Stationary Equilibrium Strategies in Repeated Games with Vanishing Actions

L.C.A. Meessen

June 5, 2009

Abstract

There are two methods for finding the equilibrium strategies in repeated games with vanishing actions. The method with one restricted player which is linear programming problem that returns the pure stationary best reply for the unrestricted player to the stationary strategy of the restricted player and the value of the game for these strategies. The second method for two restricted players depends uses the fictitious play property to find the equilibrium strategies and according to these probabilities determines what the value of the game will be.

1 Introduction

The repeated games with vanishing actions in this paper, are two-person zero-sum games where one or both players have a restriction. In zero-sum games the entries in the matrix indicate the value that player 2 has to pay to player 1. So the expected average reward for player 1 is value that player 2 is going to loose during the time the game is played. The restriction in these repeated games indicates after how many times not playing an action, the action will be unlearned. Unlearning means that the action cannot be played anymore by that player for which this action was a possible option[2]. The player looses this action from it's set of possible actions. So unlearning means making an action unavailable for that player. For finding the optimal strategies, the focus lies on zero-sum games that don't contain a saddle point. If a game contains a saddle point, this will be the value of the game and the players always play the action that will lead to the saddle point. A game without a saddle point has as characteristic that the lowest entry on one diagonal is bigger than the biggest entry on the other diagonal. So playing a certain row or column all the time is not optimal for both players. For finding the strategies in the repeated games, the repeated games will be handled as stochastic games[2]. A stochastic game consists of different states and the game will be in one of the states every stage of the game. By playing the players' actions determine to which state the game goes in the following stage.

This leads to the following problem statement: 'What is the best strategy for the players, so that they can obtain an optimal game value?'. What the stationary equilibrium strategy is, that depends on the game but also on the number of restricted players in the game. If there is only one restricted player, than the best mixed strategy for both players is to find the pure stationary best reply for the unrestricted player to the stationary strategy of the restricted player. So in that case the problem statement should be interpreted as: 'What is the pure stationary best reply from player 2 (suppose this is the unrestricted player) to the stationary strategy of player 1?'. There are some theories about the value of a repeated game and what kind of strategies the players use if both players have a restriction, but there is no algorithm for proving these theories. By finding an algorithm that can calculate the value of the game and with what strategies the players play, there will hopefully be more inside in the truth or falsity of the existing theories.

In the next section there will be some background on repeated games with vanishing actions. After that there is a section which discusses the different methods that are used for finding the optimal mixed strategies in the different set-ups (one restricted player/ two restricted players) for the game. The methods are followed by experiments in which the methods of the preceding section are tested for their working. Then the results of the experiments are discussed in the result section. To conclude the conclusion of this research is given, together with an answer to the problem statement and recommendations for further research.

2 Background

A repeated game with vanishing actions is given by an $(m \times n)$-matrix, and two natural numbers r_1 and r_2. At every stage $t = 1, 2, 3, ..$ player 1 chooses a row i and player 2 chooses a column j and the matrix entry (i, j) is the payoff the players get at that stage of the game. Player 1 has the action set $\{1, 2, ..., m\}$ and if an action i

not played for r_1 consecutive stages, then this action will be removed from the action set of player 1. For player 2 is the action set $\{1, 2, ..., n\}$ and action j will be removed from player 2's action set if player 2 has not played it for r_2 consecutive stages. A repeated game can be viewed as a special kind of stochastic game. This stochastic game has a finite state space and also finite action spaces[2]. The state space consist of all the possible states in which the game can be. The size of the state space depends on the restrictions of the players and the number of possible actions they have available. The action spaces for player 1 and 2 both consist of two actions, because the research is on (2×2)-matrices.

In those repeated games, player 1 and player 2 both play with their own strategy. Player 1 tries to maximize his expected average reward, while player 2 is trying to minimize this expected average reward. The value of the game then can be calculated according to the action sequence played. If both players have restriction $r = 1$, then the value of the game is the value of the action pair selected. For the other values of r the strategies are $max_i min_j a_{ij}$ and $min_j max_i a_{ij}$ for player 1 and player 2, respectively[2]. Those strategies only apply if the game has no saddle-point. If there is a saddle-point in the game, the players will always end up playing the action pair of the saddle point. A game has a saddle point if there is for player 1 a row in which he always gets more than in the other rows. So the entries in that row are always bigger than the entries in other rows. For player 2 there is a column in which he always has to pay less than in other columns.

For a (2×2)-matrix, player 1 has to choose between the action T for playing top and the action B for playing bottom. The actions for player 2 are L for playing left and R for playing right. If player 1 has restriction $r \geq 3$ and player 2 has restriction $r = \infty$, and the matrix is

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

the value of the game can be calculated by the following formula[3]:

$$g^1 = \frac{v \cdot (a-b-c+d)^r - d \cdot (a-b)^r - a \cdot (d-c)^r}{(a-b-c+d)^r - (a-b)^r - (d-c)^r} \tag{1}$$

In this formula, v is the value of the game calculated by taking the probability p for player 1 for playing B and probability $1-p$ for playing T, if player 2 is playing $(\frac{1}{2}, \frac{1}{2})$, and solve the equation $v = a \cdot (1-p) + c \cdot p$. The r stands in formula 1 for the restriction of player 1.

There is no such formula for a repeated game with both restricted players. A different idea about finding the value in this game depends on the fictitious play property of game theory. Fictitious play is an iterative procedure. This procedure, for a normal matrix game, can be interpreted as having the players play the game repeatedly where at each stage each player is playing a best reply to the mixed action that 'fits best' with the opponents observed behavior[5]. One can start with an arbitrary action and if there are more actions that are a best reply to the observed mixed action, then a tie breaking rule is used.

For the fictitious play process in repeated games, things work a little bit different. First there are some useful notations that help to explain how the fictitious play works. Let $i^*_\tau \in \Delta^m$ be the pure action that is selected by player 1 at stage τ of the fictitious play process and f_t denote the action frequencies of the pure actions of player 1 up to stage t. Analogously are for player 2 defined $j^*_t \in \Delta^n$ and $g_t \in \Delta^n$. For the fictitious play process the values of f and g are updated as follows[3]:

$$f_t = \frac{t-1}{t} \cdot f_{t-1} + \frac{1}{t} \cdot i^*_t \in \Delta^m \tag{2}$$

and

$$g_t = \frac{t-1}{t} \cdot g_{t-1} + \frac{1}{t} \cdot j^*_t \in \Delta^n \tag{3}$$

This fictitious play process converges if $(f_t, g_t)_{t=1}^{\inf}$ converges. If every fictitious play converges to the set of stationary equilibrium strategies than the game has the fictitious play property. So a repeated game with the fictitious play property will converge to the set of stationary equilibrium strategies[4].

The fictitious property is a useful property if the repeated game has two restricted players, but not that much if only one player has a restriction. If there is one restricted player than formula 1 will show to be a nice formula for determining the game value.

3 Methods

There are two different methods for finding the optimal strategy for a player. These methods depend on the circumstances of the game. If the repeated game only has one restricted player, then the aim is to find the pure stationary best reply for the unrestricted player to the stationary strategy of the restricted player. On the other hand, if both players have a restriction than finding the stationary equilibrium strategies depends on the best play for both players.

3.1 One restricted player

In the case of a repeated game with a (2×2)-matrix and one restricted player with the restriction $r \geq 3$, the states of the stochastic game representation of the repeated game depends on the actions played by that player. If player 1 is the restricted player and he has restriction

r^1 then the state space looks like the following: $S = \{(0,0),(0,1),(0,2),...,(0,r^1),(1,0),(2,0),...,(r^1,0)\}$. In the state space, $(0,0)$ is the initial state of the game and $(0,r^1)$ and $(r^1,0)$ are absorbing states in the game for action T and action B, respectively[3]. The game starts at the initial state and then it will never return to it. By playing the game, player 1 will avoid ending up in one of the absorbing state, because than player 2 can minimize his expected average reward. For finding the pure stationary best reply for player 2 to the stationary strategy of player 1, the initial state and the absorbing states are not important, so they can be left out.

The states $(0,1)$ till $(0,r^1-2)$ all look like the following state:

$$state(0,i) = \left(\begin{array}{cccc} a & & b & \\ & (0,i+1) & & (0,i+1) \\ c & & d & \\ & (1,0) & & (1,0) \end{array} \right)$$

When player 1 plays action T, the game goes to the state $(0,i+1)$ and whenever the action B is played then the game goes to state $(1,0)$. For the states $(1,0)$ till $(r^1-2,0)$ there is a same principle, when action T is played in state $(i,0)$ the game goes to state $(0,1)$ and the game goes to state $(i+1,0)$ when action B is played. In state $(0,r^1-1)$ player 1 has only action B left, because playing action T will lead to the absorbing state $(0,r^1)$. The same holds for state $(r^1-1,0)$ but with actions T and B reversed.

In all the states, player 1 plays with the same stationary strategy. This strategy is playing T with probability $1-p$ and playing B with probability p. Finding the pure stationary best reply for player 2 thus depends on finding the probability q with which player 2 plays R. This probability differs from state to state.

To find the probability for player 2 in every state, imagine that the state space looks like a Markov chain. In this Markov chain from every state $(0,i)$, where $i = 1,2,...,r^1-1$ there is a pointer to the state $(1,0)$. For every state $(i,0)$ there is a pointer to state $(0,1)$. From every state with $i < r^1-1$ there is also a pointer to the next state. By knowing this, the cycle in the Markov chain provide a useful tool in calculating the value and the probabilities. From the Markov chain a number of equations can be formed, namely one equation for every possible cycle in the Markov chain. If in a cycle the action T is played, the term $a \cdot (1-q) + b \cdot q$ is added to the left-hand side an equation. For action B, there is added a term $c \cdot (1-q) + d \cdot q$ to the left-hand side of the equation of the cycle. The right-hand side of the equation is the value v multiplied with the number of states for which a term is included in the equation. An examples of such an equation are:

$$c + (d-c)q01 + a + (b-a)q10 = 2v$$

$$a + (b-a)q01 + a + (b-a)q10 + c + (d-c)q02 = 3v$$

In these equation the terms for action T and B are rewritten, so the difference between the variable terms and the constants is clearer. The variable $q0i$ should be explained as the probability in state $(0,i)$, with $i = 1,2,...,r^1-2$. For the states $(0,r^1-1)$ and $(r^1-1,0)$, the probability in that state is determined by the entries in the matrix. For those states, instead of entering a constant and a variable, only a constant is entered. The value for v should equal formula 1 if this formula gives a good calculation for the game value.

From these equation, it is possible to build a linear programming problem. In this LP all the variable terms are put on the left-hand side, so term containing v is brought to the other side. All the constants are put on the right-hand side. The algorithm makes a matrix A with length and width $(r^1-2) \cdot 2 + 1$ because this is the number of variables that occur in the equations. All the coefficients of the variables are entered in this matrix. Then there is made a column vector B for all the constants. By using the Matlab operator

, the linear programming problem is solved, and the q-values for the different states in the game are returned. The value of the game is returned separately from the vector with the q-values.

The same algorithm can also be used if player 2 is the restricted player and player 1 has no restriction. For that case, instead of giving the game matrix as a parameter, give the transposed matrix. The transposed of game matrix M looks like:

$$M' = \begin{pmatrix} -a & -c \\ -b & -d \end{pmatrix}$$

The algorithm returns then a vector with the p-values for player 1 in the different states and value. The returned value is negative, so it has to be multiplied by -1 to find the original value of the game.

3.2 Two restricted players

The algorithm used in the previous section is unuseful for two restricted players. For two restricted players, the equations following from the Markov chain are not linear anymore. Another way is needed to find the probability p for player 1 and probability q for player 2 in the different states of the state space. The state space looks different for two restricted players, because all the state look like one of the following: $(0,i,0,j)$ if player 1 has played i times T and player 2 has played j times L, $(0,i,j,0)$ for playing i times T and playing j times R, states looking like $(i,0,0,j)$ stand for player 1 playing i times B and player 2 playing j times L and there is another possible state namely $(i,0,j,0)$ where action B is played i times and action R j times. The state $(0,0,0,0)$ is again the initial state of the game, and the states in which $i = r^1$ and/or $j = r^2$ are still avoid because then the game will finally end up in an absorbing state.

For the algorithm the initial state and all the states in which $i = r^1$ and/or $j = r^2$ are left out, because the players assure that they will never reach such a state. From the other states the transition matrix P of the Markov chain can build. The algorithm builds a very general transition matrix, which can be added in such a way that it can be used in the different iterations for the fictitious play process. In this matrix P is probability one entered for all the states in which player 1 and/or player 2 have played an action for $r^1 - 1$ or $r^2 - 1$ times, respectively. Probability zero is entered for all the states between which no connection exists in the Markov chain. For all the states in which both players have their two actions available, the corresponding action is entered for the connection in the Markov chain. If the connection in the matrix stands for playing action TL, then TL is entered into the transition matrix, and the same goes for the actions TR, BL and BR.

For the fictitious play process, after every time step the probabilities in the different states of the game are updated according to formulas 2 and 3. Before these updates can be done, for every possible action sequence the expected average reward in the game is calculated. For player 1, the best action sequence is the sequence with which he gets the most and for player 2, the best action sequence makes sure he pays the less. The transition matrix is, in an adapted form, used for the value calculation of an action sequence. The action sequence of player 1 consist of a combination with different T's and B's. The transition matrix is adapted in such a way that if action T belongs to a state that the entry in the matrix containing action TL is replaced by the value for $1 - q$ of that state, action TR in the matrix is replace by q for that state and the actions BL and BR are replaced by a zero. For action B in the action sequence, BR is replaced by q, BL by $1 - q$ and the others are zero. The same construction works for player 2, but his action sequence is a combination with L and R. The transition matrix then fills in p and $1 - p$ for action TL and BL, respectively, if the action in the action sequence is L. If the action is R then TR and BR are replaced by the values p and $1 - p$.

After adapting the transition matrix, the algorithm calculates the steady-state probabilities of the Markov chain. The steady-state probability is the probability of finding the process in a certain state[1]. For finding the steady-state probabilities π the following formula is applied:

$$\pi \cdot P = \pi \tag{4}$$

In this formula π is vector with the property that $\sum_{k=0}^{K} \pi_k = 1$. These π_k uniquely satisfy the steady-state equations[1]

$$\pi_k = \sum_{l=0}^{K} \pi_l p_{kl}$$

This simply means the π_k for state k is equal to the sum of the probabilities in column k of the transition matrix multiplied by the π of the other states.

The matrix game also has a certain value in every state, depending on the value of p or q for that state and the action played in the fictitious play process. By multiplying the probability π for every state with the value of that state and summing this for an action sequence, the expected average reward for that action sequence is calculated. This iterative process is repeated several times until a pre-specified number is reached or untill the probabilities in the fictitious play process do not change anymore.

At the point the fictitious play process does not change anymore, there is also found the expected average reward of the game. Since the strategies for both players are optimal if there is no strategy left with which they can optimize their play. There is a value found if convergence in the probabilities and the game value occurs. If the probabilities do not converge to a certain value, then there the fictitious play property of the game is not usable for finding the equilibrium strategies. This possibility also has to be taken into account, since if there is a cycle pattern in the fictitious play process than there is no convergence to a specific point[4].

4 Experiments

In this section the experiments for testing the different algorithms are discussed. The first experiment tests the correctness of the value calculated by the algorithm for one restricted player. The next experiment tests the relation between the probability in the different states and the restriction of the player in the case of one restricted player. At the end there is an experiment for testing the algorithm for two restricted players.

4.1 Experiment 1: Correctness of value calculation

To test the correctness of the value the algorithm returns, for different game matrix the value returned by the algorithm is compared with formula 1 if player 1 is the restricted player. Also to test the working if player 2 is the restricted player, the value is compared with the next formula

$$g^2 = \frac{v \cdot (a-b-c+d)^r - b \cdot (a-c)^r - c \cdot (d-b)^r}{(a-b-c+d)^r - (a-c)^r - (d-b)^r} \tag{5}$$

game	r	player1	g1	player2	g2
$M = \begin{pmatrix} 4 & 0 \\ 1 & 2 \end{pmatrix}$	5	1.4038	1.4038	1.7432	1.7432
	10	1.5519	1.5519	1.6098	1.6098
	25	1.5985	1.5985	1.6000	1.6000
	50	1.6000	1.6000	1.6000	1.6000

Table 1: Values for different restrictions

In this formula the r stands for the restriction of player 2, since player 1 is in that case the unrestricted player. For the correctness of the algorithm, the test is also done for different restrictions to see if the restriction has influence on the value of the game.

From table 1 it follows that there linear programming algorithm estimates the value quite well. During the test there was no game matrix with a big difference between the value calculated by the algorithm and formulas 1 and 5 for player 1 and player 2, respectively.

4.2 Experiment 2: Relation between probability and restriction

For this experiment is player 1 is always the restricted player, but the value of the restriction is different all the time. The following game matrix is used:

$$M = \begin{pmatrix} 4 & 0 \\ 1 & 2 \end{pmatrix}$$

For every restriction the q-values in the different states for player 2 are calculated and plotted in a graph, to see how theses values behave and if there is a relation between the restriction and the consecutive times player 1 has played the same action.

The experiment did run the test for different values of r, but plotted for the main aim of the experiment are only the graphs of the values $r = 5$, $r = 10$, $r = 25$ and $r = 50$ in figure 1. In the figure the red line corresponds to the value of q in the states $(0, i)$ and the blue line is the value in the states $(i, 0)$ with $i = 1, 2, ..., r$. The main result that is noticed in this figure is that the bigger the value of r, the longer it takes before the probabilities in the states differ from their standard value. For the matrix game the standard value for q equals 0.6, because this is the value q gets if the player 1 plays $(\frac{1}{2}, \frac{1}{2})$.

Another noticeable fact is that for the states $(0, i)$ it takes a lot longer before they are going away from the standard probability. There against stands that that the probabilities in the states $(i, 0)$ don't change that fast from each other. In the graphs it also looks like, the more flexible the restriction for player 1 the steeper the line in the graph goes down.

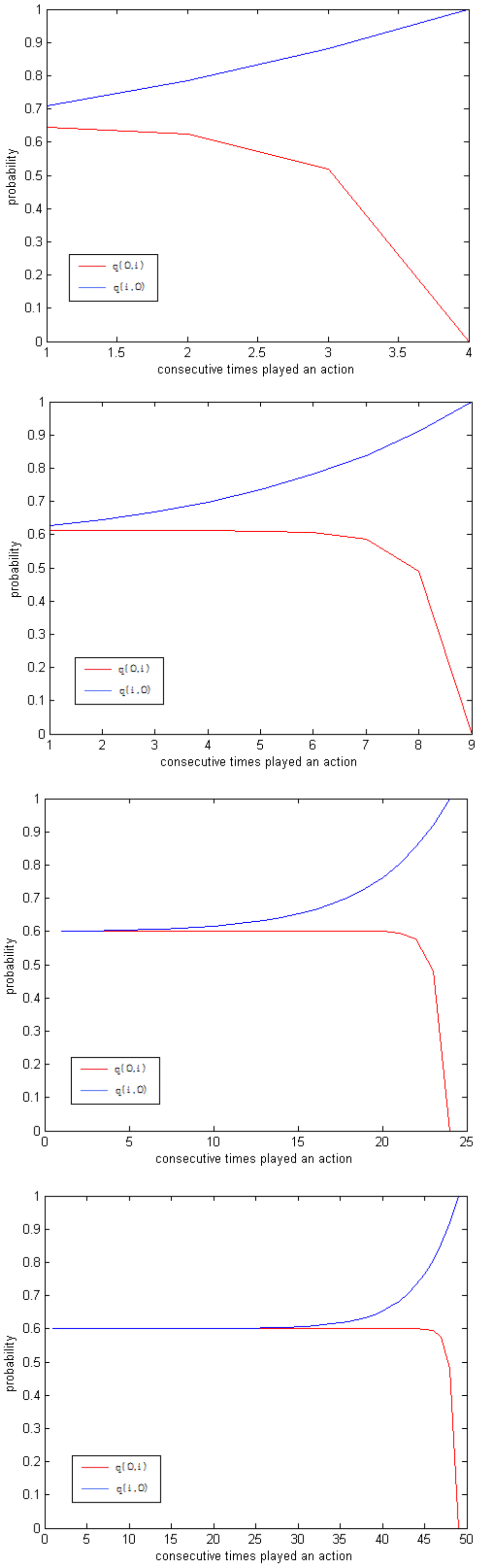

Figure 1: Values for q with different values of r

4.3 Experiment 3: Algorithm for two restricted players

The focus in the fictitious play algorithm for two restricted players lies on showing that the probabilities converge to certain values. If they do not change anymore than the equilibrium strategy is found and also the expected average reward can be calculated. The aim of this experiment is to show that there is convergence.

5 Discussion

From the first two experiments it follows that the linear programming algorithm gives good results in finding the expected average reward of a game, but also in finding the pure stationary best reply of the unrestricted player to the stationary strategy of the restricted player. Although there were some strange results, like why the probability in the states $(0, i)$ stays longer at the standard probability before moving away. Before seeing the experiments the expection was that the probabilities move away from the standard in sort of the same way, but as figure 1 show that is all but the case. The way by which the probability between the last number of states changes is the kind of the same for all the restrictions. For all the restrictions the bigger the restriction the longer the probability in a state stays near the standard probability.

6 Conclusion

References

[1] Hillier, F. and Lieberman, G. (2005). *Introduction to Operations Research*. McGraw-Hill Education, New York, NY.

[2] Joosten, R., Peters, H., and Thuijsman, F. (1995). Unlearning by not doing: Repeated games with vanishing actions. *Games and Economic Behavior*, Vol. 9, pp. 1–7.

[3] Schoenmakers, G. (2004). *The Profit of Skills in Repeated and Stochastic Games*. Maastricht University, Maastricht.

[4] Schoenmakers, G., Flesch, J., and Thuijsman, F. (2007). Fictitious play in stochastic games. *Mathematical Methods of Operations Research*, Vol. 66(2), pp. 315–325.

[5] Thuijsman, F. (2005). *To Play and To Share*. Epsilon Uitgaven, Utrecht.