БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, 50
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, 50

София . 2000 . Sofia

Evolution Convergence to Equilibrium in the Repeated Prisoner's Dilemma

Vassil Vassilev, Penka Alexandrova

Institute of Information Technologies, 1113 Sofia

1. Introduction

The paper continues the line of research initiated by R u b i n s t e i n [10] and A b r e u and R u b i n s t e i n [1] using finite automata to represent strategies in infinitely repeated games and particularly the Repeated Prisoner's Dilemma (RPD).

We study the convergence to equilibrium in a 2-player RPD when the players' strategies evolve in a genetic algorithm.

Contrary to the assumption in the classical approach ([9]) we assume that the opponent's strategy is not known to the player. During the course of play he explores it and tries to achieve the equilibrium payoff. The optimization is impeded by the fact that the player can not distinguish between non-equilibrium and equilibrium play of the opponent and thus induce his optimal strategy.

Miller[8] studied the genetic algorithms for coevolution of strategies and evolving cooperation during learning. In the present paper is used a different genetic algorithm, based on the necessary conditions for equilibrium. The finite automaton representation is simplified and asynchronous learning is assumed. This makes the analysis close to the reasoning of a player, who in search of the maximum payoff updates his strategy, which leads to change of his opponents' automaton etc.

The evolution of automata in the RPD is analyzed by K i r c h k a m p [7]. A spatial model of evolution is studied where the players evolve their strategies by copying their neighborhoods'. Alternatively we assume that the players do not know the opponents strategies and optimize using a genetic algorithm.

We will present the model in section 2. In section 3 is studied the genetic algorithm methodology used with the finite automata in the 2-player 2-strategy game. Section 4 presents the experimental design used for the simulation. In section 5 are given the results and section 6 concludes.

2. The theory

2.1. Repeated prisoner's dilemma

The stage game used for the analysis is the Prisoner's Dilemma. The payoffs associated with the game are shown in Table 1. It is a two-player symmetric game, where the only Nash Equilibrium (D,D) is Pareto inferior to the cooperative outcome (C,C). The game has important implications in social sciences, politics and biology and has been extensively studied. In the finitely repeated version of the game backward induction proves that the only equilibrium strategy supports the non-cooperative outcome (D,D). Infinite repetition of the game, however, leads to multiple equilibrium strategies. Every individually rational payoff is a Nash Equilibrium of the game, which is stated in the well-known Folk Theorem (for example [9]).

P_1 \ P_2	C	D
C	3, 3	0, 5
D	5, 0	1, 1

Table 1

2.2. Finite automata

Strategies in the infinitely repeated games can be conveniently described by using automata with finite number of states (Moore automata). An automaton of player i in a 2-player game consists of:

- set of states Q_i;
- initial state $q_i^0 \in Q_i$;
- an output function $\lambda_i: Q_i \rightarrow A_i$ (associates an action with every state, where A_i is the set of stage game actions),
- transition function $\mu_i: (Q_i \times A_j) \rightarrow Q_i$ (defines the next state depending on the actions of the other player).

The modeling of a game strategies by finite automata allows to be included complexity considerations in the repeated game analysis. The underlying assumption is that the player is not only concerned with the payoff of the strategy but also tries to reduce its complexity. This issue, which is connected to the minimization of the operating cost, is a special aspect of "bounded rationality". This term describes the need to make the models closer to human decision making.

The complexity of an automaton can be defined in a number of ways. Most often it is assumed to coincide with the number of the states of the automaton, which we will also use.

Examples of some automata are given below using transition diagrams. The initial state is the circle on the left. The letter within the circle indicates the action in the

state, The letters along the arcs indicate the conditions on the other player's action in order to move to another state(s) – the transition function.

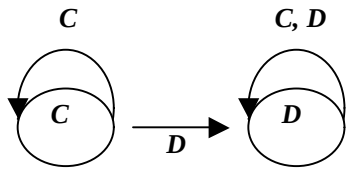

Fig. 1. "Grim" strategy

Fig. 1 illustrates the "grim" strategy: play *C* while the other player plays *C* and switch to *D* forever in case of a *D*.

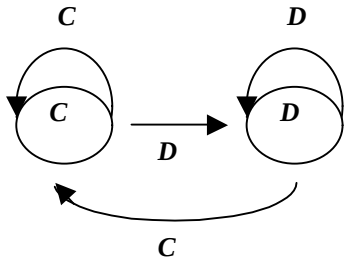

Fig. 2. "Tit for Tat" strategy

The automaton on Fig. 2 plays *C* while the other plays *C*, moves to play *D* when **D** is played and returns to the initial state in case of *C.*

2.3. The Structure of Equilibria in the Automata Game

We will cite below some results concerning the equilibrium conditions in the infinitely repeated games when the strategy space is the set of finite automata. In the repeated game the leading criterion is the payoff of the strategy, which can be calculated using different approaches (See [1]). We will apply the discounted payoff, which is defined as follows:

$$p_i = \Sigma_{t=1}^{\infty} \delta^{t-1} u_i(s^t),$$

where δ is the discounting parameter and $u_i(s^t)$ is the payoff for player i from the stage game t. The equilibrium concept, based on Nash equilibrium is defined as follows:

The automata pair (M_1, M_2) is an equilibrium of the machine game if for any fixed M_1 there does not exist another automaton M_2 for the player 2, which has higher payoff with the same number of states or provides the same payoff with fewer states. The same must hold for the M_1 automaton with M_2 fixed.

Lemma 1. In equilibrium all the states of the automaton are used.

Lemma 2. If (M_1, M_2) is an automata equilibrium:

- the states of the machines are equal,
- the sequence of states of the automata game consists of an introductory and a cycling phase and the states of the two phases are disjoint. Each state is used only once in the introductory phase, and in each cycle each state is used only once.

Proofs: See [9].

3. Genetic algorithms and finite automata

Genetic algorithms were developed by Holland[6] for optimization in complex domains and have been extensively used to characterize social learning.

In order to apply the methodology of genetic algorithms the system of output and transition functions have to be translated to binary strings.

A model used in [8] was to code sequentially the number of initial state, then the output and the transition function of the states. Let the states are N and 0 indicates "C", 1 indicates "D" action. Every state representation includes output (0,1), a number of state (1.N) to move if played 0 by the other player, a number of state (1.N) to move if played 1. So the total number of automata is $N(2.N.N)^N = 2^N N^{2N+1}$.

Here we will use a modified representation based on the assumption that the player is concerned with finding of an equilibrium strategy. As implied by Lemma 2 the automata states are distinct and all of them are reached during the play. So it is possible to permute the states so that if the equilibrium strategy is played by the opponent the next state is the number of the current plus 1, and if the last is reached the move is to the first state of the cycle. This allows us to reduce the automata number. The representation of a state includes output (0,1), a number (0,1) of the equilibrium strategy of the opponent, a number of state (1.N) to move if not played an equilibrium. The total number of automata is $N(2.2.N)^N = N(4N)^N = 4^N N^{N+1}$, which is easy to show that for N>2 increases much slower than the quantity mentioned above.

So, for example the state 0111 indicate: play C, move to the next state if the opponent plays D else move to state 4 (the numbering of the states starts from 00).

The number of bits can be changed according to the complexity of the automata.

An example of an automaton is the following (N=4):

00	0101	0110	1010	1010
cycle	state 1	state 2	state 3	state 4

This automaton plays C in states 1 and 2, and D in 3and 4. The equilibrium strategy of the opponent is the inverse – play D in states 1, 2 and C in states 3, 4. After reaching state 4 in equilibrium the automaton will move to the zero state (00). (This automaton is diagramed below).

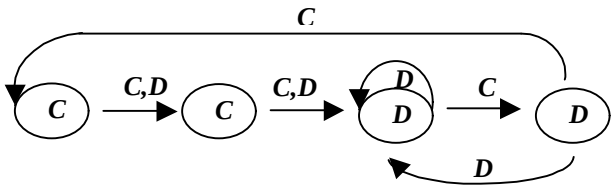

Fig. 3. "Gifts" strategy

The "Gifts" strategy achieves the highest payoff when the opponent plays the opposite symmetrically. In this kind of cooperative play the opponents make each other "gifts" – they exchange the high payoff (5) in half of the states in the cycle to the low (0) in the other.

Based on the Nash equilibrium goal we use an asynchronous genetic algorithm. The equilibrium state requires that no other strategy exists which can give a better

payoff when the opponent strategy is fixed. Therefore in searching for the equilibrium we need to fix one strategy and optimize the other and proceed analogously thereafter. In the course of analysis we will study the relative fitness of equilibrium strategies against the other.

The genetic algorithm used in the analysis of the RPD is based on a two automata population and is defined as follows (the integer g represents the number of iterations for asynchronous update):

Step 1. Initialize the two automata randomly.

Step 2. Confront the automata against each other.

Step 3. Select randomly with replacement two of the two automata of the population, with a probability proportional to their round payoff, crossover 1-point and mutate with constant mutation rate.

Step 4. Update the first of the automata

Step 5. Repeat steps 2 - 4 until the value g is reached.

Step 6. Confront the automata against each other.

Step 7. Select randomly two of the two automata of the population, with a probability proportional to their round payoff, crossover 1-point and mutate with constant mutation rate.

Step 8. Update the second automaton.

Step 9. Go to step 6 until g is reached else move to Step 2.

The algorithm describes asynchronous learning through evolution. The particular type is chosen because as clarified below the solution set contains multiple peaks of different size. The mutation rate prevents the system from converging to the nearest equilibrium and moves among the optimal payoff strategies.

In steps 3 and 7 the random choice may involve choosing twice the more successful automaton. In the course of play the opponents explore their opponent strategies. Therefore it is natural to assume that each player will use only one automaton and update it in course of play. This assumption differs from the approach used in [8] where the population size is 30.

4. Experimental design

The main goal of the research is the analysis of the convergence to the equilibrium strategy when the players asynchronously evolve their strategies using a genetic algorithm. The reasoning on the latter is that the equilibrium strategy provides the best payoff. So the players need not to check all the other strategies. The maximum size of automaton is fixed in the course of play.

While in the theory the players are optimizing their strategies in perfect knowledge of the opponents' automata we here propose an innovative approach based on evolution of the automata assuming that the players explore their opponents' strategies.

Another way to optimize the behavior for a player is to identify the opponent's strategy and after apply the knowledge of repeated games and achieve the highest payoff. The type of finite automata, however, is complex and the number of available combinations grows exponentially with the number of states. This procedure is additionally complicated by the fact that the player can not distinguish the equilibrium and non-equilibrium strategies played by the opponent.

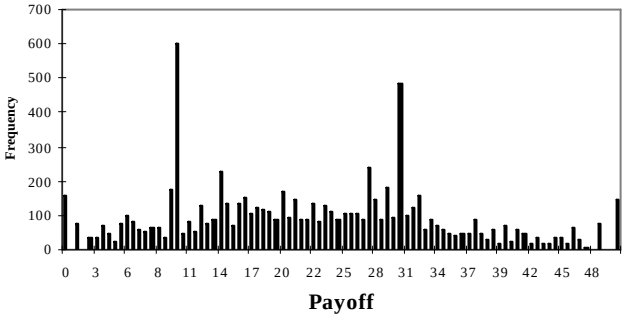

Fig. 4. Mutation rate 0.2; Complexity – 4

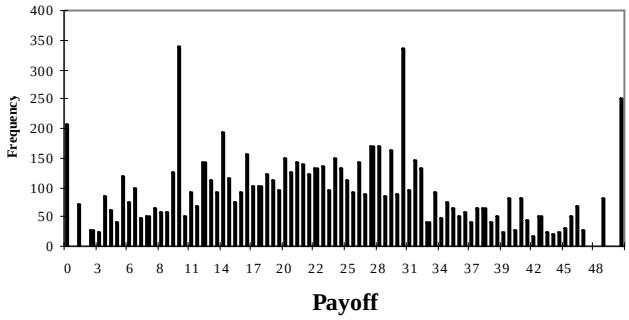

Fig. 5. Mutation rate 0.8; Complexity – 4

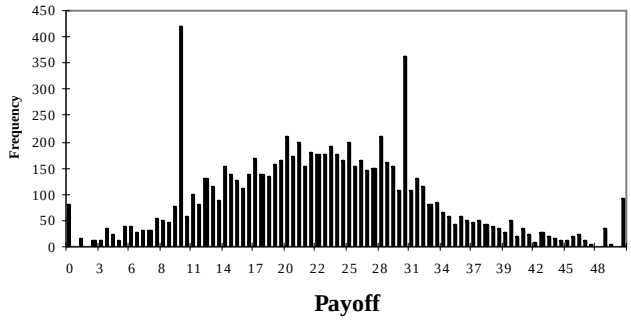

Fig. 6. Mutation rate 0.2; Complexity – 6

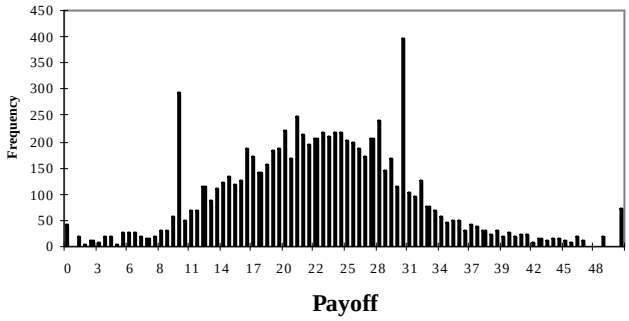

Fig. 7. Mutation rate 0.2; Complexity – 14

If a genetic algorithm procedure reaches the equilibrium (and thus highest) payoff a check the possible reduction of the complexity (number of states) of the automata. If an automaton passes this test it is inferred that it is optimal to the player. It must be noted that this is not enough in order to conclude that the pair is equilibrium. For this we need to check if fixing the new automata the opponent's is optimal. If not, another pair of automata may emerge and convergence in the process is not guaranteed.

Optimization of the automata proves to be of specific interest. As noted above the automaton's representation consists of output function, indication of the required equilibrium strategy, transition state if case of non-equilibrium play and state beginning the cycle. When the equilibrium is reached, however, the third part of the automaton is not defining. A player cannot observe in equilibrium what is the transition function in the other case. Therefore a class of N^N prove to be equivalently optimal against fixed strategy. Depending on the structure of automata other equivalence classes can emerge. The transition function, however, is useful in the course of pre-equilibrium play. It provides punishment for the player deviating from the equilibrium strategy.

An additional complication of the analysis is the existence of multiple peaks in the solution set. In the course of play different equilibria pairs can emerge. With a fixed strategy of the one player the other has some class of equivalent strategies that achieve the highest payoff. Some small perturbation, however, will cause the equilibrium to break, and cause the player to move to another equilibrium.

5. Results

In the simulation the genetic algorithm is tested using varying mutation rates and complexity of automata. Discounting parameter is set to 0,9, γ =5. For each parameter couple (mutation rate, complexity) 8000 rounds are conducted.

As predicted, play fails to converge to a steady state due to the multiple peaks existence. Equilibrium pairs can differ up to 600 %, which makes the equilibrium fragile to small perturbations. After an payoff optimal pair is reached, depending on its stability, mutation will disrupt it and the system will move to another equilibrium pair.

The results give a clear indication on the predominance of symmetric equilibrium play. Fig. 5 illustrates the results with mutation rate set to 0.2 and 4 states. The first peak on the graph indicate the constant non-cooperative play (*D, D*) with payoff 10 ($1+0.9+0.9^2+...=10$). The second shows the emergence of cooperative outcome (*C, C*) with payoff 30 ($3 + 3\times0.9+3\times0.9^2$). It is possible that the machines used by the player fail to be optimal in terms of complexity minimization. After reaching the equilibrium the players may drop some states.

Fig. 5 shows the frequencies of the payoffs when the mutation rate is increased to 0.8. This has a pronounced effect on the emerging of marginal strategies – (0, 50) (50, 0), when one of the players is certainly disadvantaged. It is intuitive that the increasing of the perturbation will widen the difference between players' payoffs.

The rise in the variation rate strengthens the vitality of cooperative outcome compared to the non-cooperative. The mutation causes the players to change faster their strategies and thus increase the fitness of the higher value equilibrium.

The only stable solution of the optimization is reached with turning off mutation. There the system converges in few steps to the best crossover between the two individuals.

With the increasing in the number of the states of the automata the 2 high peaks – 2 low peaks picture is retained (see Fig. 6 and Fig. 7). The distribution of the other payoffs approaches to the normal. The relative fitness of the cooperative outcome grows with the complexity.

6. Conclusion

In the paper is studied the convergence to an optimal strategy in the Repeated Prisoner's Dilemma modeled by finite automata. A modified genetic algorithm is used to test the evolution of the system, provided the multiplicity of equilibria and equivalent strategies. The algorithm allows the player to update their strategies asynchronously, based on the payoff equilibrium goal. The results strongly indicate the predominant use of two simple strategies by the players. The increasing of complexity and mutation rate augments the noise of the system but the two optimal pairs remain predominant. Many of the evolving strategies may not turn out to be equilibrium. The type of finite automata used provides only a necessary equilibrium condition. The analysis of the complexity minimization is, however beyond the scope of the present paper.

7. References

1. Abreu, D., A. Rubinstein. The structure of Nash equilibrium in repeated games with finite automata. – Econometrica, **56**, November, 1988, No 6, 1259-1281.
2. De Jong, K. A. Analyses of the Behaviour of a Class of Genetic Adaptive Systems. Ph.D. Dissertation, The University of Michigan, Ann Arbor, 1975.
3. Fogel, I. J, A. JOwens, A. J., M. J. Walsh. Artificial Intelligence through Simulated Evolution. New York, John Wiley and Sons, 1966.
4. Fudenberg, D, F. Maskin. The folk theorem in repeated games with discounting or with incomplete information. – Econometrica, **54,** May 1986, 533-554.
5. Goldberg, D.F., Genetic Algorithms in Search, Optimization, and Machine Learning. New York, Addison-Wesley, 1989.
6. Holland, J. Adaptation in natural and artificial systems. University of Michigan Press, 1975.
7. Kirchkamp, O. Spatial evolution of automata in the prisoners' dilemma.–Discussion Paper No B-330, University of Bonn, October 1995.
8. Miller, J.H. The Coevolution of Automata in the Repeated Prisoner's Dilemma. Working Paper, University of Michigan, 1993.
9. Osborne, M. J., A. Rubinstein. Course in Game Theory. MIT Press, 1994.
10. Rubinstein, A. Finite automata play the repeated prisoner's dilemma. – Journal of Economic Theory, **39**, June, 1986, 83-96.
11. Spears, W. The Role of Mutation and Recombination in Evolutionary Algorithms. Ph.D. Dissertation, George Mason University, 1998.
12. Spears, W. Crossover or Mutation, Working Paper. Navy Center for Applied Research in Artificial Intelligence, 1999.

Сходимость к равновесии в повторяемой дилемме заключенного

Васил Василев, Пенка Александрова

Институт информационных технологий, 1113 София

(Р е з ю м е)

Рассматривается применение генетических алгоритмов в исследовании сходимости к равновесии в повторяемой дилемме заключенного (ПДЗ). Стратегии двух участников в ПДЗ описываются при помощи конечных автоматов. Исследован ассинхронный процесс обучения автоматов, который оптимизирует игру при бесконечной ПДЗ. В работе показано, что самые расспространенные типы стратегии при симуляции симметрические и оптимальные друг к другу. Демонстрируются результаты анализа при разных коэффициентов мутации и сложности автоматов.