Comparison of Agents' Performance in Learning to Cross a Highway for Two Decisions Formulas

Anna T. Lawniczak and Fei Yu
Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario N1G 2W1, Canada

Keywords: Agents, Cognitive Agents, Autonomous Robots, Cellular Automaton, Decision Making, Learning, Knowledge Base, Computational Intelligence.

Abstract: We compare the performance of simple cognitive agents, learning to cross a Cellular Automaton (CA) based highway, for two decision formulas used by the agents' in their decision-making process. We describe the main features of the simulation model: CA based highway traffic environment, agents and their decision and learning mechanisms. The agents use a type of "observational social learning" strategy, i.e. they observe the performance of other agents and they try to mimic what worked for other agents and they try to avoid what did not work for the other agents. In the decision-making process of deciding whether to cross the highway or to wait, depending on the simulation setup, the agents use one of the two decisions formulas: the first one based only on the assessment of the agents crossing decisions (cDF), or the second one based on the assessment of the agents crossing and waiting decisions (cwDF). Our simulations show that the performance of agents using cwDF is much better than the performance of the agents using cDF in their decision making process. We measure the agents' performance by the numbers of agents: who crossed successfully, who were killed and those who are still queuing to cross at simulation end.

1 INTRODUCTION

In recent years, we have witnessed the rapid development of autonomous robots of various levels of complexity and scale, from Google driverless cars and drones to swarm robots, microrobots or kilobots. Each robot is a complex dynamical systems performance of which very often depends on many parameters. In some cases, robots must learn to adapt to dynamically changing conditions of environment in which they operate, e.g. Google driverless car. Thus, it is important to study how robots' learning performance and their outcomes may be affected by various parameters. Such investigations could be carried out through modelling and simulation, where autonomous robots are identified with autonomous cognitive agents, which are abstractions of autonomous entities capable of interacting with each other and their environment (Russell and Norvig, 2014; Poole and MacKworth, 2010; Ferber, 1999; Uhrmacher and Weyns, 2009).

In this work, we investigate the performance of a simple learning algorithm based on an *observational social learning* mechanism (Nehavin and Dautenhahn, 2007; Bandura 1977; Bandura et al., (1961; Hoppitt and Laland, 2013), in which each cognitive agent learns from observing the outcomes of 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 (Montes de Oca and Stutzle, 2008; Gong et al. 2014; Cheng and Jin, 2015; Liu et al., 2016), among others for finding more effective optimizers than swarm intelligence algorithms.

In the presented work, we consider the model of cognitive agents learning to cross a Cellular Automaton (CA) based highway introduced and described in (Lawniczak et al., 2012; Lawniczak et al., 2013; Lawniczak et al., 2014). This model is an abstraction of the situation in which an autonomous agent must learn to decide instantaneously if it is safe or not to cross a highway when it encounters an incoming vehicle and when the agent can perceive only *fuzzy* categories of the vehicle's speed and its proximity. In the model the agents are born as *tabula rasa*; i.e. a "blank slate" and they do not have a built-in knowledge base of the environmental conditions under which it is safe or not to cross the highway. The

Lawniczak A. and Yu F.
Comparison of AgentsâĂŹ Performance in Learning to Cross a Highway for Two Decisions Formulas.
DOI: 10.5220/0006193102080219
In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)*, pages 208-219
ISBN: 978-989-758-219-6
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

agents have a built-in template to classify the environmental conditions and they have a reasoning method to use this classification in deciding whether or not to cross the highway. They are capable of evaluating if a strategy of crossing the highway has been applied successfully or not and they are capable of applying this strategy again with small changes to a similar but new situation. Thus, they are capable of the adoption or rejection of the strategy through their *observational social learning* mechanism. The agents build their knowledge base representing the evolution of the model dynamics as the simulation progresses.

In the presented work, we compare agents' performance in learning to cross the highway for two decision-making formulas, the original one used in (Lawniczak et al., 2012; Lawniczak et al., 2013; Lawniczak et al., 2014), which is based only on the outcomes of the agents' crossing decisions, which we call cDF, with the new decision-making formula introduced in this paper. The new decision-making formula considers the assessment of the agents' crossing and waiting decisions and we call it cwDF. Our simulations show that the performance of agents using cwDF is much better than the performance of the agents using cDF in their decision-making process. We measure the agents' performance by the numbers of agents: who crossed successfully, who were killed and those who are still queuing to cross at simulation end.

The paper is organized as follows: Section 2 describes the model, introduces the new decision-making formula and provides mathematical formulation of both decision-making formulas; Section 3 describes simulation setup and resulting data; Section 4 compares the performance of the agents using cDF with the performance when they use cwDF instead; Section 5 reports our conclusions.

2 MODEL DESCRIPTION OF AGENTS LEARNING TO CROSS A CA BASED HIGHWAY

For completeness of the paper, we review the main features of the model introduced in (Lawniczak et al., 2012; Lawniczak et al., 2014). For the software implementation of the model we refer the reader to (Lawniczak and Di Stefano, 2014) for details. The model was developed under several assumptions about the agents that are called "creatures" in papers (Lawniczak et al., 2012; Lawniczak et al., 2014; Lawniczak and Di Stefano, 2014), their process of learning and their environment.

We assume that the environment is a single lane unidirectional highway without any intersection. A creature is an autonomous cognitive agent having a strong instinct to survive. All creatures/agents are initially located on one side of the highway and they want to cross the highway without being struck by the oncoming vehicles to get to the opposite side of the highway.

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 (i.e., class identifiers) to an approximate class; (5) understanding when another agent has been successful in crossing the highway; (6) repeating the action that has previously resulted in success. Each creature is equipped with a simple mechanism to evaluate an outcome of crossing of each creature that crossed previously. Each creature will try to imitate the successful crossings. If unsuccessful crossings outnumber the successful ones, then under similar circumstances the creature may not cross and will wait for better conditions, or will try to find a better location for crossing.

We assume that all creatures, attempting to cross the highway at the same crossing point, except the first one, have witnessed what have happened to the creatures that have previously crossed the highway at this crossing point. This allows each crossing point to build its own knowledge base during the experiment that is available to all creatures at that crossing point.

In what follows we introduce agents' new decision-making formula, called cwDF, and compare the performance of the agents using this decision formula with their performance when they use the decision-making formula of the works (Lawniczak et al., 2012; Lawniczak et al., 2014; Lawniczak and Di Stefano, 2014). Additionally, we provide mathematical description of both formulas.

2.1 Highway Model and Agents

We model the highway traffic by adapting the Nagel-Schreckenberg Cellular Automaton model and refer the reader to (Nagel and Schreckenberg, 1992; Lawniczak and Di Stefano, 2008; Lawniczak and Di Stefano 2010a; Lawniczak and Di Stefano, 2010b) for details. The model consists of four steps that are applied simultaneously to all cars: acceleration, safety distance adjustment, randomization, and change of position. The implementation of the Nagel-Schreckenberg model for this research requires a modification of the Cellular Automata (CA) paradigm 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, (Lawniczak and Di Stefano, 2008; Lawniczak and Di Stefano 2010a; Lawniczak and Di Stefano 2010b).

As customary in the traffic modelling literature, a highway is modelled by large number of adjacent cells, where each cell represents a segment of the highway of 7.5m in length, (Nagel and Schreckenberg, 1992). The cars are generated at the starting cell of the highway independently of each other with *car creation probability* (CCP), *p*, which determines car traffic density. When cars are created, they are assigned a random speed between zero and the maximum allowed speed for cars that is set in the configuration file. As some cars may start faster than others, to avoid potential collisions, a queue is used to hold each newly generated car until it is able to actually move into the highway without colliding with another car. After a car enters the highway, it speeds up until it reaches the allowed maximum velocity or until it encounters another car in front of itself. To simulate *erratic drivers*, the model allows a random deceleration of cars; i.e. it allows decreasing by one, randomly with probability 0.5, the speed of each car.

Agents/creatures are generated in a similar way as the cars. They are generated only at the crossing points set at the initialization step, and at these crossing points, they are generated with the same *creature creation probability*. In the presented simulation results we consider only one Crossing Point (CP), selected at cell 60 at the initialization setup, i.e. we consider the CP located 450m away from the beginning of the highway. The location of this CP was selected sufficiently far away from the beginning of the highway to allow emergence of the car traffic profiles for the considered CCP values. These profiles may not exit at the beginning of the highway due to the potential cars build up in the queue when they are entering the highway.

We assume that *creature creation probability* is 1, i.e. at each time step a creature is generated at the crossing point. As creatures are generated, they are placed into the queue at the crossing point. When a creature is generated two attributes, one of *Fear* and another one of *Desire*, are assigned independently to each creature, each with probability 0.5. Thus, there are actually four types of creatures being generated each with equal probability of 0.25, a creature with (1) Fear & Desire; or (2) no Fear & Desire; or (3) Fear & no Desire; or (4) no Fear & no Desire. The attributes of Fear and Desire may be interpreted as a pair of parameters describing each agent propensity to risk taking (Desire) and its aversion to risk taking (Fear). The values of these parameters are set in the configuration file. We assume that these values are between 0.00 and 1.00.

All the agents have the same goal of trying to cross the highway successfully, i.e. without being hit by a vehicle. If an agent is hit, it is killed and it will be removed from the simulation immediately. In the described model with the single lane highway, each creature crosses the highway in two time steps. The creature takes one-time step to move onto the highway and it takes the next time step to move out of the highway onto the other side of the highway.

Agents attempt to cross the highway having a limited information about the environment around them. They have a limited horizon of vision and they can perceive only *fuzzy* levels of speed (e.g., *slow*, *medium*, *fast*, *very fast*) and of distance (e.g., *close*, *medium*, *far*) of cars within this horizon. The distances and speeds that each creature is able to perceive are set in the configuration file. If a creature at some instance of time does not cross the highway, because it has become *afraid*, creatures will build up in the queue until the creature at the top of the queue decides to finally cross, or move to a different location to attempt to cross from. If the simulation setup permits, a creature may move randomly up or down along the car traffic stream, i.e. right or left along the highway, in each case with probability 1/3, to search for a new crossing point to cross from, or it may stay at the crossing point with probability 1/3. If a creature at the top of a queue moves up or down along the car traffic stream to a new location, the creature that was behind it moves to the top of the queue.

2.2 Agents' Knowledge Base

We call an agent at the top of its queue an *active creature*. Each active creature must decide whether it is safe to cross the highway or it is not safe to do so. In this case the active creature must decide whether to wait at its crossing point for better traffic conditions or to move to another crossing point, if the simulation setup permits. Thus, each active creature must make one of the following two decisions: Crossing Decision (CD) or Waiting Decision (WD). If the active creature decides to cross, it may either succeed or it may be hit. Thus, if the active creature's decision results in successful crossing, we call such decision Correct Crossing Decision (CCD), if the crossing decision results in hitting/killing the creature, we call such decision Incorrect Crossing Decision (ICD). Similarly, each waiting decision of the active creature can be assessed as: Correct Waiting Decision (CWD), or Incorrect Waiting Decision (IWD). The active creature makes CWD in the case when, if it did not wait and chose to cross, it would had been hit. The active creature makes IWD in the case when, it chose to wait but it would had crossed the highway successfully. The assessment of each active creature decision, i.e. if the decision was CCD, ICD, CWD, or IWD, is recorded as a count into the Knowledge-Based (KB) table of all the creatures waiting at the crossing point of the active creature.

The columns of the Knowledge-Based table, organized as a matrix with extra entry, store information about qualitative descriptions of velocity (e.g., such as *slow*, *medium*, *fast* and *very fast*) and the rows of the KB table store information about qualitative descriptions of the distance (e.g., such as *close*, *medium*, *far* and *out of range*). The numerical values corresponding to the qualitative descriptions of distance and velocity that the creatures may perceive are set in the configuration file. Since the creatures have limited horizon of vision, the last row of the KB table corresponds to creatures' *out of range* vision, i.e. the situation in which an active creature cannot perceive if outside its horizon of vision there is a vehicle and if it is, what is its velocity. Thus, in the last row of KB table the cells corresponding to potentially perceived velocities are all merge together. Because of this we call this row the *extra entry* of the matrix associated with the KB table.

For each time t each entry (including the *extra entry*) of the KB table contains the number of CCDs, ICDs, CWDs and IWDs made by the active creatures up to time t-1. For example, if the active creature successfully crossed the highway at time t, i.e. it made the CCD for the perceived (distance, velocity) pair at time t, then the score of CCDs for this (distance, velocity) pair recorded up to time t-1 is increased by one point in the Knowledge-Based table. If the creature was struck/killed, then the score of ICDs for this (distance, velocity) pair recorded up to time t-1 is increased by one point in the Knowledge-Based table.

The Knowledge-Based table is initialized as *tabula rasa*; i.e. a "blank slate", represented by "(0,0,0,0)" at each table entry for the assumption that the creatures can cross for all possible (distance, velocity) combinations. At the start of each simulation, creatures cross the highway regardless of the observed (distance, velocity) combinations until the first successful crossing of a creature, or five (selected for the presented simulation results) consecutive unsuccessful crossings of the creatures, whichever comes first.

After the initialization of the simulation, when a new creature arrives at the top of the queue, the creature consults the Knowledge-Based 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 parameters' values.

2.3 Agents' Decision-Making Algorithm

This section describes the creatures two types of decision formulas, which an active creature may use in its decision-making process/algorithm, when it is deciding whether to cross the highway or to wait. The first decision formula is used in the works (Lawniczak et al., 2012; Lawniczak at al. 2013; Lawniczak et al., 2014). This formula considers only the outcomes of creatures crossing decisions, i.e. the number of successful creatures and the number of killed creatures for each (distance, velocity) pair at time t. Since the number of successful creatures is equal to the number of correct crossing decisions, and the number of killed creatures is equal the number of incorrect crossing decisions, we call this formula Crossing Based Decision Formula (cDF) and provide its mathematical formulation in this paper.

The works (Lawniczak et al., 2012; Lawniczak at al. 2013; Lawniczak et al., 2014; Lawniczak et al., 2015; Lawniczak, Di Stefano et al. 2016; Lawniczak, Ly et al., 2016) show that each population of generated creatures at each simulation end is divided into three types of creatures: the successful ones, the killed ones, and the creatures still queuing to cross the highway, with over all very small numbers of killed creatures. Thus, at each simulation end there are mostly successful and queued creatures, and for some values of the model parameters the queued creatures outnumber significantly the successful ones. The works (Lawniczak et al., 2012; Lawniczak at al. 2013; Lawniczak et al., 2014; Lawniczak et al., 2015; Lawniczak, Di Stefano et al. 2016; Lawniczak, Ly et al., 2016) focus on demonstrating that the creatures' performance could be improved (i.e., the numbers of successful creatures could be increased) by passing, at the end of a simulation run, the knowledge base built by a generation of creatures to the next one within the same highway traffic environment, and/or by passing the knowledge base built in one traffic environment to the creatures in another traffic environment, i.e. the creatures would not start *tabula rasa* their process of learning to cross the highway but they would start with some pre-existing knowledge built during previous simulations, except for the first generation of creatures.

In this paper, we investigate if the improvement in creatures' performance could be also achieved by incorporating the assessment of creatures' waiting decisions into their decision-making formula, which they use to decide whether to cross the highway or to wait. In what follow we introduce a new decision-making formula, called Crossing-and-Waiting Based Decision Formula (cwDF), provide its mathematical description and compare creatures' performance when they use cwDF with their performance when they use cDF instead.

2.3.1 Crossing based Decision Formula (cDF)

After the initialization phase, at each time step t, each active creature (i.e., the one at the top of its queue), when deciding whether to cross or to wait carries several tasks, namely: (1) it determines if there is a car in its horizon of vision. If it is, then it determines the (i^{th} distance, j^{th} velocity) pair of qualitative values of the current closest car; (2) it consults the KB table associated with its crossing point to get information about the number of CCD(t-1) and the number of ICD(t-1) for the observed (i^{th} distance, j^{th} velocity) pair of qualitative values, or for the observed out of range situation, which is denoted by (0,0) pair of indexes; (3) for the observed values it calculates the value of the cDF, i.e. the value $cDF_{ij}(t)$, corresponding to the observed (i^{th} distance, j^{th} velocity) pair of qualitative values, or for the observed out of range situation (0,0). The expression $DF_{ij}(t)$ is defined below and from now on we assume that a pair of (i,j) indexes may take also the value (0,0) reserved for the extra entry in the KB table.

The active creature decides to cross or to wait based on the outcome of its calculation of cDF respective value. If $cDF_{ij}(t) \geq 0$, then the active creature will cross. If $cDF_{ij}(t) < 0$, then the active creature will wait and additionally it may move to another crossing point, if simulation setup permits.

If the active creature observed (i,j) situation, which could be the i^{th} distance type and the j^{th} velocity type, or out of range situation (0,0), then the $cDF_{ij}(t)$ for the (i,j) entry of KB table is calculated as follows:

$$cDF_{ij}(t) = cSR_{ij}(t) + v(Desire) - v(Fear), \quad (1)$$

where $v(Desire)$ and $v(Fear)$ are the values of the active creature *Fear* and *Desire* attributes/parameters, and $cSR_{ij}(t)$ is the Crossing Based Success Ratio (cSR) corresponding to the ij^{th} entry of the KB table, including the out of range entry (0,0). The $cSR_{ij}(t)$ is defined as follows:

$$cSR_{ij}(t) = \{CCD_{ij}(t-1) - ICD_{ij}(t-1)\}/CCD_{total}(t-1). \quad (2)$$

The terms $CCD_{ij}(t-1)$ and $ICD_{ij}(t-1)$ are, respectively, the numbers of CCDs and ICDs recorded in the ij^{th} entry of the KB table at time t. The term $CCD_{total}(t-1)$ is the sum of CCDs over all the entries of the KB table, i.e. it is the total number of all CCDs made by active creatures up to time $t-1$, which corresponds to the total number of successful creatures up to time $t-1$. Recall, that the KB table at each time t stores information about assessment of various decisions made by creatures up to time step $t-1$. Since the decision formula (2) considers only the assessment of Crossing Decisions, i.e. it considers only the CCDs and ICDs, we call this decision formula Crossing Based Decision Formula (cDF) to distinguish it from the decision formula cwDF introduced in this paper that is based additionally on the assessment of the active creatures waiting decisions. Recall that each CCD corresponds to creature being successful and each ICD corresponds to creature being killed. Through this identification we recognize that the cDF formula defined in (1) has been used in the works (Lawniczak et al., 2012; Lawniczak at al. 2013; Lawniczak et al., 2014; Lawniczak et al., 2015; Lawniczak, Di Stefano et al. 2016; Lawniczak, Ly et al., 2016).

Recall that each creature can be classified per its $Desire$ and $Fear$ attributes/parameters. Depending on these attributes we can express cDF explicitly for each creature type as Table 1 shows. Thus, an active creature, depending what is its type, decides to cross only when the outcome of the corresponding cDF_{ij} is greater or equal to 0. Otherwise, the active creature will wait, and additionally it may move to another crossing point, if the simulation setup allows this.

Table 1: Expression of cDF depending on an active creature attributes of *Fear* and *Desire* and their values.

Type of Creature	Decision Formula (cDF_{ij})
no Desire & no Fear	cSR_{ij}
no Desire & Fear	$cSR_{ij} - v(Fear)$
Desire & no Fear	$cSR_{ij} + v(Desire)$
Desire & Fear	$cSR_{ij} + v(Desire) - v(Fear)$

2.3.2 Crossing-and-Waiting based Decision Formula (cwDF)

The Crossing-and-Waiting Based Decision Formula (cwDF) introduced here incorporates not only the assessment of the crossing decisions of the active creatures, but also the assessment of their waiting decisions. The formula cwDF is obtained from cDF formula by replacing the term $cSR_{ij}(t)$ by the term $cwSR_{ij}(t)$ in the cDF formula, i.e. by replacing the term (2) by the term $cwSR_{ij}(t)$ in the formula (1). The term $cwSR_{ij}(t)$, called Crossing-and-Waiting Based Success Ratio (cwSR), is defined for each ij entry of the KB table at time t as follows:

$$cwSR_{ij}(t) = \{CCD_{ij}(t-1) - ICD_{ij}(t-1) - CWD_{ij}(t-1) + IWD_{ij}(t-1)\}/S(t-1), \tag{3}$$

where $CCD_{ij}(t-1)$ is the number of CCDs, $ICD_{ij}(t-1)$ is the number of ICDs, $CWD_{ij}(t-1)$ is the number of CWDs and $IWD_{ij}(t-1)$ is the number of IWDs in the KB table entry ij, where each of these numbers is being recorded up to time $t-1$. The term $S(t-1)$ is the sum of the numbers of CDs and WDs, regardless of their assessments, recorded in all the entries of the KB table up to time $t-1$, i.e.

$$S(t-1) = \sum_{ij}\{CCD_{ij}(t-1) + ICD_{ij}(t-1) + + CWD_{ij}(t-1) + IWD_{ij}(t-1)\}. \tag{4}$$

Thus, the new formula cwDF can be written as follows

$$cwDF_{ij}(t) = cwSR_{ij}(t) + v(Desire) + -v(Fear), \tag{5}$$

where the term $cwSR_{ij}(t)$ is defined in (3), and as before $v(Desire)$ and $v(Fear)$ are the values of an active creature *Desire* and *Fear* attributes/parameters.

An active creature decides to cross the highway only when the outcome of $cwDF_{ij}$ is greater or equal to 0. Otherwise, the active creature will wait and additionally it may move to another crossing point, if the simulation setup allows this. Recall that the *Desire* and *Fear* attributes are distributed uniformly and independently with probability 0.5 each among all the generated creatures. Thus, each active creature makes its decision to cross the highway or to wait based on both, the Crossing-and-Waiting Based Success Ratio formula $cwSR_{ij}(t)$ and its own values of *Desire* and *Fear* attribute/parameters as shown in Table 2.

Table 2: Expression of cwDF depending on an active creature attributes of *Fear* and *Desire* and their values.

Type of Creature	Decision Formula ($cwDF_{ij}$)
No Desire & No Fear	$cwSR_{ij}$
No Desire & Fear	$cwSR_{ij} - v(Fear)$
Desire & No Fear	$cwSR_{ij} + v(Desire)$
Desire & Fear	$cwSR_{ij} + v(Desire) - v(Fear)$

At each time t, an active creature decision-making process incorporates the assessments of all previous active creatures' decisions, through the cwSR part in cwDF formula (5), in such a way, that it encourages the active creature to cross the highway, if it is safe to do so, and it encourages it to wait, if it is not safe to cross. Additionally, to this self-feedback mechanism, i.e. incorporating the results of the assessment of crossing decisions and which was also considered in cDF formula (1), the cwDF formula (5) incorporates the self-feedback mechanism of the assessment of the waiting decisions of active creatures, i.e. it encourages each active creature to wait if it is not safe to cross and it discourages the creature to wait if it is safe to cross.

Our simulations show that the incorporation of these two self-feedback mechanisms into creatures' decision-making process contributes to the creatures' better performance, i.e. more creatures cross the highway successfully, the numbers of kills creatures stay almost the same and the numbers of queued creatures are smaller at the simulation ends. Thus, when the creatures use cwDF instead of cDF in their decision-making process some queued creatures are converted into the successful ones during simulation runs with almost no change in the numbers of killed creatures.

2.4 Model Simulation Loop

After the program reads in the configuration and knowledge base files described above, it executes the main simulation loop of the model once for every time step in the simulation. The main simulation loop of the model consists of: (1) generating cars at the beginning of the highway using the car creation probability CCP; (2) generating creatures at each crossing point CP with their attributes of *Fear* and *Desire*; (3) updating the car speeds according the Nagel-Schreckenberg model; (4) moving the creatures from their CP queues into the highway (if the decision algorithm indicates this should occur); (5) updating locations of the cars on the highway and checking if any creature has been killed; (6)

advancing the current time step. After the simulation has been completed, the results are written to output files using output functions.

3 MODEL PARAMETERS SETUP AND SIMULATION DATA

This research focuses on the comparison of the performance of creatures using cDF with their performance when they use cwDF instead in their decision-making process. Thus, two types of data sets were generated, one when cDF was used and another one when cwDF was used instead. This was the only difference between these two generated data sets, and both data sets were generated using the same software implementation, the same values of the parameters and the same number of repeats.

We consider the model parameters as factors with various levels in the sense of the experimental design paradigm (Dean and Voss, 1999). The parameters/factors that remain constant through our simulations are: (1) the single lane highway of a length of 120 cells (i.e., a stretch of a highway of the length of 900 meters); (2) the single Crossing Point (CP) set at the initialization step at cell 60; (3) each simulation run of duration of 1511 time steps; (4) 30 repetitions for each simulation set up; (5) at each CP representation of the KB table by 3 by 4 matrix with the extra entry. Each KB table has 3 groupings of distance and 4 groupings of speed. A car is perceived as: *close*, if it is 0 to 5 cells away, *medium far* if it is 6 to 10 cells away, *far* if it is 11 to 15 cells away and *out of range* if it is 16 or more cells away, regardless of the velocity of a car, and this is encoded in the extra entry of the KB table. A car is perceived as: *slow* if its perceived velocity is 0 to 3 cells per time step; *medium speed* if its perceived velocity is 4 or 5 cells per time step, *fast* if its perceived velocity is 6 or 7 cells per time step and *very fast* if its perceived velocity is 8 to 11 cells per time step. A car's max speed can be 11 cells per time step.

There are 6 parameters/factors values which vary through the main simulation loop. These parameters are: (1) *car creation probability*, i.e. CCP; (2) *Fear* parameter; (3) *Desire* parameter; (4) the KB transfer direction parameter KBT; (5) random deceleration RD and (5) horizontal movement HM of an active creature.

The car creation probability, i.e. CCP, determines the density of the cars traffic and it varies between the values: 0.1, 0.3, 0.5, 0.7, and 0.9.

The values of *Fear* and *Desire* parameters both vary between the values: 0.00, 0.25, 0.5, 0.75, and 1.00. Being a part of the decision-making formula, these values influence the creatures' decision-making process of whether to cross or not the highway. The creature's *Fear* may be interpreted as its aversion to risk taking and the creature's *Desire* may be interpreted as its propensity to risk taking.

The KB transfer direction parameter KBT can be set as: "none" (KBT=0), or "forward" (KBT=1). The parameter KBT determines if the KB table of the initial CP can be transferred or not at the end of a simulation run with lower CCP value to the beginning of the simulation run with immediately higher value of CCP. When KBT=0, the KB tables are never transferred from a traffic environment with a lower CCP value to the one with immediately higher CCP value, or any other value. When KBT=1, the KB table is always transferred at the end of a simulation run from a traffic environment with lower CCP value to the beginning of the simulation run in the traffic environment with immediately higher CCP value. In this case, each simulation in the traffic environment with CCP= 0.1 starts with the KB table *tabula rasa*, i.e. with the KB table containing all the entries of (0,0,0,0). The KB table built in this simulation is transferred next to the simulation in the traffic environment with CCP=0.3 at its beginning. This process continues until the simulation in the traffic environment with CCP=0.9 starts. Thus, the simulations with CCP=0.9 start with the KB table accumulated over the other four less dense traffic environments. This process of transferring KB tables is carried out for each simulation repeat.

To simulate *erratic drivers*, the model allows a random deceleration of cars; i.e. if RD=1 it allows decreasing by one the speed of each car, randomly with probability 0.5; if RD=0 this is not allowed.

The horizontal movement (HM) of an active creature takes value 0 or 1. This parameter is used to determine whether the active creatures can move along the highway away from their original crossing point in either direction when they decide not to cross the highway, i.e. if they decide to wait. The creatures are only allowed to move along the highway when HM equals 1. For this paper, we set the number of horizontal cells a creature may move in one-time step to 1 and the maximum distance a creature may deviate from its original crossing point in both directions to 5. When HM equals to 0 the active creatures are not allowed to move, i.e. to change their original crossing point.

Notice, that the parameter HM determines the upper bound on a potential number of successful creatures. According to the model design, it takes 2 time steps for an active creature to cross the highway successfully: in the first time step the creature moves onto the highway from its queue, in the second time step it moves away from the highway when it is not hit by a vehicle, or it is being hit/killed. When HM equals 0, the creatures are not allowed to leave the original crossing point. Thus, at most only one creature can cross the highway per each two time steps. Since a creature is generated at each time step this implies that the maximum number of possible successful creatures at each simulation end can be only half of all the generated creatures, i.e. 1511/2≈755. In other words, when the movement of the creatures is not allowed, i.e. when HM=0, at each simulation end at most 50% of all generated creatures can cross the highway successfully and 50% of all generated creatures are always in the queue waiting to cross. While setting HM=1 allows creatures simultaneously to cross the highway at potentially 11 crossing points, as they are allowed to move 5 cells away from the CP=60 in each direction. Thus, setting HM=1 removes the 50% bound on the maximum number of all possible successful creatures. This observation of the role of the parameter HM is important one for the discussion of the simulation results.

The *full simulation* means the simulation carried out for all the described values of all the discussed parameters.

4 DECISION FORMULAS PERFORMANCE ANALYSIS

In this section, we compare the creatures' performance in learning to cross the highway for cDF in their decision-making process with those when they use cwDF instead. Because of the upper bound, imposed by the model design on the maximal number of potentially successful creatures, which is at most 50% of all generated creatures when HM=0 and the fact that such bound does not exist when HM=1 we split the *full simulation* data into two subsets, one for HM=0 and another one for HM=1. For each of these subsets we analyse the average number of each creature type, i.e., successful, killed, and queued type, at simulation end, and examine how the selection of the decision formula affects these averaged numbers.

4.1 Performance of Creatures using cDF

Average of numbers, respectively, of successful, killed and queued creatures at simulation ends, expressed as percentage, and calculated from the simulation data set with HM=0 is displayed in Figure 1, and calculated from the simulation data set with HM=1 is displayed in Figure 2. In these simulations creatures used cDF in their decision-making process.

When HM=0, Figure 1 shows that on average only 14.80% of all generated creatures cross the highway successfully, which is quite low even when one takes under consideration the imposed upper bound of 50%. Furthermore, the sum of the average of numbers of successful creatures and the average of numbers of killed creatures is only 15.17%. This implies that overall only few creatures tried to cross the highway during the simulations. However, when HM=1, Figure 2 shows that on average 62.36% of all the creatures crossed the highway successfully. Thus, the parameter HM has a significant influence on the performance of the model.

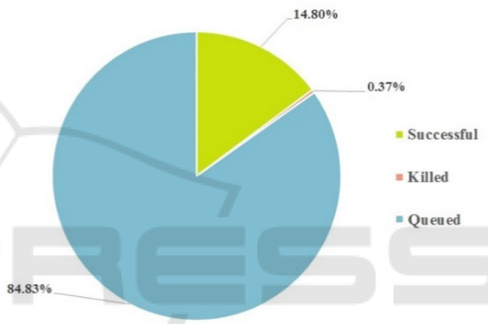

Figure 1: Average of numbers, respectively, of successful, killed and queued creatures at simulation ends, expressed in percentage, and calculated from the subset of simulation data with HM=0, when creatures used cDF.

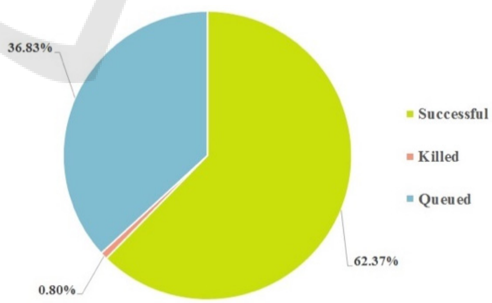

Figure 2: Average of numbers, respectively, of successful, killed and queued creatures at simulation ends, expressed in percentage, and calculated from the subset of simulation data with HM=1, when creatures used cDF.

When HM=1 more creatures simultaneously try to cross a highway, hence more of them may succeed. Though the result of 62.36% is much better than 14.80%, still there are on average 36.83% of queued creatures at simulation end when HM=1. Even though there is an improvement in the model performance when HM=1, the above results show that the creatures' performance is not the best one, regardless whether HM=0 or HM=1.

To better understand the model performance with cDF we split each of these two simulation data sets further into subsets depending on the values of the parameters KBT (Knowledge Base Transfer), RD (Random Deceleration, i.e. presence or absence of erratic drivers). For each of these new subsets characterized by the values of the parameters KBT, RD, HM, we calculate the average of numbers of each type of creatures (i.e., successful, killed and queued, respectively) at simulation ends.

Table 3 displays the average of numbers of each creature type at simulation ends, expressed as percentage, for each subset of the simulation data depending on the values of the model parameters KBT, RD and HM. The values of these parameters are listed in the rows in the first three columns of the table. For each selection of the values of the parameters KBT, RD and HM the average of numbers of successful, killed and queued creatures at simulation ends, expressed as percentage, is listed in the corresponding row in the last three columns of the table, respectively. Observe that for each creature type each entry of the last row of Table 3 is the average of the numbers listed above in the corresponding column. Thus, these averages give on average the percentage of the successful, killed and queued creatures at simulation ends in the *full simulation* data set.

Table 3: Average of numbers, respectively, of successful, killed, queued creatures at simulation ends, expressed in percentage, and calculated from the simulation data sets characterized by different values of the parameters KBT, RD and HM, when the cDF was used in these simulations. The value "0" means "without" and the value "1" means "with" KB transfer (KBT), random deceleration (RD), and creatures movement (HM), respectively.

KBT	RD	HM	Succe-ssful	Killed	Queued
0	0	0	13.22%	0.37%	86.41%
0	0	1	51.18%	0.82%	48.00%
0	1	0	13.81%	0.50%	85.69%
0	1	1	50.39%	1.03%	48.58%
1	0	0	16.80%	0.20%	83.00%
1	0	1	73.98%	0.48%	25.54%
1	1	0	15.36%	0.42%	84.22%
1	1	1	73.94%	0.90%	25.16%
Average			38.59%	0.59%	60.82%

Looking at Table 3 we can draw the following conclusions. The transfer of KB, i.e. when KBT=1 instead of KBT=0, always increases the percentage of successful creatures and decreases the percentage of queued and killed creatures. The magnitudes of these changes vary and they depend on the values of RD and HM parameters. In general, the effect of transfer of KB on the decrease in the percentage of killed creatures is very small. However, the transfer of KB has much bigger effect on the increase of the percentage of successful creatures and the decrease of the percentage of queued creatures. In some cases, this effect is quite noticeable. For example, when RD=1 and HM=1 the percentage of successful creatures increases from 50.39% to 73.94% and the percentage of queued creatures decreases from 48.58% to 25.16% when KBT takes place.

Overall the effects of erratic drivers on the changes in the percentage values of creatures' types are rather small in comparison with the effects of the other parameters. The presence of erratic drivers, i.e. when RD=1, always decreases the percentage of successful creatures, and increases the percentage of killed creatures. It also increases the percentage of queued creatures, except for the case when KBT=1 and HM=1.

Table 3 confirms that the HM parameter has the most influence on the changes in the percentage values of successful and queued creatures. Allowing creatures to move to different crossing points, i.e. when HM=1, significantly increases the percentage of successful creatures and decreases the percentage of queued creatures, because many more creatures may attempt to cross the highway at each time step. Also, Table 3 shows that the parameter HM has small effect on the changes in the percentage values of killed creatures, which is not surprising, as the numbers of killed creatures have been always very small in comparison with the numbers of the two other types of creatures.

From Table 3 we notice that the percentage of queued creatures is always high, regardless of the values of the parameters. It reaches its maximum of 86.41% when KBT=0, RD=0 and HM=0, and it reaches its minimum of 25.16% when KBT=1, RD=1 and HM=1. However, even in this case still almost a quarter of the creatures is queuing. Thus, the queuing creatures are always a significant part of all the creatures at each simulation end and if one wants to improve the model performance one needs to decrease their number. Table 3 shows that the transfer of KB always decreases on average percentage of queuing creatures at simulation ends. However, given the fact that these numbers are still relatively high one needs to look for some other mechanisms than KB transfer to improve creatures' performance.

4.2 Performance of Creatures using cwDF and Its Comparison with Performance When They Use cDF

In this section, we discuss the performance of creatures when they use cwDF instead of cDF in their decision-making process. The average of numbers, respectively, of successful, killed and queued creatures at simulation ends, expressed in percentage, and calculated from the simulation data set with HM=0 are displayed in Figure 3, and calculated from the simulation data set with HM=1 are displayed in Figure 4.

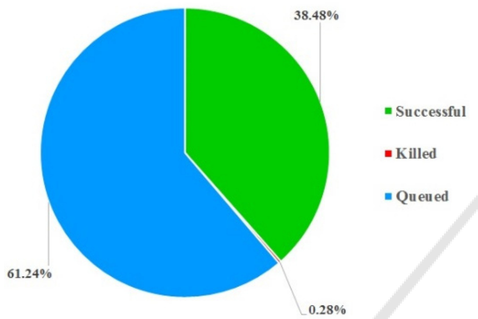

Figure 3: Average of numbers, respectively, of successful, killed and queued creatures at simulation ends, expressed in percentage, and calculated from the subset of simulation data with HM=0, when creatures used cwDF.

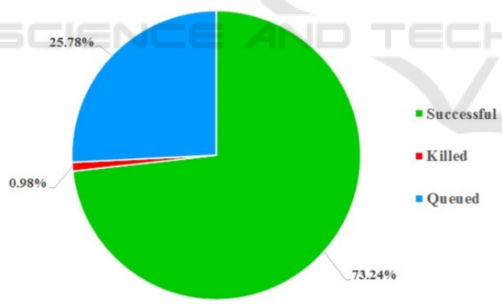

Figure 4: Average of numbers, respectively, of successful, killed and queued creatures at simulation ends, expressed in percentage, and calculated from the subset of simulation data with HM=1, when creatures used cwDF.

Comparing Figure 3 with Figure 1 and Figure 4 with Figure 2 we notice that cwDF has significant effect on the values of percentages of each creature type at simulation ends. We notice that the use of cwDF instead of cDF: (1) considerably decreases the percentages of queued creatures; (2) significantly increases the percentages of successful creatures; (3) and causes only a very small change in the percentages of killed creatures. More precisely, we notice that when HM=0 (i.e. from comparing Figure 3 with Figure 1), the percentage of queued creatures decreases by 23.59%, the percentage of successful creatures increases by 23.68%, and the percentage of killed creatures decreases by 0.09%. When HM=1, from comparing Figure 4 with Figure 2, we notice that the percentage of queued creatures decreases by 11.05% and the percentage of successful creatures increases by 10.97%. The observed changes in the percentages of queued creatures and successful ones confirm our intuition that by incorporating the assessment of WDs into DF a significant number of queued creatures would be converted on average into the successful ones. This is because the built in additional feed-back mechanism into cwDF, about the “correctness” of creatures’ waiting decisions, increases their chances of making more often correct crossing decisions in every time step, i.e. of choosing more often to cross the highway instead of to wait when the traffic conditions allow to do so.

Table 4: Average of numbers, respectively, of successful, killed, queued creatures at simulation ends, expressed in percentage, and calculated from the simulation data sets characterized by different values of the parameters KBT, RD and HM, when cwDF was used in these simulations. The value “0” means “without” and the value “1” means “with” KB transfer (KBT), random deceleration (RD), and creatures movement (HM), respectively.

KBT	RD	HM	Succe-ssful	Killed	Queued
0	0	0	32.95%	0.26%	66.79%
0	0	1	68.88%	1.03%	30.09%
0	1	0	32.32%	0.39%	67.29%
0	1	1	68.29%	1.17%	30.54%
1	0	0	44.27%	0.12%	55.61%
1	0	1	77.98%	0.56%	21.46%
1	1	0	44.39%	0.34%	55.27%
1	1	1	77.80%	1.17%	21.03%
		Average	55.61%	0.63%	43.76%

The average of numbers of each creature type at simulations ends, expressed in percentage, and calculated from the subsets of the date obtained by partitioning the *full simulation* data set in to the subsets depending on the values of the parameters KBT, RD and HM is presented in Table 4. The organization of the results displayed in Table 4 is the same one as of Table 3.

By comparing Table 4 with Table 3 one observes the increases in the percentage of successful creatures as the result of the decrease in the percentage of queued creatures and a very small change in the percentage of killed creatures for all the considered combinations of the parameters’ values. These changes can be seen better from Table 5, which displays the values of differences between the values of respective entries of Table 4 and Table 3.

Table 5: Difference in averages of numbers, respectively, of successful, killed, queued creatures at simulation ends, expressed in percentages, and calculated by taking the difference between the values of respective entries of Table 4 and Table 3. The value "0" means "without" and the value "1" means "with" KB transfer (KBT), random deceleration (RD), and creatures movement (HM), respectively.

KBT	RD	HM	Succe-ssful	Killed	Queued
0	0	0	+19.73%	-0.10%	-19.62%
0	0	1	+17.70%	+0.21%	-17.91%
0	1	0	+18.51%	-0.11%	-18.40%
0	1	1	+17.89%	+0.15%	-18.04%
1	0	0	+27.47%	-0.08%	-27.39%
1	0	1	+4.00%	+0.09%	-4.08%
1	1	0	+29.03%	-0.08%	-28.94%
1	1	1	+3.86%	+0.27%	-4.13%
Average			+17.27%	+0.05%	-17.32%

From Table 5 one can see easily that the use of cwDF in creatures' decision-making process reduces the percentages of queued creatures and allocates these gains mostly into the percentages of the successful ones, i.e. the use of cwDF converts on average some previously waiting creatures into the successful ones.

Additionally, Table 5 shows how cwDF improves the creature's performance for all the considered combinations of the parameters' values. For instance, the best improvement in the creatures' performance is achieved in simulations with KBT=1, RD=1 and HM=0. However, when KBT=1, RD=1 and HM=1, the use of cwDF does not enhance the creatures' performance too much, most likely, because the results were already quite good when the creatures used cDF in their decision-making process.

As about killed creatures, when cwDF is used instead of cDF, Table 5 shows that their percentages always increase when HM=0 (i.e., the differences are positive) and they always decrease when HM=1 (i.e., the differences are negative), regardless of the values of the other two parameters, KBT and RD. This is because when HM=0 the creatures have only two options to choose from: to cross or to wait. Thus, when the numbers of waiting creatures decrease the numbers of crossing creatures increase automatically. This increases their chances of making incorrect crossing decisions, and results in more creatures being killed. This is not the case when HM=1, because when HM=1, the creatures have 3 options to choose from: to cross, to wait or to change the crossing point. Thus, when the creatures decide not to wait they are not forced automatically to cross and they may move to another crossing point. This increase their chances of avoiding making incorrect crossing decisions that would cause an increase in the numbers of creatures being killed.

5 CONCLUSIONS

We have investigated performance of simple cognitive agents learning to cross a CA based highway for two decision formulas used by the agents' in their decision-making process. We measured the agents' performance by the numbers of agents who crossed successfully, who were killed and those who were still queuing to cross at simulation ends. We described the main features of the simulation model and the agents *observational social learning* strategy. In the decision-making process, depending on the simulation setup, the agents used one of the two decisions formulas: cDF, which was based only on the assessment of the outcomes of the agents crossing decisions, or cwDF, which was based on the assessment of the outcomes of the agents crossing and waiting decisions.

Our simulations showed that the performance of agents using cwDF in their decision-making process was much better than the performance of the agents using cDF instead. This is because in cwDF is built in the additional feed-back information about the "correctness" of the agents waiting decisions. This feed-back mechanism of cwDF increases agents' chances of making more often correct decisions in every time step. Since in cDF this feed-back mechanism is missing the agents may choose to wait even when the traffic conditions allow them to safely cross the highway.

Furthermore, our simulations showed, that transfer of accumulated knowledge base from traffic environment with lower car creation probability to the one with immediately higher car creation probability improves agents' performance, regardless what decision formula they used. Thus, accumulation of knowledge base helps the agents to be more successful. Additionally, our simulations showed: (1) that noise in the traffic environment, i.e. the presence of erratic drivers, decreases, but not so much, the agents' performance; (2) allowing agents to move to other crossing points improves noticeably their performance. Again, in both cases the agents' performance was improved when transfer of accumulated knowledge base was allowed.

Since autonomous robots may be identified with cognitive agents, thus their process of learning in dynamically changing environments can be studied through modelling and simulation of cognitive agents in such environments. The presented work contributes to this area of research and investigates how various model parameters affects the agents' performance.

ACKNOWLEDGEMENTS

The authors acknowledge helpful discussions with Bruno Di Stefano, Leslie Ly and Hao Wu. A.T.L. acknowledges partial financial support from NSERC of Canada.

REFERENCES

Russell, S., Norvig, P., 2014. *Artificial Intelligence, A Modern Approach*, Pearson Education Limited.

Poole, D.L., MacKworth, A.K., 2010. *Artificial Intelligence, Foundations of Computational Agents*, Cambridge University Press, Cambridge, US.

Ferber, J., 1999. *Multi-Agent Systems*, *An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, Longman.

Uhrmacher, A.M., Weyns, D., 2009. *Multi-Agent Systems Simulation and Applications*, CRC Press, Taylor & Francis Group, Boca Raton, Fl.

Nehavin, Ch. L., Dautenhahn, K., 2007. *Imitation and Social Learning in Robots, Humans and Animals*", Cambridge University Press, Cambridge, UK.

Bandura, A., 1977. *Social Learning Theory*, Prentice Hall, Englewood Cliffs, NJ.

Bandura, A., Ross, D., Ross, S.A., 1961. Transmission of aggression through the imitation of aggressive models, *Journal of Abnormal and Social Psychology*, 63, pp. 575-582.

Hoppitt, W., Laland, K.N., 2013. *Social Learning, An Introduction to Mechanisms, Methods, and Models*, Princeton University Press, Princeton.

Montes de Oca, M.A., Stutzle, T., 2008. 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.

Gong, Y.J., Zhang, J., Li, Y., 2014. 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, pp. 222 – 227.

Cheng, R., Jin, Y. 2015. A social learning particle swarm optimization algorithm for scalable optimization", *Information Sciences*, Vol. 291, pp. 43-60.

Liu, Z.-Z, Chu, D.-H., Song, Ch., Xue, X., Lu, B.-Y., 2016. Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition, *Information Sciences*, Vol. 326, pp. 315-333.

Lawniczak, A.T., Ernst, J.B., Di Stefano, B.N., 2012. 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.

Lawniczak, A.T., Ernst, J.B., Di Stefano, B.N., 2013. Simulated Naïve Creature Crossing a Highway, *Proceedia Computer Science* 18, 2013, pp. 2611-2614.

Lawniczak, A.T., Di Stefano, B.N., Ernst, J.B., 2014. 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, pp. 688-697.

Nagel, K., Schreckenberg, M., 1992. A cellular automaton model for freeway traffic", *J. Physique I,* 2, pp. 2221 – 2229.

Lawniczak, A.T., Di Stefano, B.N., 2008. 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.

Lawniczak, A.T., Di Stefano, B.N., 2010a. Multilane Single GCA-w Based Expressway Traffic Model, in: S. Bandini et al. (Eds.): Prof. of ACRI 2010, Springer-Verlag *LNCS* Vol. 6350, pp. 600-612.

Lawniczak, A.T., Di Stefano, B.N., 2010b. Digital Laboratory of Agent-Based Highway Traffic Model, *Acta Physica Polonica B Proc. Supplement,* Vol. 3, No. 2, pp. 479-493.

Lawniczak, A.T., Di Stefano, B.N., Ernst, J.B., 2015. Stochastic Model of Cognitive Agents Learning to Cross a Highway. In A. Steland, E. Rafajlowicz, K. Szajnowski (Ed.) The 12th Workshop on Stochastic Models, Statistics and their Applications, Springer Proceedings in Mathematics and Statistics, Vol.122, pp. 319-326.

Lawniczak, A.T., Di Stefano, B. N., Ly, L., Xie, S., 2016. Performance of Population of Naïve Creatures with Fear and Desire Capable of Observational Social Learning. *Acta Physica Polonica Series B, Proceedings Supplement*, 9 (1), pp. 95-107.

Lawniczak, A.T., Ly, L., Yu, F., Xie, S., 2016. Effects of Model Parameter Interactions on Naïve Creatures' Success of Learning to Cross a Highway, *The 2016 IEEE Congress on Evolutionary Computation (IEEE CEC 2016) at IEEE WCCI 2016*, 10 pages.

Dean, A., Voss, D., 1999. *Design and Analysis of Experiments*, Springer Science+Bussines Media, Inc.