors is considered with 30 repeats each. The “summary” files of each configuration are collected into one large dataset containing 67,500 rows. The columns include the value for each configuration parameter and also the main response variables: the number of successful creatures, the number of killed creatures, and the number of queued creatures at the end of the simulation. The dataset is created using R code.

4 Effects of KnB Transfer on Creatures' Success Rate

We investigate the cumulative number of successful creatures at every time step for both of the frameworks. Here, we examine only the configurations with crossing point 60, fear value 0.5, and desire value 0.5. The focus of this analysis is on observing the rate of the creatures' success in crossing the highway throughout time among the different configurations examined: whether or not it is constant, or perhaps increasing due to the learning/update of the KnB table after every time step.

The number of time steps for each run of the simulation is 1511 time steps. The response recorded at each time step is the number of successful creatures up to the current time step. Taking the difference of this response for adjacent time steps provides the number of successful creatures after each time step. The sums of these values are calculated for each group of 100 consecutive time steps so that the time is divided into 15 sections of time. Essentially, we computed the number of successful creatures at each section of time to examine. We ignore the data values of the last 11 time steps for each run.

This new dataset is visualized using line plots with the time sections on the x-axis and the number of successful creatures on the y-axis. The line plots are organized in a table/matrix such that the rows indicate the value of Car Prob. and the columns indicate the value of KnB Transf. that the data is filtered by. There are four groups of plots, each being a combination of Horiz. Cre. and framework. For each individual line plot, the data for all 30 repetitions are all shown by light green lines while the average of these lines at each time section is indicated by a darker green line. The dashed horizontal line is the grand average number of successful creatures.

Figure 2 and Figure 3 display the line plots for the data under Framework (I) while Figure 4 and Figure 5 display the line plots for the data under Framework (II). Figure 2 shows that the creatures are no longer successful in crossing the highway after the initial relatively small number of time intervals when Horiz. Cre. Equals 0 under Framework (I). Their success rate drops almost to zero after this initial period, regardless if KnB transfer takes place or not in most cases. However, Figure 4 shows that, when Horiz. Cre. Equals 0 under Framework (II) and when the KnB is more developed through transferring to different traffic environments in both cases, i.e. the "forward" and "backward" direction, this trend disappears and the creatures are consistently successful with crossing the highway at 50 per time section, i.e. per 100 time steps (see column two and three of Figure 4). Since a creature is generated at each time step and each creature takes 2 time steps to cross the highway, the maximum number of creatures are crossing the highway per time section, i.e. the creatures never need to wait for a car and always cross the highway successfully in those scenarios with 50 successful crossings every time section. This consistency does not hold as often when KnB transfer does not take place, see first column of Figure 4. We see that there are still some results where the success rate drops after some short initial period.

When under Framework (I) and when the creatures are allowed to move to another crossing point (i.e., when Horiz. Cre. 1), the creatures do not suddenly stop becoming successful, see Figure 3, as in the case when they are not allowed to change their crossing point (i.e., when Horiz. Cre. 0), see Figure 2. The rates of success are higher as more creatures can cross at once. It is likely the case that, in the scenarios of low success where creatures are not allowed to move to a different crossing point, a creature exhibiting fear is jamming the queue. This mechanism most likely is also responsible for the drop in the creatures' success rate under the Framework (II) when KnB "none", see Figure 4. However, both transfers of KnB, "forward" and "backward", mitigate this effect in the Framework (II), as can be seen by comparing, respectively, second and third columns of Figure 4 with those of Figure 2. Under Framework (I) when Horiz. Cre. is 1, KnB Transf. is "none", and Car Prob. is 0.9, there is an outlier repetition that has a higher number of successful creatures, achieving success rates similar to what is achieved with a forward transferred KnB from the different consecutive traffic environments.

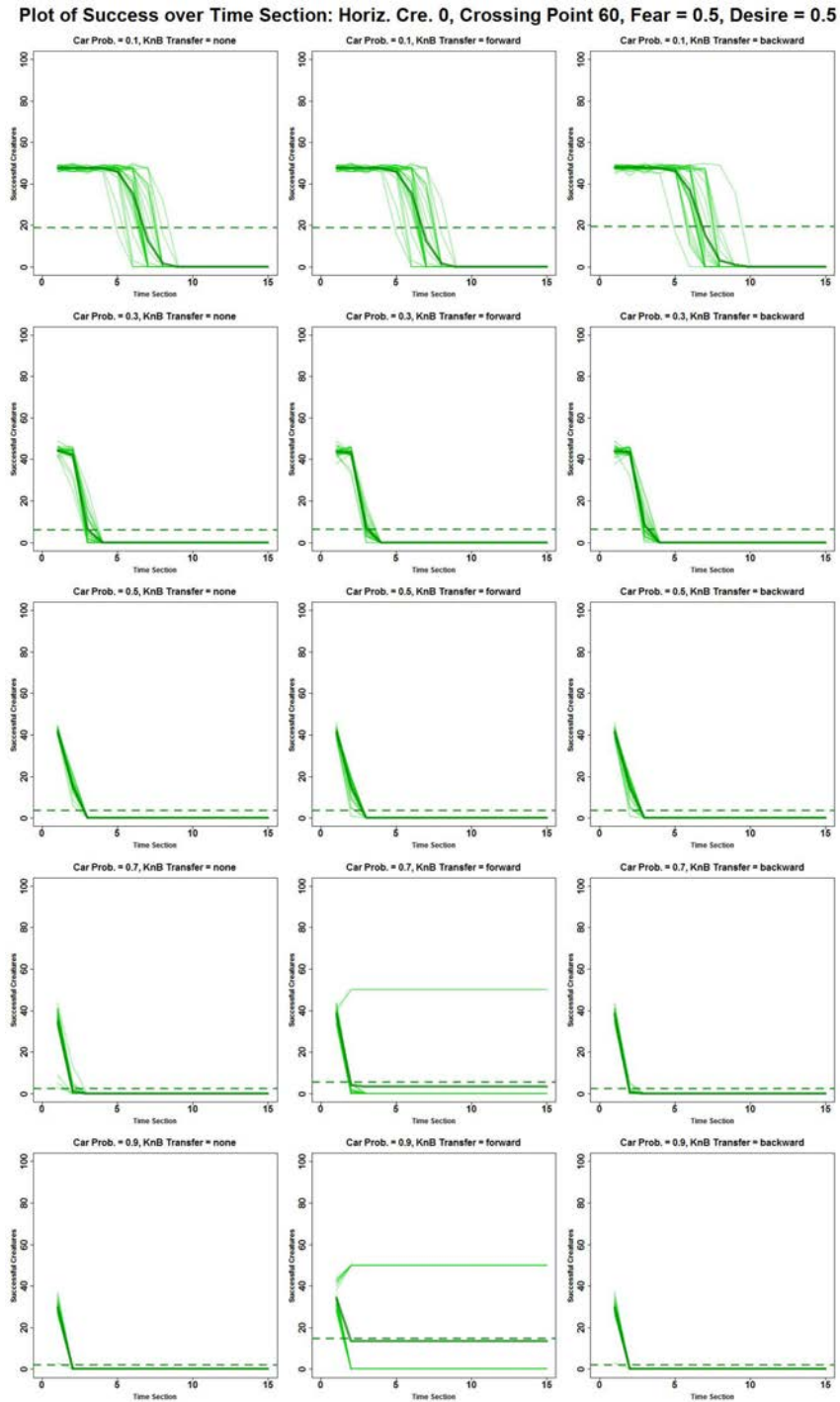


Figure 2: Line plots for data under Framework (I) with Horiz. Cre. equal to 0, CP equal to 60, fear value equal to 0.5, desire value equal to 0.5 organized by Car Prob. and KnB Transf., the values of which are indicated by the rows and columns, respectively.

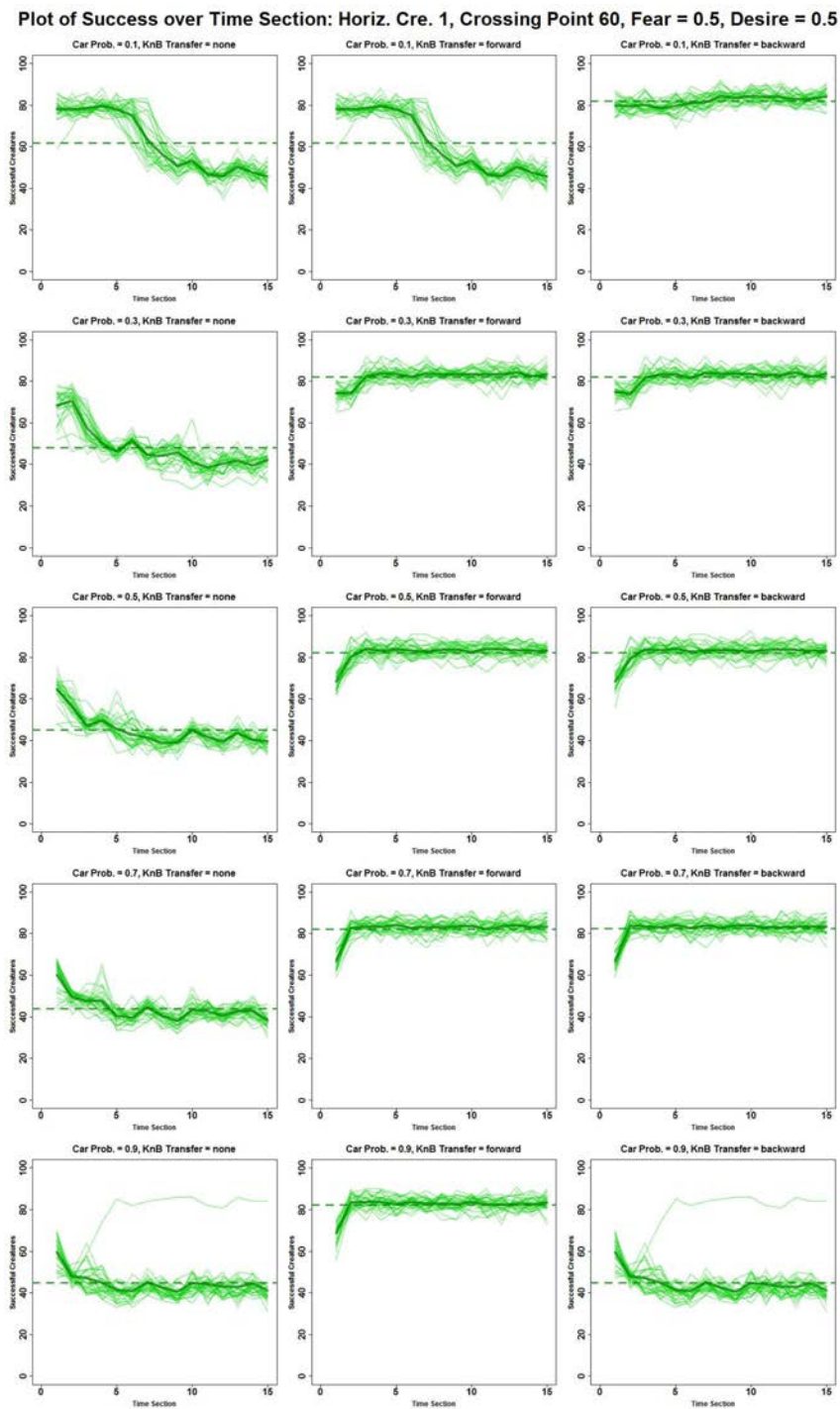


Figure 3: Line plots for data under Framework (I) with Horiz. Cre. equal to 1, CP equal to 60, fear value equal to 0.5, desire value equal to 0.5 organized by Car Prob. and KnB Transf., the values of which are indicated by the rows and columns, respectively.

Plot of Success over Time Section: Horiz. Cre. 0, Crossing Point 60, Fear = 0.5, Desire = 0.5

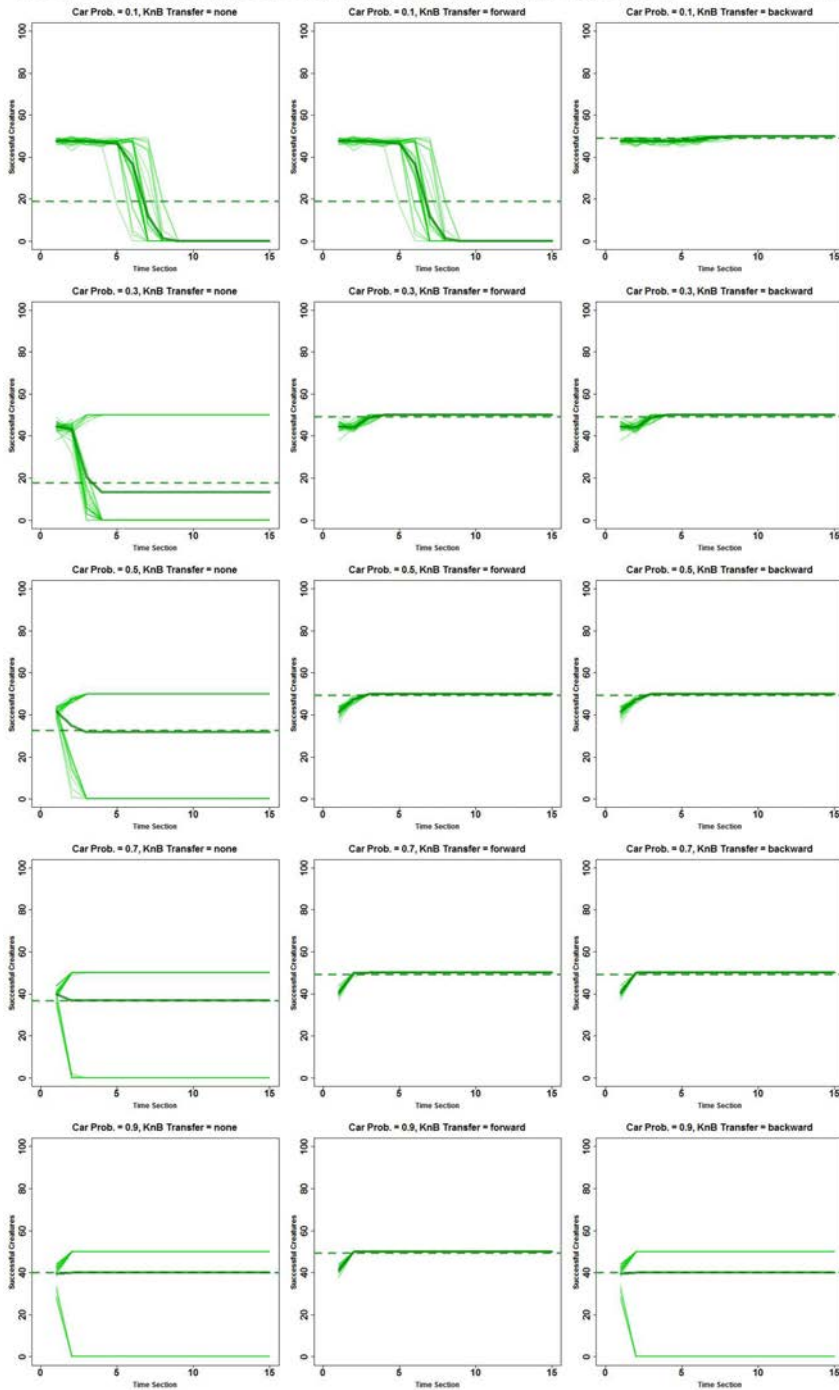


Figure 4: Line plots for data under Framework (II) with Horiz. Cre. equal to 0, CP equal to 60, fear value equal to 0.5, desire value equal to 0.5 organized by Car Prob. and KnB Transf., the values of which are indicated by the rows and columns, respectively.

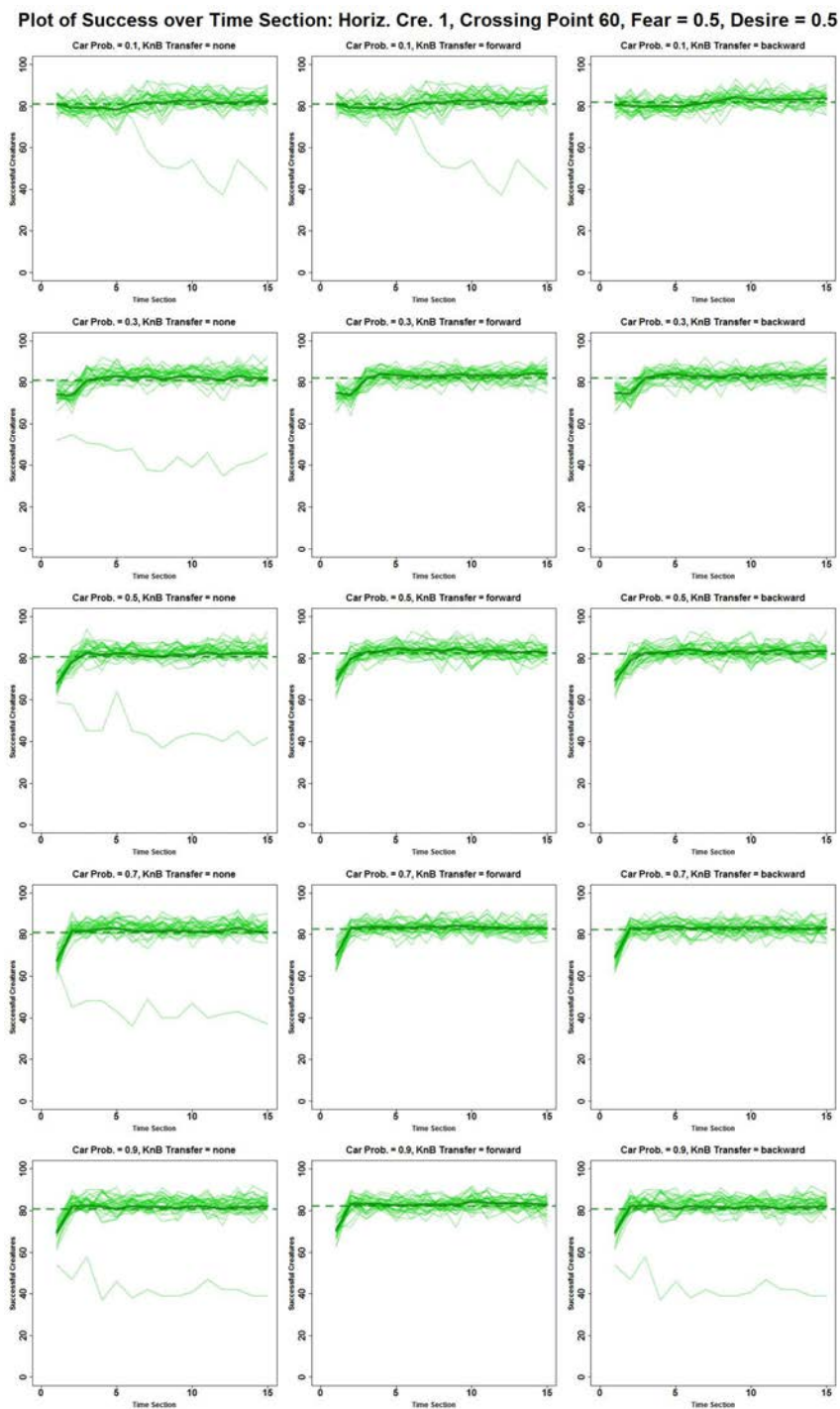


Figure 5: Line plots for data under Framework (II) with Horiz. Cre. equal to 1, CP equal to 60, fear value equal to 0.5, desire value equal to 0.5 organized by Car Prob. and KnB Transf., the values of which are indicated by the rows and columns, respectively.

The line plots for the data under Framework (II) show that the extra learning/transferring of the KnB after each repetition is allowing the simulations to achieve a higher number of successful creatures more often. This is revealed through comparing respectively, Figure 4 and Figure 5 to Figure 2 and Figure 3, and it shows that the additional training of creatures' populations within the same environment produces better results in all cases of KnB transfer scenarios. These observations are consistent with results of previous work [19] and [20].

5 Conclusions and Future Work

The presented case study shows that a more developed KnB improves creatures' success rate of crossing the highway and allowing the creatures to move to other crossing points intuitively increases the number of successful creatures. However, this creates more variability in most cases. The coupling of creatures' movement with more KnB accumulation, as the result of the KnB being passed from one environment to another one, is particularly beneficial on creatures' success rate under Framework (I). This benefit is not as noticeable under the Framework (II), as the creatures' KnB base is already much more developed through training and learning within the same environment. A limitation of this case study is the narrow scope of the analysed data, because we only considered fear and desire values of 0.5, and one selected initial crossing point. We plan to continue our analyses by exploring the effects of other model parameters and different decision making formulas on creatures' success rate of crossing the highway.

References

- [1] Ch.L. Nehavin, K. Dautenhahn, "*Imitation and Social Learning in Robots, Humans and Animals*", Cambridge University Press, Cambridge, UK, 2007.
- [2] Y. Tan, Z.-y. Zheng, "Research Advance in Swarm Robotics", *Elsevier Defence Technology*, Vol. 9, pp. 18-39, 2013.
- [3] I. Navarro, F. Matia, "An Introduction to Swarm Robotics", Hindawi Publishing Corporation, *ISRN Robotics*, Vol. 13, Article ID 608164, 10 pages, 2013.
- [4] S. Russell, P. Norvig, "*Artificial Intelligence, A Modern Approach*", Pearson Education Limited, 2014.
- [5] D.L. Poole, A.K. MacKworth, "*Artificial Intelligence, Foundations of Computational Agents*", Cambridge University Press, Cambridge, US, 2010.
- [6] J. Ferber, Multi-Agent Systems, "*An Introduction to Distributed Artificial Intelligence*", Addison-Wesley, Longman, 1999.
- [7] A.M. Uhrmacher, D. Weyns, "*Multi-Agent Systems Simulation and Applications*", CRC Press, Taylor & Francis Group, Boca Raton, Fl., 2009.
- [8] A. Bandura, "*Social Learning Theory*", Prentice Hall, Englewood Cliffs, NJ, 1977.
- [9] A. Bandura, D. Ross, S.A. Ross, "Transmission of aggression through the imitation of aggressive models", *Journal of Abnormal and Social Psychology*, 63, 1961, pp. 575-582.
- [10] W. Hoppitt, K.N. Laland, "*Social Learning, An Introduction to Mechanisms, Methods, and Models*", Princeton University Press, Princeton, 2013.
- [11] M.A. Montes de Oca, T. Stutzle, "Towards incremental social learning in optimization and multiagent systems", *Proceedings of GECCO'08, The 10th Annual Conference Comparison on Genetic and Evolutionary Computation*, pp. 1939-1944, 2008.
- [12] Y. J. Gong, J. Zhang, Y. Li, "From the social learning theory to a social learning algorithm for global optimization", *Proceedings of 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, San Diego, CA, 05-08 Oct. 2014, pp. 222 – 227.
- [13] R. Cheng, Y. Jin, "A social learning particle swarm optimization algorithm for scalable optimization", *Information Sciences*, Vol. 291, 2015, pp. 43-60.
- [14] Z.-Z. Liu, D.-H. Chu, Ch. Song, X. Xue, B.-Y. Lu, "Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition", *Information Sciences*, Vol. 326, 2016, pp. 315-333.
- [15] B.N. Di Stefano, A.T. Lawniczak, "Modeling a Simple Adaptive Cognitive Agent", *Acta Physica Polonica, B Proceedings Supplement*, Vol. 5 (1), 2012, pp. 21-29.

- [16] A.T. Lawniczak, J.B. Ernst, B.N. Di Stefano, "Creature Learning To Cross A CA Simulated Road", in: G.C. Sirakoulis, S. Bandini (Eds.) Proc. of ACRI 2012, Springer-Verlag *LNCS* Vol.7495, 2012, pp. 425-433, 2012.
- [17] A.T. Lawniczak, J.B. Ernst, B.N. Di Stefano, "Simulated Naïve Creature Crossing a Highway", *Procedia Computer Science* 18, 2013, pp. 2611-2614.
- [18] A.T. Lawniczak, B.N. Di Stefano, J.B. Ernst, "Software Implementation of Population of Cognitive Agents Learning to Cross a Highway", J. Was, G.C. Sirakoulis, S. Bandini (Eds.) Proc. of ACRI 2014, Springer-Verlag *LNCS* Vol. 8751, 2014, pp. 688-697.
- [19] A.T. Lawniczak, J.B. Ernst, B.N. Di Stefano, "Biomimicry based Learning Outcomes of Simple Cognitive Agents", *Procedia Computer Science* 36, 2014, pp. 508-514.
- [20] A.T. Lawniczak, B.N. Di Stefano, J.B. Ernst, Naïve Creature Learns to Cross a Highway in a Simulated CA-Like Environment", 2014 IEEE Symposium on Intelligent Agents (IA), Orlando, FL, 09-12 Dec. 2014, pp. 30-37.
- [21] K. Nagel, M. Schreckenberg, "A cellular automaton model for freeway traffic", *J. Physique I*, 2, 1992, pp. 2221 – 2229.
- [22] A.T. Lawniczak, B.N. Di Stefano, "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
- [23] A.T. Lawniczak, B.N. Di Stefano, "Multilane Single GCA-w Based Expressway Traffic Model", in: S. Bandini et al. (Eds.): Proc. of ACRI 2010, Springer-Verlag *LNCS* Vol. 6350, 2010, pp. 600-612.
- [24] A.T. Lawniczak, B.N. Di Stefano, "Digital Laboratory of Agent-Based Highway Traffic Model", *Acta Physica Polonica B Proc. Supplement*, Vol. 3, No. 2, February 2010, pp. 479-493.