NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

REPRESENTING TRUST IN COGNITIVE SOCIAL SIMULATIONS

by

Shawnoah Pollock

September 2011

Thesis Advisor:	Christian Darken
Second Reader:	Jonathan Alt

This thesis was done at the MOVES Institute
Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2011	Master's Thesis

4. TITLE AND SUBTITLE Representing Trust in Cognitive Social Simulations	5. FUNDING NUMBERS
6. AUTHOR(S) Shawnoah Pollock	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution is unlimited	A

13. ABSTRACT (maximum 200 words)

Trust plays a critical role in communications, strength of relationships, and information processing at the individual and group levels. Cognitive social simulations show promise in providing an experimental platform for the examination of social phenomena such as trust formation. This work is a novel attempt at trust representation in a cognitive social simulation using reinforcement learning algorithms. Initial algorithm development was completed within a standalone social network simulation and tested using a public commodity game. Evaluation of the contributions and dividends within the public commodity game shows that many of the expected behaviors of human trust formation are present. Initial results show that reinforcement learning can accurately capture the core essentials of human trust formation. Following standalone testing, the trust algorithm was imported into the Cultural Geography model for large-scale test and evaluation.

14. SUBJECT TERMS Cognition, Agent, Simulation, Trust, Reinforcement Learning, Machine Learning, Public Commodity, Experimental Economics	15. NUMBER OF PAGES 85
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

REPRESENTING TRUST IN COGNITIVE SOCIAL SIMULATIONS

Shawnoah Pollock
Lieutenant, United States Navy
B.S., University of California Los Angeles, 2002
M.S., University of California Los Angeles, 2004

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
MODELING, VIRTUAL ENVIRONMENTS AND SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL
September 2011**

Author: Shawnoah Pollock

Approved by: Christian Darken
Thesis Advisor

Jonathan Alt
Thesis Co-Advisor

Mathias Kölsch
Chair, MOVES Institute

Peter Denning
Chair, Computer Science Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Trust plays a critical role in communications, strength of relationships, and information processing at the individual and group levels. Cognitive social simulations show promise in providing an experimental platform for the examination of social phenomena such as trust formation. This work is a novel attempt at trust representation in a cognitive social simulation using reinforcement learning algorithms. Initial algorithm development was completed within a standalone social network simulation and tested using a public commodity game. Evaluation of the contributions and dividends within the public commodity game shows that many of the expected behaviors of human trust formation are present. Initial results show that reinforcement learning can accurately capture the core essentials of human trust formation. Following standalone testing, the trust algorithm was imported into the Cultural Geography model for large-scale test and evaluation.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

CG - Cultural Geography

GA – Genetic Algorithm

HSCB – Human Social, Cultural and Behavioral

ML – Machine Learning

M&S – Modeling and Simulation

PC - Public Commodity

PD – Prisoner’s Dilemma

RL - Reinforcement Learning

TRAC – TRADOC Analysis Center

TRADOC – Training and Doctrine

U.S. – United States

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Without the constant source of motivation and guidance that I found in LTC Jon Alt I would never have been able to finish this project. He always seemed to know exactly when I needed to be pushed and when I needed time to get things moving on my own. Thank you Jon!

I would also like to thank my thesis advisor, Dr. Chris Darken. He allowed me the latitude to undertake a fairly difficult thesis project and work on it independently so that I had the freedom to really express my inner creativity. Thank you, Chris!

During my research, I always received ample support from TRAC-Monterey. In particular I would like to thank Harold Yamauchi and MAJ Francisco Baez who both took the time to help me learn and understand Cultural Geography model and were indispensible parts in the success of this project. Thank you Cisco and Harold and everyone at TRAC-Monterey!

There were many NPS students who I owe a big thanks to yet to thank them all would take all the pages in this thesis. Therefore, I will simply say thank you to those who specifically contributed in some way to this project: Ozkan Ozcan, Dan Mckaughan, Rob Zaborowski, Tommy Getty and all my colleagues at NPS. Thank you all!

Without the tremendous resources of the MOVES and NPS faculty and staff, there was absolutely no way I could have completed this work. In only brushing the surface of the creative depths of these individuals I have learned more

about Modeling and Simulation than I ever imagined possible. All I can say is that I stand in awe at the talent that is here at MOVES and at NPS. Thank you all!

Lastly, I would like to thank my wife and kids for the love and support they provide. To each of them I also want to say thank you for the sacrifices they make to help defend this country, from moving around every two years to having to watch their dad or husband sail off for the far side of the world. To each of them I say thank you and I love you!

I. INTRODUCTION TO THE PROBLEM OF MODELING TRUST

A. PROBLEM STATEMENT

Social simulation, and in particular human behavior modeling, has become an extremely important element of modern warfare. This thesis is an attempt to solve one small piece of the problem of modeling human behavior by answering the following question:

> How can trust formation be modeled within human, social, cultural and behavior (HSCB) based simulations?

In answering this broad question, it was necessary to answer several related questions. These questions are:

> What is an appropriate working definition of trust as it applies to HSCB models and in particular agent communications within these models?
>
> What trust based effects do we expect to see from a properly implemented trust model within HSCB applications?

Recent advancements in human behavior modeling, cognitive agent simulations and artificial intelligence have made the goal of predictive HSCB modeling attainable in the foreseeable future. In the last ten years in particular, it has become possible to begin tackling the individual problems that are the stepping stones in achieving this modeling and simulation goal. This research is directed specifically at the problem of modeling trust formation in a society and how it effects communication within that society. If a society can be successfully modeled,

including accurate human behavior models, it may become possible to understand how insurgencies form and operate. Being able to predict the actions of an insurgency or even being able to prevent it from ever forming will be an immeasurably powerful tool in modern security and stability operations.

The problem of modeling a society and more importantly of modeling an insurgency within that society is one of understanding political power. Political power stems directly from the will of the populous and their opinions of those wielding the power (regardless of whether the power is implemented through fear or through proper civil discourse). The ebb and flow of political power is directly related to the communications that take place within that society, or more accurately how information is disseminated through the social network of that society. Within social networks is a flow of information that begins with either an individual or a group and is transmitted to others with whom they have ties. The recipients of this information will make a trust evaluation in order to determine whether the information is actionable. Information that is trusted can be used by the recipient in several ways. Primarily, the recipients will adjust their beliefs based on this new piece of trusted data. In some cases, the information that is received is something that the recipient feels his closest friends should also know about, and he can then resend this information further into the social network.

It is easy to see how simple person-to-person communications are the building blocks of information flow in a society. Furthermore, modeling the flow of information

is the most fundamental step in modeling political power in a society. The particular problem addressed within this work is how best to model a system by which agents in a social network can evaluate information that they receive and how they can determine who in their local group of agents are trusted enough to receive this new piece of information.

In the following section, there is a further exploration of the problem of trust modeling, including a discussion of why it is important to develop these kinds of models.

B. EXPLORING THE PROBLEM

When viewed at the national level, the objectives of warfare have never significantly changed for as long as there have been wars in society. Carl von Clausewitz said in his unpublished treatise *On War*, "The political object is the goal, war is the means of reaching it, and the means can never be considered in isolation form their purposes" (1832, p. 87). In other words, the motivation to go to war is political in nature and therefore combat is just another means of achieving political ends. In conventional warfare, the forces involved are typically evenly matched and the conflict is resolved by means of kinetic combat until the forces decide to cease this action usually by means of a treaty or surrender. In the 21st century, the United States has developed a nearly unmatched military power in conventional kinetic ability. This power encourages the enemies of the United States, especially those that are nongovernment actors, to resort to insurgencies and

nonconventional methods rather than meeting the United States in open combat (Department of the Army, 2006).

The new flavor of warfare is one in which a technologically and numerically superior force is engaged with an inferior one that is willing to resort to insurgent and terrorist tactics. The United States' policies when involved in foreign internal defense are designed to protect the population of the host nation, as well as aid in dealing with insurgencies and other opposing forces that would prevent the development of an independent and free nation (Joint Chiefs of Staff, 2004). The ultimate nature of warfare itself has not changed in that in these operations the U.S. still seek to win and use political power. In conventional warfare, it is the choice between capitulation and combat that drives a nation to submit to the will of their opponent. In recent wars, such as in Iraq and Afghanistan, it has been found that kinetic military force alone is not enough to gain and wield political power. If the U.S. is to allow these nations to build themselves up in the ways that they see fit in order become a free and independent, it simply cannot be done by sheer force alone.

Following major military operations such as in Iraq and Afghanistan, there is a likely to be a much longer period of stabilization and rebuilding. It is during this time that to find success a legitimate government that is widely supported by the populace and capable of dealing with counterinsurgents on its own must be established. It is also during these beginning times that the counterinsurgency has the best opportunity to undermine this goal.

When dealing with kinetic operations, the efforts in modeling and simulation dealt mostly with the objective elements of the problem. It is easy to model how a shell from a tank is going to travel and what kind of damage it could do to an enemy tank. It is also well within the capability of modeling and simulation (M&S) to model conventional warfare through simulation and analysis using fairly straightforward laws of military conflict. The reason these types of problem paradigms are well understood and easily modeled is that there is little human cognition involved. Modeling insurgencies and public opinion has very little to do with physics and concrete laws and rather has everything to do with modeling the human mind. Modeling an insurgency requires an understanding of the changing motivations of the insurgents and of the population in which they are hiding and operating.

Understanding the complexities of human behavior is still in its infancy. Furthermore, looking at a socially connected system of human beings and allowing them to freely interact makes modeling the behavior of that system even more difficult. Take as an example the U.S.-led coalition force conducting stability operations in Iraq. If the military decides to improve the roads in the city, it would be of great advantage to the populace. However, if an insurgency group does nothing more than spread rumors that the coalition force's intentions are not to benefit the population but rather to ease their own military vehicle's travel through the city, the popular sentiment may turn against coalition forces. If the insurgents then destroy some of the roads and possibly even kill contractors building the roads it may end up fueling hostility toward

the coalition. In this situation, military commanders will be faced with troubling dilemmas. They will have to decide if it is best to conduct operations against the insurgency, such as searches and arrests of those involved, or will it be more beneficial to rebuild the roads, or will it be better to move onto another public works project. They will have to know if it is better to try and sway public opinion by interacting with the populace or if it is better to go through city officials. In order for the U.S. to make justifiable decisions in the situations described above, it is necessary that leaders be given tools that can aid in those decisions.

The actions taken by the U.S. during stability operations will have broad and far reaching consequences. This work is attempting to further development models that can track how particular actions might sway public opinion. The type of HSCB model that this work is applicable is one in which the agents within the model are connected in some kind of social network within which information can flow, e.g., the Cultural Geography (CG) model developed by TRAC-Monterey, which will be discussed in detail in a later section. When agents in the social network witness an event, they will form some kind of opinion and may choose to share this information with its closest neighbors. The key to modeling this information flow is in understanding the processing mechanisms of the individual agents. The particular goal of this work is to model the trust decision that agents make when they receive information and also the decision of who to trust enough to resend vital information on to. In the following sections there are more complete

discussions of the particular methodologies employed in modeling these trust decisions.

C. METHODOLOGY

This research uses Reinforcement Learning (RL) algorithms for modeling trust. The trust algorithm was first implemented in a standalone simulation intended to be a vastly simplified version of the Cultural Geography (CG) model. Following initial testing it was then transitioned to the CG model for full scale test and evaluation. In later sections there will be in depth introductions to both RL and CG.

The first step in the development of a successful trust algorithm was to build an extremely scaled down test bed that mimics some of the social networking behavior of the CG model. This was done by modeling a simple social network of agents and applying modern social network analysis to see how the network evolves and how the agents communicate. Once an operable test bed social network simulation was developed, the trust algorithm was implemented and the agents were made to play trust games. In particular, the economics based game called Public Commodity (PC) was used in our analysis, which will also be discussed in depth in later sections.

The results of testing the trust algorithm with the PC game led to several revisions until a satisfactory outcome was reached. At this point the trust algorithm was transplanted into the CG model. A similar game of trust was developed within the CG model to test this early implementation of the algorithm.

THIS PAGE INTENTIONALLY LEFT BLANK

II. RELEVANT BACKGROUND INFORMATION

This section provide background information that is needed to develop a RL based trust algorithm as well as relevant background information needed to test the algorithm with the CG model using the PC game.

A. FOUNDATIONS OF TRUST

The first rule of simulation is to know what is being simulated. If you want to draw an 800-pound gorilla, you must first know what an 800-pound gorilla looks like.

Trust is a concept that is easy to talk about casually but extraordinarily difficult to define specifically. This is especially true when it must be described in precise terms that can lead to a useful computer algorithm. The concept of trust is important to a very broad spectrum of academic disciplines including psychology, sociology, philosophy, computer science, political science, and much more. Certainly, there are no one-size-fits-all definitions for trust. Therefore, in order to establish a useful working definition of trust one must first be careful to define exactly how it will be used and in what context it can be graded. For this work the following definition of trust will be used:

> **Trust = The perception of one agent (trustor) that other agents (trustees) will adhere to an unspoken social contract and will faithfully conform to preconceived actions based on their past actions, perceived characteristics or position in the social hierarchy.**

The motivations for the definition of trust will be outlined in the following sections. The ultimate goal for this project is the implementation of a trust based filter within the CG model, which is an agent based simulation currently being developed by TRAC-Monterey. Therefore, the first subjects that must be discussed are the basics of agent modeling and communication and how trust affects communications.

1. Agent Communications and Trust

Agent based modeling can most simply be described by being made up of a group of agents that each receive percepts and take actions. These agents base its actions on those percepts as dictated by its internal set of rules and processes. The process of perception and action can be seen as a cyclic process between the agent and the environment, especially when all the other agents are viewed as part of the environment instead of actors within it. This cyclic relationship is shown in Figure 1.

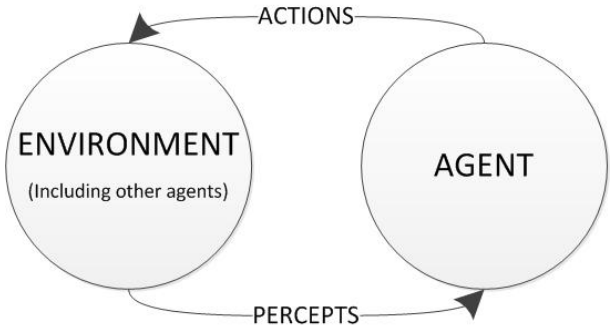

Figure 1. A General Agent Model

If we treat the other agents as part of the environment then the communications that are received from those agents

can be treated in exactly the same as percepts from the environment. Looking at it in this way, we see how a general model of agent communication should look. Figure 2 shows a communication that is passed on from one agent to another and then repeated on to another agent, i.e., a "telephone game."

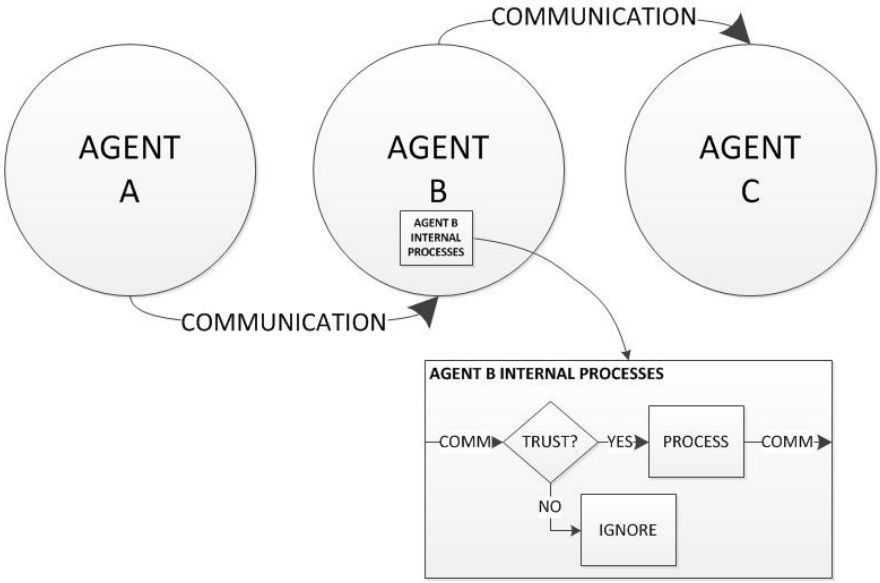

Figure 2. General Trust/Communication Model

In the simple model above, the overall communications process is complicated by the existence of many similar agents operating simultaneously within the same environment. In the example of the telephone game, each agent in the line of communication must accurately receive and then retransmit the information. As soon as one agent in this link fails, the information is changed. The evolution of the message driven by individual mistakes can lead to hilarious results, which is why the telephone game is so much fun to play. In contrast to the telephone game, real societies have lines of

communication that are not linear. Real communications within social networks transmit by taking multiple routes and therefore it is likely that important pieces of information that transmit through the entire social network will be received and retransmitted many times over by each agent. Despite this redundancy of communication, the particular processes that the individual takes to receive, process and retransmit communications are the foundation of information flow in the network. In models such as CG, the flow of information in the social network is the prime mover for agent belief revision, and therefore understanding the communication process is of vital importance. The following section contains an in depth introduction to the CG model.

2. The Cultural Geography Model

The Cultural Geography model is a discrete event simulation developed in Simkit that comprises a small society of agents that are seeking basic commodities for living such as water, fuel and food. As these agents acquire these needed items, they can experience shortages or long queues that can influence its beliefs and therefore can drive some of its actions. In particular, shortages in commodities may drive an agent to discuss its views on the situation with its neighbors. As a parallel to the real world, if there were shortages of gasoline, and high prices and long lines where gasoline was available, this would certain be a center of many conversations. In addition to dislike of the situation, people are also likely to express dissatisfaction for the current administration and how it might be their fault for the current shortage. In addition

to commodities, the agents can also witness events such as a terrorist attack that can also affect its beliefs and become a topic of conversation.

The agents each have a belief structure encoded as a Bayesian network that defines the agent's issue stances. The point of this model is to monitor issue stances, such as positive feelings toward coalition forces, or satisfaction in the government. As the simulation progresses the flow of commodities, or certain events will have an impact on the issue stances of the population and these effects can be tracked and analyzed.

Trust in this model falls primarily into the area of inter-agent communications. Simplifying this a bit, we can say that the agents will communicate with each other, update its beliefs, then, based on the updated beliefs take some actions. The choice of who to trust guides the agent in updating its beliefs, which in turn guides the actions of the agents and therefore will have a direct impact on the happiness of the agent.

The implementation of trust in CG will have an effect on all aspects of the simulation. In order to weed out non-trust phenomena and really see the effects of trust, we will be allowing the agents to interact in trust based economic games, such as the public commodity game.

In parallel with the development of this algorithm, the CG model underwent a major overhaul with the addition of a cognitive architecture. The agents utilize a working memory that can take a sequence of percepts, limited to a specific number, typically 7 percepts simultaneously. These percepts lead the agent to form a cognitive determination of its

situation based on preplanned characteristics. These characteristics include things like basic human needs and potential opportunities. It is from this situation formation that the agents can determine the most appropriate action to take.

The trust algorithm being developed within this work is based on RL, which relies heavily on the identification of the particular state the agent finds itself in. In the case of inter-agent communications, the state could be as simple as the sender of information or could be much more complicated. For example, the state could be a combination of the disposition of the sender, the sender's name, the subject, the disposition of the receiver, and more. Unfortunately, the more complicated the state space becomes in RL, the longer it takes the algorithm to converge to an optimum. There will be more discussion of the determination of states in a later section.

In addition to knowing what state the agent is in, an RL algorithm must also have a method for mapping certain states and actions to rewards. The CG model contains a cognitive self-appraisal that can be used as a reward signal. The drawback here is that this method of reward does not specifically address the issue of trust, only the agent's overall well-being. Rewards are easily defined for games where the reward structures are built directly into the game's rules. Before returning to the goal of defining trust, there is first an introduction to the trust based games that these agents will play to test the trust algorithm.

3. The Public Commodity and Economic Games of Trust

The Public Commodity Game (sometimes referred to as the Public Trust or Public Goods game) is a very common test bed within the Game Theory and Economics communities. The rules of the game are simple, although several variants do exist. This section will focus on the particular variant used in development of this trust algorithm.

The PC game starts with a group of players. In each round, every player is given an amount of income with which to play, 100 units, for example. The player will then decide a portion of that money, ranging from no contribution to full contribution, to put into a public commodity. The public commodity is meant to represent some kind of public good that benefits from cooperation en-masse, such as civil services (fire, police, hospitals, etc.).

Other variants of the public commodity game include versions where the play must be either all or nothing, giving us two broad categories of players, contributors and defectors. Although total defection is not allowed in this model, social defection is fundamental to the development of a meaningful definition of trust. Because of this, the term defector will not be used for a player that completely opts out but for one who is taking advantage of the group by merely minimizing his contribution.

After each player has contributed his selected amount in the blind, the total pot is then multiplied and then redistributed equally amongst all players regardless of its initial contribution. The following shows the steps played in each round of the PC game:

1. Each player receives 1.00 units of commodity (utility) each round.
2. Players have the chance to communicate once per round with close neighbors, including both sending and receiving information.
3. The information transmitted between players does not directly pertain to the PC game, rather communications were about beliefs that indirectly contributed to a players decision on how much to contribute.
4. The public commodity is collected once per round, multiplied by 3.0 and redistributed.
5. Players are not allowed to know the contributions of the other players, nor did they explicitly know how many players are involved in the game.

In this version, the payout to each player is private (i.e., no communications regarding payouts). The agent's contribution strategies, therefore, will be based on trust of the other agents in general, not based on trust of declared contributions.

Looking at the PC game from a theoretic standpoint, we see that if the pot is multiplied by any factor greater than 1.0, it is clear that the most mutually beneficial strategy would be for all players to commit all of its income to the pot. This situation is an unstable state because any defector from this mutually beneficial strategy will individually benefit from opting out of a contribution. In the case of a large number of players, each player will receive nearly the same public commodity payout, except the

defector in that without having had to pay into the pot he is now richer than the others by that contribution amount. The defector now has no monetary motivation to reinstate his contribution. As the game progresses, more and more players begin to opt out and eventually the pot lowers in value. As the pot lowers in value, many additional players will begin to opt out as they perceive the game to not be worth as much as they feel it should be based on past experience.

As a game of pure strategy, the only stable equilibrium will be when all players opt out. Look at the case where in a given round, there were no contributions from any player. In subsequent rounds, any single player that choses to begin contributing will find his contribution returned substantially reduced as it is divided up amongst all the other players. This negative reinforcement will urge the player to once again defect from the game, returning the stable equilibrium of zero contributions.

When PC games are used on real players in the laboratory, there are consistently higher payoff levels than what would be predicted within game theory (Hoffman, McCabe & Smith, 1998). Based on the results seen in experimental economics, it is expected that the average dividend will approach a small but non zero value.

Take, for example, a large social network of players, such that there is a sufficiently large population of players so that direct communication between all players is impossible. In the language of network theory, this social network will have a relatively low ratio of average closeness-centrality to the total number of nodes in the network. As an example, take a city of 250,000 people — any

one person is likely to have direct communication with between 130 and 250 people, but typically cited is 150 as a good representative value (Dunbar, 1996). There will also be a high degree of overlap; in other words, given any two persons in close contact with each other, their combined group of close neighbors will not be closer to 150 than 300. In other words, many of the 150 friends of each of these individuals will be the same as the other. From these relationships comes the age-old axiom of "Six Degrees of Separation," where even in a large population nearly all individuals are connected by six or fewer indirect relationships. When this kind of group is examined within the PC scenario, it is expected to initially find a trend toward little or no contribution, but then it is expected to see close groups of individuals changing their contributions nearly at the same time. What one should expect from an accurate model of agents playing a PC game is that trust groups make decisions nearly at the same time and as a whole change their contribution to the game.

4. Getting Back to the Issue of Trust

Trust is more than a prediction of an agent's actions based on their past actions. In the PC game, an agent will develop a trust of the other agents based mostly on the past performance of the public commodity in general, rather than the specific actions of any one player. In other trust games, such as the prisoner's dilemma (PD), the trust decision is likely not entirely based on reputation. In the PD game, two players make a decision, usually pertaining to the confession of a crime, in which betrayal of their partner could stand to bring them reward. Additionally, if

both players opt out of betraying the others then they will receive some smaller reward. For example, if neither confess, they both get one month in jail, if one confesses and the other does not, the confessor may be set free and the other spends a year in prison. And, if they both confess, they both spend a year in prison. In particular, what is being asked is if the opposing player(s) will adhere to some unspoken rules or a code of conduct. These rules depend highly on the relationship of the players as well as the value of the objects involved. As an example, look at the story "Button, Button," by Richard Matheson, in which a strange man gives an unsuspecting person a box with a big red button. If the button is pressed, the owner of the box receives a million dollar prize, but somewhere a stranger that they could not possibly ever know will fall dead. This is the kind of situation wherein the social contract would dictate that the contestant should not press the button. This concept of a social contract is central to the notion of trust (Mistzal, 1996).

When human beings in modern society view each other and make trustworthiness evaluations, there is clearly more than reputation involved in the decision process. There will always be a constant baseline trust that exists between individuals. This baseline will include such things as the potential trustees position in society, such as doctors and police officers, who garner automatic trust amongst most people. The baseline will also include unconscious biases such as racial biases that make us inherently trust people who appear to be similar to ourselves (Stanley et al., 2011).

In summary, what we have is a concept of trust that can be used to help predict the actions of other agents (such as their possible contribution to a public pot) and help to guide the actions of the trustor. This trust will be based on three key elements, reputation, characteristics and position of the trustee. These characteristics lead to the definition of trust being used in this research.

B. MACHINE LEARNING

As the goal for this project is to model trust in a small group of cognitive social agents in a computer simulation, it is necessary to dive into the subject of machine learning (ML) and how it played a central role in the development of trust algorithms.

1. Reinforcement Learning

The particular brand of machine learning utilized in this project is reinforcement learning (RL). RL is an appealing approach in that the very idea of reputation is built right into it. Additionally, RL has been shown to be a fantastic tool for solving problems and captures many of the reinforcing phenomena that occur naturally in the human brain.

The basic idea of RL is that agents will seek to select actions within their environment based on their experience and learn from those selections. Based on the permissiveness of the environment, agents are eligible to receive percepts from the environment that inform them on the state of the environment at a given point in time. The basic elements of reinforcement learning are: a policy that maps states to

actions; a reward function that maps a state of the environment to a reward; a value function that maps states to long term value given experience; and an optional model of the environment. The policy provides a set of actions that are available in a given state of the environment; the agents leverage its prior knowledge of the environment, informed by the value function, to determine which action will provide the greatest reward, as defined by the modeler. Agents must strike a balance between exploration, behavior to explore the reward outcomes of state action pairs that have not been tried, and exploitation, behavior that takes advantage of prior knowledge to maximize short term rewards, in order to avoid converging to local minima (Sutton & Barto, 1998). The ability to control this balance makes reinforcement learning an attractive approach for representing human behavior. The reinforcement learning technique used in this work is Q-learning in conjunction with a soft-max function (the Boltzmann distribution).

2. Q-Learning Using Boltzmann Selection

The basic reinforcement equation of Q-Learning is as follows in Figure 3:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a} Q(s_{t+1}, a_t) - Q(s_t, a_t) \right]$$

Figure 3. Q-Learning Reinforcement Equation.

Q-learning falls into a class of model free reinforcement learning methods that have the property that the learned action-value function, Q, approximates the optimal action-value function, Q*, requiring only that all

state action pairs be updated as visited (Sutton Barto 1998). For each state action pair, (s,a), the Q-learning function updates the current estimate based on new information received from recent actions, r, and discounted long term reward. In general, an action is selected from a given state, the reward outcome is observed and recorded, and the value function updated. The value associated with each action is used during each visit to a particular state to determine which action should be chosen using the Boltzmann distribution, shown in Figure 4 below.

$$\frac{e^{Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}}$$

Figure 4. Boltzmann (Soft-Max) Selection Probability

The Boltzmann distribution uses the temperature term, τ, to control the level of exploration and exploitation. A high temperature translates into exploratory behavior, a low temperature results in greedy behavior. Although the algorithms presented here utilize constant temperatures, it has been shown that temperature scheduling is far superior to constant temperature methods (Ozcan, 2011).

A good example of reinforcement learning is an agent that plays the game, "N-Armed Bandit." In this game, the agent is faced with a slot machine with n arms, where n is greater than 1 and can be as large as necessary to serve the purpose of the experiment: in this case, choosing n to be 2. The agent is made aware that the payout probabilities are fixed and necessarily unequal, although the RL algorithms do not require this information to function properly. In other

words, each arm pays out at a different fixed rate. It is easy to see how most humans would approach this problem, and this is typically how the RL algorithms handles it as well. Most would pull each lever a fixed number of times, say ten pulls each, and record the results. If lever A hits 2 times out of ten, and lever B hit out 5 times out of ten, most human players would favor lever B in the next round of play. The amount the player would favor depends on his particular attitude and is something that can be controlled within RL.

So, we say that our player will play 15 times on lever B and only 5 times on A, but then something unusual happens -lever B only hits 2 times and lever A now hits 4 out of the 5 times. Most human players would put this one back to step 1 and play equal amounts the next few rounds to settle once and for all which lever is better. For RL, this added reward from lever A adds to its likelihood of being chosen in a very precise, although probabilistic, way as discussed previously. The advantage here is that the agent utilizing proper RL algorithms will find the optimum, where a purely greedy algorithm may not. In fact, in many situations it is easy for a simple algorithm to find a local optimum that is not the global optimum. RL algorithms, when properly set up, are very good at not getting foiled into local optimums and most often find the true global optimum. More importantly, it is possible to configure these algorithms so that its search for the global optimum is very similar to human behavior. RL is particularly well suited for dynamic repeated environments wherein measuring past actions against unexplored opportunities yields the best overall results (Dutt, 2011).

As previously discussed, the basis of trust likely has evolutionary motivations and is driven at its core by simple concepts such as reciprocation and reputation. RL provides a fantastic platform for designing a trust algorithm, in that its fundamental processes are perfectly suited to model these concepts and therefore it is the best choice for the basis of a trust algorithm. RL does, however, highly rely on its inputs in order to function properly. In particular, it is up to the designer in RL to define what reward signal should accompany given states. It is also up to the designer to define how percepts will combine to form a hashable state and possible courses of action that the algorithm can use to determine the action-value function. An overview of this is shown in Figure 5 below.

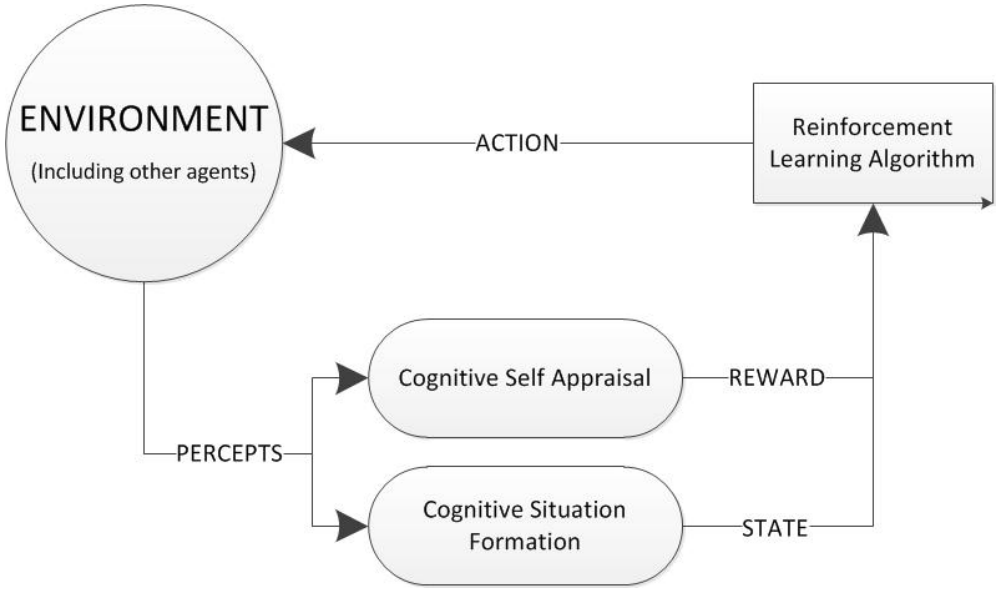

Figure 5. Overview of the Q-Learning Cycle

There is a further discussion of reward signals and state formation in Chapter III. For now, it suffices to say

that, even though RL can be demonstrated to be a useful core algorithm for trust modeling, there is difficulty in implementing this model in that for each situation we choose to implement it, we must determine complete mappings for reward signaling and state formation. Fortunately, the CG model has a built in cognitive self-appraisal that can be fed into RL as a reward signal.

C. SOCIAL NETWORK ANALYSIS

In order to see the effects of trust on a society of agents, it is necessary to look at the society in a network format. Social networks are interconnected groups of individuals represented by nodes on a network graph and connected by edges that represent social relationships primarily based on communication. Within the social network, there are a few attributes that can be used to evaluate the overall character of the network. These are defined below:

Degree Centrality is a measure of the direct connections of a node. The average degree centrality tells us how connected the individuals are in this social network. This factor depends on the particular social network that is being looked at. For example, in a small village of 100 people or fewer, chances are that each individual will have connections to nearly all the others as this is a very small tight-knit community. In a large city, a person will likely have between 130 and 200 connections, with 7 to 12 close personal ties. In the last case, the degree centrality that will be modeled into the network depends on what strength of social ties wished to be modeled. It may be that only close

personal ties are included in the model, whereas simple acquaintances are not. Degree centrality formulae are given in Figure 6.

n *is the number of nodes in the network graph*
N *is the set of all nodes in the network graph*
a *is a node in the network graph*
c_a *is the number of connections from node* a

$$Degree - Centrality_a = \frac{c_a}{n-1}$$

$$Average - Degree - Centrality = \frac{1}{n}\sum_a \frac{c_a}{n-1}$$

Figure 6. Degree Centrality Formulae.

Betweenness Centrality is a measure of a node's importance to information flow within a social network. In other words, given a node, sometimes called a broker, which is the only connection between two other nodes that are each hubs with high degree centrality, then any information that flows between the two sub networks formed by the two hubs must pass through the broker node. The broker node in this case has a high betweenness centrality. Another way to think of betweenness centrality is as a measure of the fraction of shortest paths that pass through the node in question. Betweenness centrality formulae are given in Figure 7.

$$n \text{ is the number of nodes in the network graph}$$
$$N \text{ is the set of all nodes in the network graph}$$
$$a \text{ is a node in the network graph}$$
$$\sigma(s,t) \text{is the number of shortest paths between } s \text{ and } t$$
$$\sigma(s,t|a) \text{is the number of shortest paths from } s \text{ to } t \text{ through } a$$

$$Betweenness - Centrality_a = \sum_{s,t \in N} \frac{\sigma(s,t|a)}{\sigma(s,t)}$$

$$Average - Betweenness - Centrality = \frac{1}{n} \sum_{a \in N} \sum_{s,t \in N} \frac{\sigma(s,t|a)}{\sigma(s,t)}$$

Figure 7. Betweenness-Centrality Formulae

Closeness Centrality is a measure of how close a node is in a social network to all the other nodes by both direct and indirect connections. The average closeness centrality of the network tells us how connected the network is where the maximum possible case is a social network in which all nodes are connected to all others. Closeness centrality formulae are given in Figure 8 below.

$$n \text{ is the number of nodes in the network graph}$$
$$N \text{ is the set of all nodes in the network graph}$$
$$a \text{ is a node in the network graph}$$
$$d(s,t) \text{is the distance between } s \text{ and } t$$

$$Closeness - Centrality_a = \left[\frac{1}{n}\sum_{t \in N} d(a,t)\right]^{-1}$$

$$Average - Closeness - Centrality = \frac{1}{n}\sum_{s \in N}\left[\frac{1}{n}\sum_{t \in N} d(s,t)\right]^{-1}$$

Figure 8. Closeness-Centrality Formulae.

Clique is a maximal group within a network in which all the nodes in the group are interconnected. The formation of cliques is an expected part of most social networks (Hallinan & Smith, 1989).

Every social network is different and therefore we will not find that there are any single values that we would expect for any of these characteristics that apply to all social networks.

III. THE TRUST ALGORITHM IN DETAIL

As discussed in previous chapters, we are working with a biologically inspired model of trust in which the effects of reputation and reciprocity are central. This trust algorithm was developed first as a simplified model based loosely on Cultural Geography. The simplified version is a social network simulation that was developed in the Python programming language using the NetworkX network analysis tool pack from Los Alamos National Laboratory. The next sections are a detailed description of how the algorithm was developed as well as the results of test and evaluation.

A. REINFORCEMENT LEARNING AS A TOOL TO DRIVE DYNAMIC SOCIAL NETWORKS

One of the archetypal game scenarios for which RL is perfectly suited is the "N-Armed Bandit" problem, as discussed in the last chapter. In this game, the agent is faced with a series of actions that are state independent, which means that the payouts do not depend on the history of actions of the agent, or the state of the environment. That is to say, the environment in which the agent operates is unchanging. In the case of the n-armed slot machine, this means that the probability of hitting a jackpot never changes, whether it is the first or the millionth pull, whether the jackpot has just hit or has never hit. Therefore, the agent simply must choose which arm to pull. Thinking about how a human being would determine which arm is best, the agent can more accurately be said to select a strategy for a series of pulls. This strategy might be something like, first pull each arm 10 times and see which

one pays out better. Then, based on the payouts, start favoring one arm over the other. After a long period of play, the agent will become fairly confident which arm is better and will play that arm nearly exclusively. In simplified form, this just means the agent will choose what percentage of a series of pulls that go to each arm. The percentage of pulls can be called the emphasis that the agent places on each of the arms.

If we think of slot machine arms as connections in a social network, and the payoff as the benefit of spending time with agents on the other side of those connections, we start to see how this n-armed bandit problem can be generalized and used in social networks. Essentially, the agent has a choice of his nearest neighbors in the social network that he can choose to place some emphasis. It may be instructive to think of this emphasis as a fraction of the day (or week, or whatever time period is relevant) that he would like to spend with this other person. Wanting to spend time with someone is not enough to garner a reward; the other person must also want to spend time with you. The game now becomes a multiplayer integrated n-armed bandit game in which each of the agents will decide which of its k nearest neighbors it chooses to spend its time with, and specifically how much time to spend with each. If two neighbors both choose to spend time with each other, then they will both receive a positive reward from this interaction. When one neighbor wishes to spend time and the other does not reciprocate, then little or no reward is given. It is left for future work to determine if and how to utilize the idea of negative rewards in this situation.

This model has been implemented and turned into a turn-based simulation in which the agents and their relationships are represented by a single network graph with the agents as the nodes and their social relationships as the non-directional edges, weighted by the mutual value of the relationship. The value of the relationship is dynamic and calculated once each round. It will be a combination of a base static value and a dynamic value that is controlled by the agents. The static is based entirely on the concept of homophily (E_H) (McPherson Smith-Lovin, & Cook 2001), in other words, that demographically similar persons associate more frequently. The homophily calculation is a simple Euclidean distance from the agents demographic characteristics. In this simulation the agents have multiple demographic dimensions, including age, sex, race and others which have fixed constant values. When two agents share a common demographic value, a demographic character score of 1.0 is inserted into the Euclidean distance formula and 0.0 if they are different. The demographic character scores are squared, summed and the square root is taken. This value is then divided by a normalization factor to make the maximum homophily value 1.0. For example, looking at the case of 3 demographic dimensions, we have two agents that have the following demographic characters, Agent1 = [Caucasian, 24–34 years, Male] and Agent2 = [Caucasian, 64+ years, Male]. For these agents the homophily calculation would be as in Figure 9 below (where E_H is the homophily value and δ_n are the demographic character scores for demographic characteristic n):

$$E_H = \frac{\sqrt{\delta_1^2 + \delta_2^2 + \delta_3^2}}{\sqrt{3}}$$

$$E_H = \frac{\sqrt{1.0^2 + 0.0^2 + 1.0^2}}{\sqrt{3}}$$

$$E_H = 0.82$$

Figure 9. Sample Homophily Calculation

The value calculated above will be the baseline emphasis between the two agents for the entire simulation run. It is left for future work to determine a viable means of altering the baseline homophily.

The dynamic portion of the emphasis is completely under the control of the agents involved ($E_{A \to B}$). In the case of k completely connected agents, each agent has k-1 choices of neighbors with whom to spend time with. Based on the emphasis the agents places on each relationship, it will receive some unknown reward from time spent with the other agents. For every simulation round, each agent will choose to increase, decrease or maintain its contribution to its relationships with the other agents. This contribution can be viewed as a fraction of time spent with the others in that it is represented as a floating point number from 0.0 to 1.0 and such that the sum of all these components (i.e., the sum of all edge weights leaving the node representing the agent) always sums to 1.0. At the end of each turn, the agent is rewarded based on the strength of its relationships, which only hold value if the emphasis on the

relationship is reciprocal with the other agent. This reward takes the form in Figure 10 below:

$$Reward = E_H \min(E_{A\to B}, E_{B\to A})$$

Figure 10. Basic Reward Calculation

In the equation above, the $E_{A\to B}$ terms are the emphasis value that A places on the relationship with B. Additionally, recall that the E_H term is a homophily derived base emphasis between agents A and B that is based on the Euclidean distance of their demographic characteristics. This equation uses the minimum of the variable contributions from each agent. In this way, it can more accurately be said that the variable portion is the fraction of time that the agent would like to spend with the other agent, but if this sentiment is not reciprocated, no reward is earned. In fact, since the total emphasis is constant in this simulation, placing emphasis on an agent that does not reciprocate comes with an opportunity cost that can be seen as a form of punishment.

The result of this basic model is the development of a simple dynamic social network. The network tends to become highly centralized around 1 or 2 agents. In particular, in runs consisting of 50 agents, the final network graph consisted of nearly every agent with a strong connection to a single central agent with nearly no other connections present (Figure 11).

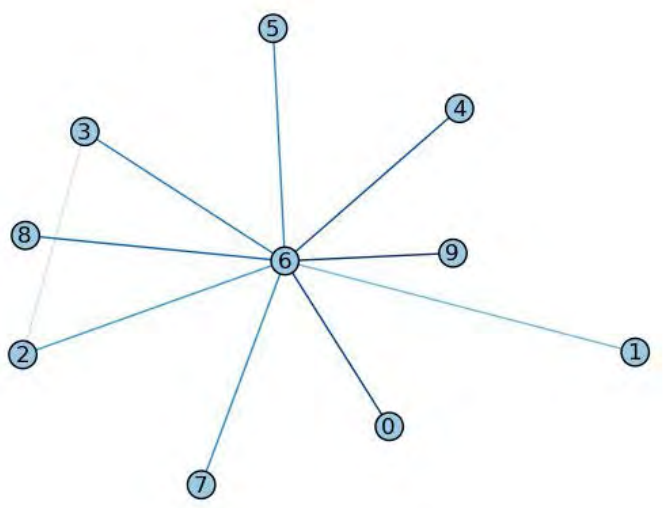

Figure 11. Highly Centralized Social Network

This centralization of the social network is explained by the fact that the algorithm is based solely on one-on-one interaction and neglects any effects due to larger groups and also due to the fact that the homophily calculation is a Euclidean distance formulation. One might expect that the best the network can do is to pair up into closest neighbors. In other words, each agent finds the one other agent it shares the closest demographic characteristics to and places total emphasis on this one relationship. Given the way that E_H is calculated, this will likely never be the case. This is mostly due to the fact that the RL algorithm is not trying to maximize utility for the entire network, which can easily be done by forcing this kind of pairing. Rather, the RL algorithm, or better yet, the RL algorithms are working on one direction of one edge at a time, independently of all others. So, instead of optimizing the network it is competitively optimizing all the connections

simultaneously. Optimization can also come in the form of an opportunity cost, which is to say that a relationship that is emphasized but not reciprocated is the same as taking a penalty, which is equivalent to what the agent could have received from emphasizing some other relationship that would have been reciprocated and thus produced its relationships both by emphasizing some and ignoring others.

Due to the fact that the homophily calculation used in this case is a Euclidean distance formula, we can say that each demographic characteristic is like a linearly independent value that can be treated like a coordinate access in a Cartesian coordinate system. In this case, the agent's particular demographic characteristics can be seen as coordinates in a 5 dimensional space (due to the fact that there are 5 demographic characteristics). Each agent can then be represented as a single point in this space. As the network evolves and the agents optimize their network connections, the agents will effectively try to minimize the distance they must cover in the 5 dimensional demographic space in order to make their best relationships. The group of agents, all represented by points in a 5 dimensional space, will work most smoothly together by forming relationships closest to the geometric center of all the points. Whichever agent, or agents, occupies the point closest to this center will inevitably end up being the central agent, depicted as agent #6 in the simplified graph above. Any deviation from this will clearly be less than optimal for nearly every agent involved (save the one exception of the central agent).

In order to solve the problem of excessive network centralization, it was necessary to consider the effects of cliques on the network. There was a need to give the RL algorithms some benefit to forming a network that allowed the formation of cliques like the one shown in Figure 12.

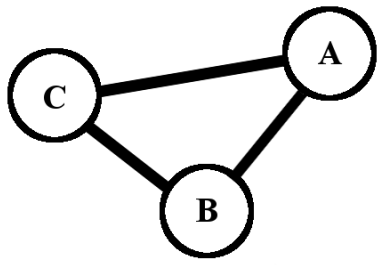

Figure 12. An Example of a clique of three agents or 3-clique

In the example above, there are three agents that form a tightly bound clique. In other words, the three agents all place equal emphasis on all the relationships. If the emphasis is equivalent to an amount of time desired to spent with the target agent, then in the case above, there should be extra time allotted simply due to the fact that it is likely that agent A will spend time with agents B and C simultaneously. In other words, if agent A only has 1.0 units of emphasis to dole out, the time spent mutually with B and C should not count twice. However, the secondary reward from spending time with B and C together will not be the same as if the agent could spent equal time with B and C separately. This simply suggests that time spent in a pair is ultimately more personally rewarding. This is obviously not always the case in every relationship, but as an average, it is likely to mimic actual social interactions quite well.

The second order reward factors are based on the same reward function as used in the first order above. In this case, the reward is reduced by dividing by a distribution factor and is subsequently squared in order to emphasize the importance of first order relationships. For the case of agents A B and C above, the additional reward looks as in Figure 13 below:

$$2nd - Reward = \min\left(E_H \min(E_{A\to C}, E_{C\to A}), E_H \min(E_{B\to C}, E_{C\to B})\right)/D$$

Figure 13. Secondary Rewards in the Social Network

Once the second order terms are added similar network properties to what we would expect to see in real social situations emerge; namely subdivisions into cliques, pairings and the exclusion of certain individuals from these cliques (Wellman, Carrington, & Hall, 1988). Figure 14 below shows a less centralized network than the previous example. The effect is much more dramatic in larger social networks but significantly more difficult to visualize and in printed form, therefore only a small social network is shown.

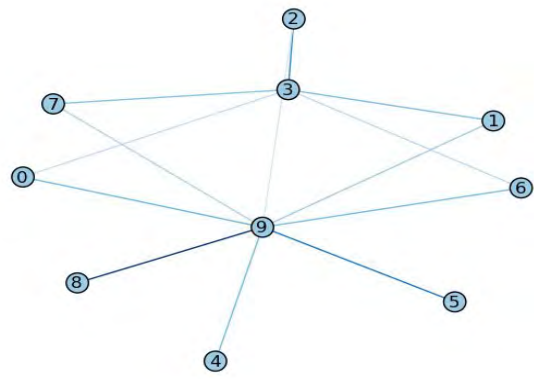

Figure 14. An Example of a Less Centralized Social Network

We find that the parameter D is highly influential on the closeness-centrality of the social network in such a way that D can be used to tune this factor to fit with the network being modeled. The distribution factor was varied and showed a fairly steep "S" curve (Figure 15) that was centered between D = 14 to D = 24. There are several widely varying sources on what a real human social network should look like in terms of closeness centrality that range from 0.20 to 0.60 (Krebs, 2002; Dekker, 2008). Therefore, for the purposes of the remainder of this initial experimentation D = 18.4 is used in order to target the fractional closeness centrality to around 0.30. The exact nature of these values is irrelevant for this initial model and only serves as a baseline for further work.

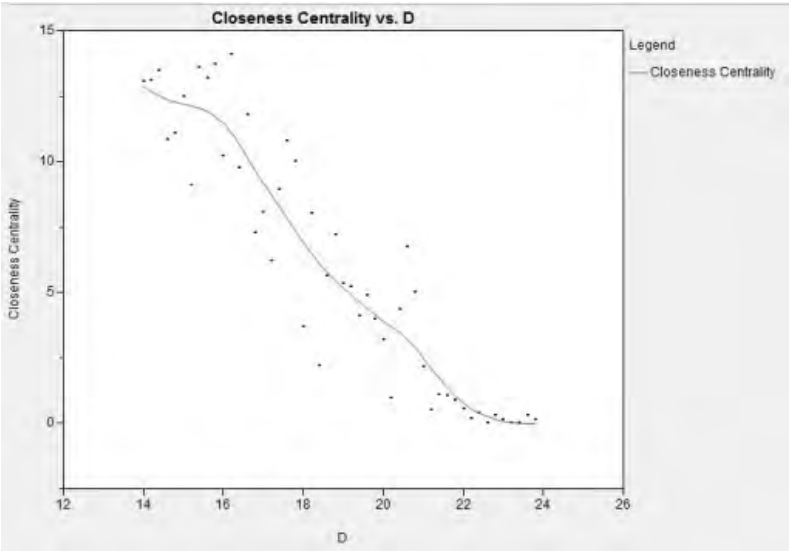

Figure 15. Average Closeness-Centrality Versus Distribution Factor

It is obvious that these features are not intended to actually model the internal and external processes that form

real human social networks; rather, it is just a starting block. This dynamic social network is a simple stage that roughly mimics the dynamics seen in real social networks and which allowed the development of a trust algorithm within it.

B. ADDING COMMUNICATIONS AND TRUST TO THE SOCIAL NETWORK

Now that there was a functioning social network on which to operate, it was necessary to implement a rudimentary belief structure for the agents. Each of these agents had a finite set of beliefs, five in this case, represented by a single floating point number from 0.0 to 1.0. These beliefs combine in a simple linear combination (i.e., each value multiplied by a weighting factor from -1.0 to 1.0) to provide a single issue-stance, also as a floating point that ranged in value from 0.0 to 1.0. For these purposes, the interpretation of these numbers to actual real world beliefs is irrelevant; it only mattered that the agents had some kind of belief structure that is roughly heterogeneous across the population. The social network for this simulation is allowed to evolve, initially for 1000 rounds for a simulation of 15 agents. Once the network was stabilized, the agents began choosing topics of conversation. This choice was based on a probabilistic Boltzmann distribution. The agents discussed this topic with its k nearest neighbors with the caveat that any neighbor above the communication-threshold, set initially to 0.90, will automatically receive a communication and likewise any neighbor below the ignore-threshold, initially set to 0.10, will never receive one.

Initially, the communications consisted of each agent telling its closest neighbors exactly what its value was on a selected belief. At this point, the receiving agent made a trust evaluation of the information received in order to determine the best action. The receiving agents used a reinforcement learning algorithm to determine whether or not belief revision was merited. The state space that the RL algorithm operated in is a pairing of subject and sender for the communication. As an example, if agent A tells agent B that he feels belief 5 has a value of 0.75, the receiver agent, B will use "Agent A discussing Belief 5" as the unique identifier of this state. As will be seen in the next chapter, for early implementation into CG it will be necessary to confine the state space to just the sender of the information. Each state in the state space has 2 corresponding actions, "Trust" or "Do not Trust." For information received and trusted, the agent will update its beliefs according to the new information and this will define their future actions. The method of belief revision used was to simply shift the agents own belief in the subject of the communication 1/10 of the way to the value stated by the other agent.

In order to utilize reinforcement learning in this way, it is necessary to define some concept of a reward that the agent will receive based on its beliefs and therefore directly related to its trust and belief revision mechanisms. Our inspiration for a reward model was the Public Commodity (PC) game from experimental economics. As discussed in the previous chapter, within the PC game, each agent has an option to contribute some fraction of its income (1.0 per round) to a public pot of money each round.

Following the round, the money in the pot is multiplied by some amount (in this case 3.0) and then redistributed to each agent regardless of contribution.

In the current model, agents are given 1.0 possible units to play such that an agent that contributes nothing is guaranteed a reward of at least 1.0 for opting out and an unknown reward ranging from nearly 0.0 to 3.0 for full contribution. Game theory tells us that without cooperation the expected equilibrium for rational players would be exactly 0.0 contributions from all agents; in other words, all agents take the guaranteed 1.0 and opt-out of the public commodity all together. As discussed in the previous chapter, we expect that in reality some people would always contribute at least some small amount irrespective of their losses. In order to simulate this, we have developed "Faith in the Public Commodity" as the issue-stance and is used to directly control the level of their contribution to the public commodity. During each simulation round, agents communicate with one another and attempt to bring other agents closer to their beliefs.

Now that there is a concrete idea of reward in this model, it is possible to begin applying a simple model of belief revision. The agents will communicate and the information will either be trusted or distrusted. Trusted information will cause the recipient to alter its beliefs some fraction of the distance between its starting belief and the value of the belief being told to it. The effect of this style of communication and belief revision will result in a local optimum of play by all players. Effectively, all the beliefs will average out until all players believe the

same things, then no further belief revision is possible and the game will become static. This is highly unrealistic for two major reasons: 1) Changing a belief is not free, there are internal psychological costs to changing beliefs (especially those that are deeply ingrained in the believer). 2) Beliefs are not always constant. One of the key features of CG, for which this algorithm is intended, is that events such as food shortage or terrorist attacks can have an effect on beliefs.

In order to model a penalty for straying beliefs away from the agent's normal beliefs, the following penalty (Figure 16) is assessed to the reward signal received by the agents each round.

$$NormPenalty = e^{F \cdot BeliefVariance}$$

Figure 16. Penalty Assessed to the Reward Signal from Straying from the Agents Normal Beliefs.

In the above, the Belief Variance is a simple Euclidean distance measure from the agents current beliefs to what it started with at the beginning of the simulation. The norm penalty is applied to the reward signal, by reducing the net dividend the agent can receive from the pubic commodity as shown in Figure 17 below.

$$Net\ Dividend = (1.0 - Contribution) + 3.0\ x\ Contribution - NormPenalty$$

$$Adjusted\ Net\ Dividend_A = \left[(1.0 - C_A) + \frac{3.0}{K} \sum_{n \in agents} C_n\right] - NormPenalty$$

Figure 17. Application of Belief Revision Penalty

In the above equation, C_A represents the contribution to the public pot from agent A and K represents the number of agents in the social network. The following summarizes the way this PC game is carried out within this social network:

1. Each agent first decides whether to raise, lower or maintain its social emphasis on each of the agents they are connected to.
2. Each agent will then conduct communications with a selection of its closer friends that consist of a basic statement about their value of a specific belief. (e.g., Agent A communicates to Agent B that it feels 0.72 about belief # 1).
3. Each agent that receives a communication will then choose to either totally trust the received information or totally distrust it.
4. Each agent that has chosen to trust a piece of received information will adjust that particular belief 1/10 of the way to the announced belief value in the communication.
5. After communications and belief revision have been processed, each agent will take the appropriate linear combination of its beliefs to produce its single issue stance, "Faith in the Public Commodity," which will be a single floating point number from 0.0 to 1.0.
6. Each agent will receive 1.0 units of income and from that contribute to the public pot in the

amount equal to the value of its issue stance on "Faith in the Public Commodity."

7. After all contributions have been collected the money in the public pot is multiplied by 3.0 and divided amongst each player equally as the net dividend.

8. Each agent will assess the norm penalty by taking the Euclidean distance of its 5 beliefs from the values it had at the start of the simulation and applying it to the equation shown previously.

9. Each agent will reduce its net dividend by the value of the norm penalty as shown in the equation in Figure 15 to produce the adjusted net dividend which will in turn be used as the reward signal for the agents RL algorithm for trust. (note: the trust RL algorithm is independent of the RL algorithm for network emphasis)

What is surprising is that when the factor F in the NormPenalty equation is varied, there is no marked difference in the outcome of the simulation from a purely statistical point of view. In other words, the average PC play, contributions and dividends do not change. What is seen, however, is an interesting structure of PC play over time, shown in Figure 18.

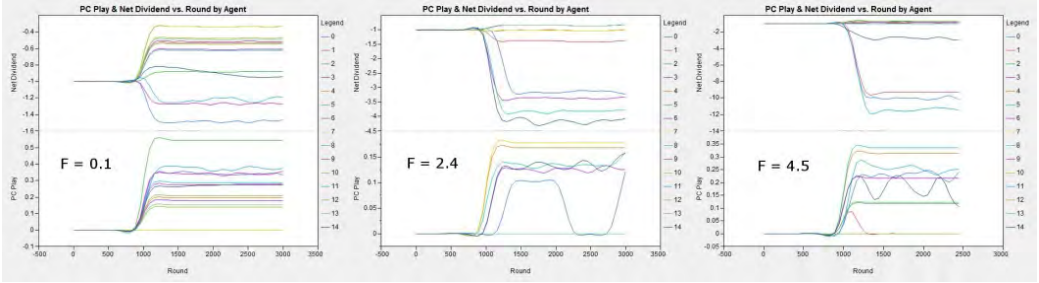

Figure 18. Graphs Showing Increasing Belief Revision Penalty

What is seen in the above is the higher the factor F becomes, the more unstable the Public Commodity game is. In other words, with a small norm penalty the agents will tend to find a stable equilibrium and remain there with fairly significant stability. As F is increased, the stability is decreased.

The intriguing thing is that this behavior appears to be similar to actual human interaction. For example, if we look at a society there is a sense of a norm although it will really change over time it will remain fairly constant over small enough time periods. In this society there will be people or factions that challenge the social norm causing brief unstable equilibrium away from the norm that seem to return to the social norm after some time. Often this can be seen as an individual or coalition breaking from the social norm. The concept of coalitions in economic game theory is well understood and is an expected outcome games such as this (Von, Neumann, & Morgenstern, 1944). The level at which coalition behavior takes place in societies can be a tunable parameter of this model. It should be pointed out however that this version of the trust algorithm utilizes a simple reward-penalty structure. Implementing this model in CG, which has its own reward structure built in, makes this

feature very difficult to reproduce. There will be more in depth discussion of this in the next chapter.

C. TUNING THE REINFORCEMENT LEARNING PARAMETERS

One of the issues in implementing RL algorithms is that there are a few parameters that are used to define how the algorithm functions. There was a detailed description of these parameters in Chapter II. To apply a RL algorithm to the CG model, there are a few things that must be taken into account. The most important thing is that this model is intended to run for relatively small groups of agents up to about 300, and meant to run for relatively short periods of time depending on the specifics of the scenario. These limitations are due to complexity and the limits of computing power available. There is no perfect theory that will identify the best parameter inputs for a given scenario. Additionally and more importantly there is now way to map human behavior directly to these input parameters. This is due to the fact that every person and every situation is vastly different. RL is not a perfect match for human problem solving and therefore our prime motivation in selecting inputs to the RL algorithm is speed of learning. We sought to find inputs that would yield global optimums in the minimum amount of time.

In order to optimize the inputs to the RL algorithm we developed a program that would allow a group of 100 agents to compete at the tasks identified above in the PC game. In particular, we left the input parameters as individual genes in a genetic algorithm. The population was allowed to play the PC game for 2000 rounds including a 300 round stabilization period. During play the agents were only

allowed to communicate about every 5 rounds in order to approximately mimic the communications that go on in the CG model. Each round the total utility of the agents was ranked and the bottom half of all agents lobotomized, in that their RL algorithms were stripped away. New RL algorithms for these agents were inserted as a genetic cross between two surviving parents selected at random using a Boltzmann selection method were the agent with higher utility has the better chance of breeding. In addition to this the agents had a 3% chance per gene of random mutation.

The following is an example of this methodology also shown graphically in Figure 19:

1. **Start of the simulation**: 30 agents are created with random values for their Lambda, Gamma, Default Utility and Temperature which are used to define the RL algorithm at the core of their trust and communications behaviors.
2. **Stabilization**: These agents will be allowed to randomly communicate for 300 rounds which will give the social network enough time to stabilize from its initially random values.
3. **PC Game Play**: Following the stabilization period the agents will continue communicating randomly, but will also be forced to play the PC game once per round for 1700 additional rounds.
4. **Ranking and Culling**: At the end of the PC game play, the agents are ranked according to their total score (utility) in the PC game. The bottom

50% of these agents are culled as they are poor performers and their genes are undesirable.

5. **Breeding**: The remaining agents are then selected 2 at a time for breeding. The selection is random and is weighted based on the agent's utility score from the PC play portion. One of the culled agents is regenerated with genes randomly selected from each parent weighted by each parents utility score. For each gene there is also a 3% chance that the gene will be from neither parent and will take on a new random value. This process is repeated until the agent list is repopulated.

6. **Repeat**: Steps 1–5 are considered 1 generation. For most processes very few generations are required, for this work 100 is used, just to be certain all the genes have reached stable values or have otherwise been shown to be unstable.

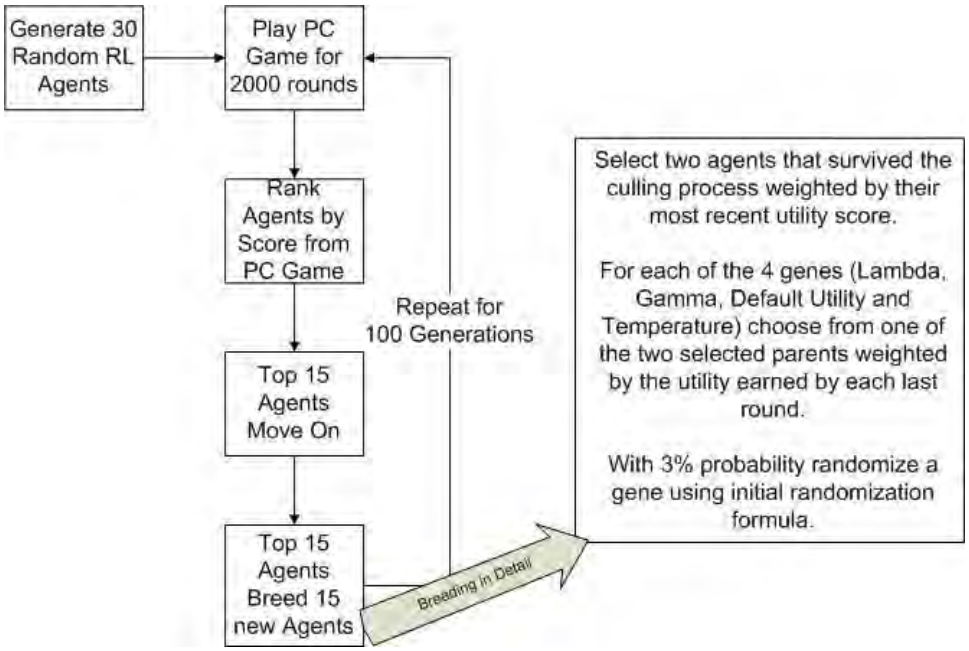

Figure 19. Overview of Using Genetic Algorithms to Breed Effective Social Network Agents

Allowing the agents to evolve for 100 generations we have found that some of our learning parameters form a pretty tight distribution, while others do not. Those that do not, indicate that they do not have high importance in making these agents fast learners. The results of the parameters that were significant are included in Figure 20 below.

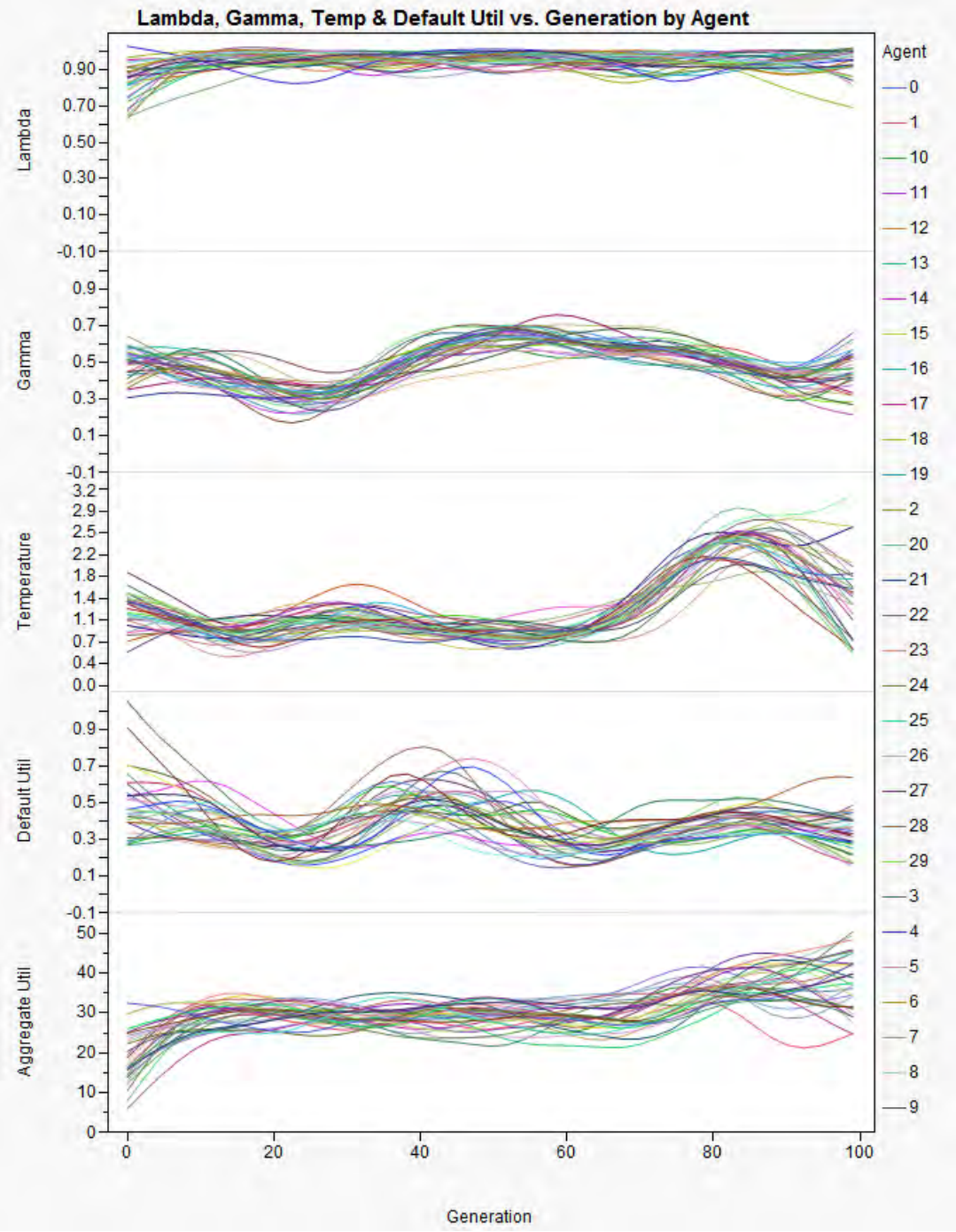

Figure 20. The Genes of the RL Algorithm by Generation

The first and most obvious result here is that the gene for Lambda or the learning rate in the model tends to be very high. This indicates that the agents do not need a

very long memory in order to be successful in this model and look primarily at the most recent information. Looking at the figures below, it can be seen how the Lambda gene evolved (Figure 21) very rapidly to an optimum within only a few generations.

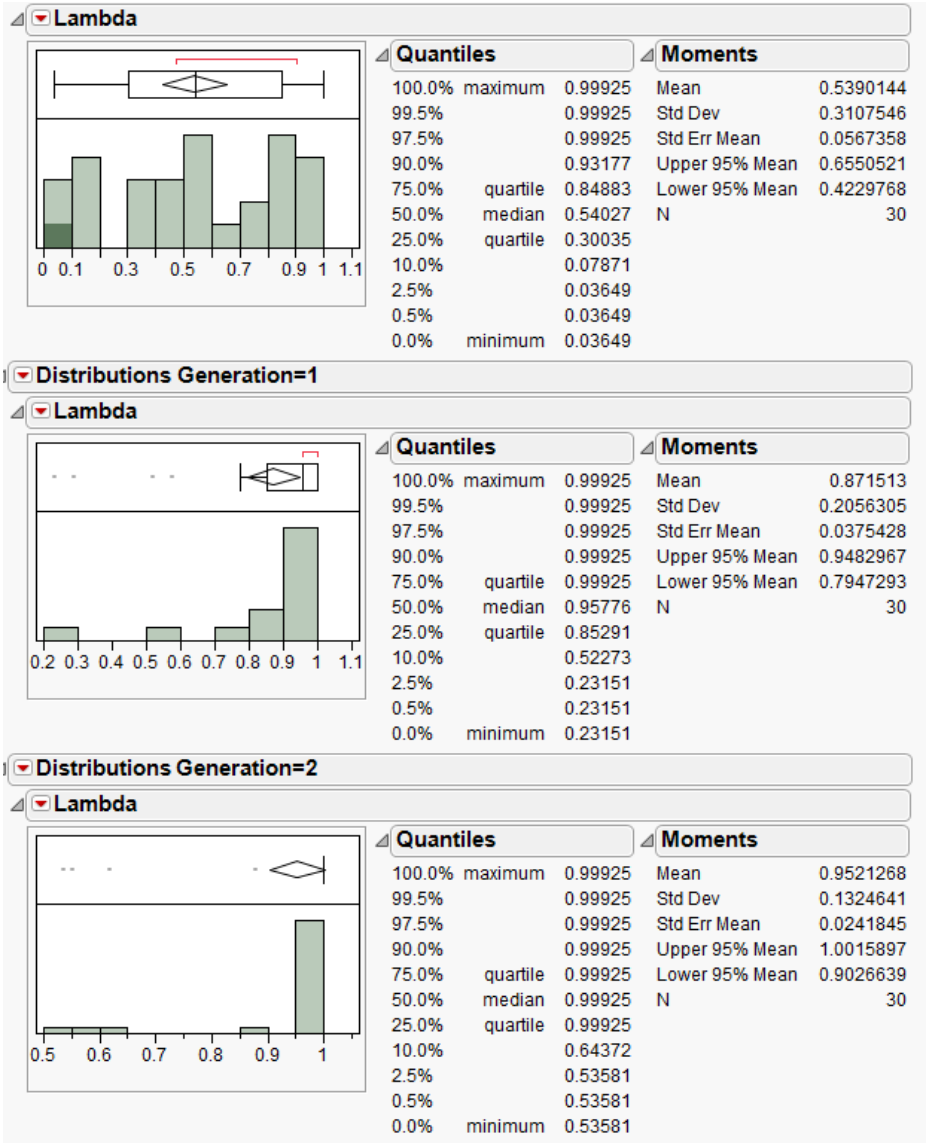

Figure 21. Statistics on the Lambda Gene in the First Three Generations

The other genes in this model all seem to fit only loosely into an optimum. The Gamma gene seemed to show some population pressure to stay close to the range of 0.3 to 0.5. The other genes did not seem to come stable to any particularly tight grouping.

Assuming the gene pool has roughly stabilized by 50 generations, the remaining 50 generations were used to produce random agents for testing within the CG model. The results of testing are discussed in the next chapter.

IV. APPLICATION TO CULTURAL GEOGRAPHY

Cultural Geography is an extremely complex model and there were changes to be made to the trust algorithm in order to fit it within the existing CG architecture. In a typical simulation run there simply are not enough communications for the agents to have a state space that is two dimensional and inclusive of subject matter and sender. For this initial implementation, it was necessary to limit the state space to include only the sender and be completely independent of the subject matter. In future versions, the subject matter will likely be brought back in, in such a way that will require the simulation to undergo an initial learning period prior to the actual simulation run. More of this will be discussed in the future work section in Chapter V. Additionally, the CG model is a discrete event simulation in which there are distinct and separate events for sending and for receiving communications. These events occupy different portions of the code and therefore it was vastly simpler to develop separate inbound and outbound trust models. In future versions, a method for linking these separate algorithms can be implemented. However, for this early version, no such link was developed. Lastly, in order to help track trust development during future testing it was decided that a binary trust decision would not suffice. In lieu of this, the agents choose to raise or lower a trust value each time a communication is sent or received. The trust value, compared to a threshold determines whether or not a communiqué is to be trusted.

This will also allow future implementations to model the concept of trust as a floating value rather than a binary decision.

A. THE TRUST MODEL WITHIN CULTURAL GEOGRAPHY

As was discussed previously the process from going from the received percepts of the environment to the hashable state to be used in the RL algorithm is not able to be generalized and must be specifically implemented for each simulation this algorithm is used in. For CG, the agent has an internal process that forms the current situation as described in Chapter II. This trust algorithm is attached to CG at this point. In particular, if the current situation involves the receipt of a communication, the trust algorithm is applied. In the simplified test program, it was possible to use both the sender's name and the subject as state variables; however, due to the complexity of CG, it is necessary to only use the subject at this time. In addition to reducing complexity, it is also realistic to remove the subject from the state determination because in the observations of real human interactions, the subject of discussion does not often effect trust. The person sending the information seems to have a much more significant impact.

Within CG, the agents choice of actions are themselves based an RL algorithms. The agents determine its state using a cognitive approach described in Chapter II and based on that make a choice of possible actions. If the agent chooses to communicate, it is at this point that the outgoing trust algorithm is tied in. The agent has a selection of its closest neighbors in the social network.

The algorithm will choose to trust the agents to receive the communication on an individual basis based on the trust level of that agent. Prior to making the actual trust decision, the agent will choose to either raise, lower or keep constant the level of trust in each of the potential nearest neighbors with whom it may communicate. These decisions are the result of an RL algorithm.

One of the crucial pieces to developing a successful RL based algorithm is definition of a reward structure. Concurrently with the development of this trust algorithm was the development of a cognitive architecture. The cognitive architecture allows the agent to receive percepts from the environment into the agents short term memory which contains a tunable limit to the number of percepts that can be simultaneously stored in short term memory. Periodically the short term memory is evaluated and a situation is cognitively determined that tells the agent essentially what is going on in the world. The agents then use this situation to determine motivation. For example, in CG all the agents are commodity seekers, obtaining items such as food, water and fuel as they are needed. If an agent has been without water for a while, its basic need for water will be at the forefront of his motivations. The cognitive architecture also has a long-term memory that can give the agent a sense of how they are prospering. The CG cognitive architecture provides a built-in function for agent satisfaction that is easily used as the prime reward signal for the trust algorithms as shown in Figure 22 below.

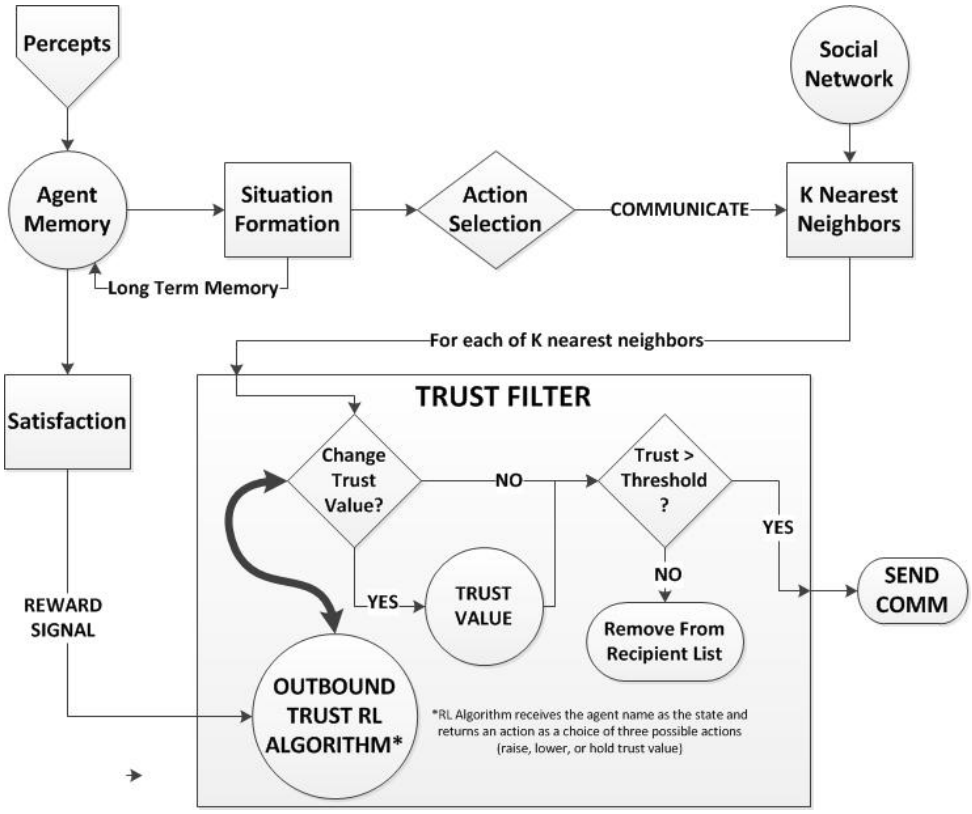

Figure 22. Overview of Outbound Trust Algorithm Within Cultural Geography

B. ATTEMPTING TO PLAY PUBLIC COMMODITY GAMES IN CULTURAL GEOGRAPHY

In order to develop a simple enough PC scenario within CG and remain in the scope of this research project it was necessary to patch the PC game into an existing scenario. The scenario chosen was a simple model of 30 agents modeled after the population of the United States. These agents will communicate for approximately 300 rounds with injects from the environment that are information pertinent to their national satisfaction. These injects include economic factors. With the generalized nature of the algorithm being applied it is felt that the particulars of this model would

not affect the outcome of the trust algorithm. Particularly if the national satisfaction was assumed to be equivalent to the faith in the public commodity, we can easily play the public commodity game within this model.

C. DISCUSSION OF EXPERIMENTAL RESULTS

The PC game within the CG model had similar results to early testing in the standalone version. Figure 23 shows the individual agents PC contributions and satisfaction over the run of the simulation.

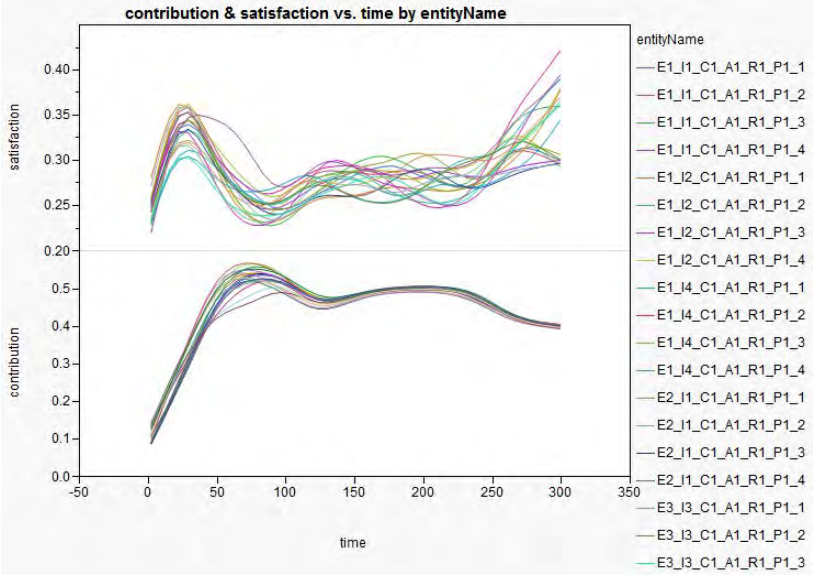

Figure 23. Initial Results of Public Commodity in Cultural Geography Showing PC Contributions Over Time by Agent and Agent Satisfaction.

Both contribution and satisfaction are very tightly grouped, which is not a feature to be expected from a real population of individuals in a similar situation. To understand what is going on here, recall that the contribution of the agents were identical to the issue

stance of national satisfaction. What this graph shows is that with this scenario and this particular implementation of the CG model, all the agents in this scenario have very little difference in their beliefs. This is likely due to the fact that the belief structure for this test case was very simple. More significant than this however, is the likelihood that the agents can too easily change their beliefs. Looking to the standalone data, it was not until a very sharp exponentially driven penalty function was applied to the model that realistic trust group formation began to occur. There will be more discussion of this and other recommended improvements in the future directions chapter to follow.

V. FUTURE DIRECTIONS

In summary, there were very promising results observed from the initial testing of the algorithm. Applying this algorithm to the CG model produced less dramatic results, however has pointed to some obvious areas for improvement. In the following sections there are several recommendations for future work with this trust model.

A. FUTURE TESTING WITH TRUST GAMES

The results of testing within Cultural Geography show that the agents too readily allow their beliefs to come in line with each other and thusly allow their contributions to nearly all be the same. With such behavior, the game is not dynamic enough for there to be any trust formation. Additionally, the game is being played as if all the agents in the society are within one large trust group. Without separate groups competing, there are no really good data from which to validate this model of trust within Cultural Geography. This situation is strikingly similar to what was first observed in the standalone version. Essentially in that instance all the players converged to the same beliefs and the same PC contribution. When a significant penalty was added for changing beliefs, the model became far more realistic. It is recommended that to test this algorithm further within CG, a more significant penalty be implemented for belief revision.

B. GENETIC ALGORITHMS FOR IN-SITU MODIFCATION OF AGENT LEARNING

It may also be possible to make the learning parameters of the agents self adaptive. If we follow the same basic genetic algorithms approach described previously, we might find that the agents could be made to alter their own learning parameters to adapt to different environments. The fundamental difference would be that instead of 30 competitively bred agents, there would instead be 30 agents that are static that have multiple learning techniques applied internally. For example, each agent would have 7 different RL algorithms operating in its internals that would also give it up to 7 different possible courses of action. The choice of which of these 7 to utilize can also be RL driven in a similar situation to a 7-armed bandit scenario, in that it is a stateless RL algorithm with a static set of possible actions. This way, when an RL algorithm leads to poor performance, another one may be selected. The 7 RL algorithms could periodically be culled and bred in order to take advantage of the benefits of genetic techniques.

C. SITUATION IDENTIFICATION AND LAYERED APPROACH TO TRUST

Due to the scarcity of communications in the average CG run, there is not a lot of time to allow RL algorithms to work and develop trust. This is the primary reason why the state input to the RL algorithm for CG had to be limited to just the identity of the sender. If the state were more complicated, learning behavior simply would not have enough time to find any sort of optimums in the state-action space. However, if an adequate method could be found to prime the

learning engines of these agents that would not be computationally untenable, the state space could be much more complicated and make the model much more realistic.

Obviously the first item would be to bring back the subject of communications back into the trust decision. For future work it would be useful to determine if the trust decision should be a single decision based on the sender-subject as a single entity, or should it be two sequential decisions? As a sequential decision is would be that the agent would first ask if they trust the sender and then if the sender is trusted, should they trust the sender to discuss the particular subject matter?

The next level in adding complexity to the state space would be to allow all the perceptual information that a person would normally use to develop a notion of trust in others. The need for this is clear when we have an established social network of agents and then we add a newcomer agent. If that newcomer or stranger enters into conversation with an established agent, the initial trust decision can become an important part of social phenomenon. Therefore, it is not enough to just allow the RL algorithm to develop the sense of trust of this agent by its name. Rather, the agent's physical (i.e., demographic) characteristics will play a huge role in the initial trust determination.

The state space would be represented by a series of percepts of the received communication. Those percepts could include the identity of the sender, the subject matter, and other characteristics including race, demeanor, appearance, age, apparent social status, or many more

possibilities. The obvious problem is that if the state space is too complex, the probability that any two states the agents find themselves are the same would be small. Therefore, if we employ a standard single RL algorithm, it could take millions of interactions before the RL algorithm has located any optimums within which to operate.

One possible way around the problem with overly complex state spaces would be to build a layered dynamic algorithm where the trust decision is a series of decisions. The state space for each decision is based on a single characteristic just like proposed above where first we trust the sender and then the subject matter. There is an added problem to this approach in that it is unrealistic to assume that this complex layered trust decision will be the same from the time an agent is first met to when they are an old friend and have been for 20 years. These decisions do not have to be sequential. In fact, they could be a combination of serial and parallel weighted decisions as shown in Figure 24.

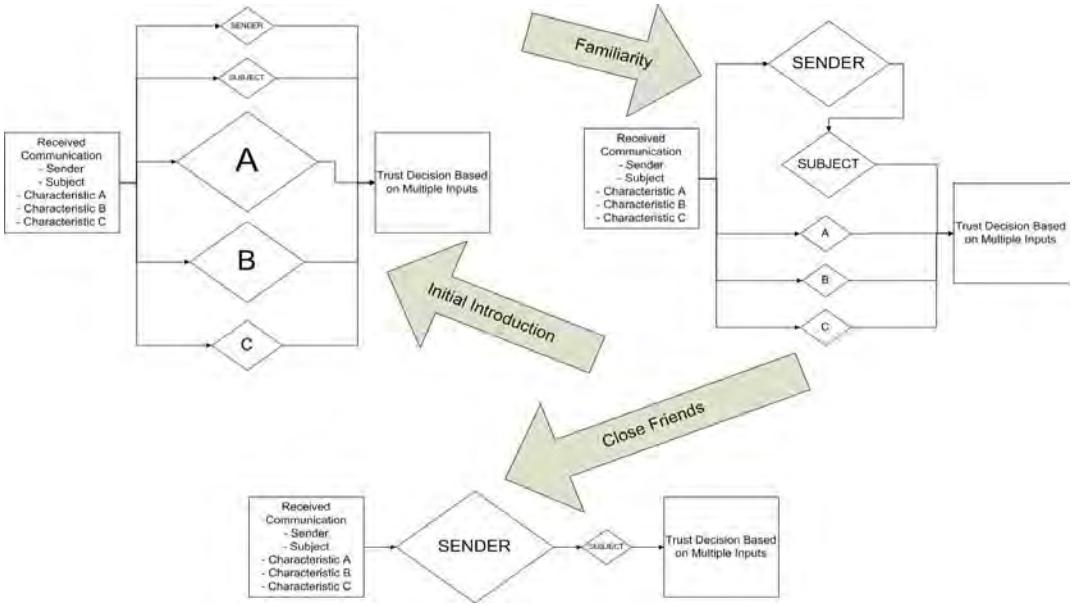

Figure 24. Layered Approach to Trust Decisions

As shown in Figure 24, the decision to trust could be based on a group of series and parallel decisions. The pathways and weights of these decisions could modified based on utility gained from the agent, or could be based on how long and how often the agent is communicated with. The updating of the pathways and weights could also be part of an RL algorithm in itself.

As part of the layered approach to the trust decision we could also include some internal percepts. This would allow the agents "emotional" state to possible effect its trust decision. As an example, a real person who is feeling extremely happy and fortunate is likely to be far more trusting of people than someone who is not.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Allport, G. W. (1954). *The Nature of Prejudice.* Cambridge, MA: Addison-Wesley Publishing Company.

Chudler, E. H. (2001, March 10). A Computer in Your Head? *Odyssey Magazine*, 6-7.

Clausewitz, C. von. (1832). *On War.* Translated by Michael Howard and Peter Paret. Princeton, NJ: Princeton University Press.

Dejmal, S., Fern, A. & Nguyen, T. (2008). Reinforcement Learning for Vulnerability Assessment in Peer-to-Peer Networks. *World Academy of Science, Engineering and Technology, 51*(44): 256-261.

Dekker, A. H. (2008). Centrality in social networks: Theoretical and simulation approaches. *Proceedings of SimTecT 2008,* Melbourne, Australia.

Demarest, G. (2011). *Winning insurgent war: Back to basics.*, Fort Leavenworth, KS: Foreign Military Studies Office.

Department of the Army. (2006). *FM 3-24 / MCWP 3-33.5 Counterinsurgency.* Washington, DC: Headquarters of the Department of the Army.

Dunbar, R. (1996). *Grooming, gossip, and the evolution of language.* London, UK: Faber Limited.

Dutt, V. (2011). Explaining human behavior in dynamic tasks through reinforcement learning. Accepted for publication in *The Journal of Advances in Information Technology*.

Hallinan, M. T. & Smith, S. S. (1989). Classroom characteristics and student friendship cliques. *Social Forces, 67*(4): 898-919.

Hoffman, E., McCabe, K. A., & Smith, V. L. (1998). Behavioral Foundations of Reciprocity: Experimental Economics and Evolutionary Psychology. *Economic Inquiry, 36*(3): 335-352.

Joint Chiefs of Staff. (2004). ***Joint publication 3-07.1: Joint tactics, techniques, and procedures for foreign internal defense (FID).*** Washington, DC: Office of the Chairman of the Joint Chiefs of Staff.

Krebs, V. (2002). Uncloaking terrorist networks. ***First Monday, 7***(4)

McPherson, M., Lynn S-L., & Cook, & J. M. (2001). Birds of a feather: Homophily in social networks. ***Annual Review of Sociology, 27***: 415-444

Mistzal, B. A. (1996). ***Trust in modern societies: The search for the bases of social order.*** Cambridge, UK: Polity Press.

Ostram, E., & Walker, J. (2003). ***Trust and reciprocity: Interdisciplinary lessons from experimental research.*** New York, NY: Russell Sage Foundation.

Pollock, S., Alt, J., & Darken, C. (2011). Representing trust in cognitive social simulations. ***Lecture Notes in Computer Science: Proceedings of the (2011) Social Computing, Behavioral-Cultural Modeling and Prediction Conference.*** College Park, MD: Springer

Shenk, D. (2006). ***The immortal game.*** New York: The Doubleday Broadway Publishing Group.

Stanley, D. A., Sokol-Hessner, P., Banaji, M. R., & Phelps, E. A. (2011). ***Implicit*** race attitudes predict trustworthiness judgements and economic trust decisions. ***Proceedings of the National Academy of Science of the United States of America, 108***(19): (7710)-(7715).

Striedter, G. F. (2006). Précis of principles of brain evolution. ***Behavioral and Brain Sciences, 29***(1): 24-25.

Sutton, R. S., & Barton, A. G. (1998). ***Reinforcement learning: An introduction.*** Cambridge, MA: MIT Press.

Sveningsson, M. (2007). *Taking the girls' room online: Similarities and differences between traditional girls' rooms and computer-mediated ones. Proceedings of INTER: A European Cultural Studies Conference in Sweden.* Linkoping University, Sweden.

Von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior.* Princeton, NJ: Princeton University Press.

Wellman, B., Carrington, P. J., & Hall, A. (1988). *Networks as personal communities. Social Structures a Network Approach, 2*: 130–184.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Christian Darken, PhD
   MOVES Institute
   Naval Postgraduate School
   Monterey, California

4. LTC Jonathan Alt
   TRAC-Monterey
   Naval Postgraduate School
   Monterey, California

5. MAJ Francisco Baez
   TRAC-Monterey
   Naval Postgraduate School
   Monterey, California