Genetic Programming in the Coordination Game with a Chaotic Best-Response Function

Shu-Heng Chen
Department of Economics,
National Chengchi University,
Taipei, Taiwan 11623
E-mail: chchen@cc.nccu.edu.tw

John Duffy
Department of Economics,
University of Pittsburgh,
Pittsburgh, PA 15260
E-mail: jduffy+@pitt.edu

Chia-Hsuan Yeh
Department of Economics,
National Chengchi University,
Taipei, Taiwan 11623
E-mail: g3258501@grad.cc.nccu.edu.tw

Key words: experimental economics, coordination game, multiple equilibria, equilibrium selection, genetic programming, chaotic attractors

Abstract

By modeling the *coordination game* as GP (Genetic Programming)-based adaptive multiagent systems, this paper analyzes the coordination experiments with human subjects conducted by (Van Huyck et al. 1994). In the model on which their experiments were based, the coordination pattern in the equilibrium crucially depends on the learning schemes adopted by the interactive agents in the society. While, in general, we cannot exclude the possibility of chaotic-like coordination, such a result did not occur in their experiments. In our simulations, we find that GP-based learning agents can always coordinate their choice of actions so that chaotic possibilities are eliminated, which is consistent with the results of the experiments. Therefore, a GP-based multiagent adaptive system can be a starting point to facilitate our understanding towards the adaptive behavior of coordination games.

1 Introduction

One of the most unsettled issues in the current state of macroeconomics and game theory is *equilibrium selection.* In the literature, there are many examples of the rational expectations model in which equilibria are not unique. When there are multiple equilibria, it means that the physical description of the economy together with the notion of equilibrium is not sufficient to pin down to a unique predicted outcome. The connundrum of unpredictability will be even more puzzling when the qualitative properties of different equilibria are fundamentally so different that the comparative static analysis of policy effect is almost impossible. For example, it is not rare to see models with coexisting equilibria of fixed points, limit cycles and strange attractors. In this case, to know which class of equilibria is more likely to occur is certainly very useful.

Recently, researchers have started to address this issue using the experimental approach. By simulating an almost identical environment of a theoretical model, macroeconomists attempt to find their answers from experiments with real people. Some interesting findings indicate that the likelihood of different classes of equilibria might not be uniformly distributed. For instance, in (Marimom, Spear and Sunder 1993), while there are two equilibria in their model, one being a *fixed point* and the other a *periodic cycle with period 2*, the likelihood of the occurance of the latter seems to be far less than the former. Also, in a recent study of the *coordination game*, (Van Huyck, Cook, Battalio 1994) conducted experiments whose corresponding theoretical model can have fixed points, limit cycles and chaotic attractors as its equilibria. In all their experiments, however, only the interior fixed point was chosen.

While experimental approaches can contribute to the selection of equilibria, it helps us little to understand why some classes of equilibria are more likely to be chosen than others. For example, why the equilibria with lower *fractal dimension* will be more likely chosen than those with higher *fractal dimension*. Van Huyck et al. (1994) suggest that a model of the adaptive multi-agent system (MAS) would help but that it is considered a "daunting" task. In this paper, we challenge this daunting task with *the genetic programming paradigm* developed by (Koza 1989).

The use of GP in this paper is motivated by a series of recent successful applications of genetic algorithms and genetic programming to the modeling of the adaptive MAS to explain results observed in exprimental economics and experimental game theory. For instance, (Arifovic 1994) used GA-based MASs to explain the *non-smooth convergent price patterns* observed in Wellford's experiments of the cobweb model (Wellford, 1989), which cannot be explained by other existing adaptive schemes used in economics. This work was further extended by (Chen and Yeh 1995), who used genetic programming to show that the market behavior observed by Wellford can be explained as well without endowing firms with any prior information about the market. Moreover, (Arifovic 1995) used genetic algorithms to explain successfully the experimental outcomes of the coordiantion failure in the *average opinion game* conducted by (Van Huyck, Battalio and Beil 1991). Therefore, it seems natural to explore the potential of this paradigm to tackle other unsolvable problems left in other experiments such as the one by (Van Huyck et al. 1994).

The rest of this paper is organized as follows. In Section 2, the coordination game proposed by Van Huyck et al. (1994) and its important properties and associated experimental results are briefly reviewed. Section 3 describes the genetic programming applied in the context of the coordination game. Computer simulations of the GP-based coordination game are detailed in Section 4. Concluding remarks are given in Section 5.

2 The Analytical Framework

Consider the following coordination game porposed by Van (Huyck, Cook and Battalio 1994). Let $e_{i,t}$,...,$e_{n,t}$ denote the action taken by n players. Let e_t denote this action combination, and M_t the mean of e_t. The game $\Gamma(\omega)$ is defined by the following *payoff function* and *action space* for each of these n players, which are indexed by i:

$$\pi_{i,t} = c_1 - c_2|e_{i,t} - \omega M_{i,t}[1 - M_{i,t}]| \qquad (1)$$

where $\omega \in$ (1,4], $e_{i,t} \in \mathbf{E}$ = [0,1]. c_1 and c_2 are positive parameters, and |.| is the absolute value function. Furthermore, we will use the notation $e_{-i,t}$ to denote $\{e_{1,t}, ..., e_{i-1,t}, e_{i+1,t}, ..., e_{n,t}\}$. Assume that the payoff functions and feasible actions are common knowledge.

Since at time t, M_t is not available, the decision about optimal $e_{i,t}$ must be based on the expectation of M_t, i.e., $\widehat{M_{i,t}}$. Given $\widehat{M_{i,t}}$ and the payoff function $\pi_{i,t}$, the expected payoffs of firm i at time t ($\widehat{\pi_{i,t}}$) can be expressed as follows:

$$\widehat{\pi_{i,t}} = c_1 - c_2|e_{i,t} - \omega\widehat{M_{i,t}}[1 - \widehat{M_{i,t}}]| \qquad (2)$$

Given $\widehat{M_{i,t}}$, $e^*_{i,t}$ is chosen at the level such that $\widehat{\pi_{i,t}}$ can be maximized and is given by

$$e^*_{i,t} = \omega\widehat{M_{i,t}}(1 - \widehat{M_{i,t}}) \qquad (3)$$

Once $e^*_{i,t}$ is decided, M_t is determined as follows.

$$M_t = \frac{\sum_{i=1}^{n} e^*_{i,t}}{n} \qquad (4)$$

Given M_t, the actual payoff of player i at time t ($\pi_{i,t}$) is:

$$\pi_{i,t} = c_1 - c_2|e^*_{i,t} - \omega M_t[1 - M_t]| \qquad (5)$$

An action combination e^* constitutes a strict equilibrium if it satisfies the following *mutual best-response condition*:

$$\pi_i(e_i, e^*_{-i}) < \pi_i(e^*_i, e^*_{-i}) \qquad (6)$$

for all $e_i \in$ [0,1] and for all i. An observed action combination is a mutual best-response outcome if it satisfies (6). An action combination is a *symmetric equilibrium* if it satisifies condition (6) and assigns the same action to all players. Notice that *all the strict equilibria of* $\Gamma(\omega)$ *are symmetric*. Hence, it is convenient to denote the equilibria by the ordered pair (e, M). The requirement that $\omega \in$ (1,4] results in two strict equilibria: a *corner equilirbium* (0,0) and an *interior equilibrium* (1-$\frac{1}{\omega}$, 1-$\frac{1}{\omega}$).

There is an interesting intuitive interpretation of this coordination game. Suppose player i can foresee M_t, i.e., the mean choice of action, then according to Equation (3) the best action chosen by player i should not be M_t unless M_t is exactly 0 or $1 - \frac{1}{\omega}$. Therefore, except in a set with *Lebsgue measure* 0, the rational player i will not choose the "popular" action; instead, he wants to be unique in the way described by Equation (3).

Given the coordination game described above, Van Huyck et al. (1994) address the issue of equilibrium selection by using the models of the *adaptive reprsentative agent* associated with different adaptive schemes. They find that, given $\Gamma(\omega)$, the long-run behavior will crucially depend on the adaptive scheme chosen to model the representative agent. For some adaptive schemes, M_t will converge to $1-\frac{1}{\omega}$, for others, schemes,

M_t will converge to a *chaotic attractor.* To see the possibility of the later, consider the *myopic best-response dynamic scheme* analyzed by Van Huyck et al. (1994):

$$\widehat{M_{i,t}} = M_{t-1} \quad \forall\ i \tag{7}$$

$$e^*_{i,t} = \omega M_{t-1}(1 - M_{t-1}) \quad \forall\ i \tag{8}$$

Equations (7) and (8) together state that all players expect that the mean choice of action will remain constant to its previous level M_{t-1} and they react to this expectation in an optimal way. Given Equations (7), (8) and (4), the realized mean choice of action M_t can be written as a *logistic map.*

$$M_t = \frac{\sum_{i=1}^{n} e^*_{i,t}}{n} = \omega M_{t-1}(1 - M_{t-1}). \tag{9}$$

The mathematical properties of the logistic map are well known. The long-run behavior is determined by the choice of ω. The property known as *bifurcation* implies that different choice of ω will, qualitatively, result in quite different dynamics of M_t. For example, if ω is set to be 2.47222, then M_t will converge to 0.59551. But, when ω is set to be 3.86957, M_t will very likely converge to a chaotic attractor. This explains why, theoretically, the chaotic attractor could be a possible result of this coordination game. However, despite the theoretical possibility of the existence of chaotic attractors, in all their eight experiments with human subjets, including two of $\Gamma(2.47222)$ and six of $\Gamma(3.86957)$, M_t converges to the interior strict equilibrium $1 - \frac{1}{\omega}$ and the chaotic attractor is not observed in any of these experiments.

The observed inconsistency between theory and practice demands an explanation. The explanation proposed by Van Huyck et al. (1994) is that players do not follow the myopic best-response dynamic scheme and they are using some other adaptive schemes such as the *recursive simple average adaptive scheme.* However, this explanation does not go without criticisms. A minor one is offered by (Arifovic 1994), who has shown that, despite its convergence to a steady fixed point, the convergent patterns simulated according to the *recursive simple average learning scheme* are too smooth compared with the actual patterns observed in experiments with human subjects. This problem, known as *excessive smoothness*, can also be found in Van Huyck's experiments by a careful inspection of the convergence paths presented in their Figures 9-14.

The major criticism comes from the fact that if all players use the *recursive simple average learning scheme*, then given the same history of M_t, they should come up with identical expectations and thus take identical actions. But, this is not what we observe in their experiments (See their Appendix B). On the contrary, players always take different actions at the early stages of each experiment, which evidences that players do not hold identical expectations. In sum, models based on the representative agent system with identical adaptive schemes are simply so unrealistic that it is hard to take any explanation from them seriously. By contrast, models based on multiagent systems are preferable in that they allow for heterogeneity and interactive exploration. This is also noticed by Van Huyck et al. (1994), who, however, consider this direction a "daunting " task.

In the following, we shall show that modeling the coordination game by adaptive multiagent systems (MASs) is not that daunting and that adaptive MASs can provide us with a solid foundation to understand the adaptive behavior observed in the experiments. To do this, genetic programming developed by (Koza 1989) is used to model adaptive MASs.

3 Population Learning via Genetic Programming

This section provides a brief description of the way we apply genetic programming to modeling population learning in the coordination game. The version of genetic programming used here is the *simple genetic programming* which has been detailed in (Koza 1992).

Let GP_t, a population of trees, represent a collection of players' forecasting functions. A player i, $i = 1, ..., n$, makes a decision about its action for time t using a tree, $gp_{i,t}$ ($gp_{i,t} \in GP_t$), a *parse tree* written over the *function set* and *terminal set* which are given in Table 1. In this paper, all simulations conducted are based on the terminal set which includes the ehpemeral random floating-point constant R ranging over the interval [-9.99,9.99] and the mean choice of action combination lagged up to h periods, i.e., M_{t-1},...,M_{t-h}. Therefore, the forecasting functions that players may use are the linear and nonlinear functions of M_{t-1},...,M_{t-h}, $gp_{i,t}(M_{t-1}, ..., M_{t-h})$. The parameter h determines players' ability to recall the past. To endow players with the capability to learn not to be myopic, h must be set large enough. In this paper, h is set to be 5.

The decoding of a parse tree $gp_{i,t}$ gives the forecasting function used by player i at time period t, i.e., $gp_{i,t}(\Omega_{t-1})$ where Ω_{t-1} is the information of the past means up to M_{t-1}. Evaluating $gp_{i,t}(\Omega_{t-1})$ at the realization of Ω_{t-1} will give the mean action predicted by player i at time period t, i.e., $gp_{i,t}$. Without any further restriction, the range of $gp_{i,t}$ is $(-\infty, \infty)$. However, since the action space for each firm is only [0,1],

Table 1: Tableau of GP-Based Learning

Population size	500
The number of trees created by complete growth	50
The number of trees created by partial growth	50
Function set	$\{+, -, \times, \%, EXP, RLOG, Sin, Cos\}$
Terminal set	$\{M_{t-1}, M_{t-2}, ..., M_{t-5}, R\}$
The number of trees created by reproduction	50
The number of trees created by crossover	350
The number of trees created by mutation	100
The probability of mutation	0.0033
The maximum length of tree	17
The probability of leaf selection under crossover	0.5
The number of generations	1000
The maximum number in the domain of Exp	1700
Criterion of fitness	Payoffs

it is not reasonable if $gp_{i,t} \notin [0,1]$. We therefore restrict $gp_{i,t}$ to [0,1] by using the *symmetric sigmoidal activation function* to map $(-\infty, \infty)$ to [0,1], i.e.,

$$\widehat{M_{i,t}} = \frac{1}{1+e^{-gp_{i,t}}} \qquad (10)$$

The *raw fitness* of a parse tree $gp_{i,t}$ is determined by the value of the player's payoffs earned at the end of time t based on Equation (5). To avoid a negative fitness value, each raw fitness value is then adjusted to produce an *adjusted fitness* measure $\mu_{i,t}$ and is given as follows.

$$\begin{aligned} \mu_{i,t} &= \pi_{i,t} + 0.25 \quad if \;\; \pi_{i,t} \geq -0.25, \\ &= 0 \qquad\qquad\; if \;\; \pi_{i,t} < -0.25. \end{aligned} \qquad (11)$$

By doing this, we are assuming that the forecasting functions $\widehat{M_{i,t}}$ which make players lose more than \$0.25 will be immediately deleted in the following genetic operations. The choice of "-0.25" is due to the following consideration. Since at the early stage of the game, players have very limited knowledge about the market, their expectations are sort of random guessing and, as a result, it is very likely that most of them could lose money. If we only consider players with positive payoffs, them the selection process can easily be dominated by those few players who luckily earn positive payoff at the initial stages. After few generations when most of players start to take off and earn positive payoffs, this protection no longer plays any effective role.

Each such adjusted fitness value $\mu_{i,t}$ is then normalized. The *normalized fitness* value $p_{i,t}$ is given in Equation (12).

$$p_{i,t} = \frac{\mu_{i,t}}{\sum_{i=1}^{n} \mu_{i,t}} \qquad (12)$$

It is clear that normalized fitness is a *probability measure*. Moreover, $p_{i,t}$ is greater for a better parse tree $gp_{i,t}$. Once $p_{i,t}$ is determined, GP_{t+1} is generated from GP_t by three primary genetic operators, i.e., *reproduction, crossover*, and *mutation*. They are described below.

1. **Reproduction:**

   Reproduction makes the copies of individual parse trees. The criterion used in copying is the normalized fitness value $p_{i,t}$. If $gp_{i,t}$ is an individual in the population GP_t with the normalized fitness value $p_{i,t}$, then in each run of selection, it will be copied into the next generation with probability $p_{i,t}$. The operation of reproduction does not create anything new in the population and the offspring generated by reproduction constitute only part of the population GP_{t+1}. As specified in Table 1, reproduction is performed on only 10% (50 out of 500) of the population. The rest of the offspring are generated by the other two operators, *crossover* and *mutation*.

2. **Crossover:**

   The crossover operation for the genetic programming paradigm is a sexual operation that starts with two parental parse trees which are randomly selected from population GP_t in accordance with the normalized fitness described above. Next, by exchanging the parts of these parents, two offspring are produced. This exchange begins by randomly and independently selecting one point in each parental parse tree using a uniform distribution described below.

   By the syntax of LISP, each point (atom) of a parse tree could be either a *leaf* (terminal) or a *inner code* (function). Thus, the point (atom) selected could either be a terminal or a function. As specified in Table 1, the probability that the crossover point is a terminal or a function is the same, i.e., one half. Given that a terminal or function is to be the point chosen for crossover, the probability that any terminal or function is chosen as the crossover point is uniformly distributed. For example, if the crossover point is to be a terminal, and then there are three terminals in the parse tree, the probability that any one of the three terminals is chosen for the crossover point is one-third (1/3). Unlike reproduction, the crossover operation creates new individuals in the populations. As shown in Table 1, 70% (350 out of 500) of the new generation is created in this way.

3. **Mutation:**

   The operation of mutation also allows new individuals to be created. It begins by selecting a parse tree $gp_{i,t}$ from the population GP_t based on $p_{i,t}$. Once a particular $gp_{i,t}$ is selected, mutation is a process of a random change of the value of a point (atom) within $gp_{i,t}$. Each point (atom) has a small probability of being altered by mutuation, which is independent of other points (atoms). As specified in Table 1, the probability used throughout this paper is 0.0033. To be a syntactically and semantically valid LISP S-expression, terminals can only be altered by the member from the terminal set and functions can only be altered by the member with the same number of argunments from the function set. The altered individual is then copied into the next generation of the population. 20% (100 out of 500) of the new generation is created in this way.

Given the GP-based adaptive agents, our computer simulations are implemented by using the values of payoff parameters from Van Huyck et al. (1994) (See Table 2). In order to induce the chaotic possibility, they considered two different coordination games, namely $\Gamma(2.47222)$ and $\Gamma(3.86957)$. They are referred to as CASE 1 and CASE 2 in this paper.

4 Results of Simulations

Simulations are conducted for Cases 1 and 2 in accordance with Tables 1 and 2. For each case, we run five simulations and each simulation is conducted for one thousand periods (generations). The time series of the mean choice of action M_t for two out of five simulations are exhibited in Figures 1.1-1.2 (corresponding to CASE 1) and 2.1-2.2 (corresponding to CASE 2). In addition, basic statistics such as the average and the standard deviations of M_t for all cases are given in Table 3. The results of our simulations is briefly descibed as follows.

Table 2: Parameter Values of the Coordination Game Used in Genetic Programming Simulations

Set	CASE 1	CASE 2
ω	2.47222	3.86957
c_1	0.5	0.5
c_2	1	1
n	500	500
e_I^*	0	0
e_{II}^*	0.59551	0.74157

e_I^*: The optimal action under the strict equilibrium $(e, M) = (0, 0)$.
e_{II}^*: The optimal action under the strict equilibrium $(e, M) = (1 - \frac{1}{\omega}, 1 - \frac{1}{\omega})$.

4.1 The Dynamics of the Mean Choice of Action

Firstly, Figures 1.1-2.2 indicate that the time series of the M_t of GP-based coordination games have a tendency towards the value predicted by the strict interior equilibrium $1-\frac{1}{\omega}$, i.e., 0.59551 for CASE 1 and 0.74157 for CASE 2. In addition, the transition to $1 - \frac{1}{\omega}$ is remarkably brief. Consider $(0.99 - \frac{1}{\omega}, 1.01 + \frac{1}{\omega})$ as a neighborhood of $1 - \frac{1}{\omega}$, for all simuations, it takes no more than 50 generations to move into this area.

Secondly, while, due to the effect of mutation, M_t does not converge to $1 - \frac{1}{\omega}$ in a strict sense, there seems to be a force to stabilize the movement of M_t in a very small niche of the interior equilibrium. In other words, GP-based coordination games have a self-stabilizing feature. These properties are also revealed by Table 3. Based on the average of the price from generation 201 to 1000, i.e., $\overline{M_b}$, the $\overline{M_b}$ of almost all simulations does not deviate from $1 - \frac{1}{\omega}$ by more than 0.5%. Moreover, by the *Lindeberg-Levy central limit theorem*, they are all statistically insignificantly different from $1 - \frac{1}{\omega}$. Also, comparing $\delta_{M,a}$ with $\delta_{M,b}$ or $\delta_{M^*,a}$ with $\delta_{M^*,b}$ for each simulation, we can see that after 200 periods of learning, the stability of all GP-based coordination games improve.

Thirdly, the chaotic attractor of $\Gamma(3.86957)$ studied by Van Huyck et al. (1994) did not happen in any simulation of CASE 2. However, by comparing $\delta_{M,b}$ or $\delta_{M^*,b}$ accross CASE 1 and CASE 2 in Table 3, we find that the standard deviations of CASE 2 seem to be larger than those of CASE 1. This difference is also revealed by visually comparing Figures 1.1-1.2 with Figures 2.1-2.2.

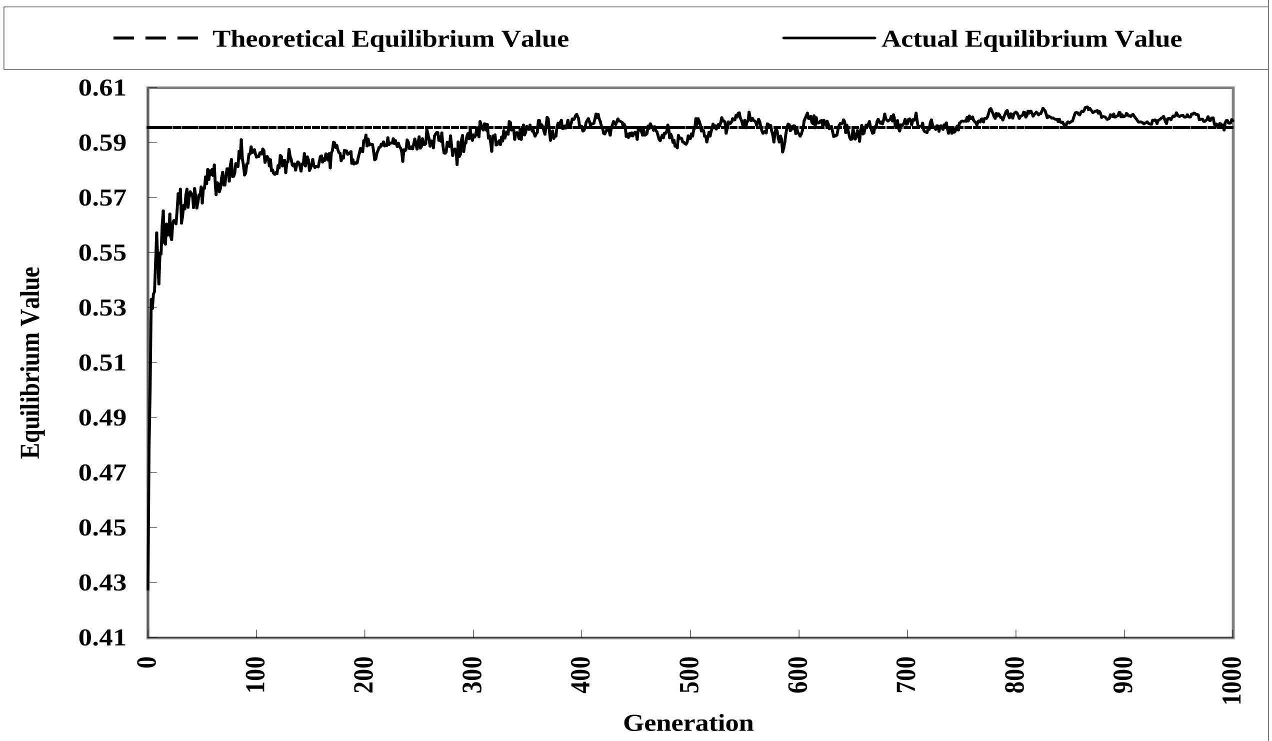

Figure 1.1 : The Time Series of the Mean Choice of Action of the Coordination Game (CASE 1) (Simulation 1.1)

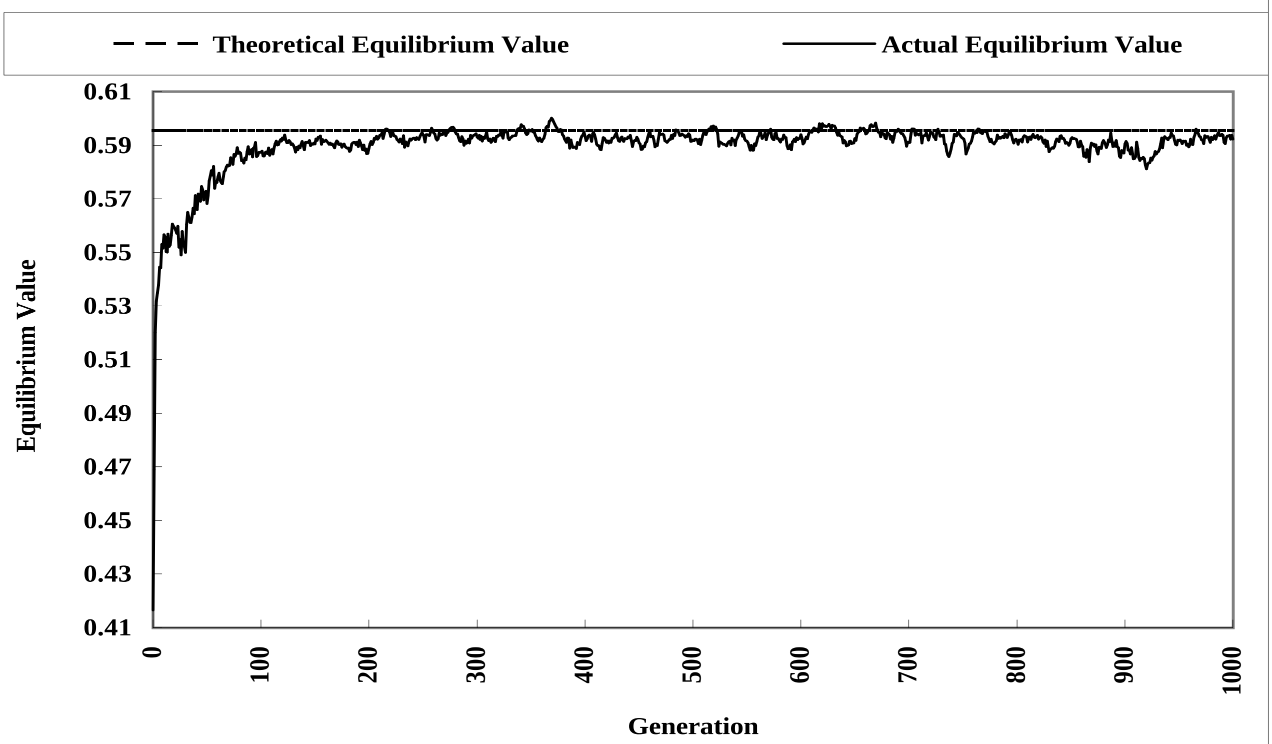

Figure 1.2 : The Time Series of the Mean Choice of Action of the Coordination Game (CASE 1) (Simulation 1.2)

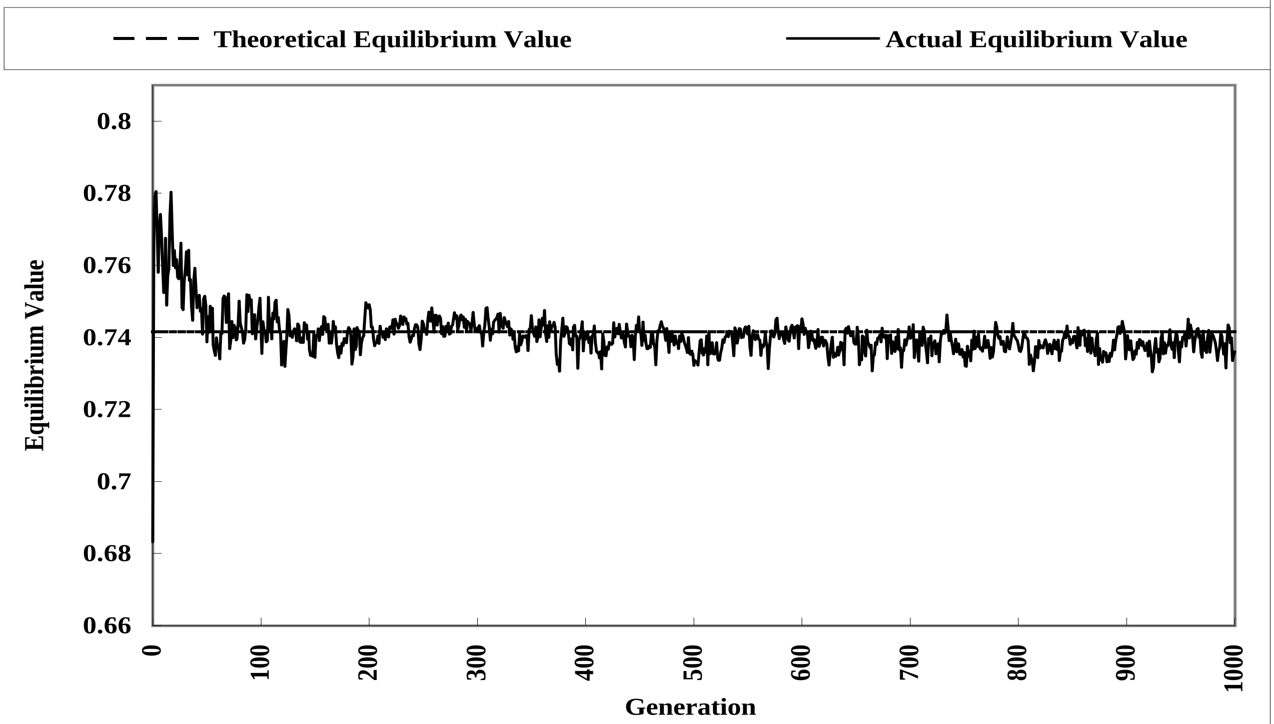

Figure 2.1: The Time Series of the Mean Choice of Action of the Coordination Game (CASE 2) (Simulation 2.1)

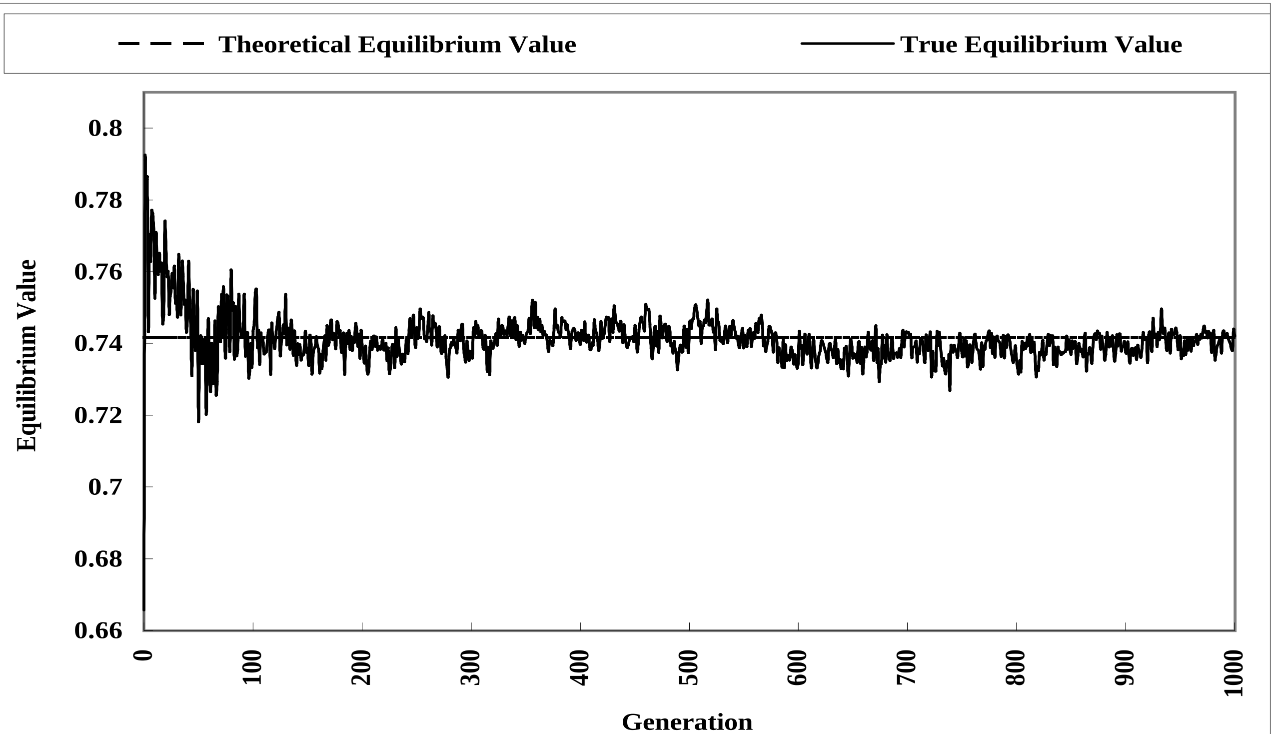

Figure 2.2 : The Time Series of the Mean Choice of Action of the Coordination Game (CASE 2) (Simulation 2.2)

Table 3: Results of the simulations of GP

CASE		Simulation 1	2	3	4	5
1	$\overline{M_a}$	0.5917	0.5897	0.5910	0.5907	0.5901
	$\delta_{M,a}$	0.0118	0.0107	0.0091	0.0094	0.0122
	$\delta_{M^*,a}$	0.0124	0.0122	0.0102	0.0105	0.0133
1	$\overline{M_b}$	0.5958	0.5925	0.5936	0.5933	0.5946
	$\delta_{M,b}$	0.0037	0.0026	0.0033	0.0028	0.0021
	$\delta_{M^*,b}$	0.0038	0.0039	0.0038	0.0035	0.0023
2	$\overline{M_a}$	0.7406	0.7411	0.7415	0.7450	0.7447
	$\delta_{M,a}$	0.0061	0.0067	0.0069	0.0064	0.0064
	$\delta_{M^*,a}$	0.0061	0.0067	0.0069	0.0073	0.0071
2	$\overline{M_b}$	0.7394	0.7403	0.7399	0.7434	0.7431
	$\delta_{M,b}$	0.0033	0.0039	0.0040	0.0034	0.0024
	$\delta_{M^*,b}$	0.0039	0.0041	0.0043	0.0039	0.0029

$\overline{M_a}$ = the average of M_t of a simulation (from Generation 1 to 1000).

$\overline{M_b}$ = the average of M_t of a simulation (from Generation 201 to 1000).

$\delta_{M,a}$ = standard deviation about the M_a of a simulation (from Generation 1 to 1000).

$\delta_{M,b}$ = standard deviation about the M_b of a simulation (from Generation 201 to 1000).

$\delta_{M^*,a}$ = standard deviation about the *strict interior equilibrium* 1-$\frac{1}{\omega}$ (from Generation 1 to 1000).

$\delta_{M^*,b}$ = standard deviation about the *strict interior equilibrium* 1-$\frac{1}{\omega}$ (from Generation 201 to 1000).

4.2 The Evolution of Population Learning

In addition to the dynamics of M_t, it is also interesting to see the evolution of players' forecasting functions, i.e., GP_t. There is a wide range of the program size of the best-of-generation forecasting function, and the program has a slight tendency to get more complicated. Take Simulation 2.5 as an example. The length of the shortest best-of-generation program is 15 and it appeared in Generations 16, 23, 27, 29, 32, 35, 37, 40, 41 and 50 (Equations (13)- (22)). Programs of this small size still frequently appeared after Gen. 50 but none of them was selected as the best individual. Instead, more and more complex programs with length over 200 were selelcted to be the best individuals and the longest best-of-generation program with length 459 appeared in Gen. 554 (Equation (23)). Also, in all simulations, there is no evidence that players were becoming homogenous. Take Simulation 2.5 as an example. At the end of Generation 1000, the size of individual programs ranges over an interval [3, 297].

$$gp_{best,16} = ((M_{t-5} + M_{t-4}) * M_{t-4}) \quad (13)$$

$$gp_{best,23} = ((M_{t-5} + M_{t-4}) * M_{t-4}) \quad (14)$$

$$gp_{best,27} = ((M_{t-5} + M_{t-3}) * M_{t-4}) \quad (15)$$

$$gp_{best,29} = (M_{t-3} * (M_{t-5} + M_{t-4})) \quad (16)$$

$$gp_{best,32} = ((M_{t-1} + M_{t-3}) * M_{t-3}) \quad (17)$$

$$gp_{best,35} = (M_{t-5} * (M_{t-5} + M_{t-4})) \quad (18)$$

$$gp_{best,37} = (M_{t-3} * (M_{t-4} + M_{t-5})) \quad (19)$$

$$gp_{best,40} = (M_{t-3} * (M_{t-4} + M_{t-5})) \quad (20)$$

$$gp_{best,41} = (M_{t-2} * (M_{t-5} + M_{t-5})) \quad (21)$$

$$gp_{best,50} = (M_{t-5} * (M_{t-4} + M_{t-5})) \quad (22)$$

$$gp_{best,554} = (SinCos((CosCosCos(CosM_{t-3} * M_{t-3}) CosSin(CosCos(Cos(CosM_{t-4}\%M_{t-4}) * Cos(M_{t-2} + SinSinM_{t-1})) + (M_{t-5} * CosM_{t-2}))) * CosCosCos Sin((M_{t-3} * M_{t-3}) - CosM_{t-3}))\%CosSinCos(((Cos (CosM_{t-1} * Cos(ExpSinCosM_{t-4} * M_{t-2}))\%CosSin CosSin(M_{t-4}\%CosSinCosM_{t-1})) * CosSinLog((Sin (M_{t-4} * CosCosM_{t-1}) * Cos(M_{t-4}\%((CosM_{t-5} + M_{t-2}) * (CosM_{t-3} * M_{t-4})))) * Cos((Cos(M_{t-3} * Cos M_{t-5}) * CosCosM_{t-4}) * M_{t-1}))) * Cos(Sin(SinCos(CosSinCosM_{t-1} * (CosSinM_{t-2} * CosCosM_{t-4})) + ExpSinCosCosSin((M_{t-2} * M_{t-3}) - CosM_{t-3})) * Cos((SinM_{t-3} + M_{t-5}) * ((M_{t-4} + (M_{t-3} + ((M_{t-2} * SinM_{t-2}) * M_{t-1})))\%CosM_{t-3}))))) \quad (23)$$

The features that the best-of-generation forecasting functions get complicated and that heterogeneous players remain to be heterogeneous may result from our choice of the *symmetric sigmoidal activation function* to map $gp_{i,t}$ into [0,1]. This map makes simple functions such as M_{t-1}, M_{t-2},... far less competitive when the mean choice of action moves closer to the interior equilibrium. This also explains why other simple functions such as (Equations (13)-(22)) can be the best only at the early stages of the simulations. Once the simple functions fail to be the best-of-generation program, it gives the tree a chance to grow up and the combinatoric nature of this expansion will make the number of potential offspring increase at an exponential rate. But, most of the potential offspring will not be given a birth if their ancestors differ widely in fitness. However, the choice of sigmoidal transformation shrinks the range of functions over the domain [0,1], which makes them not very distinguishable in fitness. This explains why after generations of evolution, there is still a wide range of size of the best-of-generation programs. In other words, it is difficult to coordinate players' forecasting functions into identical or similar ones, and so diversity remains at a significant level.

To confirm this conjecture, we replace the sigmoidal function with the *truncated linear transformation* writ-

Table 4: Results of Simulations under Truncated Linear Transformation

CASE	1	2
$\overline{M_a}$	0.5928	0.7407
$\delta_{M,a}$	0.0270	0.0171
$\delta_{M^*,a}$	0.0272	0.0171
$\overline{M_b}$	0.5954	0.7415
$\delta_{M,b}$	0.0003	0.0004
$\delta_{M^*,b}$	0.0003	0.0004

For the meaning of each notation, see Table 3

ten as follows.

$$\begin{aligned} \widehat{M_{i,t}} &= gp_{i,t} \quad if \quad 0 \leq gp_{i,t} \leq 1, \\ &= 1 \quad if \quad gp_{i,t} > 1, \\ &= 0 \quad if \quad gp_{i,t} < 0. \end{aligned} \tag{24}$$

Given this transformation, Simulations 3.1 and 3.2 were conducted for Cases 1 and 2 separately and the results are exhibited on Table 4, Figure 3.1 (CASE 1) and Figure 3.2 (CASE 2). Comparing these tables and figures with those obtained from sigmoidal function, we can see that, while the result on the aspect of equilibrium selection remains unchanged, the speed of M_t approaching the strict interior equilibrium under *truncated linear transformation* is dramatically improved. Moreover, M_t moves even in much smaller niche of the interior equilibrium. Based on Table 4, the $\overline{M_b}$ of these two simulations does not deviate from 1-$\frac{1}{\omega}$ by more than 0.01%.

As to the evolution of player's forecasting function, we also experience a quite different result. The sizes of the best-of-generation programs in general are much smaller compared with those discovered under the sigmoidal function. Take Simulation 3.2 as an example. The longest best-of-generation program appeared in Gen.2 and is given in Equation (25). The last occasion when the size of the best-of-generation program is larger than 3 is in Gen.100 (Equation 26) and after that all the best-of-generation programs are in one of the following simplest forms, M_{t-i}, i=1,2,...,5.

$$gp_{best,2} = Exp((((M_{t-3} * M_{t-5}) - ExpM_{t-1}) * M_{t-1}) - (M_{t-2}\%Log(-2.66020 + M_{t-5}))) \tag{25}$$

$$gp_{best,100} = Cos(Cos(M_{t-1})) \tag{26}$$

5 Concluding Remarks

By using Koza's simple genetic programming to modeling the coordination game as an adaptive multiagent system, we find that the behavior observed in the experiments conducted by Van Huyck et al. (1994) can be well approximated by the GP-based coordination game, especially in the aspect of equilibrium selection. The interior fixed point chosen as the equilibrium in their experiments was also chosen as the equilibrium in GP-based coordination games. Furthermore, the patterns of non-smooth convergence to $1 - \frac{1}{\omega}$ observed in their experiments were also observed in GP-based coordination games, whereas they cannot be simulated by the *recursive simple average adaptive scheme.*

Recently, by simulating a GA-based overlapping generation model, (Bullard and Duffy 1995) also found that the population of artificial agents is able to eventually coordinate on steady state and low-order cycles of inflation rates, but not on higher-order periodic equilibria. In light of the evidence of (Marimon, Spear and Sunder 1993) and (Bullard and Duffy 1995), maybe the most impressive result of this paper is that the population of artificial players does not coordinate on a chaotic attractor. This is certaintly a continuation of their evidences. However, to what extent we can generalize the conjecture that *it is easier for a population of agents to coordinate on equilibria with low fractal dimension than on those with high fractal dimension* demands further research.

References

[1] Arifovic, J. (1994), "Genetic Algorithm Learning and the Cobweb Model," *Journal of Dynamics and Control*, vol.18, no.1, January, pp. 3-28.

[2] Arifovic, J. (1995), "Strategic Uncertainty and the Genetic Algorithm Adaptation," in *Proceedings of the First International of the Society for Computational Economics.*

[3] Bullard, J. and J. Duffy (1995), "On Learning and the Stability of Cycles," In *Proceedings of the First International Conference of the Society for Computational Economics.*

[4] Chen, S and C. Yeh (1995), "Genetic Progamming Learning abd the Cobweb Model", forthcoming in Angeline, P. and K. E. Kinnear, Jr. (eds.) *Advances in Genetic Programming 2*, MIT Press.

[5] Holland, J. H. and J. H. Miller (1991), "Artifical Adaptive Agents in Economic Theory," *American Economic Reviwe*, Vol. 81, No. 2, pp. 365-370.

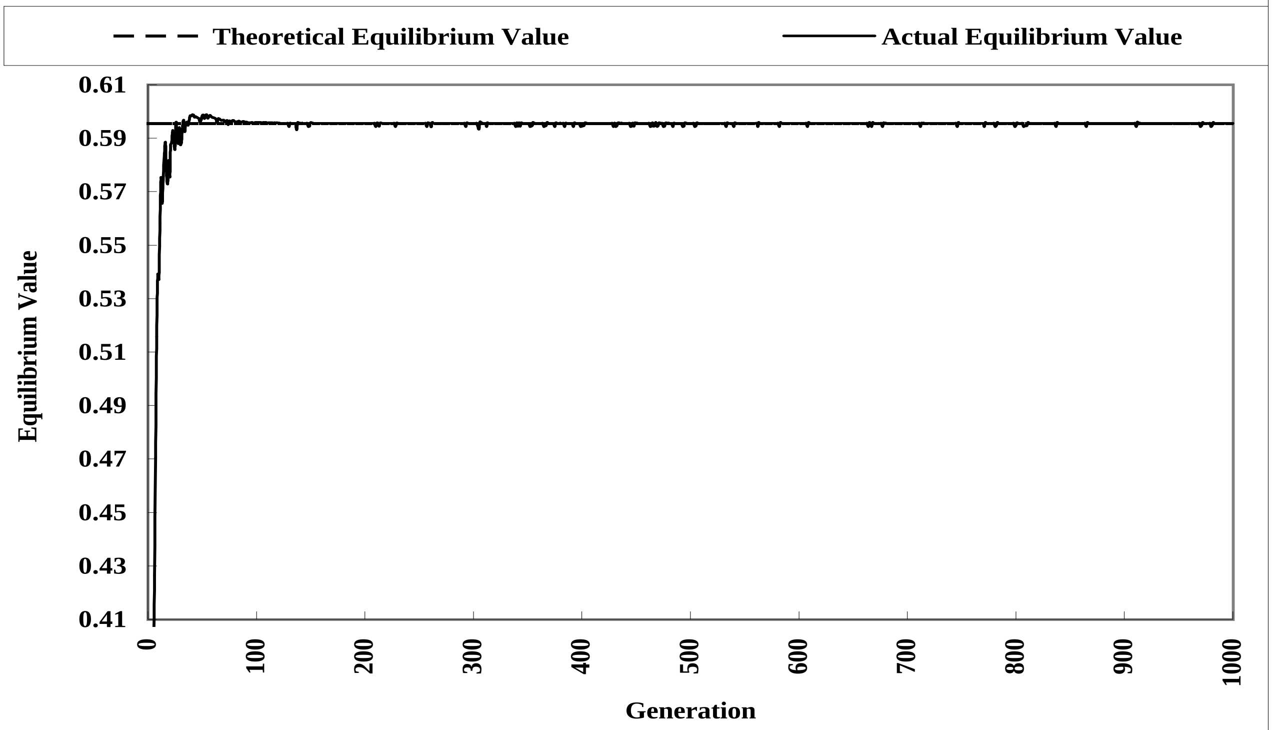

Figure 3.1 : The Time Series of Mean Choice of Action of the Coordination Game (CASE 1) (Truncated Linear Transformation)

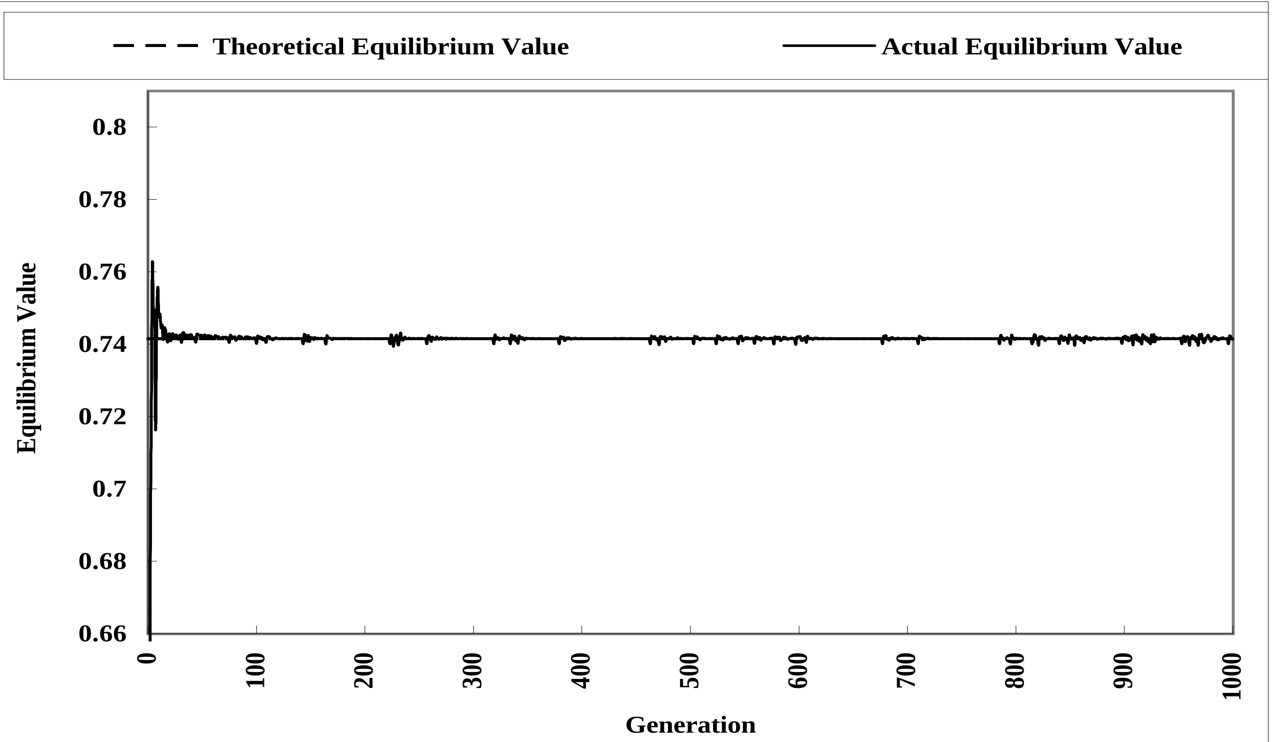

Figure 3.2 : The Time Series of Mean Choice of Action of the Coordination Game (CASE 2) (Truncated Linear Transformation)

[6] Koza, J. R. (1989), "Hierarchieal Genetic Algorithms Operating on Populations of Computer Program", in *Proceedings of the 11th International Joint Conference on Artifical Intelligence*, Morgan Kaufmann.

[7] Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

[8] Lucas, R. E., Jr. (1986), "Adaptive Behavior and Economic Theory," *Journal of Business* 59, pp. 401-426.

[9] Marimon R., S. Spear, S. Sunder (1993), "Expectationally-Driven Market Volatility: An Expremental Study," *Journal of Economic Theory*, Vol 61, No.1, pp. 74-103.

[10] Van Hyuck, J. B., J. R. Battalio, and R. Beil (1991), "Strategic Uncertainty, Equilibrium Selection Principles, and Coordination Failure in Average Opinion Games," *Quarterly Journal of Economics*, 106, pp.885-910.

[11] Van Hyuck, J. B., J. P. Cook, and R. C. Battalio (1994), "Selection Dynamics, Asymptotic Stability, and Adaptive Behavior," *Journal of Political Economy*, Vol. 102, No. 5, pp. 975-1005.

[12] Wellford, C. P. (1989), "A Laboratory Analysis of Price Dynamics and Expectations in the Cobweb Model," Discussion Paper 89-15 (University of Arizona, Tuscon, Arizona).