Social Simulations with Reinforcement Learning

A Research Proposal
Presented to the
Faculty
Philippine Science High School
Eastern Visayas Campus

In Partial Fulfillment
of the Requirements
in STEM Research

by

Franz Louis T. Cesista

August 2017

CHAPTER I
The Problem and Its Setting

Background of the Study

One of the prevalent models of human economic behavior, *Homo economicus*, portrays people as consistently rational, self-interested agents driven to maximize their own welfares (Henrich, et al., 2001). Economists believe that this model is adequate to explain most human social behaviors. In the perspective of psychology and neurosciences, however, this is not satisfactory: humans are prone to biases (Tversky & Kahneman, 1975) and also intentionally discard the Darwinian rules of “survival of the fittest”, to some degree. Economical models may not have the foundational rigor as in pure mathematics, but they work – and for most, that is what matters most. This research aims to solidify (or strengthen, at least) some of these foundations.

This study applies the concept of emergence to define social behaviors. Essentially, complex social behaviors just naturally *emerge* from simple agent behaviors. Most migratory animals do not plan out their movement patterns but, somehow, they travel efficiently as groups and safely end up on their destinations. Neither would this study focus on the macro-scale (as done in economics) nor the micro-scale (as done in psychology); instead, certain social behaviors would be simulated *from* the micro-scale *to* the macro-scale. This approach is also called generative social sciences (Epstein, 2006).

The earliest social simulations include Sugarscape. In this simulation, the agents only followed hard-coded rules (such as “farm when not hungry, else trade”) but were able to show several, human-like social behaviors (such as price fixing in large-scale barter systems) (Epstein

& Axtell, 1996). There is a catch. This type of simulations has a fundamental flaw: those hard-coded rules are still written by the researches and thus are prone to the latter's personal biases. The agents must learn to behave with minimal to no input from the researchers (Busoniu, Babuska, & De Schutter, 2010). The challenged to simulate learning should then be addressed.

Psychologists believe that *learning* is an achievement over millions of years of evolution (Buss, 2015). It cannot simply be replicated. But when *learning* is defined as 'changing one's behavior based on experiences', then it can easily be modeled. One such model is Reinforcement Learning. Reinforcement Learning is a subset of Machine Learning which is a subset of Artificial Intelligence. It can act the brain of an agent. Essentially, this model uses statistics to *balance* the agent's long-term and short-term *goals* when deciding which *action* to take based on what the agent *sees* at the moment.

In summary, this research aims to simulate certain social behaviors emerging from simple, learned, agent behaviors with reinforcement learning.

Objectives

Main Problem:

- To simulate certain social behaviors.

Sub-problems:

- To simulate learning**,** specialization, migration, and cooperation.
- To compare the agents' macro-level factors such as population size, wealth distribution, etc. to real world data.
- To study the relationship between the productivity and the amount of information being managed by an agent.

Significance of the Study

Societies are complex nonlinear systems (analogous to turbulence in physics) which are difficult to study using conventional mathematical models. Thus, simulations are necessary replacements for such models. That mentioned, this study would provide various insights regarding social behaviors which would only be most useful to further researchers.

Scope and Limitations

The human brain is complex and so is human behavior. Thus, not all of the latter's aspects would be simulated - only learning through reinforcement. Aside from using learning algorithms to model the agents' brains and tweaking their environments, no other inputs from the researcher would be processed by the agents. Only the classic reinforcement learning algorithm, q-learning, would be used instead of the more advanced ones with neural networks. This is to speed up the experiments. Additionally, the environment would be a 2-dimensional discrete grid-world to minimize extraneous variables.

Since the simulations are not designed for use by the general public, the graphical user interface (GUI) would also be minimal.

On the simulations, the concepts of *specialization*, *migration*, nor *cooperation* would be introduced to the agents. The environments would be designed for each experiment and the agents would just be left to adapt to these environments.

The wealth distribution would be studied in the context of wealth inequality. This is measured by calculating the agents' Gini Coefficient as a whole. And lastly, a *simple* mathematical model would also be developed between the relationship of an agent's productivity and the amount of information it manages. This can be done with excel and would not be pursued further.

Definition of Terms

Action Selection Method – An algorithm that takes in the compressed information from the Q-learning algorithm and uses probability techniques to search for the optimal policy.

Agents – Artificial intelligence units that can adapt to the environment. Here, they can safely be thought of as ants since they are not as intelligent as humans are, yet. This term does not carry its connotations in sociology and psychology.

Boltzmann Method – This action selection method is generally biased on exploration. This method normalizes the immediate rewards using an exponential function, dampened by a "temperature" variable, before calculating the probabilities of selecting the actions.

Exploration – When an Action Selection Method, and consequently, the agent, is biased on exploration, it sacrifices immediate rewards in hopes of finding long-term and high-reward strategies.

Exploitation – When an Action Selection Method, on the other hand, is biased on exploitation, it prioritizes its immediate rewards and heavily relies on its already learned experiences.

Machine Learning – A set of mathematical optimization techniques, loosely inspired by neurosciences, that focuses on the design of systems that can learn from, make decisions, and/or make predictions based on data.

Mod_random or Epsilon-Greedy Method – This action selection method is biased on exploitation. In usual implementations, the algorithm takes the action with the maximum immediate reward with probability ε and the others with equal probability $\frac{(1-\varepsilon)}{|A|-1}$, where |A| is the total number actions the agent can take.

Policy – A series of actions an agent does based on a certain series of events. A policy can either be hard-coded by the researcher or learned by the agents themselves. The latter is preferred as to minimize the effects of the researcher's biases on the results.

Q-Learning – a Reinforcement Learning algorithm which utilizes Bayesian probability mechanics to search for the optimal agent policy.

Reinforcement Learning – A Machine Learning technique which allows the agents to learn through trial and error, as most living organisms do.

Specialization – In this study, specialization is defined as the certainty of doing a series of actions in a certain series of events. Or, mathematically, the certainty of following a certain policy, optimal or not. In this case, it is the certainty of choosing to gather a certain resource over time.

CHAPTER II
Review of Related Literature

Multiagent Systems

According to Sawyer (2003), until the development of Multiagent Systems in the 1990s, social simulations primarily used analytics, or *Equation Based Modeling* which mainly consists of evaluating a set of equations. This includes the application of rational choice theory, the Nash equilibrium (Coleman, 1987), and other macro-level sociological and organizational models (Forrester, 1968). They are not concerned with the system's properties as they change through time but only with the system's end state. For example, in Belovsky's (1987) *Subsistence Behavior Model*, only the optimal subsistence strategy is of interest, not on how it was arrived at. In light of this, it can be generalized that Equation Based Models are descriptive and are, on themselves, meaningless.

A decade on, many sociologists still thought that Equation Based Modeling is the only computational tool available to them and social simulation is even categorized as a branch of mathematical sociology. This may be due to the neglect of Multiagent Systems (or mention them only in passing) by survey articles in the late 1990s (e.g. Halpin, 1999; Hannerman et.al., 1995). This perception was gradually overturned as this subfield matured (Axtell, 1999). By the early 2000s, it already has the potential to be useful tools for sociologists.

A Multiagent System is a group of interacting entities sharing a common environment. These entities are called agents, each having control over their own behavior and can act without the intervention of a human or other systems. Interest in this subfield was first driven by the development of multiprocessors in the 1980s and the rapid expansion of the internet in the 1990s

(Sawyer, 2003). A few years later, the development of desktop computers decreased computational costs thus making the subfield accessible to more researchers. Recently, it has been applied in a variety of domains including robotic teams, resource management, data mining, social simulations, etc. (Busoniu, Babuska, & De Schutter, 2010; Yan, Y., Kuphal, T., Bode, J., 2000).

Although widely used, 'agent' is a loose term which does not carry the same connotations as it does in sociology and psychology. To understand its connotations, a brief history of Multiagent Systems is helpful. In the 1990s, this subfield emerged from precursor systems with multiple interacting processes but were still not autonomous. The advent of *Object Oriented Programming* allowed researchers to design fully automated '*objects*' that can maintain its own data structures and procedures. These Agents can also communicate with each other and with the environment and reacts by executing a corresponding '*method*'.

An example of this is the *Sugarscape Model*. In *Sugarscape*, as stated earlier, Epstein and Axtell (1996) were able to simulate "emergent" patterns such as cultural segregation, economic markets, intergroup warfare, etc. The model has also been extended to include various aspects of human behavior: One is the *Village Ecodynamics Model* (VEM) which was used by Kobti et.al. (2004) to better understand the kinship behavior of ancient tribes; another is Epstein's (2002) Civil Violence Model which presents two variants of civil violence (civil war and rebellion); and etc. Sugarscape also served as the foundation of *Multi-Agent Reinforcement Learning* research which will be discussed later.

Going back, OOP also allowed scientists to build *Distributed Artificial Intelligence* systems (O'Hare & Jennings, 1996). Essentially, these systems are just AI agents which are hierarchically organized around a centralized controller; thus, the agents are never truly autonomous. After noticing this trend, researchers gradually began to experiment on decentralization. This shift to autonomy led to the use of the term 'agent'. Although these systems are only notably useful in robotics, its concept led to the development of Multiagent Reinforcement Learning.

Multiagent Reinforcement Learning

Multiagent Reinforcement Learning is a subfield of Multiagent Systems wherein the agents learn by trial-and-error interactions with its dynamic environment. Eventually, the behaviors of these agents converge into optimal strategies which maximizes their gained rewards. To be able to simulate this, researchers turned to probability principles and cognitive sciences. They eventually found RL algorithms with good convergence and consistency properties that are easily applicable to *Single-Agent Reinforcement Learning* tasks (Mahadevan, 1996) and generalized it for use in this subfield. So, how does it work?

Through the application of *Bayesian probability principles* and the reduction of the system to a *Markov web;* in which, time is discrete and the agents decide which action to execute next based on the observed state of the environment in after each time-step. The agents will not only take into account its immediate reward but also its reward in the long run; thus, the agents must be able to learn from delayed reinforcement (Kaelbling & Littman, 1996).

That said, the most common approach is the use of the Q-Learning algorithm. In Q-Learning, actions are given weights (Q-values) and one is selected depending on the ad-hoc exploration method being applied. These Q-values are initially equal but varies as the agent adapts to its environment which effects a positive feedback loop: the greater the Q-value, the greater is the probability of the corresponding action be executed, the greater the Q-value will become, and so on. This ensures that the agent will learn the optimal strategy (policy) after a finite timeframe.

In addition, Q-Learning is *exploration insensitive*: The Q-values will converge regardless of how the agent behaves. This simply means that the details of the exploration strategy will not affect the convergence of the learning algorithm. These exploration strategies can be viewed as heuristics each having distinct trade-offs between *exploration* and *exploitation* (should the agents take the best course at hand or wander off for a little while longer?). Thrun (1992) has surveyed a variety of these techniques and Kaelbling et.al. (1996) classified them into three categories based on their *exploration-exploitation* tradeoffs:

In the first, generally called *Greedy Strategies*, always choose the action with the highest estimated payoff. The Q-values in these heuristics typically converge faster with the risk of being stuck in suboptimal policies. The *Epsilon-greedy* heuristic solves this by allowing the agent to choose another random action based on a decreasing probability function. The second includes heuristics that chooses actions based on probability distributions which are biased in favor of actions having relatively greater Q-values then the others. An example is the *Boltzmann Exploration* in which a *temperature* parameter T determines the "rate of exploration" of the agents. And the third, *Interval-Based Techniques,* are greedy strategies that allows the agents to explore using another heuristic. In Kaelbling's (1993) *Interval Estimation* algorithm, another heuristic calculates the success probability of executing each of the actions and returns the data to the algorithm which then chooses the action with the highest upper bound.

Going back, these reasons make Q-learning the most popular and apparently the most effective algorithm for reinforcement learning. In addition, its simplicity and generality make it attractive for Multiagent Reinforcement Learning. Shibata and Okuhara (2006) extended

Sugarscape by applying Q-learning to the agents. Not only did the agents learn the optimal policy in static environments, they also performed better in dynamic settings such as seasonal rotations, etc. There are a lot more examples that the scope of this subfield is now too wide for a single survey to comprehensively analyze. The following is Busoniu, et.al. (2010) taxonomy of these algorithms:

Agent Awareness	*Task Type*		
	Cooperative	*Competitive*	*Mixed*
Independent	Coordination-free	Opponent-independent	Agent-independent
Tracking	Coordination-based	---	Agent-tracking
Aware	Indirect coordination	Opponent Aware	Agent-aware

Task Type	*Static or Dynamic?*	*Algorithms*
Fully Cooperative	Static	Joint Action Learners (JAL), Frequency Maximum Q-value (FMQ)
	Dynamic	Team-Q, Distributed-Q, Optimal Adaptive Learning
Fully Competitive	NA	Minimax-Q
Mixed	Static	Fictitious Play, MetaStrategy, Infinitesial Gradient Ascent (IGA), Win-or-Learn-Fast-IGA (WoLF-IGA), Generalized IGA, GIGA-WoLF, AWESOME, Hyper-Q
	Dynamic	Single-agent RL, Nash-Q, Correlated Equilibrium Qlearning (CE-Q), Asymmetric-Q, Non-Stationary Converging Policies (NSCP), WoLF-Policy Hill Climbing (WoLF-PHC), PD-WoLF, EXORL

Task Type	*Open Issues*
Fully Cooperative	• Rely on exact measurements of the state • Many also require exact measurements of the other agents' actions • Communication might help relax these strict requirements • Most suffer from the curse of dimensionality
Mixed	• Static, repeated games represented a limited set of applications

	• Most static game algorithms assume the availability of an exact task model, which is rarely the case in practice • Many suffer from the curse of dimensionality • Many are sensitive to imperfect observations

Specialization

Specialization, or division of labor, allows individuals to maximize their fitness by exploiting their skill set and environment (Murciano, del Millan, & Zamora, 1997). They do this by cooperating with others in a community of mutual interest (Spencer, Couzin, & Franks, 1998). While there are many definitions of specialization, this study defines specialization as the choice to produce excessive amounts of some goods relative for subsistence requirements, while simultaneously under-producing others. They must then trade these excesses for the goods that they need (Evans, 1978). That said, specialization can be treated as a spectrum: some agents can be fully specialized (whereby they prefer to perform only one task), and the others partially specialized (whereby they perform all tasks to varying degrees) (Cockburn, Kobti, & Kholer, 2010).

In addition, specialization is directly related to the economics of the system. A specialized agent will simultaneously increase the supply for some resources (its products) while increasing the demand for others (its needs) (Young, 1928). This demand will then encourage other agents to specialize on its production leading to a positive feedback loop of specialization. Eventually, the market reaches a state of equilibrium and, consequently, so does specialization: if the supply is already saturated, it would be inefficient for other agents to produce it – forcing them to supply other resources.

In Cockburn et. al.'s *Village Ecodynamics Model,* the agents can adjust their specialization among given tasks using reinforcement learning. The model also presented a complicated trading system. In this system, the agents are classified into five states, *wealthy, trading, satisfied, critical*, and *starving*, respectively. When an agent reaches either the *critical* or *starving* state, they will immediately find viable trade partners in either *trading* or *wealthy* states and trade through the *Barter Exchange Network*. If it is still *starving,* then it'll *beg* for resources from its kin through the *Generalized Reciprocal Exchange Network.* And *borrow* through the *Balanced Reciprocal Exchange Network* as a last resort. This trading system allowed the agents to survive through droughts and famine periods.

CHAPTER III
Methodology

Phase 1: Framework of the Simulations

Coding Environment

Python 3.5 was the primary programming language used in the study. This is because it is fast to write, already has an active open-source community, and is compatible with C++. Replacing the crucial parts of the simulation with C++ can drastically speed it up.

The python libraries used were PyGame, Numpy, and Matplotlob. The first for the display and the other two for the development of the data analysis tools used in the study. OpenAI's Gym environment was used as a layer of abstraction to make the simulation easier to scale and/or customize.

Git was used to do revision control over the code so new features could be implemented without the risk of the entire project breaking down. This system was hosted in GIthub.

Environment Design

Vilches' (2016) open-sourced repository on Basic Reinforcement Learning was forked as a platform for the simulations. This was then immediately ported from python 2.7 to python 3.5, modularized, and made compatible with OpenAI's Gym environment.

Overall, 5 separate modules were developed: *agent.py, gridworld.py, cell.py, display.py,* and *start.py. agent.py* manages the agents' behavior, *gridworld.py* manages the environment, *cell.py* manages the dynamics of the cells in the environment, *display.py* manages the Graphical User Interface, while *start.py* serves as the bridge between the other four. With this, the overall design of the simulation can easily be tweaked by swapping out the unneeded modules with the necessary ones.

To reduce computing cost, the environment was designed to be a 20-by-20 gridworld surrounded by impassable walls. Each cell in the gridworld contains five types of resources which the agents could gather. When harvested, the resource grows back at a rate of 2% per time-step.

Development of the Data Analysis Tools

Overall, there were 9 types of datasets gathered. A data visualization tool was developed for each. Additionally, each dataset is unique on their own right; therefore, 65 versions of these tools were used in the study. All of these tools are available at https://github.com/LE-LOY/cultural-evolution/tree/master/miscSims.

Phase 2: Simple Social Behaviors

Reinforcement Learning

The agents were designed to behave individually; they have different *brains* (but running the same reinforcement learning algorithm) and different memories. The agents were then tested in various gridworlds and compared their behavior to that of humans. With a stable learning environment in place, the two action selection methods, mod_random and boltzmann were then compared.

Agent Specialization

Next followed a series of experiments on specialization. In each experiment, there are five types of resources the agents had to gather to survive. But without trading, the agents would just opt to collect all of those resources themselves (Rotemberg & Saloner, 2000). So to allow them to specialize, they must also be allowed to trade. And allowed they were. After that, the distribution of specialists and their degree of specialization were then analyzed. The latter was measured by calculating the average Residual Entropy of the agents' *brains*.

The Village Ecodynamics Project (Varien, Ortman, Kohler, & Glowacki, 2007) were then successfully attempted to be improved.

Phase 3: Analysis of Miscellaneous Data

The agents' wealth inequality was measured by calculating the Gini Coefficient of their wealth distribution. The fluctuations in the agents' population size over time were also tried to be explained. And lastly, the relationship between the agents' productivity and the amount of information they managed was also studied.

Phase 4: Complex Social Behaviors

Migration

The agents were then tested in environments with seasonal cycles for their movement patterns to be studied. In these environments, some resources get scarce in certain places in certain periods of the simulation.

Cooperation

The simulations in this section are yet to be conducted. Here, the agents will be tested in environments where both cooperation and competition will not be rewarded explicitly. Indirect approaches would be used to encourage them to cooperate (Busoniu, Babuska, & De Schutter, 2010).

References

Axtell, R. (1999). Why Agents? On the Varied Motivation for Agent Computing in the Social Sciences. *Workshop on Agent Simulation: Applications, Models, and Tools*, 3-24.

Belovsky, G. (1987). Hunter-gathering Foraging: A Linear Programming Approach. *Journal of Anthropological Archeology*, 29-76.

Busino, G. (2000). The Significance of Vilfredo Pareto's Sociology. *European Journal of Social Sciences*, 217-228.

Busoniu, L., Babuska, R., & De Schutter, B. (2010). *Multi-Agent Reinforcement Learning: An Overview.* Delft: Delft Center for Systems and Control.

Buss, D. (2015). *Evolutionary Psychology: The New Science of the Mind* (5th ed.). Psychology Press.

Cockburn, D., Kobti, Z., & Kholer, T. (2010). A Reinforcement Learning Model for Economic Agent Specialization. *Florida Artificial Intelligence Research Society Conference*, (pp. 20-25). Florida.

Coleman, J. (1987). Microfoundations and Macrosocial Behavior. *The Micro-Macro Link*, 153-173.

Epstein, J. (2002). Modeling Civil Violence: An Agent-Based Computational Approach. *Proceedings of the Natural Academy of Sciences* (pp. 7243–7250). Washington, DC: Highwire Press.

Epstein, J. (2006). *Generative Social Sciene: Studies in Agent-Based Computational Modeling.* New Jersey: Princeton University Press.

Epstein, J., & Axtell, R. (1996). *Growing Artificial Societies - Social Science from the Bottom Up.* Washington, D.C.: Brookings Institution Press.

Evans, R. (1978). Early Craft Specialization: An Example for the Balkan Chalcolithic. *Social Archaeology: Beyond Subsistence and Dating*.

Forrester, J. (1968). *Principles of Systems.* Cambridge: MIT Press.

Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., Gintis, H., & McElreath, R. (2001). In Search of Homo Economicus: Behavioral Experiments in 15 Small-Scale Societies. *Papers and Proceedings of the Hundred Thirteenth Annual Meeting of the American Economic Association*, 73-78.

Kaelbling, L. (1993). *Learning in Embedded Systems.* Cambridge: The MIT Press.

Kaelbling, L., & Littman, M. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 237-285.

Kobti, Z., Reynolds, R., & Kohler, T. (2004). The effect of kinship cooperation learning strategy and culture on the resilience of social systems in the village multi-agent simulation. *Evolutionary Computation*, 1743-1750.

Lappenschaar, J. (2010). *On the Evolution of Cooperation, Trust, and Diversity.* Nijmegen: Radboud University Press.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J., & Graepel, T. (2017). *Multi-agent Reinforcement Learning in Sequential Social Dilemmas.* Retrieved from https://storage.googleapis.com/deepmind-media/papers/multi-agent-rl-in-ssd.pdf

Mahadevan, S. (1996). Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. *Machine Learning*, 159-195.

Murciano, A., del Millan, J., & Zamora, J. (1997). Specialization in Multi-Agent Systems Through Learning. *Biological Cybernetics*, 375-382.

O'Hare, G., & Jennings, N. (1996). *Foundations of Distributed Artificial Intelligence.* New York: Wiley.

Rotemberg, J., & Saloner, G. (2000). Competition and human capital accumulation: a theory of interregional specialization and trade. *Regional Science and Urban Economics*, 373-404.

Sawyer, K. (2003). Artificial Societies: Multiagent Systems and the Macro-Micro Link in Socioloical Theory. *Sociological Methods and Research*, 325-363.

Shibata, J., & Okuhara, K. (2006). Acquirement of Agent Movement Rule which Adapt to Change in Environment Condition. *Asia Pacific Management Review*, 383-388.

Spencer, A., Couzin, I., & Franks, N. (1998). The dynamics of specialization and generalization within biological populations. *Journal of Complex Systems*, 115-127.

Thrun, S. B. (1992). The Role of Exploration in Learning Control. In D. A. White, & D. A. Sofge, *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches.* New York: Van Nostrand Reinhold.

Traulsen, A., & Nowak, M. (2006). Evolution of Cooperation by Multilevel Selection. (S. Levin, Ed.) *Proceedings of the National Academy of Sciences of the United States of America, 103*(29), 10952–10955.

Tversky, A., & Kahneman, D. (1975). *Judgment under Uncertainty: Heuristics and Biases* (Vol. 11). Dordrecht: Springer. doi:https://doi.org/10.1007/978-94-010-1834-0_8

Varien, M., Ortman, S., Kohler, T., & Glowacki, D. (2007). Historical Ecology in the Mesa Verde Region: Results from the Village Ecodynamics Project. *American Antiquity*, 273-299.

Vilches, V. M. (2016). *Basic Reinforcement Learning.* Retrieved from Github: https://github.com/vmayoral/basic_reinforcement_learning

World Bank. (2017). *World Development Indicators: Distribution of income or consumption*. Retrieved from The World Bank: http://wdi.worldbank.org/table/1.3

Young, A. A. (1928). Increasing returns and economic progress. *The Economic Journal, 38*(152), 527-542.