Aalto University
School of Science
Degree programme in Engineering Physics and Mathematics

Computing equilibria in repeated games

Barchelor's thesis
18.2.2013

Markus Kärki

Aalto University

AALTO UNIVERSITY SCHOOL OF SCIENCE PO Box 11000, FI-00076 AALTO http://www.aalto.fi	ABSTRACT OF THE BACHELOR'S THESIS
Author: Markus Kärki	
Title: Computing equilibria in repeated games	
Degree programme: Degree programme in Engineering Physics and Mathematics	
Major subject: System sciences	Major subject code: F3010
Supervisor: prof. Harri Ehtamo Instructor: TkT Kimmo Berg	

Abstract:

Infinitely repeated games in game theory provide a way to model long-term competition and co-operation. The equilibrium of the game is a combination of strategies, where no player has an incentive to change his strategy. The outcomes of game when players are adopting equilibrium strategies are under interest in this study.

The aim of this study is to introduce and develop an algorithm for solving all the outcomes of equilibrium strategies. These outcomes are called equilibrium paths of the game. The algorithm is based on the idea of elementary subpaths from which all the equilibrium paths can be constructed. Having these pahts provide also a possibility to examine achievable payoffs when players adopt equilibrium strategies. There is also a new method to examine the payoff sets of 2x2 games with different discount factors.

Computing all the equilibrium paths of the game turned out to be a hard task especially with large discount factors. Despite of this, it is possible to solve part of the equilibrium paths and an approximation of the payoff set. This study also presents a method to solve the equilibrium path, which gives the smallest payoff to each player. These minimum payoffs are used in many proofs of the game theory.

Date: 18.2.2013	Language: English	Number of pages: 26

Keywords: Game theory,infinitely repeated games, payoffs set, elementary subpaths

Contents

1 Introduction

Infinitely repeated games are a way to model and explain long-term cooperation and competition. This study focuses on equilibrium strategies of the games with players who observe perfectly each others actions, discount their future payoffs and use only pure strategies. The aim of the study is to introduce and improve computational methods to find out equilibrium outcomes of repeated games. Also, the payoffs produced by these equilibrium paths are of interest.

This study discusses the computation of equilibria in infinitely repeated games by using Bergs and Kittis idea of elementary subpaths [2]. They proved that an enormously large sets of equilibrium paths are generated by a collection of elementary subpaths and introduced an algorithm for computing them. In their study, some fundamental questions remain still open. Is there a more efficient way to compute the elementary subpaths of the game? With large discount factors, the number of these paths tends to be large, but the study does not tell if the number of the paths is finite? Also, the question about finding the equilibrium paths that give the smallest payoffs to the players' remains open. The question is fundamental, because we need to know the payoff of that path in Bergs and Kittis algorithm. There is also a theoretical motivation to find this path, because the path appears at the folk theorems of repeated games [6].

Knowing the equilibrium paths provides one way to generate payoff set of the game. The payoff set of a game contains all payoff pairs (pairs on game with 2 players) which can be achieved by using the equilibrium strategies. These payoff sets are fractals [3], which become more dense when the discount factor grows. The point where the fractal covers the whole feasible payoff area is of special interest. The folk theorem of infinitely repeated games [6] states that this happens for every game when the discount factor is close to one, but according Stahl [10] this seems to happen even with much smaller discount factors for some games. This motivates developing a tool for investigating to not only equilibrium paths, but payoff sets too.

The focus of this study is on working principles of the improved algorithm for computing equilibrium paths. The algorithm is introduced in Section 3. Section 4 analyses the efficiency of the algorithm and introduces some examples of using algorithm to find the equilibrium and punishment paths of the game. There is also an example where we solve the minimum factor where the payoff set is full dimensional in Prisoners dilemma.

2 Theory and background

In this section we first shortly introduce the main concepts of the repeated games and previous studies. The most essential part of this section is the idea of the elementary subpaths and using them for computing equilibrium paths. Previous theory about payoff sets and folk theorem are outlined in the end of this section.

2.1 Introduction to repeated games

2.1.1 Stage game

The left of Table 1 below presents a stage game in normal form. Player 1 chooses between T and B (row) and player 2 chooses between L and R (column). The payoffs of players can be read from the cell so that first number is the payoff of player 1 and the second number payoff of 2. The size of the matrix and the number of player depends on the game. The outcomes of the game are denoted by alphabets a-d in the right of Table 1.

Table 1: Game matrix of Prisoner's dilemma and notation of actions

	L	R
T	3,3	0,4
B	4,0	1,1

	L	R
T	a	b
B	c	d

Minmax payoff for each player i is the smallest payoff, which player i can be forced by the other players. In the example game, this payoff is 1 for both players because when one player tries to minimize the other's payoff, he chooses second alternative which lets the other player choose between payoffs 1 and 0.

2.1.2 Nash equilibrium

Nash equilibrium of a strategic game with ordinal preferences is defined in Equation 1 [9]. Said less formally, Nash equilibrium is such a pair of strategies (one to each player), which no player can achieve better payoff by changing his strategy when the other players use the equilibrium strategies. Pure strategy equilibrium means that the players use fixed actions and for example not randomize between multiple actions.

$$u_i(a_i^N, a_{-i}^N) \geq u_i(a_i, a_{-i}^N), \forall i, a_i \in A_i, a_i \neq a_{-i} \tag{1}$$

In the example game the only Nash equilibrium outcome is payoff (1,1), because if one or another player defects his payoff drops to 0. In every other cell at least one player could improve his payoff by changing his choice.

2.1.3 Infinitely repeated games

Repeated game consists of a series of stage games. This study considers about infinitely repeated games, where the stage game remains the same all the time. The average payoff of these game is received by Formula 2, where δ_i is the discount factor of player i, σ current strategy profile and $a^k(\sigma)$ is the action played after history k when the strategy σ is adopted

$$U_i(\sigma) = (1 - \delta_i) \sum_{k=0}^{\infty} \delta_i^k u_i(a^k(\sigma)). \tag{2}$$

According to the definition, the average payoff is a discounted sum of payoffs of all stage games. Discounting means that the weight of a payoff is smaller when it actualizes further in future. This formula allows the comparison of average payoffs with different discount factors.

The outcome of a game is called a path and they are denoted by $cda(bbd)^{\infty}$, which for example means playing first actions c, d and a and then repeating infinitely actions b, b and d.

2.2 Equilibrium of repeated games

The strategy σ tells to the player what to do in current situation, technically, with current history of played actions. We say that the strategy is a best response to the other players' strategy if the player could not earn better payoff by using any other strategy. If the players' strategies are best responses to each other, they are an equilibrium of the repeated game. In this study equilibrium of a game means subgame perfect equilibrium (SPE), which demand that the current strategy is a best response in every subgame of the infinitely repeated game. For example $dbba^{\infty}$ is SPE path if and only if every subgame ($dbba^{\infty}$,bba^{∞}, ba^{∞}, a^{∞}) also satisfies equilibrium condition in Equation 3 [1]. In this equations σ is the adoted strategy, v^k is the payoff achieved after current stage-game when adoting σ and v^- is the punishment payoff. This condition is called the incentive compatibility condition and the path which satisfies this conditions in every subgame is called an equilibrium path.

$$(1 - \delta_i)u_i(a^k(\sigma)) + \delta_i v_i^k \geq \max_{a_i \in A_i} [(1 - \delta_i)u(a_i, a_{-i}(\sigma)) + \delta_i v_i^-] \; \forall i \tag{3}$$

According to the condition, the player must achieve at least as good payoff by following the equilibrium path (left side) than choosing one-shot deviation and then receiving punishment: minimum equilibrium payoff v_i^- for the rest of game [1](right side). A path is an equilibrium path only if it satisfies this condition for every player in every subgame. The discount factor δ_i has often critical influence whether a path is SPE or not.

In Prisoner's dilemma example, d is the Nash equilibrium of the stage game and d^∞ is the only SPE path when the discount factors of both player are small. When the discount factor grows, path a^∞ becomes also equilibrium path at the point where payoff of a^∞ ($\sum_{k=0}^{\infty} 3\cdot\delta_i^k$) exceeds payoff of betraying, which comes from the path bd^∞ for player 1 ($4 + \sum_{k=1}^{\infty} 1 \cdot \delta_i^k$).

2.3 Idea of elementary subpaths

Berg and Kitti present the idea of elementary subpaths in their study [2]. They show that "all the equilibrium paths will be composed of fragments called elementary subpaths". However, as seen in Section 4, the amount and the length of elementary subpaths are not finite for sure. With these fragments of equilibrium paths it is possible to build a graph, which present all the equilibrium paths of the game.

2.3.1 Definition of elementary subpaths

To determine if a path is an elementary subpath, Berg and Kitti define two other types of paths: first-action feasible (FAF) and first-action infeasible (FAI) paths. Definitions are based directly on Equation 3 of SPE. v^k in inequality is replaced by $con(a)$, which is the least payoff that the player must get in future. For each player i we can solve $con_i(a)$ from the incentive compatibility condition:

$$con_i(a) = \frac{\max_{a_i \in A_i}[(1-\delta_i)u(a_i, a_{-i}) + \delta_i v_i^-] - (1-\delta_i)u_i(a)}{\delta_i} \; \forall i \tag{4}$$

Moreover, the least continuation payoff $con(p)$ is recursive as equation 5 [2], where p is a path, a is the last actions of p and p^{k-1} is path p without its last action

$$con_i(p) = \frac{con_i(p^{k-1}) - (1-\delta_i)u_i(a)}{\delta_i} \;, \forall i. \tag{5}$$

After solving the continuation payoff requirement of a path, it can be sorted to FAF or FAI path according to the following condition. A finite path p is an FAF path, if

$$con(p) \leq v^{-}, \forall i. \tag{6}$$

This means that it is impossible to gain so small payoff after playing path p, that the path is not an equilibrium path. On the other hand, a finite path p is FAI path, if

$$con_i(p) > max\{v_i | v_{-i} = con_{-i}(p)\} \tag{7}$$

This is satisfied when the players' continuation payoffs demands are so high that all of them (if any) cannot be reached at the same time.

In the example, we have the same punishment payoff $v^{-} = 1$ for both players. If the continuation payoff $con(p)$ is less than 1, which is the smallest payoff that the player can accept, the path is a FAF path. If the continuation payoff requirement is too high, it cannot be achieved and the path is an FAI path. Between these limits there are still paths which cannot be sorted to these categories. Berg and Kitti call these as Neutral paths (N).

It is possible that FAF path contains FAI path. In Bergs and Kittis example bdd is FAF path and dd FAI path and that is why bdd cannot be part of an equilibrium path. When paths, that contain FAI paths, are removed from FAF path list, we get the list of elementary subpaths of the game.

2.3.2 Equilibrium paths and elementary subpaths

All the equilibrium paths can be constructed from the elementary subpaths. To get the SPE paths it is essential to consider all subgames. Every infinity path which can be constructed from consecutive elementary subpaths is an equilibrium path.

If we have elemetary subpaths aa, ba, and bbaa then bb(a)$^\infty$ is an equilibrium path because we can build bb(a)$^\infty$ from bbaa and aa after that, b(a)$^\infty$ using ba and aa infinitely and of course (a)$^\infty$ with just using aa. The path aab(a)$^\infty$ would not be an equilibrium, because we have no way to build subpath ab(a)$^\infty$. Playing action a demands always playing a again.

2.4 Payoff sets

Equilibrium paths of a game provide a possibility to examine the payoff set. A large set of equilibrium payoffs can be computed from the graph presentation of equilibrium paths. Plot of the sets are certainly approximative due to finite number of payoffs which is possible to generate by computer. Still, plots of payoff set are often very illustrative.

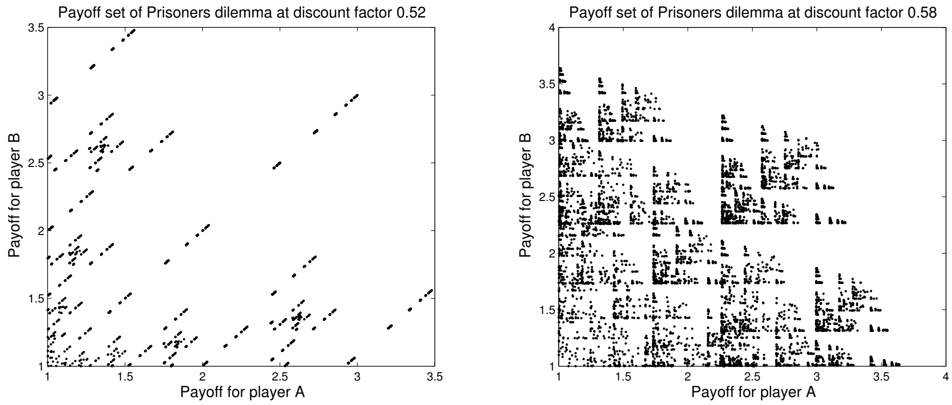


Figure 1: The payoff set of Prisoners Dilemma with discount factors 0.52 and 0.58

Now, we concentrate on payoff sets when players use the same discount factor. Payoff points lay in feasible area, which is convex hull formed from actions of the game matrix then cut by punishment payoff by players. Only convex hull area is possible to reach by actions of the game even with not equilibrium paths. Cutting by punishment is essential because players do not accept smaller payoffs. By definition the punishment produce the smallest payoff, which is possible to achieve by equilibrium strategies.

2.4.1 Payoff sets of repeated games

When discount factor is large enough, all strictly rational payoff points in feasible area, are achieved by some equilibrium path. Feasible payoffs are payoffs greater than minmax payoff and in the area limited by points of the game matrix. This statement is know as folk theorem [5]. The critical discount factor is denoted as $\bar{\delta}$ and it is the smallest discount factor, which leads to the full-dimensional payoff set. Figure 1 presents Prisoners Dilemmas payoff sets with two different discount factor and it can be seen that payoff set becomes more dense when the discount factor grows. Folk theorem does not tell what is the smallest discount factor a specific game in which theorem is valid. It does not tell either if the theorem is valid with any discount factor which is smaller than one.

2.4.2 The critical discount factor

Payoff sets of repeated games are fractals more specifically sub-self-affine sets. Bergs and Kittis show in their study [3] how the fractal becomes more dense when the discount factor grows. Their study leads to conclusion that using continuation payoff sets we can find the critical folk discount factor. The idea is to find the smallest discount factor, where the continuation payoff sets cover the whole feasible area. The sets are computed by a illustrative way in Section 3.2.2, where's also presented a tool to examine these sets of given game. Gaps and holes in payoff set become filled only if some assumptions are made. Either players have to use correlated strategies or continuation payoff sets must be convex [4].

3 The computational methods

This section is for detailed description of the algorithm made for computing equilibrium paths using elementary subpaths. Algorithm also finds the punishment paths and payoffs and returns the graph of equilibrium paths of the game. Section 3.2 describes the methods for plotting the payoff sets and finding the critical folk discounting.

3.1 Algorithm for computing equilibria paths of repeated game

This algorithm returns the graph of equilibrium paths of given game. The basic structure is presented in Algorithm 1. Algorithm input consists of payoff vectors and a vector of players discount factors. The output is a graph of equilibrium paths and an approximation of a payoff set, which is generated from graph. It is notable that algorithm solves and returns also the punishment paths, which is essential for solving all other equilibrium paths.

The main structure of Algorithm 1 is to first find FAF paths, when punishment payoff are given, this is described in Section 3.1.1. At the start point we do not know the punishment, so the minmax payoff is used instead. The minmax is a lower bound to punishment payoff and if the punishment payoff is higher, collections of FAF paths contains some excessive paths. Excessive paths are removed after solving the punishment payoffs later in run. When we have a collection of FAF paths, it is possible to construct a graph by an algorithm describer in Section 3.1.2.

Punishment paths are found by searching from the graph for each player the path, which gives the smallest payoff. After finding these paths the algorithm checks if the paths satisfies the equilibrium condition. If not, the algorithm starts over but now using the smallest payoffs, which found from previous graph, as the punishments. Anyway, we know that the payoff found from the graph is smaller than the punishment. Often, there is no need to run the algorithm several times, because there is no difference between FAF path collections computed with the minmax or the punishment payoffs.

Algorithm 1: Structure of algorithm finding equilibrium paths

Input: Payoffs and discount factors
Output: Graph, punishment paths and payoffs, vizualisation of payoff set
```
begin
    punisment ← minmax;
    while Any punishment path is not SPE do
        Search FAF paths // Section 3.1.1
        Make the graph from FAFs // Section 3.1.2
        Find and update punishment // Section 3.1.3
    Plot approximation of payoff set; // Section 3.2.1
```

3.1.1 Finding FAF-paths

Finding FAF-paths is the first step of searching the equilibrium paths of a game. Pseudocode of this algorithm is presented in Algorithm 2. The algorithm sorts paths to FAF, FAI and N category by bread-first search starting at one length paths (a,b,c and d in 2x2 game). Child paths of every N path are added to a queue of unsorted paths (if a∈N, aa, ab, ac, ad goes to the queue of unsorted paths). Only child paths of N paths are added, because the children of FAI paths are always FAI paths and children of FAF paths are FAF paths. The result can be thought as a tree where leaf nodes are FAF or FAI paths and inner nodes are Neutral paths. This is demonstrated in Figure 2.

Checking conditions for FAF or FAI paths comes from the Equation 6. The idea of the condition is that a payoff outside of feasible payoff area cannot be received. The algorithm uses tighter conditions than necessary making the computation more simple. That is why some of FAF paths are sorted to Neutral paths, but the child paths of these missorted paths should be sorted correctly sooner or later. The condition used is the same than Berg and Kitti[2] use in their algorithm.

A finite path p is a FAF path if

$$con(p) \leq con(\textit{final action of } p), \ \textit{length of } p > 1 \tag{8}$$

$$con(p) \leq v^{-}, \ \textit{length of } p = 1 \tag{9}$$

and a finite path p is a FAI path if

$$con_i(p) > \bar{v}_i, \quad \textit{for some } i \tag{10}$$

where $\bar{v}_i$ is the maximum payoff for player i in the stage game.

Sometimes the fourth category for paths is needed. If a path p (length k) is such a FAF path that the demand for the continuation payoff of a path p ($con(p^k)$) is exactly the same than the path's parents continuation payoff ($con(p^{k-1})$) for some player, we say that the path is an Infinite path (INF). A infinite path is an FAF path when the last action is repeated infinitely, but we can prevent the infinite loop by removing it from the search in this stage. This check recognize only simple INF paths, but it should be possible to recognize more complicated infinity paths too. Implementing a proper check for more complicated infinity paths, could help when the set of elementary subpaths is enormously large.

When the algorithm reaches the point where are no Neutral paths in queue, then the search is ready. At this point every single path is possible to sort to feasible or to infeasible. Path is infeasible, if it contains any FAI path anywhere as a part of the path. That is why some of FAF paths may be infeasible. Removing infeasible FAF paths is done when making the graph in the Algorithm 3.

As seen in FAI and FAF conditions whether the path is FAF of FAI depends critically on the discount factors and the punishment payoffs. The example game used previously in this study, Prisoner's dilemma, has 17 FAF paths when the discount factor for both players is 0.51. These paths are used in the next algorithm and they are: bdcdaaab, d, cb, ca, ba, bdcdaaaa, bdca, cdba, bdaaac, bdaaaa, cdbdaaaa, aa, adaaaa, cdaaaa, bc, cdbdaaac and cdaaab. There are no infinity paths with this discount factor, but with discount 1/2 there would be infinity paths like ad(a)$^\infty$.

Algorithm 2: Searching FAF paths

Input: Payoff vectors, discount factors, punishment payoffs
Output: Collection of FAF paths
begin
    Set action set (ex. a, b, c, d) to queue of untested paths;
    while *Number of FAFs* $<$ *limit* & *queue of untested is not empty* **do**
        p $\leftarrow$ next path from queue of untested paths;
        if *con(p) satisfies condition 9* **then**
            p $\rightarrow$ Collection of FAF paths;
        else if *con(p) satisfies condition 10* **then**
            continue;
        else if $con_i(p) = con_i(p^{k-1})$ *for some* i **then**
            p $\rightarrow$ Collection of INF paths;
        else
            add childs of p to qeue of untested paths

3.1.2 Constructing graph from elementary subpaths

Algorithm for making the graph is presented in Algorithm 3. The first phase in this algorithm is collecting FAF path as a tree as in Figure 2 which presents the FAF paths from the example in the previous chapter. The tree of FAF paths is the figure used as a base of the graph of all equilibrium paths. AFter that the final graph is constructed by adding every possible connection from a branch to another.

It is essential that the path satisfies the equilibrium condition not only for the first subgame but also for every subgame after that. The algorithm is a bit different than Bergs and Kittis, but the result is equivalent. The algorithm checks every FAF path separately and find possible connections to other FAF paths. Because the equilibrium condition must be valid for every subgame, we have to mind about FAI paths in the middle of FAF paths. If, for example, bda is an FAF path, the connection must be feasible for subpaths *da* and *a* too. This is done by checking that subpaths of current FAF paths are also FAF paths. If a subpath is the beginning of another FAF path, then the current path can be connected to that other path. When every possible connections are added, graph is like in Figure 3.

The last, trivial, phase is removing dead-end from graph. The FAFs, which are not elementary subpaths are removed, because they simply cannot be connected to anywhere. The final graph is in figure 3.1.2.

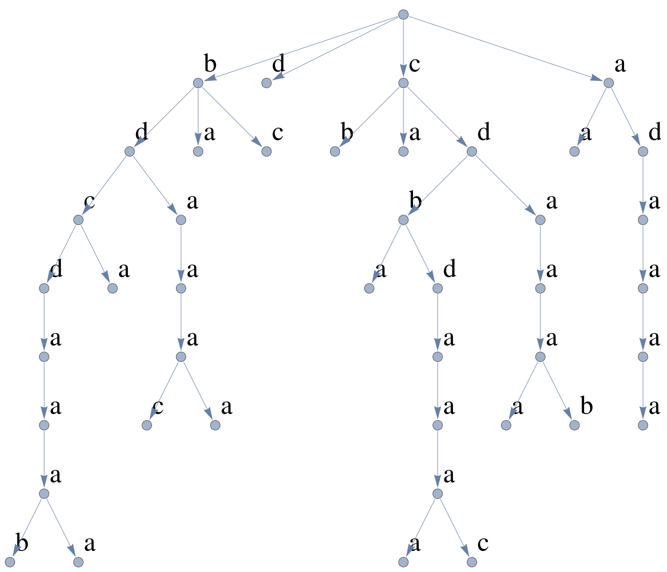


Figure 2: A Reprentation of FAF paths of Prisoner's dilemma at discount factor 0.51

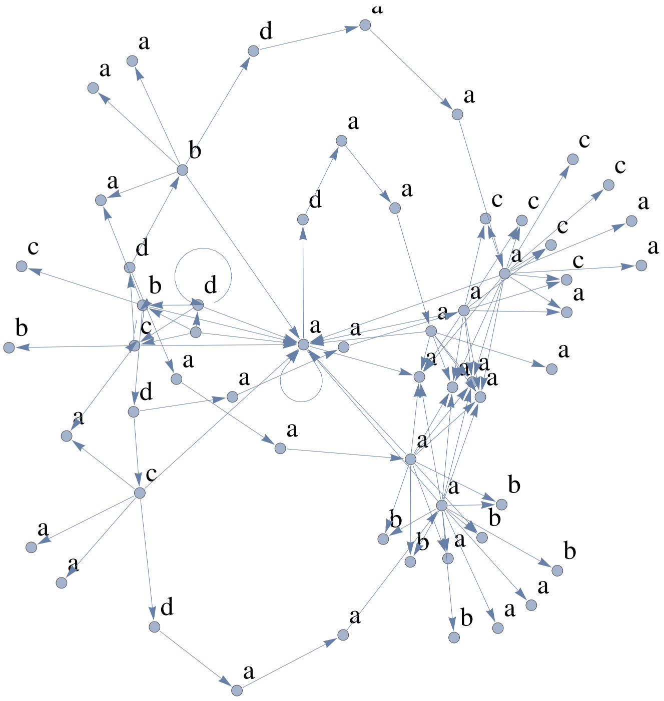


Figure 3: A graph made from tree by adding possible connections between FAF paths

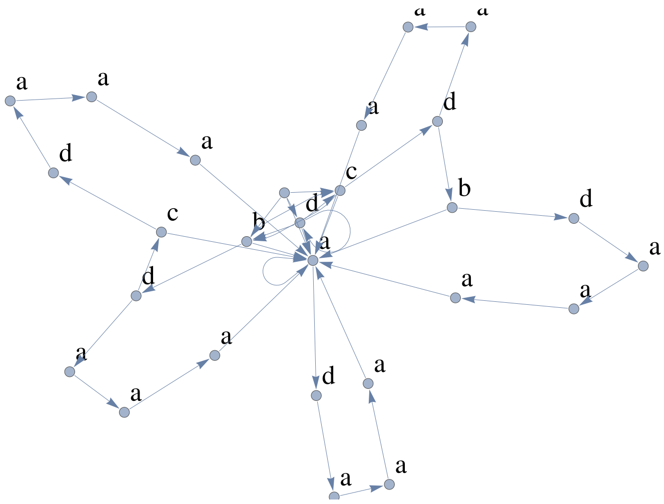


Figure 4: The final graph after removing dead-ends

Algorithm 3: Make graph

Input: Collection of FAF paths
Output: Adjacency matrix and node list of the graph
begin
    Make a tree from the collection of the FAF paths;
    for *Collection of the FAF paths* **do**
        $p_f \leftarrow$ current FAF path;
        for *sub-endparts of* p_f **do**
            $p_{f,e} \leftarrow$ current sub-endpart of p_f;
            if $p_{f,e}$ *starts with a FAF path* **then**
                Continue to sub-endpart after found FAF;
            if $p_{f,e}$ *is start of a FAF path* **then**
                Add connection to the graph from start of $p_{f,e}$ to start of found FAF;
    Remove dead-ends from the graph;

3.1.3 Finding punishment

One very interesting question, when talking about repeated games, is punishment paths and payoffs of a game. Generally, this is very untrivial question. If repeating minmax action is a Nash equilibrium of repeated game then it is also the punishment, because the minmax payoff of the repeated game cannot be lower than minmax of the stage game [11]. A rational player never accept a payoff lower than minmax. In every other cases it is possible that punishment paths are complicated and there have not been any systematic method to find them. This algorithm gives the answer to this question anytime when the graph of the game is finite, practically when the discount factor for both players is small enough.

The punishment path may be different for every player and the search shall be done for every player differently. The main idea is to look at graph and find the infinite path with the smallest payoff. Second part is the check if the found paths are equilibrium paths. If not, the reason is that the assumed punishment payoff is too low and there are excessive paths in the graph. We can still use the payoff of found path as punishment of next iteration, because it should be greater than previous assumption. With this method the found punishment approach the real punishment by every iteration.

The main principle used to find graph minimum payoff path is to do breadfirst search for the graph. The search continues until a branch ends to a node which the branch already contains. After finding a loop, the algorithm assumes an infinite repetation of the loop. This is done for every possible branches of the graph. The method finds all possible loops of the graph among which is the path producing the smallest payoff.

When the found path ends to an infinite loop, the algorithm computes payoff for it and it is every infinite subpath for every player. They are stored to a N x l matrix, where N is number of the players and l is the length of path repeating loop once. Also, the punishment demands v_i^- matrix will be computed using the incentive condition in form

$$v_i^k \geq \max_{a_i \in A_i}[u(a_i, a_{-i}(\sigma))] + \delta_i v_i^- \forall i \tag{11}$$

where v_i^k is the payoff for player i when starting at k:th action. The greatest punishment, which makes subgame path as equilibrium is the equal case and v_i^- is solved for every player and subgame. Now we can compare each player punishment payoff demand and possible payoffs from graph. If a players punishment path candidate demands smaller payoff than other players accept, then we can say that the punishment path candidate is not SPE and we have to search new elementary paths with higher punishment assumption.

There is also some cutting conditions which make the search faster. One is that if a payoff of a branch grows greater than some found infinite paths payoff, investigating this brach is useless. On the other hand if the payoff of a path with loop is equal to the minmax payoff, there is no need to search for a better punishment path candidate.

3.2 Visualizing payoff set

Visualizing the payoff set is an illustrative way to examine payoffs of SPEs of a game. For this work two method to visualize the set are developed. One is based on the graph of SPEs and the other on the feasible areas of the game. First is meant for visualizing actual payoff points and its fractalic shapes. The other is based on feasible areas and it is made for finding the critical discount factor.

3.2.1 Plotting payoff set from equilibrium paths

Plotting a payoff set is always approximative, because the number of possible paths is infinite except in a few special cases. This algorithm just takes random steps in the graph. After taking certain amount of the steps, the algorithm stops when it arrives to the next loop. Then it calculates payoffs for every player and then we have one point which is ready to plot. Usually a few thousand points are needed to an illustrative figure. Two payoff sets of the Prisoner's dilemma was presented earlier in Figure 1. These figures can contain two or three players payoffs at once.

Algorithm 4: Search punishment paths

Input: Graph, games payoffs, discount factors
Output: Punishment path and payoff for each player
begin
    for *each player i* **do**
        Put start node to qeueu of untested paths;
        while *queue of untested path is not empty AND bestCandidateForI* $\neq$ *minmax* **do**
            p $\leftarrow$ next path from queue of untested paths;
            for *every chilf of p* (p_c) **do**
                if *payoff of p* $>$ *payoff of bestCandidateForI* **then**
                    continue;
                if *end of* p_c *is loop* **then**
                    $u^c \leftarrow$ Compute payoffs of for all players from every subgame of p_c assuming repeating loop infitely;
                    $u^c_{demand} \leftarrow$ Compute con. payoff demands for all players from in every subgame;
                    if $u^c = minmax$ **then**
                        bestCandidateForI $\leftarrow p_c$;
                        break;
                    else if $u^c <$ *minmax for some i* **then**
                        continue;
                    else if $u^c <$ *bestCandidateForI* **then**
                        bestCandidateForI $\leftarrow p_c$;
                    else
                        $p_c \rightarrow$ queue of untested paths;

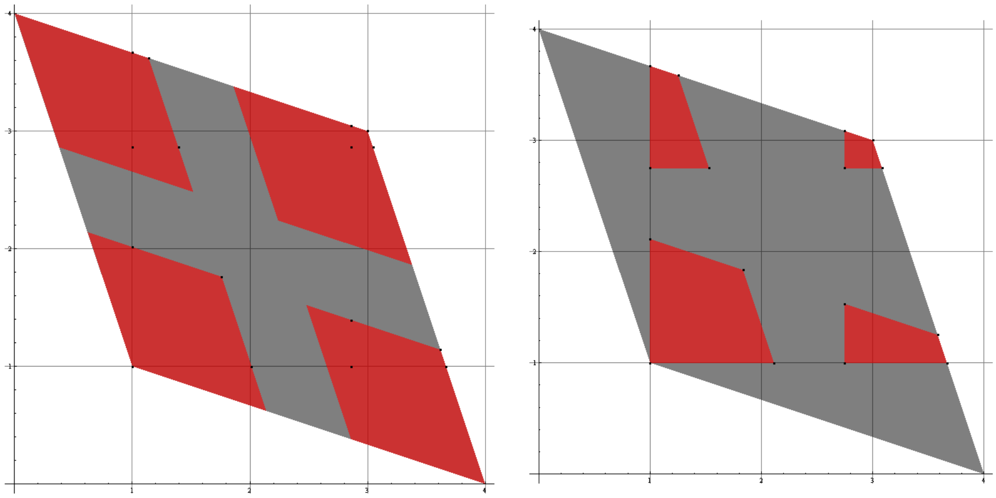


Figure 5: Two steps of making figure of the payoff set with the tool

3.2.2 A tool visualizing the critical folk discount factor

Another tool for examining the payoff sets was made especially to find the critical folk discounting. The working principles of the tool are described in this section. The tool needs the game matrix and discounting as input and gives instantly figure of area covered by the payoff set. Third necessary input is the punishment payoff, which can be different for both players. If punishment payoffs are not trivial they can be solved by the algorithm presented before. It is also important manually update punishment respond to discounting.

Making the figure consist on two phases:

1. First step is to create the convex hull from points of the game matrix and scale that hull among discounting for each action. The scaling is done by affine transform y=Ax+b, where

$$A = \begin{bmatrix} \delta & 0 \\ 0 & \delta \end{bmatrix} \text{ and b } = \begin{bmatrix} 1-\delta \\ 1-\delta \end{bmatrix} * \begin{bmatrix} u_1(a) & u_2(a) \end{bmatrix} \tag{12}$$

   These areas are those which can be achieved even theoretically when one action has been played. This is presented for Prisoner's dilemma in Figure 5 left.

2. Second step is to cut the convex hull by punishment payoff and get feasible payoff area. Also each subarea need to be cut. Cutting points of those are get by scaling original punishment point by the same affine

transforms than subareas in the previous step. The unfeasible areas of the subareas as removed in Figure 5 right.

In Figure 5 left is four areas which are the basic structure of the fractal. The fractal consist on these four areas when they are copied in smaller scale over and over again. The area becomes full at the time when these four areas cover the whole feasible area. It is easy see where is the last uncovered area with this kind of figure.

4 Analysis and examples

There is analysis and result from testing algorithm to various problems in this section. In addition there are examples of punishment paths and finding the critical folk discounting of the game in this section. The punishment path example is about No-Conflict game and critical discounting example about Prisoner's dilemma, but the idea of these works as well in any repeated game. The algorithm's ability to compute elementary paths with high discount factors seems to be very limited so there is no final answer whether the collection of elementary subpaths is finite or not. Instead some new knowledge about the punishment paths is presented with the example of computing punishment paths. The method to compute the critical discount factor is brand new.

4.1 Analyzing equilibrium paths

When the discount factors grow, the number of elementary subpaths and the length of the paths usually explode after some discount factor much lower than one. In this case the algorithm stops searching paths after the limits set by user. In this case the graph remains incomplete: there are some missing equilibrium paths and if returned punishment payoff is not minmax payoff for all players, it is possible that punishment path is not found. The main problem of equilibrium paths is the huge amount of elementary subpaths when the discounting grows over some critical point depending on the game. The figures in Appendix 1 show amounts and lengths of elementary subpaths paths of the Prisoners Dilemma and No-Conflict game. We would get same kind of figures for other 10 symmetric 2x2 games.[7] The limit for the number of FAF paths of those figures was 1000, but the rapid increasing both number of paths and length of longest path is perceptible. Increasing the limit has only a little effect, because in tested cases length of paths grows so rapidly. It is notable that rapid growth of computing task happens often a much before of the critical folk discounting.

When looking for computation elementary subpaths games seem to divide to easy and futile. The complete computation with every discount factor probably requires at least a different approach. Anyway these runs are made by assuming minmax payoff for punishment and in some special cases knowing the real punishment could reduce computing task to doable.

4.2 An example of punishment paths

The question about the punishment path and payoff of a game is fundamental because of it is used as well as computing elementary subpaths as in this study, but also in the proofs of folk theorems for different game types [8] [11] [6].

Among 12 symmetric 2x2 there are three games under special interest: No-conflict, Anti-No-Conflict and anti-Stag Hunt. Punishments paths of all other 9 games are just repeating equilibrium action of the stage game, which is easy to detect from the game matrix. In three exceptional games the minmax payoff is achieved only at some higher discount factor.

Punishment payoff of the three special games follows the same pattern. When the discount factors are small, punishments are just repeating the only equilibrium action. With discount factors near one punishment are either repeating minmax action or some series generating payoff very close minmax as folk theorem predicts. The interesting region is between them: in this region punishment paths are complicated, payoffs are close to minmax and new punishment path appear often when discounting grows even a bit.

Figure 6 presents punishment payoffs of No-Conflict game as function of discount factor. The payoff matrix of the game is in Table 2 below. A more detailed representation of these paths is found at Appendix 2. As seen in the figure, the punishment between discount factors 0.44 and 0.49 are only best found solution and we cannot say for sure if there are some equilibrium paths with smaller payoffs. Despite that it is clear that punishment payoff is not monotonic, which lead to situation where a path can be SPE at some discount factor, but not at some higher discount factor.

Table 2: The payoffs of No-Conflict game

	L	R
T	5,5	3,4
B	4,3	2,2

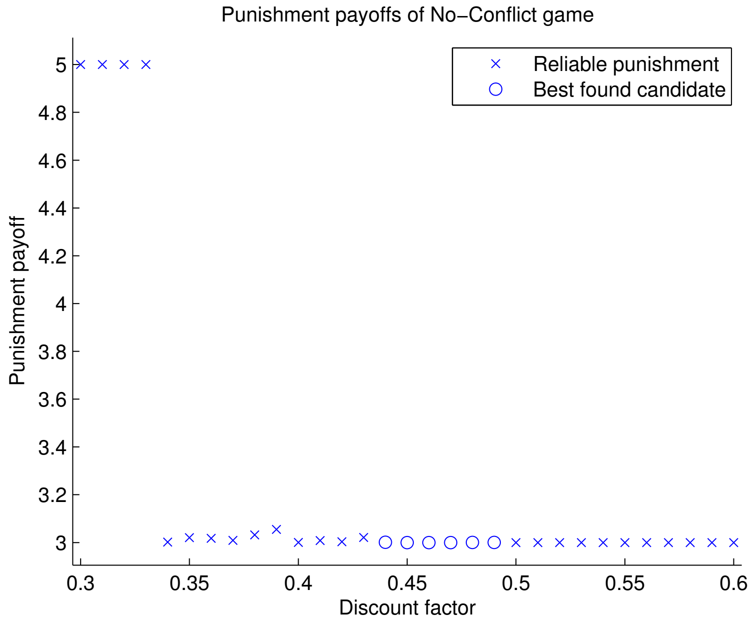


Figure 6: Punishment (average) payoffs of No-Conflict game with different discount factors

4.3 Finding the critical discount factor of the payoff set

Fractalic nature of the payoff provide possibility to solve the critical folk discount factor. The method is based on geometric areas of payoff set and that is why it can be only used if all player's are discounting with the same factor. With different discount factors, the payoff sets tend to be malformed and there is no exact area to fill.

The idea of this method was presented previously in Theory and background section 2.4.2. For further study, we have developed a visualization tool with Mathematica software. The same phenomena are seen in payoff sets made from fractalic points, but the Mathematica tool makes possible to instantly visualize where the unreachable payoffs are. The tool can handle 2x2 games with different discount factors. By this way it is easy to find the last not-covered part of the fractal. Solving the critical folk discount factor is usually easy, because areas the are scaled linearly by the discount factor.

4.3.1 Finding the critical discount factor for Prisoners Dilemma

In this example fulfilling point of Prisoner's dilemma is solved. With mathematica tool we see that the last uncovered area is at the edge of the possible payoff area. The areas are presented in figure 7. The critical discounting seems to be somewhere near 0.65. At the critical discount factor the corner points of two areas come together. Because the both points move along the same line, we can only look at x- or y-coordinates of points.

The idea of Equations 13 and 14 is to find the discount factor where the points at edge of gap area meet each others. Equation 13 is about x-coordinates of gap on the upper edge and Equation 14 solves y-coordinates of gap on the right edge. When the game is symmetric, it is necessary to solve only one of these equations. Notation of payoffs is basically same than before: a_1 means payoff for player 1, when action a is played.

The left side of equation Equations 13 is the x-coordinate of left corner point of the upper edge of the gap. It moves linearly from b_1 to d_1 when the discount factor goes from 0 to 1. This comes straight from Affine scaling 12. The right side is same kind of formula for the right edge of the gap. When these point are the same, we get Equation 13 where the discount factor can be solved. At asymmetric game we need to look area which is the last uncovered area, so we can write the critical discount factor in form presented in Formula 15. Setting payoffs like in Table 1 gives the critical discount factor 2/3, which corresponds to previous results [10].

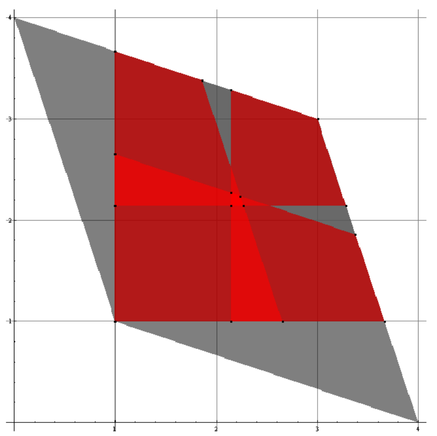


Figure 7: Almost full payoff set. The last uncovered area is easy to detect.

$$b_1 + \delta(d_1 - b_1) = c_1 - \delta(c_1 - a_1) \tag{13}$$

$$c_2 + \delta(d_2 - c_2) = b_2 + \delta(b_2 - a_2) \tag{14}$$

$$\bar{\delta} = \max[\frac{c_1 - b_1}{d_1 - a_1 + c_1 - b_1}, \frac{b_2 - c_2}{d_2 - a_2 + b_2 - c_2}] \tag{15}$$

Case where the last uncovered area is at the edge limited by punishment payoff is more complicated to solve, if the punishment changes in surrounding of the critical discounting. Equations are tricky, but they should be doable at least in all 2x2 cases.

5 Conclusions and discussion

The study provide an improved method for solving the equilibrium paths of the game with given discount factors. The method is based on Berg and Kittis method [2]. The code works regardless of the size of the game matrix or the number of players, but computing work seems to explode at some point when discount factors increase. For that reason, most of the cases can be divided to easy or impossible to be practical. There is no proof that amount of the paths is infinite, but computing all the paths demand at least a new kind of approach when discount factors are high. Approximative solution can still be computed and payoff sets of these can be still illustrative.

As presented in Section 3.1.3 the algorithm uses an iterative method for solving the punishment paths and payoffs. Abreu shows how the threat of punishment payoff makes a path to equilibrium[1], but he does not have a method to find that payoff unless in special cases. The payoffs of punishment paths are used in proofs of folk theorems, but there have not been method to solve these paths or their payoffs. This study discovers that the punishment payoffs are not monotonic when discount factor grows, so some SPE path may not be a SPE anymore when discount factor is a bit higher.

Having a way to compute elementary and punishment paths can support further research. Obvious questions are what happens when the number of equilibrium paths explode and are the collections of these paths still finite? This study shows that punishment paths can be repeating long pattern, but could these paths be infinitely long without a pattern? A punishment like this cannot be found with described algorithm, but the question is essential in theoretical mean.

Stahls provided that payoff sets may become full even if the discount factors are far smaller than one [10]. This expands folk theorem [6], which says that every feasible payoff are achieved by some equilibrium path when discounting is close to one. This study presents an additional tool to visualize payoff sets as fractals. Unlike the algorithm based on equilibrium paths, this method is suitable only for equal discount factors, but makes possible to visualizing sets when computing is too troublesome. This leads also to possibility to solve the exact the smallest discount factor when feasible sets is full at least when mixed strategies are accepted. This is also demonstrated in Prisoner's dilemma-game.

Equilibrium paths are a seldom used approach to examining to payoff sets in game theory, but the methods used in this study leads to exactly same result than in previous studies. Therefore the methods, based on computing equilibrium paths, may be a suitable alternative when solving problems in theory of repeated games.

References

[1] Dilip Abreu. On the theory of infinitely repeated games with discounting. *Econometrica*, 56(2):383, March 1988.

[2] Kimmo Berg and Mitri Kitti. Computing equilibria in discounted 2x2 supergames. *Computational Economics*, pages 1–18.

[3] Kimmo Berg and Mitri Kitti. Fractal geometry of equilibrium payoffs in discounted supergames, 2012.

[4] Mark B. Cronshaw and David G. Luenberger. Strongly symmetric subgame perfect equilibria in infinitely repeated games with perfect monitoring and discounting. *Games and Economic Behavior*, 6(2):220–237, March 1994.

[5] J. W. Friedman. A non-cooperative equilibrium for supergames. *The Review of Economic Studies*, page 1–12, 1971.

[6] Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting or with incomplete information. *Econometrica*, 54(3):533, May 1986.

[7] David Goforth and David Robinson. *Topology of 2x2 Games*. Routledge, 1 edition, September 2005.

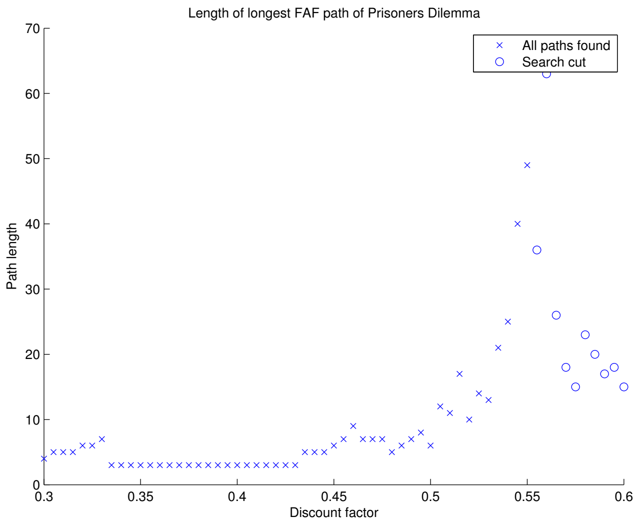
[8] Yves Guéron, Thibaut Lamadon, and Caroline D. Thomas. On the folk theorem with one-dimensional payoffs and different discount factors. *Games and Economic Behavior*, 73(1):287–295, September 2011.

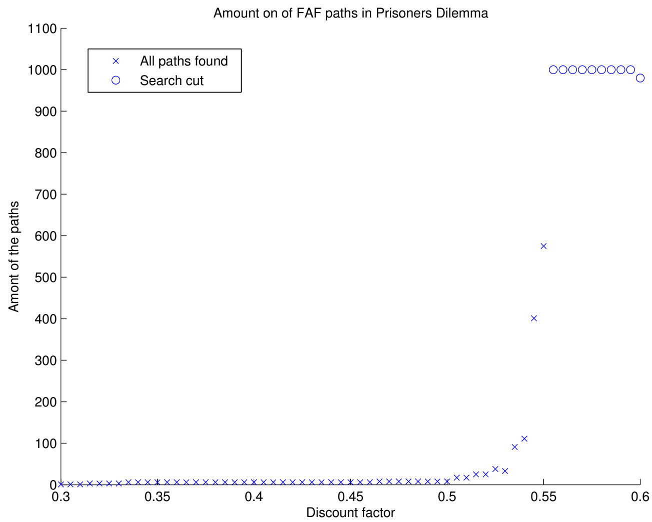
[9] J. F. Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

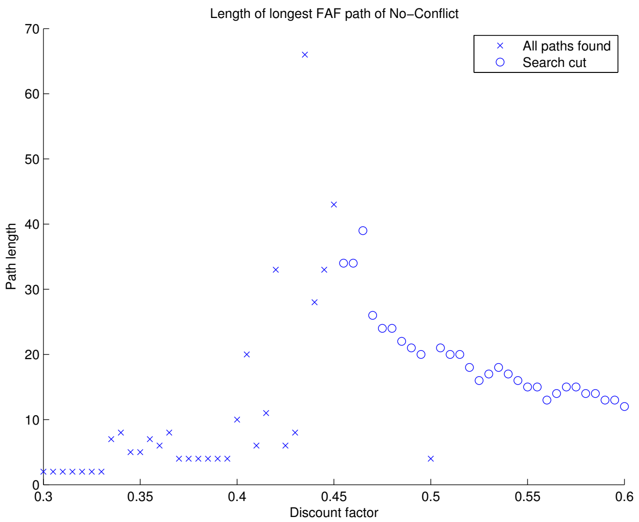
[10] Dale O Stahl II. The graph of prisoners' dilemma supergame payoffs as a function of the discount factor. *Games and Economic Behavior*, 3(3):368–384, August 1991.

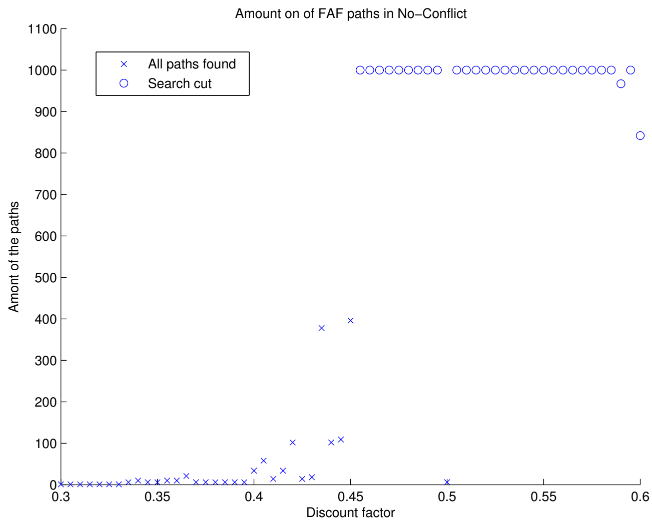
[11] Quan Wen. The "Folk theorem" for repeated games with complete information. *Econometrica*, 62(4):949, July 1994.

A Figures of amount and length of FAF paths in No-Conflict and Prisoners Dilemma games









B Table of punishment paths in No Conflict-game

Discount factor	Payoff	Type	Path, player A
0,3	5	Reliable	a^{∞}
0,31	5	Reliable	a^{∞}
0,32	5	Reliable	a^{∞}
0,33	5	Reliable	a^{∞}
0,34	3,002121447	Reliable	daaaca$(aaca)^{\infty}$
0,35	3,020292007	Reliable	$(daaa)^{\infty}$
0,36	3,01735768	Reliable	daac$(aac)^{\infty}$
0,37	3,009157874	Reliable	$(daa)^{\infty}$
0,38	3,032012595	Reliable	$(daa)^{\infty}$
0,39	3,054600869	Reliable	$(daa)^{\infty}$
0,4	3,000102939	Reliable	$(dacaaadaa)^{\infty}$
0,41	3,008570049	Reliable	$(dacaa)^{\infty}$
0,42	3,003441404	Reliable	dacacca$(aacca)^{\infty}$
0,43	3,021507848	Reliable	daca$(ccaa)^{\infty}$
0,44	3,001380772	Best found	$(daccaaa)^{\infty}$
0,45	3,000000098	Best found	$(daccacadaaccaaaa)^{\infty}$
0,46	3,000000717	Best found	$(dadaacccaaaaaacaa)^{\infty}$
0,47	3,000000173	Best found	$(dadacababaaaacaa)^{\infty}$
0,48	3,000002097	Best found	$(daccccaacacaabaa)^{\infty}$
0,49	3,00000003	Best found	$(dadadacaaaaadabaa)^{\infty}$
0,5	3	Minmax	c^{∞}
0,51	3	Minmax	c^{∞}
0,52	3	Minmax	c^{∞}

C Summary in Finnish

Peliteoria tarkastelee strategista toimintaa tilanteessa, jossa toimijan saamat hyödyt ja haitat riippuvat paitsi toimijan omista, myös muiden toimijoiden, valinnoista. Tällainen tilanne voisi olla esimerkiksi yrityksen hinnoitteluongelma, jossa yritykset valitsevat myyntihintansa kahdesta vaihtoehdosta: kalliista ja halvasta. Yrityksen myynti ja voitot riippuvat paitsi sen omasta valinnasta varmasti myös kilpailevan yrityksen valinnasta. Pitkäaikaisen yhteistyön ja kilpailun mallintamiseen käytetään peliteoriassa äärettömästi toistettuja pelejä.

Kandidaatintyössä tarkastellaan pelejä, joissa pelaajat valitsevat jonkin vaihtoehdon äärellisistä määrästä vaihtoehtoja tietäen kunkin valinnan seuraukset sekä itselleen että muille. Valintansa pelaajat tekevät yhtäaikaisesti ja toisistaan tietämättä, mutta päätösten jälkeen he havaitsevat seuraukset. Tämän jälkeen he tekevät uudet valinnat samojen vaihtoehtojen väliltä. Pelaajan strategia määrää, minkä vaihtoehdon hän valitsee kunkin pelihistorian jälkeen. Pelaajien käyttämät strategiat muodostavat tasapainon, mikäli kukaan pelaaja ei voi saavuttaa itselleen parempaa tulosta poikkeamalla käyttämästään strategiasta, ja usein tällaisia tasapainostrategioita on samassa pelissä useita erilaisia . Tasapainopoluksi kutsutaan tasapainostrategian käyttämisestä syntyvää lopputulosta. Eräs tasapainopolku voisi yritysesimerkissä olla sellainen, jossa molemmat yritykset valitsevat toistuvasti halvat myyntihinnat ja sen seurauksena saavuttaisivat jonkin euroissa mitattavan hyödyn.

Pelaajat arvottavat kauempana tulevaisuudessa saatavat hyödyt vähemmän arvokkaiksi kuin pian realisoituvat. Tätä talousteoriasta tulevaa hyödyn nykyarvon laskentaa kutsutaan diskonttaukseksi ja pelaajilla ajatellaan olevan erityinen nollan ja yhden välillä oleva diskonttauskerroin, jonka avulla he arvottavat tulevat hyötynsä. Käytännöllinen tulkinta tälle kertoimelle on kärsivällisyys – mikäli pelaajan diskonttauskerroin on riittävän suuri, hän saattaa suostua huonoon tulokseen lähitulevaisuudessa saadakseen myöhemmin suuremman palkkion. Pelaajien käyttämät diskonttauskertoimet vaikuttavat ratkaisevasti siihen, onko jokin polku pelin tasapainopolku.

Kandidaatintyön tavoitteena oli kehittää laskenta-algoritmi annetun pelin kaikkien tasapainopolkujen löytämiseksi. Kun tasapainopolut tunnetaan, voidaan tietysti tarkastella myös hyötyjä, joita pelaajat saavuttavat niitä käyttäen. Erityisen kiinnostavaa on tasapainopolkujen määrän ja niillä saavutettavien hyötyjen vertailu eri diskonttauskertoimilla.

Työssä toteutettu algoritmi perustuu aiempaan tutkimukseen, mutta pyrkii vastaamaan siinä havaittuihin puutteisiin. Yksi ongelmista on tasapainopolkujen määrän voimakas kasvu ja siitä aiheutuvat laskennan ongelmat diskonttauskertoimen kasvaessa. Lisäksi aiempi menetelmä tarvitsee alkutiedokseen pienimmän tasapainopolun avulla saatavan hyödyn kullekin pelaajalle. Koska tätä ei yleensä tiedetä ennen kuin tasapainopolut on selvitetty, on tämän ongelman ratkaisu laskennan kannalta välttämätöntä. Polkujen lukumäärän kasvaessa kasvaa luonnollisesti myös mahdollisten hyötyjen lukumäärä. Nämä hyödyt voidaan piirtää kuvaksi siten, että kunkin pelaajan hyöty määrittää hyötypisteen yhden koordinaatin. Riittävän suurilla diskonttauskertoimilla saavutetaan tilanne, jossa jokainen periaatteessa mahdolliseen hyötypisteeseen päästään jollain tasapainopolulla. Tämä lause tunnetaan peliteoriassa folk teoreemana. Kiinnostava kysymys tähän liittyen on, mikä on pienin diskonttauskerroin, jolla tämä täysi hyötyjoukko saavutetaan.

Vaikka tasapainopolut ovat äärettömän pitkiä ja niitä voi olla ääretön määrä, ne voidaan usein koota joukosta äärellisiä polun pätkiä. Toteutetun algoritmin toiminta perustuu juuri näiden elementaaristen osapolkujen etsintään. Lopuksi osapolut kootaan graafiksi, joka sisältää kaikki pelin tasapainopolut. Graafista löytyy myös pienimmän höydyn tuottava tasapainopolku, jota kutsutaan rangaistuspoluksi. Nimi johtuu siitä, että muut pelaajat voivat käyttää tätä polkua painostuskeinona. Rationaalinen pelaaja voidaan pakottaa mihin tahansa polkuun, josta hän saa paremman hyödyn kuin rangaistuspolusta. Siihen, onko jokin polku tasapainopolku, vaikuttaa siis olennaisesti kaksi asiaa: pelaajien käyttämät diskonttauskertoimet sekä lisäksi rangaistushyöty, jolla muut pelaajat voivat uhata pelaajaa. Graafin avulla voidaan myös piirtää kuva kaikista tasapainostrategioilla saatavista hyötypisteistä. Nämä hyötypisteet muodostavat fraktaalin, siis kuvion, jossa samat rakenteet toistuvat jatkuvasti aina pienemmässä mittakaavassa ja diskonttauskertoimen kasvaessa kuvioon tulee lisää pisteitä uusien tasapainopolkujen myötä. Kuvion täyttymisen tutkimiseksi tehtiin myös työkalu, jonka avulla löydetään helposti viimeinen alue, joka ei ole hyötypisteiden peittämä. Käytännössä algoritmi ei löytänyt kaikkia tasapainopolkuja, kun niiden määrä on hyvin suuri, mutta pystyi silti tuottamaan havainnollisia kuvia hyötypisteistä. Sen sijaan rangaistushyötyjen laskeminen onnistui useissa tapauksissa hyvin ja niihin liittyen tehtiin useita uusiakin havaintoja. Ensinnäkin pelaajat voidaan yleisesti pakottaa sitä huonompiin rangaistuksiin, mitä kärsivällisempiä he ovat. Tämä kehitys ei kuitenkaan ole täysin monotoninen, vaan toisinaan rangaistuksesta saatava hyöty kasvaa hieman, kun diskonttauskerroin kasvaa vähän. Lisäksi rangaistuspolut osoittautuivat joissain tapauksissa hyvin monimutkaisiksi. Tällaisten polkujen löytämiseksi ei aiemmin ole tehty mitään järjestelmällistä menetelmää.

Rangaistusten lisäksi saatiin mielenkiintoisia tuloksia hyötyjoukoille ja kyettiin löytämään useille peleille tarkkoja diskonttauskertoimen alarajoja, joista lähtien hyötypistejoukko peittää koko mahdollisen alueen. Tämä tarkoittaa sitä, että mikä tahansa rangaistusta suurempi, mutta pelin puitteissa mahdollinen hyöty voidaan saavuttaa jollain tasapainostrategialla.

Kandityössä avoimeksi kysymykseksi jäi, olisiko periaatteessa mahdollista löytää kaikki tasapainopolut myös tapauksissa, joissa niiden määrä on hyvin suuri. Kuitenkin rangaistuspolkujen ja höytyjoukkojen osalta päästiin tavoiteltuihin tuloksiin. Toistettujen pelien teoriassa hyötyjoukkoja on harvoin tarkasteltu tasapainopolkujen kautta, mutta työssä käytetyt menetelmät johtavat lopulta samoihin lopputuloksiin kuin aikaisemmissa tutkimuksissa. Näin ollen tasapainopolkujen etsimiseen perustuvia menetelmiä voidaan pitää varteenotettavana vaihtoehtona ratkaistaessa äärettömästi toistettuihin peleihin liittyviä ongelmia.