

An Automatic Simulation Framework to Find Loopholes in Regimes

Yun Wu*, Yiqiao Li

College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China

Abstract

Loopholes exist in nearly every regime, which leads to invalidity and instability in the running process of a regime. In this paper, we propose a new framework based on agent method and game theory to find loopholes in a regime, which contains two players with their clear payoffs. Following the principle of “winners stay, losers change”, agents make their own choices among strategies, and their choices can affect the payoffs of other players, resulting in a dynamic equilibrium that has obvious features in decision path, which can be easily found in the final figures after simulation. We further exploit typical cases, such as prisoner's dilemma, anti-coordination game, coordination game and harmony, of which the results have been mathematical proved, to illustrate the validity of our method. Finally, we make some regimes in this paper based on the classical case, “Boxed Pigs”, and find the loopholes in these regimes. By utilizing the framework proposed in this paper, managers are able to detect possible problems in advance at the time when they make regimes, which helps reduce the loss of costs caused by management and improve team relations.

Keywords: Agent Method; Automatic Simulation Framework; Game Theory

(Submitted on April 11, 2017; Revised on June 13, 2017; Accepted on August 21, 2017)

© 2017 Totem Publisher, Inc. All rights reserved.

1. Introduction

A rational and efficient regime can collect a flock of selfish individuals into a group with strong productivity because it gives individuals the payoff reference to deal with problems. When builders of a regime neglect some details, invalidity and instability occurs. On the one hand, no one wants to obey the regime to make efforts in the right direction; on the other hand, the inner friction increases quickly. In the typical case of prisoner's dilemma, for example, no matter how players communicate with each other before the game, the result of defection is unavoidable, which leads to the lowest total payoff of a group. In the running process of a practical regime, this kind of defection can be represented as low job enthusiasm and disharmony. Regimes with loopholes, therefore, will seriously hamper the normal and efficient operation of a company.

Game theory is the study of decision among strategies between two or more players in a game. In natural evolution and social development, cooperation always plays a very important role. Some cooperation is for the purpose of altruism, but most cooperation reaches a win-win relationship. Through the framework of game theory, we can find the conditions to make more efficient cooperation to reach win-win status. As reference of cooperation, regimes make the win-win status last longer. But any loophole in regimes can attract selfish individuals to pursue their own best interests, which damages the total payoff of group. Therefore, to some degree, these kind of regimes are invalid and unstable, even harmful.

In this paper, we propose an automatic simulation framework to find loopholes in a regime. Base on the principle of “winners stay, losers change”, we use a flock of agents to do evolution simulation according to the regime, actually, each player's payoffs. Losers will change their choice of strategies in next turn randomly, and winners stay their strategies. After several competitions, we get a figure, in which we can find the loopholes in the regimes by tracing the decision path of selfish players. Different results have different meanings in actual regimes, so we need to analyze the results to find the actual loopholes. We will make some regimes in this paper based on the classical case, “Boxed Pigs”, and use our method to find loopholes in given regimes.

* Corresponding author.

E-mail address: Wuyun_v@126.com.

2. Framework

In our framework, there are two main classes: game class, which controls the whole process of the game, and agent class, which controls the decisions among strategies of each agent. Then, we give a simple introduction to our framework, in which we assume that there are two agents in the game, and each agent makes a choice between two strategies.

2.1. Basic Parameters

2.1.1. Payoff Matrix

We need extract the main payoff relationship from given regime, because the regime itself cannot participate in computation [3,5]. Payoff matrix is show as Table 1.

Table 1. Payoff Matrix

Payoff_Matrix		Player B	
		Strategy 0	Strategy 1
Player A	Strategy 0	3, 1	4, 0
	Strategy 1	0, 4	0, 0

In the payoff matrix, if Player A uses Strategy 0, and Player B uses Strategy 0 at the same time, then the payoff of Player A is 3, and the payoff of Player B is 1. And we need change the form of payoff matrix into list, because the program cannot deal with the matrix directly. The list form of payoff matrix is show as below:

$$Payoff_Matrix = [(3,1),(4,0)],((0,4),(0,0))] \tag{1}$$

In the basic description of our framework, we use determined values to describe the payoff relationship in regimes. But in practical applications, it is often hard to know all determined values. Instead we can find some functional relations between the payoffs and strategies, so there is still much room to enrich our framework. In this paper, we just make a description about the simple form of our framework, and focus on the architecture of our framework.

2.1.2. Repeated Times of the Game in Each Generation

When the repeated times of the game in each generation is set as 1, it means the strategy of each agent will be changed after one time of game [2,11]. But in practical applications, players often use Strategy 0 at some probability a , and use Strategy 1 at probability $1 - a$, so it forms a mixed strategy $(a, 1 - a)$. In this paper, we mainly discuss this kind of mixed strategy. For an instance, when we talk about the case of cooperation, players in each generation are more likely not to cooperate with others at the rate of 100%, maybe they can choose defection in a low rate, maybe 10%, which means that in one generation of game, one can choose cooperation at the rate of 90% and choose defection at the rate of 10%. This kind of description in probability is more fit for modeling regimes, and it turns discrete choices into continuous problems, which is benefit for finding potential laws in regimes. Our framework will generate a series of game steps by the probability in player’s strategy randomly. For example, if the repeated times of the game in each generation is set as 10, and the players use the strategy of $(0.2, 0.8)$, then the game steps can be portrayed as 1111100111, or 1110101111, or other sequence that meets probability in the strategy. Each side of the game uses one’s own strategy, and our framework will generate game steps randomly, and calculate the final payoffs as the results of this generation of game.

2.1.3. Generations of the Game

In each generation of game, losers will change their strategies in the next generation, and winners will remain their strategies to their offspring [14]. This is called “winners stay, losers change”, which is a common principle in most forms of game. In practical applications, we can follow different principles in different problems. In this paper, we make a description about this simple principle. After several generations, we can get the history of agent’s strategies, which is called decision path, from which we can find the loopholes in the given regimes. To some degree, in our framework, we take regimes as games, and take participators as players. We can find the loopholes in the regimes by tracing the decision path of selfish players. It is just like finding a broken egg by tracing a fly.

2.2. Game Class

Game class has three main functions.

2.2.1. Control the Process of the Game

First of all, our framework will set initial strategies of agents, for example, (0.4,0.6), which means agent will use Strategy 0 at probability 0.4, and use Strategy 1 at probability 0.6. Different agents will get one's own initial strategies, and our framework will generate game steps according to initial strategies. After one generation of game, losers change their strategies and winners remain their strategies in the next generation [4,10]. This repeats until the number of generations reaches a given value. Each agent will record the history of its own strategies in the process.

2.2.2. Normalize Payoffs

For different problems, the payoffs may be very large or very small [9]. We can map the payoffs into a range of [0,1], because we focus on the change process of the payoffs. Hence, this kind of normalization will not affect the relative change process of the payoffs, and there is no need to change the parameters when the given regime is changed. In this paper, we choose a kind of normalization is show as below:

$$\text{Current_Payoff} = \frac{\text{Real_Payoff} - \text{Min_Payoff}}{\text{Max_payoff} - \text{Min_Payoff}} \quad (2)$$

2.2.3. Draw Figures as Needed

Our framework can draw figures about decision path and the change of payoffs by using matplotlib, a python plotting library, which can produce publication quality figures. In this paper, we just talk about the meaning of decision path, and give some reasonable explanations on given problems. More kinds of figures can be drawn in further work.

2.3. Agent Class

Agent class has two main functions.

2.3.1. Change Strategies Based on the Results of Former Generation of Game

In this paper, we use the method of random adjustment to change strategies. If an agent loses in one generation of game, it will change its strategy into any other floating numbers in the range of [0,1]. For example, if the former strategy is (0.2,0.8), and the agent loses in former generation of game, then its strategy can be changed into (0.5,0.5). This process is random. In practical applications, we can use different adjustment methods in different situations. For instance, we can use the method of small-scale random adjustment, considering the low speed of human's reaction. We will propose more viable strategy adjustment methods in future work, and we can also do some research on how agents learn to change their strategies based on the history of game, which will be good to the development of machine learning.

2.3.2. Record the Information of Agents in the Process of Game

Agents can record the new strategy when the strategy is changed, and record the payoffs of the new strategy in one generation of game [15]. As for different problems, it can record more details. In this paper, it only records the process of the adjustment of its strategies, namely decision path. The extra record of payoffs lets the results more expressive. In more complicated cases, agent can also record the strategies of its opponents. As long as the information can be obtained in the process of game, it can be recorded by agents, which will be used in loopholes identification or any other practical applications. It is more significant when we use data mining methods to dig the records of agents to find valuable clues in some typical fields.

3. Our Framework in Symmetrical Games

The results of symmetrical games have been proved mathematically [12], so we can use the cases of symmetrical games to observe the rationality of our method. Since our method does not apply to whether the game is symmetrical and whether the players are rational, our method can be applied on a larger scale since it only considers the payoffs of the game,. Without complicated mathematical tools, it just needs to set the payoffs list and run, and the result figures can explain the problems

to a certain extent. In this section, we will apply our method to symmetrical games to illustrate the rationality of our method. It should be emphasized that we explain the rationality of our method through the way of example verification, instead of mathematical proof, so the preciseness is not enough but can deal with some sorts of problems. We will use four typical cases in symmetrical game to verify our method, and they are prisoner's dilemma, anti-coordination game, coordination game and harmony.

First of all, we assume that we are discussing symmetrical game, and the players in game are totally rational. In addition, we set the repeated times of the game in each generation as 100, and the number of generations in the game is 500. The circles in the final figures represent the decision path of player A, and the squares represent the decision path of player B. In order to make the figures clear, we just record the probability of each player's choice on strategy 0, because the sum of probability of player's choices on strategy 0 and strategy 1 is 1.

Then, since we are talking about symmetrical game, we just focus on the payoffs of Player A. The payoff matrix of player A is show as Table 2.

Table 2. Payoff Matrix of Player A

Payoff_Matrix		Player B	
		Strategy 0	Strategy 1
Player A	Strategy 0	a	b
	Strategy 1	c	d

Since the payoffs only depend on the selected strategy in symmetrical game, we can change payoff matrix into the form of Table 3.

Table 3. After Change of Payoff Matrix of Player A

Payoff_Matrix		Player B	
		Strategy 0	Strategy 1
Player A	Strategy 0	0	b-d
	Strategy 1	c-a	0

In Table 3, we let the first column of payoff matrix minus a , and the second column of payoff matrix minus d . Since the payoffs only depend on the selected strategy in symmetry game, such changes will not affect the tendency of the selection of the players on strategies. Then, we do a further abstraction and get a payoff matrix as

Table 4.

Table 4. Abstraction of Payoff Matrix

Payoff_Matrix		Player B	
		Strategy 0	Strategy 1
Player A	Strategy 0	0	n
	Strategy 1	m	0

In

Table 4, if $n < 0$ and $m > 0$, the game is called prisoner's dilemma in game theory, and if $n > 0$ and $m > 0$, the game is called anti-coordination game, and if $n < 0$ and $m < 0$, the game is called coordination game, and if $n > 0$ and $m < 0$, the game is called harmony. Then, we discuss the four cases respectively.

3.1. Prisoner's Dilemma

Prisoner's Dilemma[1] is a classic case in game theory, and it was formalized by Albert W. Tucker, who presented it as follows:

Police arrest two suspects, but there is no enough evidence to charge the two suspects guilty. So police imprison the two suspects separately, and they cannot exchange messages with each other. Then, police give them choices as follows:

- A) If the two suspects stay silent (we take this choice as cooperation), they will be imprisoned for 1 year.
- B) If the two suspects confess their crime (we take this choice as defection), they will be imprisoned for 3 years.
- C) If one suspect confesses his crime, and the other stays silent, the former will be released immediately and the latter

will be imprisoned for 5 years. The payoff matrix is show as Table 5.

D)

Table 5. Payoff Matrix of Prisoner's Dilemma

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	-1, -1	-5, 0
	S1(defection)	0, -5	-3, -3

Since imprisonment is a loss to human, we take payoffs as non-positive numbers. Then, for each player, we know:

A) If the other side stays silent, I will choose to confess my crime to let myself be released, so I will choose defection.

B) If the other side confesses his crime, I will choose to confess my crime to minimize my loss, so I will choose defection.

So, both rational players will choose defection. We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path show as Figure 1.

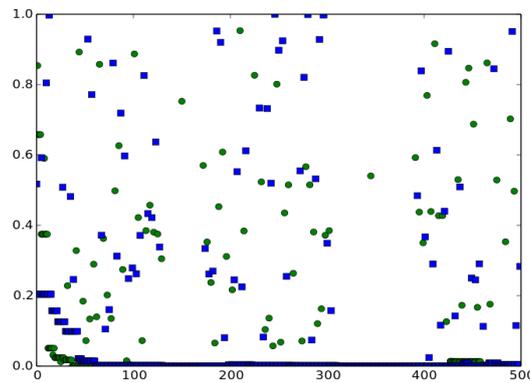


Figure 1. Decision Path of Prisoner' s Dilemma

In Figure 1, we find that although players try to cooperate with each other (let the probability of their choice on strategy 0 close to 1), but the result is most of their choices near 0, which means the two players both tend to strategy 1, namely defection. So, we can get expected results by using our framework.

3.2. Anti-coordination Game

The typical case of anti-coordination game is snowdrift game.

One night in a snowstorm, two persons went home together, and they were blocked by a big snowdrift[8]. We assume the payoff of shoveling snow is c , and the clear road will bring each person benefit of b , then:

A) If the two persons cooperate with each other to shovel snow (we take this choice as cooperation), each person's payoff will be $M = b - c / 2$

B) If the two persons both don't want to make efforts to clear the road (we take this choice as defection), each person's payoff will be $P = 0$

C) If one person shovels snow, and the other lays off, the former's payoff will be $N = b - c$, and the latter will be $L = b$.

Obviously, in normal situation, $L > M > N > P$. We assume b is equal to 10, and c is equal to 8, then the payoff matrix is show as Table 6.

Table 6. Payoff Matrix of Anti-coordination Game

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	6, 6	2, 10
	S1(defection)	10, 2	0, 0

Then, for each player, what we know is shown as below:

A) If the other side shovels snow, I will choose to lay off to enjoy the fruits of others' work, so I will choose defection.

B) If the other side lays off, I will choose to shovel snow, because I want to go home quickly, so I will choose cooperation.

So, rational players will look at the choice of each other. If the other side shovels snow, I will have a rest; if the other side lays off, I will shovel snow. We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path show as Figure 2.

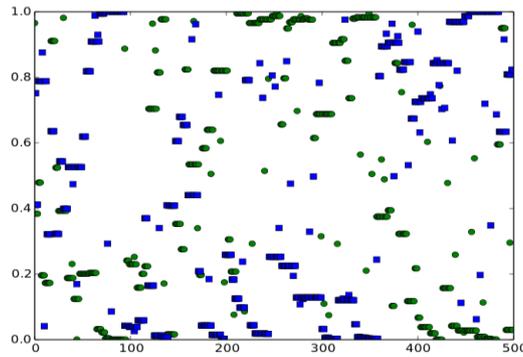


Figure 2. Decision Path of Anti-Coordination Game

In Figure 2, we find that, in most cases, players will take opposite actions, although sometimes they enjoy a short cooperation. Therefore, we can get expected results by using our framework.

3.3. Coordination Game

A case about coordination game is battle of the sexes:

There are a couple going on a vacation, the husband wants to watch a basketball game, and his wife wants to go shopping, but they want to stay together, instead of being separated[6]. So:

A) If the couple both watch a basketball game (we take this choice as cooperation), the payoff of the husband is 10, and the wife is 5.

B) If the couple both go shopping (we take this choice as defection), the payoff of the husband is 5, and the wife is 10.

C) If the husband watches a basketball game, and his wife goes shopping, they will be both dissatisfied, so their payoffs are all 0, and vice versa.

D) The payoff matrix is show as Table 7.

Table 7. Payoff Matrix of Coordination Game

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	10, 5	0, 0
	S1(defection)	0, 0	5, 10

Then, for each player, what we know is shown as below:

A) I can get the maximum benefit by making the same choice with the other side.

So, rational players will look at the choice of each other. If the other side watches a basketball game, I will watch a basketball game; if the other side goes shopping, I will go shopping. We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path show as Figure 3.

In Figure 3, we find that, in most cases, players will make the same choice with each other, although sometimes they try to make different choices. So, we can get expected results by using our framework.

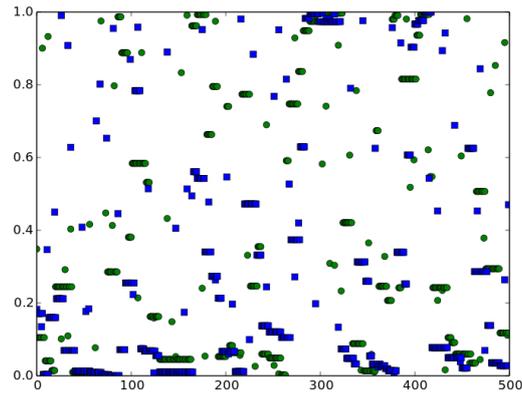


Figure 3. Decision Path of Coordination Game

3.4. Harmony

The typical case of harmony[7,13], which is relative to prisoner’s dilemma, is deadlock, whose payoff matrix is show as

Table 8.

Table 8. Payoff Matrix Of Harmony

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	-1, -1	0, -3
	S1(defection)	-3, 0	-2, -2

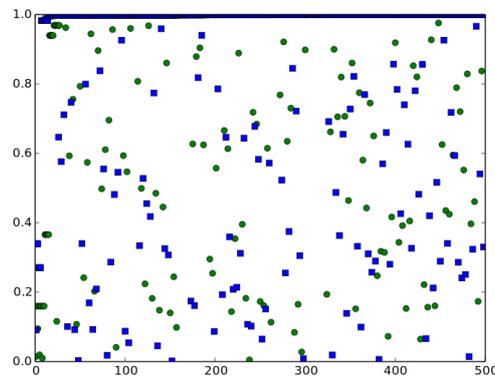


Figure 4. Decision Path of Harmony

Then, for each player, what we know is shown as below:

A) If the other side cooperates with me, I will choose cooperation to minimize my loss, so I will choose cooperation.

B) If the other side defects, I will choose cooperation to avoid any loss, so I will choose cooperation.

So, both rational players will choose cooperation. We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path show as Figure 4.

In Figure 4, we can easily find that, the two players both tend to cooperation. So, we can get expected results by using our framework.

4. Experiments in Loopholes Identification

In this section, we are going to make some regimes based on the classic case “Boxed Pigs”, and use our framework to find

the loopholes in these regimes.

Each company has strong technical experts, and it also has new recruits. So how to make effective regimes to let employees work together and share the fruits is worthy of study. Awful regimes let technical experts do not want to work, and let new recruits have more chance to thumb a ride. But good regimes can let them carry out their duties and finish their own work well.

In the traditional version of “Boxed Pigs”, there are two pigs, a big one and a small one, in a pigsty, which has a pedal on one side and a manger with a nozzle on another side. Pushing the pedal will cost energy, which is equal to 2 units of hogwash, and the nozzle will eject 10 units of hogwash on the other side. If the big pig reaches the manger firstly, it can eat 9 units of hogwash, and if the small pig reaches the manger firstly, it can eat 4 units of hogwash. If they reach the manger at the same time, the big one can eat 7 units of hogwash.

We take the big one as player A, and the small one as player B. And we assume the action of pushing the pedal is cooperation, and the action of waiting is defection.

So the payoff matrix is show as Table 9.

Table 9. Payoff Matrix of Boxed Pigs

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	5, 1	4, 4
	S1(defection)	9, -1	0, 0

We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path show as Figure 5.

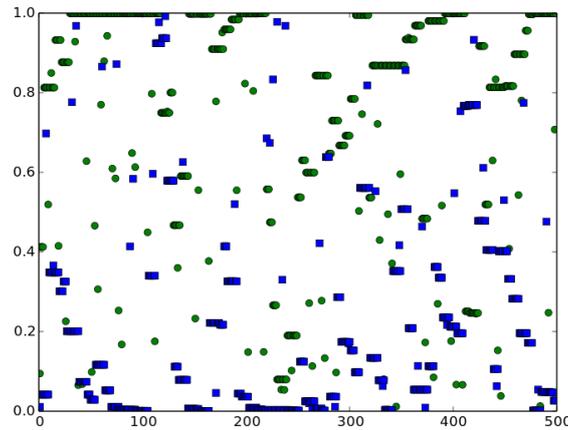


Figure 5. Decision Path of Boxed Pigs

In Figure 5, we find that, in most situations, the big one pushes the pedal while the small one waits for hogwash. Sometimes, the big one wants to have a rest but the small one still waits; both pigs have nothing to eat so the big one has to push the pedal again. The result is consistent with previous work. Then, we will research the effects of mechanism for sharing food on pigs’ choices among strategies.

Now, we slightly adjust the original model.

The pigs are still in that pigsty, and pushing the pedal will cost energy, which is equal to b units of hogwash, and the nozzle will eject a units of hogwash on the other side. But if the two pigs cooperate with each other to push the pedal, the action of pushing the pedal will cost $b/2$ units of hogwash. If the big one reaches the manger firstly, it can eat c units of hogwash, and if the small one reaches the manger firstly, it can eat d units of hogwash ($c > d$). If both of the two pigs wait, they have nothing to eat. The most important is that the other food will be allocated by pig-breeder, namely allocation

regimes.

We assume that a is equal to 10, b is equal to 2, c is equal to 5, and d is equal to 3.

Then, we make some regimes to prevent the small pig thumbing a ride, and find the loopholes in these regimes by using our framework.

Regime 1: The big one can get 80% of the other food, and the small one can get 20%. The payoff matrix is shown as Table 10.

Table 10. Payoff Matrix of Regime 1

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	7, 1	3.6, 4.4
	S1(defection)	9, -1	0, 0

We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path show as Figure 6.

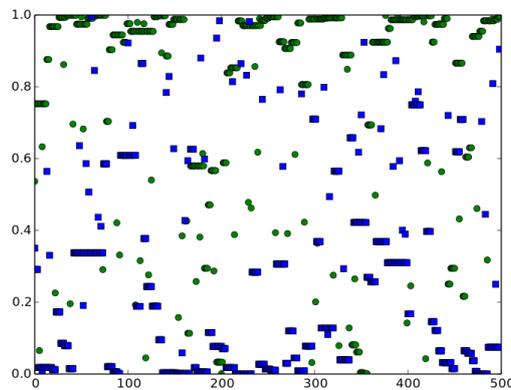


Figure 6. Decision Path of Regime 1

In Figure 6, we can easily find that the regime cannot change the situation, and the small one still thumbs a ride, which means that, if we do not think about the actual situation, we may easily make an invalid regime. Obviously, this regime almost causes no effect on the actual benefit relationship.

Regime 2: The big one can get 20% of the other food, and the small one can get 80%. The payoff matrix is shown as Table 11.

Table 11. Payoff Matrix of Regime 2

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	1, 7	-0.6, 8.6
	S1(defection)	6, 2	0, 0

We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path shown as Figure 7.

In Figure 7, we can find that the regime changes the situation totally. However, the big one does not want to push the pedal because the payoff is too low. Instead, since the small one works hard, the big one just needs to wait for food. So, just like regime 1, this regime leads to polarization, which means labor force is idle due to the unreasonable design of the regime.

Regime 3: The one who pushes the pedal can get 20% of the other food, and the other one can get 80%. If both of them push the pedal, each one can get 50%.

The payoff matrix is shown as Table 12.

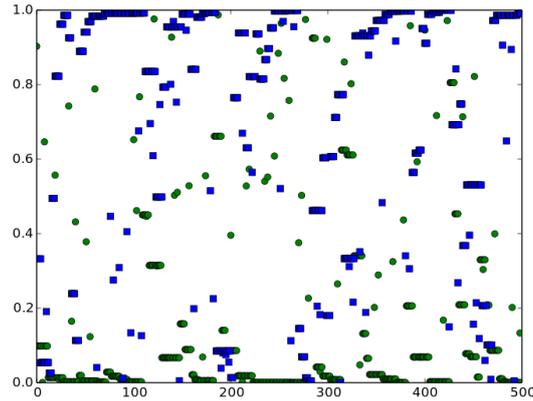


Figure 7. Decision Path of Regime 2

Table 12. Payoff Matrix of Regime 3

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	4, 4	-0.6, 8.6
	S1(defection)	9, -1	0, 0

We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path shown as Figure 8.

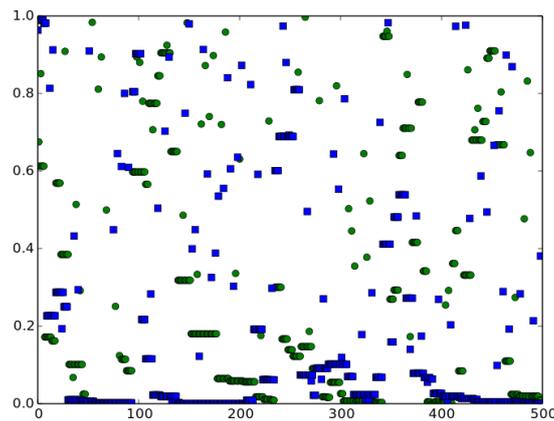


Figure 8. Decision Path of Regime 3

In Figure 8, we can find that both two pigs do not want to work. Although cooperation is good for both two pigs, but the unreasonable regime of “work more to earn less” is a hit to each pig, so both of them do not work. It is just like the typical case of prisoner’s dilemma.

Regime 4: The one who pushes the pedal can get 80% of the other food, and the other one can get 20%. If both of them push the pedal, each one can get 50%.

The payoff matrix is shown as Table 13.

Table 13. Payoff Matrix of Regime 4

Payoff_Matrix		Player B	
		S0(cooperation)	S1(defection)
Player A	S0(cooperation)	4, 4	3.6, 4.4
	S1(defection)	6, 2	0, 0

We put the payoff matrix into our framework to simulate the repeated game, and get the figure of decision path shown as Figure 9.

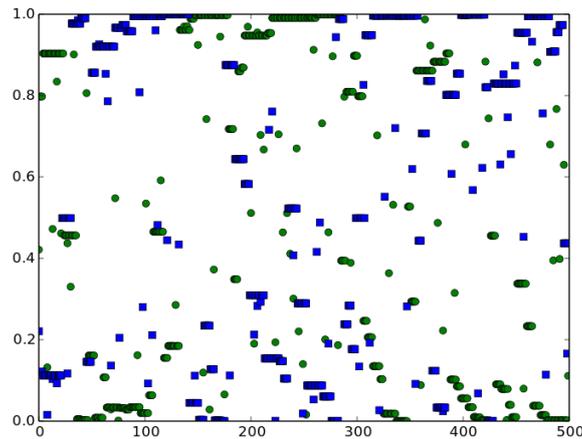


Figure 9. Decision Path of Regime 4

In Figure 9, we can find that the regime of “work more to earn more” makes both of the two pigs begin to work alternately, and sometimes they cooperate with each other, which changes the situation of “one pig always works, while another thumbs a ride”. But a reasonable and efficient regime should encourage everyone to work together, rather than alternately. Just like what we have discussed in the aforementioned section, working alternately occurs in anti-coordination game, which means there is always someone who wants to have a rest and enjoys fruits, but working alternately improve the situation of “one pig always works, while another thumbs a ride” to some extent, which is good news to managers.

5. Conclusion and Prospective Work

A simple regime simulation framework based on agent method and game theory has been proposed in this paper. We implement a simple automatic verification system to find the tendency of selfish individuals’ choices among strategies, so as to find loopholes in given regimes. The rationality of our framework has been verified by four classic cases in symmetrical game. And we try to make some regimes based on “Boxed Pigs” to prevent the small pig from thumbing a ride, use our framework to verify these regimes, and make some gains. Through our framework, managers can detect possible problems early when they make regimes, which helps reduce the loss of costs caused by management and improve team relations. However, it is just a basic framework, and there are some spaces that can be enriched and improved in future research work.

First of all, our current framework is to find the possible loopholes in regimes rather than optimize regimes, which means that we have no idea in terms of what kind of regimes has no loophole. Therefore, in the future, we will try to introduce optimization mechanisms into our framework to improve its performance. In addition, agents just consider their own payoffs when they choose their strategies, which cause constraints on practical applications. Thus, we will add mechanisms to control the relations among agents, such as comparisons and justice, which will further introduce concept of income satisfaction, which maybe have a great influence on the individual decision-making process. Furthermore, the principle of “winners stay, losers change” oversimplifies the individual decision-making process, which can lead to distortion in irrational simulation. We also need to add some psychology mechanisms into our framework to enrich our framework in the field of irrational simulation, because the fundamental purpose of our automatic verification framework is to use agent method and game theory to deal with problems which are not easy to handle in mathematical level, so as to extend the application field of computer science in management science and psychology.

Acknowledgements

This work is implemented when the authors study at Guizhou University. We thank the anonymous reviewers sincerely for their significant and valuable feedback. We further want to extend our sincere gratitude to Ruizhang Huang (Professor) for her insightful comments on this paper. Also, we are grateful to all authors of the references for their great contributions.

References

1. K. Behrstock, M. Bena ĩn, and M. W. Hirsch, "Smale Strategies for Network Prisoner's Dilemma Games," *Journal of Dynamics & Games*, vol. 2, no. 2, pp. 141-155, February 2017
2. K. Berg, "Extremal Pure Strategies and Monotonicity in Repeated Games," *Computational Economics*, vol. 49, no. 3, pp. 387-404, March 2017
3. E. R. Canterbury, "Game Theory: an Introduction," *Reference Reviews*, vol. 14, no. 7, pp. 35-36, July 2015
4. T. Chen, X. Dang, J. Han, and D. Jia, "How to Realize the Optimal Allocation of the Control Rights of the Enterprise?—Based on Evolutionary Game Theory," *Journal of Xian University of Technology*, (online since March 3, 2016) (DOI: 10.19322/j.cnki.issn.1006-4710.2016.03.022)
5. G. B. Hale and M. Obstfeld, "The Euro and The Geography of International Debt Flows," *National Bureau of Economic Research, Inc.*, 2014
6. M. Hofer, "Algorithmic Game Theory," *Springer Berlin Heidelberg*, 2015
7. M. J. Kearns, M. L. Littman, and S. P. Singh, "Graphical Models for Game Theory," *Computer Science*, pp. 253-260, 2001
8. F. Kojima, and S. Takahashi, "Anti-Coordination Games and Dynamic Stability," *International Game Theory Review*, vol. 9, no. 04, pp. 667-688, April 2007
9. D. F. Li and F. X. Hong, "Solving Constrained Matrix Games with Payoffs of Triangular Fuzzy Numbers," *Computers & Mathematics with Applications*, vol. 64, no. 4, pp. 432-446, April 2012
10. J. Lisowski, "Sensitivity of the Game Control of Ship in Collision Situations," *Polish Maritime Research*, vol. 22, no. 4, pp. 27-33, April 2015
11. H. Mintzberg, "Patterns in Strategy Formation," *Management Science*, vol. 24, no. 9, pp. 934-948, September 1978
12. M. J. Osborne and A. Rubinstein, "A Course in Game Theory," *Economica*, vol. 63, no. 249, 2008
13. C. Paternotte and J. Grose, "Social Norms and Game Theory: Harmony or Discord?" *British Journal for the Philosophy of Science*, vol. 64, no. 3, pp. 551-587, March 2013
14. E. Rovithis, A. Floros, and A. Mniestris, "Educational Audio Game Design: Kronos, a Role-playing Game," *Hellenic Conference of Education in Arts*, 2015
15. R. Tan, Y. Liu, K. Zhou, L. Jiao, and W. Tang, "A Game-theory Based Agent-cellular Model for Use in Urban Growth Simulation: A Case Study of the Rapidly Urbanizing Wuhan Area of Central China," *Computers Environment & Urban Systems*, vol. 49, pp. 15-29, 2015

Yun Wu received the Ph.D. degree from Guizhou University, Guizhou, China, in 2009. Now he is an associate professor, graduate supervisor, and the member of China Computer Society. His research interests include Distributed Computing, Game Theory, Recommender System, Big Data and its Application.

Yiqiao Li is a master student in the College of Computer Science and Technology, Guizhou University. Her current research interests include Recommender System, Distributed Computing and Data Mining.