Public Goods Game Simulator with Reinforcement Learning Agents

ManChon U
Department of Computer Science
The University of Georgia
Athens, GA 30602, USA
manchonu@cs.uga.edu

Zhen Li
Department of Computer Science
The University of Georgia
Athens, GA 30602, USA
zhen@cs.uga.edu

***Abstract*—As a famous game in the domain of game theory, both pervasive empirical studies as well as intensive theoretical analysis have been conducted and performed worldwide to research different public goods game scenarios. At the same time, computer game simulators are utilized widely for better research of game theory by providing easy but powerful visualization and statistics functionalities. However, although solutions of public goods game have been widely discussed with empirical studies or theoretical approaches, no computational and automatic simulation approaches has been adopted. For this reason, we have implemented a computer simulator with reinforcement learning agents module for public goods game, and we have utilized this simulator to further study the characteristics of public goods game. Furthermore, in this article, we have also presented a bunch of interesting experimental results with respect to the strategies that agents used and the profits they earned.**

Keywords-Reinforcement Learning, Game Theory; Public Goods (PG), Decision Making, Simulation, User Interface

I. INTRODUCTION

Game theory is a branch of applied mathematics that is used in the social sciences, most notably in economics, as well as in biology (most notably evolutionary biology and ecology), engineering, political science, international relations, computer science, and philosophy. Game theory attempts to mathematically capture behavior in strategic situations, in which an individual's success in making choices depends on the choices of others. While initially developed to analyze competitions in which one individual does better at another's expense (zero sum games), it has been expanded to treat a wide class of interactions. Today, "game theory is a sort of umbrella or 'unified field' theory for the rational side of social science, where 'social' is interpreted broadly, to include human as well as non-human players (computers, animals, plants)" [1]

Traditional applications of game theory attempt to find equilibria in these games. In an equilibrium, each player of the game has adopted a strategy that they are unlikely to change. Many equilibrium concepts have been developed (most famously the Nash equilibrium) in an attempt to capture this idea. These equilibrium concepts are motivated by different fields of applications, although they often overlap or coincide with each other. This methodology is not free from criticism, and debates continue over the appropriateness of particular equilibrium concepts, the appropriateness of equilibria altogether, and the usefulness of general mathematical models.

The public goods game is a standard of experimental economics; subjects in the game secretly contribute none or more of their private tokens to put into the public pot. Each subject keeps the tokens they do not contribute plus an even split of the discounted tokens in the pot. The group as a whole does best when everyone contributes all of their tokens into the public pool. If everyone puts every token they start with into the pot then the group will extract the maximum total reward from the economists running the test. However, the Nash equilibrium in this game is simply zero contributions by all; if the experiment were a purely analytical exercise in game theory it would resolve to zero contributions because any player does better contributing zero than any other amount regardless of whatever anyone else does. Following game theory, those who contribute nothing are called "defectors", as opposed to the contributors who are called "cooperators". The defector is also a "free rider". In fact, the Nash equilibrium is rarely seen in experiments; people do tend to add something into the pot. The actual level of contribution found in individual subjects varies widely, anywhere from 0% to 100% of initial endowment can be chipped in - subjects are heterogeneous. Some experiments [2][3][4][5][6] have been conducted to try proving this situation, and some are to try obtaining more statistical data for other relative researches. But all of these researches face to the same problem, which is that taking people to participate into this sort of experiments is time consuming and it is costly. Therefore, it is always a good idea to have a computer simulator to mimic the human behaviors like they are playing the game.

There are many different simulators available for the purpose of researching game theory scenarios and settings. Yet most of them deal with the Prisoners' Dilemma game which could be viewed as a simplified or specified version of public goods game. Most of them are web-based small java applets for visualization or statistic purpose. [7][8][9] However, few simulator focus on building agents module and using this module to research game settings as a supplement of empirical study and theoretical works. For this reason, we have built a computer simulator for public goods game with reinforcement learning agents.

The rest of the paper is organized as follows. In Section II we introduce the background of this research work. Section III shows the overall structure and outlook of our simulator. In Section IV we illustrate the detail implementation of our system, which includes how we formulate our reinforcement learning agents. In Section IV.A we present some interesting researching results and their analyses. Finally, we present the conclusion and future work in Section VI.

II. BACKGROUND

A. Related Work & Motivation

A large number of research works have been conducted based on public goods game: Simon P. Anderson et al. [10] formalize an equilibrium model in which altruism and decision-error parameters determine the distribution of contributions for linear and quadratic public goods games. The empirically examination in [11] showed evidence supporting that humans will readily and knowingly behave altruistically. Reviva Hasson et al. [12] utilizes public goods game to emphasize the important role of trust in enhancing cooperation in climate change problem. Two experiments in [13] using a real-time version of the voluntary contribution mechanism were conducted to investigate the hypothesis that players are generally willing to contribute public goods conditional on beliefs that others are doing so at similar levels. Andreas Glöckner et al. [14] describe the importance of intentions for cooperation in public-goods environments also, confirming the intuition that a sacrifice provides an encouraging signal to followers. Marco A. Janssen et al. [15] compares the empirical performance of a variety of learning models and showed the potential of using laboratory experiments to develop empirically tested agent-based models. Ming Tan [16] demonstrates that reinforcement learning agents can learn cooperative behavior in a simulated social environment. Robert Kurzban and Peter Descioli [17] introduced how people would behave if they could pay to see information during the game. They found that were willing to incur costs to acquire information, particularly those using a reciprocal strategy. Alexis Belianin and Marco Novarese [18] reports a cross-cultural public goods game experiment played in real time through Internet. The results show that the degree of cooperation is rather high, but does not vary significantly with nationalities of the group members, while communication tends to enhance contributions to public goods.

A common drawback of above studies is that they all required the participation of human beings, which means that all of these studies require a good few amount of resources, in terms of time and money. Actually, all of these amounts of time and money can be reduced if a computer simulator for public goods game is available. For this reason, we have implemented our public goods game simulator with reinforcement learning agents.

B. Reforcement Learning

Reinforcement learning is a sub-area of machine learning, it learns by interacting with an environment. Reinforcement learning allows a virtual agent to automatically determine its next action in an environment in order to obtain the maximum long-term reward. For making the best decision in next action, an agent is required to learn from the consequences of its actions without any taught (unsupervised learning). The reinforcement signal is required for an agent to learn its behavior; it is a numerical reward, which stands for an action's outcome. Based on this, an agent then tries to learn to select actions that can lead to the maximum accumulated reward over time.

The above environment is typically formulated as a finite-state Markov Decision Process (MDP). A basic reinforcement learning model consists of a set of environment states S, a set of actions A, and a set of scalar rewards in R. At each time t, an agent knows about its state s_t and a set of possible actions $A(s_t)$. It then selects an action a $\in A(s_t)$, followed by updating its current state to s_{t+1} with a reward r_t. To go further and say, consider a sequence of states followed by rewards: $s_t, r_t, s_{t+1}, r_{t+1}, ..., r_T, s_T$. The final expected return R_t in the future from state s_t is: $R_t= r_{t+1}+\gamma^1 r_{t+2}+\ldots+ \gamma^{T-t-1} r_T$ where γ is a discount factor whose value is less than one. Reinforcement learning assumes that the value of a state V(s) is directly equivalent to the expected return: $V(s) = E_\pi(R_t|S_t=s)$, where π is here an unspecified action policy. Therefore, the value of state s_t can be iteratively updated with: $V(s_t) \rightarrow V(s_t) + \alpha[R_t - V(s_t)]$, where α is a step-size (often =1). More details are described in [20].

C. Variants of Public Goods Game

In this section, we are going to introduce several mainstream public goods game [20]:

1) *Iterated Public Goods Game.* This type of game simply involves the same group of subjects playing the basic game over a series of rounds. The typical result is a declining proportion of public contribution, from the simple game (the "One-shot" public goods game). When trusting contributors see that not everyone is giving up as much as they do they tend to reduce the amount they share with the group if the game is repeated to another round. If this is again repeated the same thing happens but from a lower base, so that the amount contributed to the pot is reduced again. However, the amount contributed to the pool rarely drops to zero when rounds of the game are iterated, because there tend to remain a hardcore of 'givers'. One explanation for the dropping level of contribution is inequity aversion; once it is realized that others are receiving a bigger share for a smaller contribution the sharing members react against the perceived injustice (even though the identity of the "free riders" are unknown, and it's only a game). Those who contribute nothing in one round, rarely contribute something in later rounds, even after discovering that other people are.

2) *Open Public Goods Game.* If the amount contributed isn't hidden it tends to be higher. In a typical public goods game there might be six subjects contributing to the pot so concealing the level of contribution isn't difficult. In "pairwise iterations" with only two players the other player's contribution level is always known.

3) *Public Goods Game with Punishment.* Famously, the option to punish non-contributors after a round of the public

goods game is widely exercised (although costly and technically "irrational"). In most experiments this leads to greater group cooperation, and fewer defections in subsequent rounds.

4) Public Goods Game with Reward. The option to reward co-operation (rather than punish defection) is less often exercised by players, but some studies have shown that it can be more effective at enforcing co-operation than punishing (e.g. Rand 2009). The evidence comparing reward with punishment is mixed, with Sefton (2007) finding that rewards could not sustain long-term cooperation.

III. DESIGN OVERVIEW

A. Overview Structure

Our simulation system are designed and built mainly in three parts. One is the core functional part including game engine and agent modules. Another part is the graphical user interface which enables user to conveniently setting game scenarios (Global Setting). The third part includes a bunch of auxiliary tools such as random generator, STD calculator, and log file writer. Random generator generates datasets according to user specified distribution type such as normal distribution, uniform distribution, and polarized distribution. STD calculator calculates standard deviation of dataset, and log file writer writes game history detail into log files for a user's further usage.

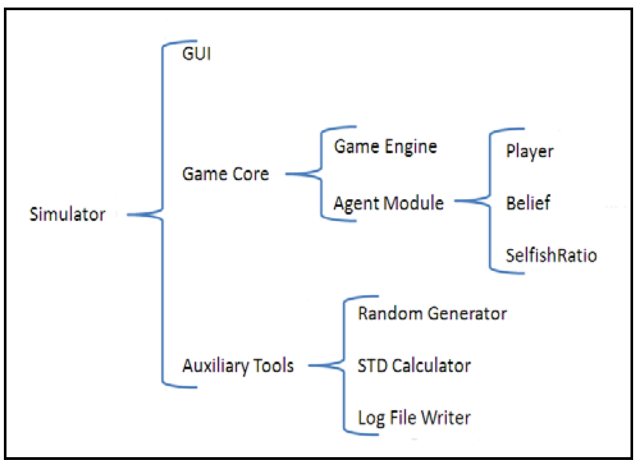

Figure 1. Overall Code Structure

B. User Interface

Our simulator is implemented with Java, which is platform independent. With our simulator, a user is allowed to set a brunch of parameters in order to simulate the situation that there is a large group of people are participating into the game. A user is not only allowed to set all same parameters (individual characteristic) to all of the virtual players (agents) by once, but is also allowed to set these parameters for every single individual players separately. Figure 2. shows the outlook of our simulator GUI.

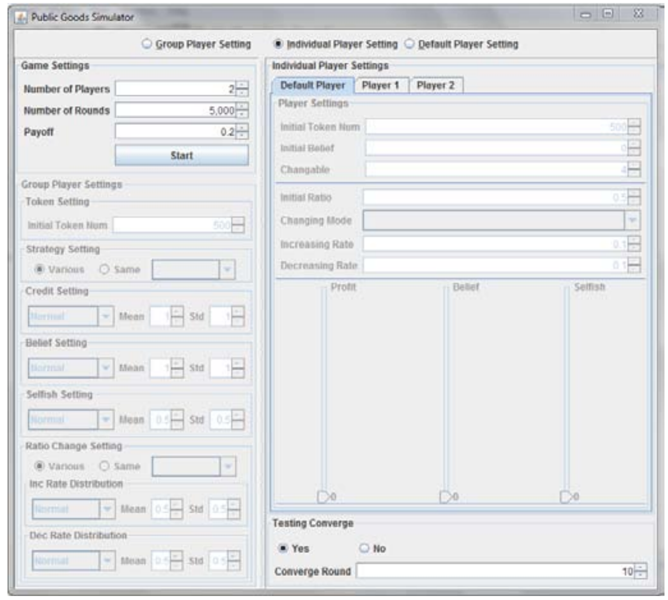

Figure 2. Simulator GUI

IV. IMPLEMENTATION DETAILS

A. Game Engine

Basically, game engine in the simulator performs the roles of economist in the real public goods game experiments. It has following functionalities:

- Update game status such as rounds.
- Collect and re-distribute agents' tokens.
- Record game details in log file.

B. Agent Module

Agent modules in this simulator are designed to simulate rational players with different preferences and strategies. Basically, the agent module has following functionalities:

- Update agent status such as number of tokens owned.
- Update profit according to maximum earn and loss.
- Update beliefs, which is agent's estimation of the number of tokens contributed to public pot in next round.
- Update selfish ratio, which is agent's contribution willing in next round.
- Calculate the decision for next round based on profit, belief and selfish ratio.

C. Learning Model

In this section, we introduce our learning model by mapping it to the formal reinforcement learning format which we introduced in section II.B. Our model could be expressed as in Figure. 3, in which s_t^i is the current state of a player that describes whether a player is earning or losing at round t. a_t^i is the action that a player choose to perform at round t. γ_t^i is the discount factor of player i. π_t^i is the implicit policy that shows how other factors in the game affect a player's decision in next round. r_t^i is the reward of player i at round t, which is directly equal to $profit_t^i$. R_T^i is the final reward of player i when the game ends.

$$
\begin{aligned}
&i: player\ i \\
&t: current \\
&T: termination \\
&s_t^i = \{earning, losing\} \\
&a_t^i = \{contribute\ token, not\ contribute\ token\} \\
&\gamma_t^i = \theta_t^i + \beta_t^i + \delta_t^i \\
&\pi_t^i = \theta_t^i \times profit_t^i + \beta_t^i \times belief_t^i + \delta_t^i \times (1 - selfish_t^i) \\
&r_t^i = \{profit_t^i \mid profit_t^i \ \epsilon\ R^{\pm}\} \\
&R_T^i = r_t^i + r_{t+1}^i + \cdots + r_{T-1}^i + r_T^i
\end{aligned}
$$

Figure 3. Learning Model of our Reinforcement Learning Agents

This is a simple but quite powerful learning model. In Section A, a bunch of tests and analysis results demonstrate this point. It is worth noting, all three core items in this model: profit, belief and selfish are normalized. We will introduce the calculation and updates of these items in the following sub-sections.

D. Profit Calculation

To calculate a normalized profit, we adopt formulas in Figure 4. As we can see, the profit are normalized with respect to maximum possible earn or loss respectively. First we calculate at time the difference D_t^i between player i's current number of tokens C_t^i and initial number of tokens I. Then we calculate the maximum possible number of tokens earned E_t^i and maximum possible number of tokens lost L_t^i according to number of rounds already played $O_t$as well as different game settings including payoff F and number of players S. Finally, we calculate the current profit P_t^i by normalizing token difference D_t^i with maximum possible earn E_t^i or loss L_t^i.

$$
\begin{aligned}
D_t^i &= C_t^i - I \\
E_t^i &= \begin{cases} F \times O_t, & F > S \\ \dfrac{S-1}{S} \times F \times O_t, & F < S \end{cases} \\
L_t^i &= \frac{S-F}{S} \times RP_t \\
P_t^i &= \begin{cases} \dfrac{D_t^i}{E_t^i}, D_t^i > 0 \\ \dfrac{D_t^i}{L_t^i}, D_t^i < 0 \end{cases}
\end{aligned}
$$

Figure 4. Calculation for Normalized Profit

E. Belief Update

Belief is another important item in our learning model which is defined in the following manner. Belief is a specific agent's prediction of the number of tokens finally contributed to the public pot in next round. In our agent module, belief is calculated together with another metric credit which characterizes the correctness of agent's belief. Alike human beings, an agent's belief follows the actual contribution situation with a variation of credit. And the update of credit makes this variation getting smaller along with the process of the game. Besides, the belief is also normalized according to the max possible number of tokens can be contributed as shown in Figure 5. Figure 6. is the visualization of how the belief is being updated.

$$
\begin{aligned}
&A_t: Acutal\ number\ of\ tokens\ contributed\ in\ round\ t \\
&B'^i_t: Meta\ Belief\ in\ round\ t\ of\ player\ i \\
&B_t^i: Belief\ in\ round\ t\ of\ player\ i \\
&C_t^i: Credit\ of\ Belief\ in\ round\ t \\
&C_{t+1}^i = |A_t - B_t^i| \\
&B'^i_{t+1} = A_t \\
&B_{t+1}^i = Rand(B'^i_{t+1} - C_{t+1}^i,\ B'^i_{t+1} + C_{t+1}^i)
\end{aligned}
$$

Figure 5. Details of Belief Update

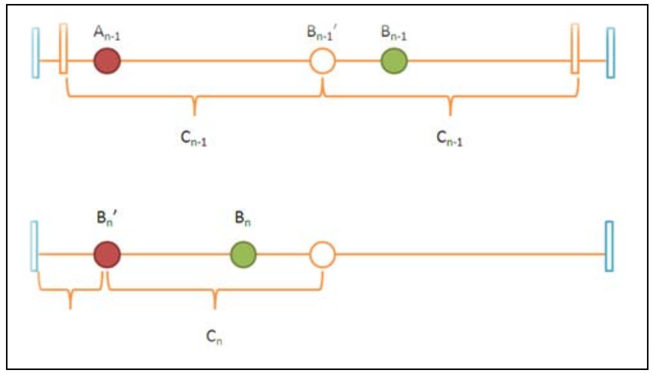

Figure 6. Visualization of Belief Update

F. Selfish Ratio Update

Different agents in our system have various self ratios. According to the actual situation in which human participate into this game, we have implemented three different update types of selfish ratio in our system:

- Fixed Ratio: Selfish ratio of an agent will not change from rounds to rounds in the game
- Mono Increasing: Selfish ratio of an agent increases when token lost in the previous round, or keep unchanged otherwise
- Bi-directional: Selfish ratio of an agent increases when token lost in the previous round, and decreases when tokens earned in the previous round.

The increase and decrease ratio could be specified by the user of the simulator.

V. EXPERIMENT RESULT & ANALYSIS

In our experiment, we simulate 10 players play the game simultaneously. Each of them is given 1000 tokens initially, and the game runs 100 rounds. On each round a player can either contribute tokens into the public pot or keep them for himself. If a player invests a token, it costs him money. For example, let's set the payoff as 40%, whenever a player invests one token, and he personally earns only 0.4 tokens. But every other member in the group gets 0.4 tokens as well. So the group as a whole gets 4 tokens for every one that's invested. Therefore, when individuals contribute into the public pot, the whole group becomes richer, beyond the tokens contributed.

If all players were to contribute all their tokens (1000 * 4 = 4000) then the whole group would be greatly enriched (4000 * 4 = 16000). Each individual would have a wealth of 1600 (16000/ 10= 1600). Yet, if everyone were to keep their tokens and contribute nothing into the public pot then they each would have a wealth of 1000 as beginning. If one person were to contribute nothing, and all others were to contribute their entire token into the public pot then the "free-rider" would become the richest in the end of the game.

A. *Equilibrium Test*

We first implement a series of equilibrium test. With the learning model we described in the previous section, equilibrium are found quickly reached no matter with what kind of distribution (normal , uniform, polarization) other parameters are, including initial beliefs, initial credits, initial selfish ratios, selfish ratio increasing / decreasing rates, different strategy weights, etc. However, this is only the general case. With our strategy model (π_t^i), some special cases are also found which beliefs are converged while some agents' actions are changing in pattern of periodic contribution.

1) *General Case*

TABLE I. EQUILIBRIUM TEST (GENERAL CASE)

		Payoff			
		5%	*10%*	*20%*	*50%*
Distribution	Normal	Figure 7.	Figure 12.	Figure 15.	Figure 8.
	Uniform	Figure 10.	Figure 13.	Figure 16.	Figure 18.
	Polarized	Figure 11.	Figure 14.	Figure 17.	Figure 19.

a. See appendix for more images

Table 1 illustrates our plan of equilibrium test. Basically, we implement tests with the combination of two aspects: payoff and distribution types of parameters. We run 50 times for each of the different combination with the number of agents set to 10. Then we randomly select the results of 5 runs to present in this article. As we analyzed before, because of the same linear strategy model shared among all agents, all experiments show a quick convergence to equilibrium. Figure 7. and Figure 8. illustrate the convergence with payoff equals to 5% and 50% while other parameters set to normal distribution. The y-axis of the upper half figure is the number of tokens contributed in each round and the y-axis of the bottom half figure is the standard deviation of each agent's token in each round. It is clear to see all agents action reach the equilibrium of no contribution and the standard deviation of five repeated experiments are ranged within 2 with 1000 initial tokens for each agent.

Figure 10 to Figure 19 illustrates further details of other different payoffs and distributions type combination. It is worth noting, when the payoff is equal to or less than 10%, as there are 10 agents play in our game, the actions are converged to no contribution for every agent which is the theoretical Nash Equilibrium. When the payoff is greater than 10%, as reasonable agents, all of their actions converge to contribution which gains them best profit.

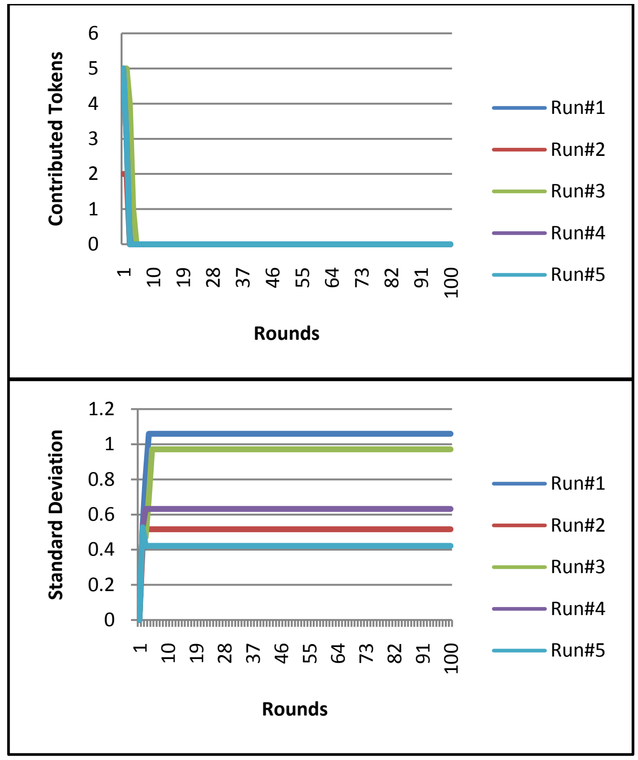

Figure 7. Results with Normal Distribution & Payoff is 5%

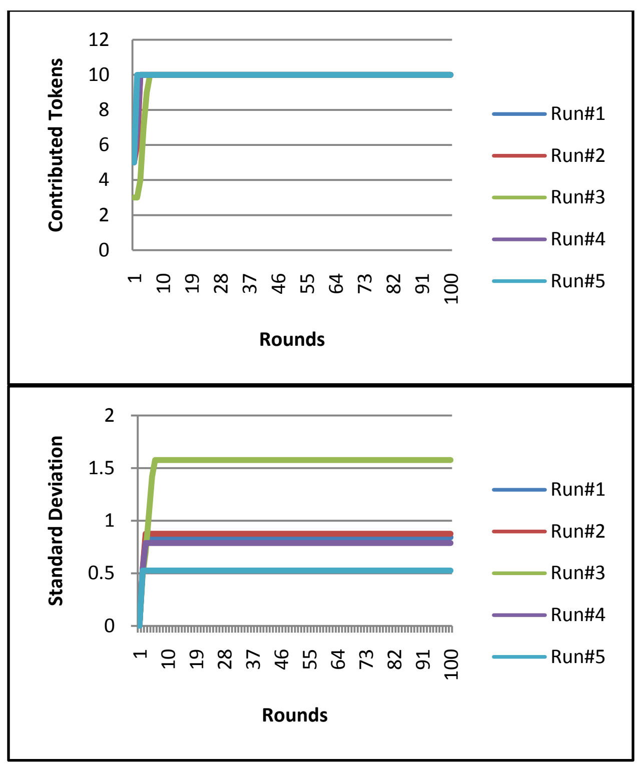

Figure 8. Results with Normal Distribution & Payoff is 50%

2) Special Case

As we mentioned before, some special case come up during our equilibrium tests. For these special cases, we organize them into another set of experiments illustrated in TABLE II.

TABLE II. EQUILIBRIUM TEST (SPECIAL CASE)

		Payoff	
		5%	***10%***
Distribution	Normal	Figure 9.	
	Uniform	Figure 10.	
	Polarized	Figure 11.	

a. See appendix for more images

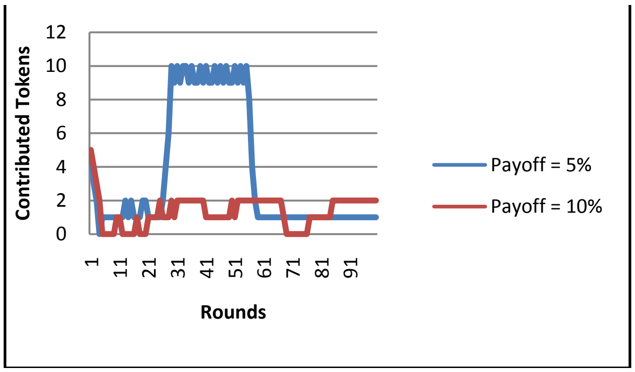

Figure 9. Special Case Result with Normal Distribution

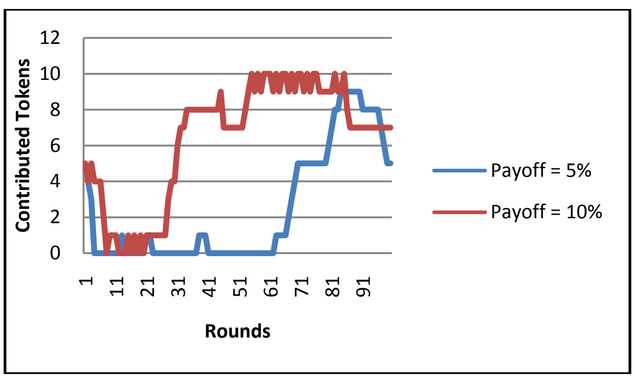

Figure 10. Special Case Result with Uniform Distribution

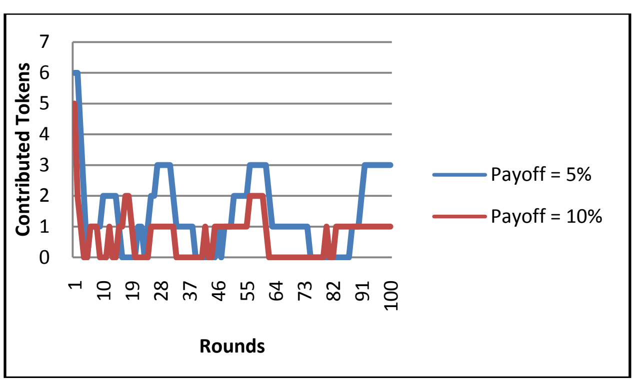

Figure 11. Special Case Result with Polarized Distribution

Figure 9. to Figure 11. illustrate the special cases with combination of different distributions and payoff equals to 5% and 10%. Contrast to Figure 7. , we could see a monotonous increment of standard deviation instead of a convergence. By intensive study of the details of game history, we found this is due to the oscillation of some agents' actions that are illustrated in the upper half image of Figure 9. Further investigation shows that these agents share some common characteristics: they all used fixed selfish ratio and their initial selfish ratio are relatively small. By looking at our learning model in Figure 3. , we could infer that with the gradually reaching of equilibrium and all agents' beliefs converge to 0, the belief items of the strategy model becomes almost 0; and for those with consistent small selfish ratio, the selfish item becomes almost 1 which means an agent is very selfish. Therefore, the only working part is the first item which is related to the profit. Although a loss for a player in the game history means a negative value of profit item, however, since we calculate profit according to max possible earning and losing which are closely related to the number of rounds already passed, thus, the normalized profit is changing in a periodic pattern which causes some extreme optimistic agents (those with consistent small selfish ratios) contribute tokens from time to time.

B. Competitive Test

We also implement some competitive tests. Since our simulator makes all agents share one strategy model, we only conduct this test for the selfish ratio, and all other parameters are the same between different agents. We run 20 experiments with payoff equals to 10 among 10 agents of uniform distributed initial selfish ratio, selfish ratio update type and increasing / decreasing rates. Different meanings of the three update types are as following:

1) Update Type1: An agent always keeps his selfish ratio in the same value as initial no matter he lost or earned in the previous game.

2) Update Type2: An agent increases he selfish ratio in certain extents when he lost in the previous game, while he would never changes his selfish ratio when he earned in the previous game.

3) Update Type3: An agent increases he selfish ratio in certain extents when he lost in the previous game, while he would decreases his selfish ratio in certain extents when he earned in the previous game.

The results are illustrated in Figure 12. It is clear to see that agents with update type 1 (consistent selfish ratio) and type 2 (monotonous increment selfish ratio) have the greatest chance to win. This also explains the real truth that has been proposed for a long time: a selfish person earns the most in this kind of games while any other players are not that selfish as he is. It is not difficult to conclude from these series experiments that, although the total benefits are the same for all players to contribute or not contribute, those who are more selfish and who do not contribute earn more of the game. Furthermore, besides these systematic tests and experiments, we also implemented some individual tests with some interesting payoff settings such as 10.3%. It turns out that with this payoff setting, the agents actions are converged to a oscillation of total 9 out of 10 contribute or 10 out of 10 contribute. This also reinforces the fact that the most interesting part of the game is payoff settings that gives agents just a little bit rewards but not too much.

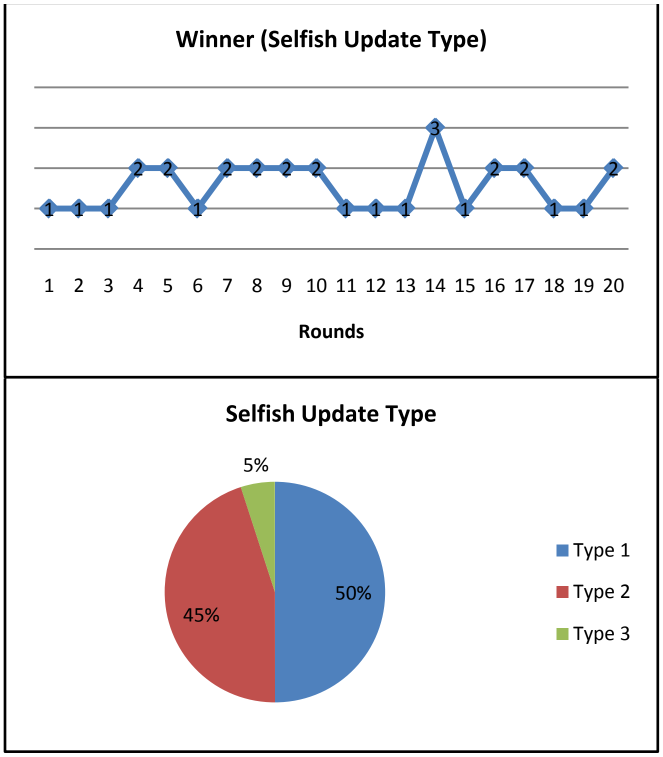

Figure 12. Competitive test between different update types of selfish ratio

VI. CONCLUSIONS & FUTURE WORK

In this paper, we proposed and built a platform independent computer simulator for the specific game “public goods” in the domain of game theory. We also proposed and implemented reasonable agents’ module with the reinforcement learning technique. Alike human’s actual behaviors in this game, we have implemented three core components: profit, belief and selfish that are capable for self-adaptively update according to prior user settings and the actual game status at different rounds. Also, we have implemented and provided a series of auxiliary tools for the usage of this simulator for further research of public goods game. Furthermore, we have performed a bunch of different experiments to demonstrate that our simulator functioned as expected, which means that our simulator can help those researches in the domain of public goods game with much lesser human resources involved. In another hand, our current simulator are based on the iterated public goods game, in future, we could expanded it to become a general purpose simulator which is capable of simulating different variations of public goods game. Last but not least, we also target to improve the graphical user interface (GUI) of our simulator to support more convenient and easy usage.

APPENDIX

All supplemental figures in this article are available online at http://ecml.uga.edu/paper/ICMLA202.pdf

REFERENCES

[1] Aumann, Robert J. (1987), "game theory,",The New Palgrave: A Dictionary of Economics, 2, pp. 460–82 .

[2] Isaac, Walker, and Williams, 1994 Group Size and the Voluntary Provision of Public Goods: Experimental Evidence Utilizing Large Groups. Journal of Public Economics, 54(1)

[3] James Andreoni, William Harbaugh and LiseVesterlund 2003 The Carrot or the Stick: Rewards, Punishments, and Cooperation The American Economic Review, 93(3)(2003)pp. 893–902

[4] Rand, 2009 Positive Interactions Promote Public Cooperation. Science. 2009, 325

[5] Sefton, Shupp, and Walker, 2007 The Effect of Rewards and Sanctions in Provision of Public Goods, Economic Inquiry. , 45 (4): 671-690

[6] Xu, B., Liangcong Fan, QiQi Cheng, and ZhiJian Wang, 2008 Asymmetries of Reward and Punishment: Evidence from Public Goods Experiments, 8th Annual Conference of Economics in China

[7] Game Theory Tool - a utility for Finite Mathematics and Finite Mathematics & Applied Calculus - www.hofstra.edu/~matscw/gametheory/games.html

[8] Simulator for Prisoner's Dilemma - www.iterated-prisoners-dilemma.net/

[9] Simulators for Game Theory - www.gametheory.net/applets/evolution.html

[10] Simon P. Anderson, Jacob K. Goeree and Charles A. Holt, "A theoretical analysis of altruism and decision error in public goods games," Journal of Public Economics. Volume 70, Issue 2, 1 November 1998, Pages 297-323

[11] O'Gorman, R., Henrich, J., & van Vugt, M. (2009). Constraining free-riding in public goods games: Designated solitary punishers can sustain human cooperation. Proceedings of the Royal Society B, 276, 323-329.

[12] Hasson, Reviva, Åsa Löfgren, and Martine Visser (2009), "Climate Change in a Public Goods Game: Investment Decision in Mitigation versus Adaptation", EfD Discussion Paper 09-23, Environment for Development Initiative and Resources for the Future, Washington DC, October 2009.

[13] Robert Kurzban, Kevin McCabe, Vernon L. Smith., and Bart J. Wilson, “Incremental Commitment and Reciprocity in a Real Time Public Goods Game,” Personality and Social Psychology Bulletin, 27(12), December 2001.

[14] Glöckner, Andreas, Irlenbusch, Bernd , Kube, Sebastian, Nicklisch, Andreas and Normann, Hans Theo, Leading With(Out) Sacrifice? A Public-Goods Experiment with a Super-Additive Player (March 2009). MPI Collective Goods Preprint, No. 2009/8.

[15] Janssen, M. A., and T. K. Ahn., "Learning, signaling, and social preferences in public-good games," Ecology and Society 11(2): 21. 2006.

[16] Ming Tan., "Multi-agent reinforcement learning: Independent vs. cooperative agents," Proceedings of the Tenth International Conference on Machine Learning, pages 330–337, Amherst, MA, 1993.

[17] Robert Kurzban and Peter DeScioli, "Reciprocity in groups: information-seeking in a public goods game," European Journal of Social Psychology, Volume 38 Issue 1, Pages 139 - 158, May 2007.

[18] Alexis Belianin & Marco Novarese, 2005. "Trust, communication and equlibrium behaviour in public goods," Experimental 0506001, EconWPA.

[19] Reinforcement learning – http://www.scholarpedia.org/article/Reinforcement_learning

[20] Public Goods Game - http://en.wikipedia.org/wiki/Public_goods_game