**xx manuscript No.**
(will be inserted by the editor)

# An algorithm for computing the minimum pure-strategy payoffs in repeated games

**Kimmo Berg and Markus Kärki**

Received: date / Accepted: date

**Abstract** This paper presents a method for computing the minimum pure-strategy subgame-perfect equilibrium payoffs in repeated games. These optimal punishments play an important role as they provide the players' credible threats and the required incentives to stay on the equilibrium path of play. The algorithm is based on the idea of branch-and-bound, and it produces lower and upper bounds for the minimum payoffs. The optimal punishment paths may be long in general and finding them is a difficult computational problem. It is also shown that approximations of bounded length can be obtained by relaxing either feasibility or optimality.

**Keywords** repeated game · minimum payoff · subgame perfection · pure strategy · Nash equilibrium

## 1 Introduction

Repeated games provide a foundation for studying rational behavior in dynamic interactions (Mailath and Samuelson 2006). These models also help understanding better how autonomous agents behave in a multiagent systems (Shoham and Leyton-Brown 2008). The main solution concept in these games is the subgame-perfect Nash equilibrium, which requires that the strategies have to form an equilibrium in all the possible decision nodes in the game. Thus, the players can only use credible threats in supporting equilibrium behavior. The structure of equilibrium strategies in infinitely repeated games has been characterized in Abreu (1988) and Abreu et al. (1990), where it is shown that all the equilibrium outcomes can be obtained in simple strategies. These strategies consist of an equilibrium path that is followed and a punishment path for each player that is implemented if the player unilaterally deviates

Department of Mathematics and Systems Analysis, Aalto University School of Science
P.O. Box 11100, FI-00076 Aalto, Finland
E-mail: kimmo.berg@aalto.fi

from the current path of play. Due to subgame-perfection, the punishment paths are equilibrium paths and each one of them provides that player's minimum payoff. This paper focuses on finding these paths and payoffs. They are especially important since they are required when the set of equilibria is computed.

The computation of Nash equilibria has received a lot of attention recently (Sandholm et al. 2005; Littman and Stone 2005; Daskalakis et al. 2006; Sandholm 2007; Porter et al. 2008; Herings and Peeters 2010; Sandholm 2012). The methods for infinitely repeated discounted games are usually based on the fixed-point characterization of Abreu et al. (1986, 1990). However, many of the methods rely on the use of public randomization and correlated strategies (Cronshaw and Luenberger 1994; Cronshaw 1997; Judd et al. 2003; Burkov and Chaib-draa 2010; Salcedo and Sultanum 2010; Abreu and Sannikov 2013), which simplifies the model considerably by making the payoff set convex; see also the extensions to the dynamic and stochastic games (Judd and Yeltekin 2011; Hörner et al. 2011). These methods produce fast approximations to the payoff set without providing the action sequences that generate the payoffs.

This paper examines a model that does not require the use of public randomization, but we restrict our analysis to pure strategies. The more general model is examined in Berg and Schoenmakers (2014), where the players can use randomized strategies but they only observe the realized pure actions. Moreover, we assume perfect monitoring, i.e., the players can observe all the past actions. Thus, the model can be seen as a problem of designing a deterministic path of pure actions such that no player wants to deviate from the plan at any stage. This model has been examined in Berg and Kitti (2012, 2013), where it is shown that the equilibrium paths consist of fragments called elementary subpaths; which also generalize to stochastic games (Berg 2012). The method of Berg and Kitti has made it possible to compute subgame-perfect equilibria by constructing the equilibrium paths from smaller fragments, and all the equilibria can be obtained if the game has a finite number of elementary subpaths, which is the case when the discount factors are small enough. A problem with this method is that it requires that the minimum equilibrium payoffs are known. This paper offers a solution to this problem by providing an algorithm for computing the minimum payoffs.

The task can be seen as an optimization problem where we try to find a path for each player that minimizes the player's payoff and no player should have a profitable deviation. These constraints mean that the punishment paths depend on each other, which makes the problem more complicated to solve since the paths should be searched simultaneously. We develop a method that finds the punishments paths systematically using the branch-and-bound method. It provides bounds for the minimum payoffs and a feasible equilibrium path as an upper bound.

We find that in many games the punishment paths have a particular finite structure, where the path consists of a starting sequence and a loop that is infinitely repeated. Our method enumerates all the possible paths of this type by increasing the length of the structure. We have noticed that the punishment

paths may be long in some games and this makes it difficult to find them. For this reason, we examine approximations where the punishment paths are guaranteed to be of finite length, which bounds the computational complexity.

The paper is structured as follows. The repeated game model and the notion of subgame-perfect equilibrium is defined in Section 2. The branch-and-bound algorithm is developed in Section 3. Numerical results are presented in Section 4. Section 5 develops approximations by relaxing feasibility and optimality. Section 6 is the conclusion.

## 2 Repeated games and definitions

In repeated games, the same stage game is played over and over again either finitely or infinitely many times. We examine the latter case and these models are sometimes called as supergames. The stage game can be defined with a tuple $(N, \{A_i\}_{i\in N}, \{u_i\}_{i\in N})$, where $N = \{1, \ldots, n\}$ denotes the finite set of players, $A_i$ is the finite set of actions and $A = \times_{i\in N} A_i$ is the set of action profiles. If the players choose actions $a = (a_1, \ldots, a_n)$, i.e., an action profile $a \in A$, then player $i$ receives the payoff $u_i(a)$. As usual, player $i$'s opponents' action profiles are denoted by $a_{-i} \in A_{-i} = \times_{j\neq i} A_j$, $j \in N$. Let $\overline{v}_i = \max_{a\in A} u_i(a)$ be the player $i$'s maximum payoff in the stage game and the minimax value is

$$\underline{v}_i = \min_{a_{-i}\in A_{-i}} \max_{a_i\in A_i} u_i(a_i, a_{-i}). \tag{1}$$

The best possible deviation by player $i$ from action profile $a$ is

$$v_i^*(a) = \max_{a_i\in A_i} u_i(a_i, a_{-i}). \tag{2}$$

We assume perfect monitoring, which means that the players observe and remember all the past action profiles that have been played in the game. The set of length $k$ histories is given by $A^k = \times_k A$, and the set of all possible histories is $\mathcal{A} = \bigcup_{k=0}^{\infty} A^k$, where $A^0 = \{\varnothing\}$ is the empty set, which corresponds to the beginning of the game where no actions have been played yet.

Note that the sets $A^k$ and $A^\infty$ contain the $k$-length and the infinitely long paths; $\mathcal{A}$ is the set of all paths. The length of path $p$ is denoted by $|p|$. Moreover, let $p_j$ denote the path that starts from the element $j+1$ of $p$ and $p^k$ is the path of first $k$ elements of $p$. For example, if $p = a^0a^1\ldots$, then $p_1 = a^1a^2\ldots$, $p^k = a^0\ldots a^{k-1}$ and $p_j^k = a^j\ldots a^{j+k-1}$. Let $f(p)$ denote the first action profile of path $p$. Similarly, $A^k(a)$, $A^\infty(a)$ and $\mathcal{A}(a)$ denote the set of paths that begin with an action profile $a \in A$.

We denote the action profiles by alphabets; e.g., the action profiles are $\{a, b, c, d\}$ in a two-player game with two actions. Now, we can denote an infinitely-long path by $d^7(cb)^\infty$, which means that the players first play the action profile $d$ seven times and then repeat infinitely the sequence of action profiles $c$ and $b$.

We assume that the players use pure strategies, i.e., we do not allow randomized nor correlated strategies. A pure strategy of player $i$ is a sequence of

mappings $\sigma_i^0, \sigma_i^1, \ldots$, where $\sigma_i^k : A^k \mapsto A_i$. The set of strategies for player $i$ is $\Sigma_i$, and the strategy profile composed of $\sigma_1, \ldots, \sigma_n$ is denoted by $\sigma$. Given a strategy profile $\sigma$ and a path $p$, the restriction of the strategy profile after $p$ is $\sigma|p$. The outcome path induced by $\sigma$ is $(a^0(\sigma), a^1(\sigma), \ldots) \in A^\infty$, where $a^k(\sigma) = \sigma^k(a^0(\sigma) \cdots a^{k-1}(\sigma))$ for all $k$.

We examine the case where the players discount the future payoffs with the discount factors $\delta_i \in [0, 1)$, $i \in N$. Note that this model also has an interpretation where the players have uncertainty about the length of the game and the discount factor reflects the probability that the game will continue. When the players choose strategies $\sigma_1, \ldots, \sigma_n$, i.e., they play a strategy profile $\sigma$, player $i$ receives the utility given by the discounted average payoff

$$U_i(\sigma) = (1 - \delta_i) \sum_{k=0}^{\infty} \delta_i^k u_i(a^k(\sigma)), \tag{3}$$

where the term $(1 - \delta_i)$ normalizes the repeated game payoffs such that they correspond to the stage game payoffs. A strategy $\sigma$ is a subgame-perfect equilibrium (SPE) of the supergame if

$$U_i(\sigma|p) \geq U_i(\sigma_i', \sigma_{-i}|p) \text{ for all } i \in N,\ p \in A^k,\ k \geq 0, \text{ and } \sigma_i' \in \Sigma_i.$$

This means that no player can gain by deviating from the given strategy $\sigma$ at any stage of the game. From now on, we refer equilibrium as subgame-perfect equilibrium. This paper focuses on the paths that can be played when the SPE strategies are used.

**Definition 1** A path $p \in A^\infty$ is a subgame-perfect equilibrium path if there is an SPE strategy profile that induces it.

It has been shown in Abreu (1988), see also Abreu et al. (1986, 1990), that the only thing that matters from the players' strategies is the induced path of play and what the players do if a unilateral deviation occurs. Abreu has shown that it is enough to study simple strategies when analyzing the set of equilibria. A simple strategy consists of $n+1$ paths $(p0, p1, \ldots, pn)$: an equilibrium path $p0$ that is played and an optimal punishment path $pi$ for each player $i \in N$. The punishment paths are equilibrium paths themselves that give the players' minimum equilibrium payoffs. The play follows the current path unless a single player $j \in N$ deviates from it. In that case, the punishment path $pj$ is restarted and it becomes the new path to be followed. The deviations by more than one player are neglected and they need not be considered as we examine non-cooperative games. The current path is initially $p0$ and after deviation(s) one of the paths $pi$, $i \in N$.

The set of SPE payoffs has been characterized in Abreu et al. (1986, 1990). Let $V$ denote the compact set of SPE payoffs. The minimum SPE payoff of player $i$ is denoted by $v_i^-(\delta)$ when the players discount factors are $\delta = (\delta_1, \ldots, \delta_n)$, if $V$ is non-empty. It should be noted that the set of equilibria may be empty in pure strategies, but it is assumed that this is not the case. We assume that the stage game has at least one pure-strategy Nash equilibrium,

which guarantees that $V$ is non-empty. The equilibrium conditions for the SPE paths are given by the following one-shot deviation principle. A path $p = a^0(\sigma)a^1(\sigma)\cdots$ induced by a strategy $\sigma$ is an SPE path if and only if

$$(1-\delta_i)u_i\left(a^k(\sigma)\right) + \delta_i v_i^k \geq \max_{a_i \in A_i}\left[(1-\delta_i)u_i\left(a_i, a_{-i}^k(\sigma)\right) + \delta_i v_i^-(\delta)\right], \quad (4)$$

for all $i \in N$, $k \geq 0$, and where

$$v_i^k = (1-\delta_i)\sum_{j=0}^{\infty}\delta_i^j u_i\left(a^{k+1+j}(\sigma)\right)$$

is the continuation payoff after $a^k(\sigma)$. The incentive compatibility (IC) condition (4) means that the players should prefer the payoffs given by path $p$ to any deviations at any stage that are followed by the punishment paths with payoffs $v^-(\delta)$. We say that a path is feasible if it is incentive compatible.

The set of equilibria is recursive in the sense that all the equilibrium paths depend on and are supported by the punishment paths and their payoffs $v^-(\delta)$. Note that the punishment paths may depend on each other. In general, the minimum payoffs are not known, but with perfect monitoring they are above the minimax values $v_i^-(\delta) \geq \underline{v}_i$. The aim of this paper is to find the punishment paths and the corresponding minimum equilibrium payoffs for different discount factors. Note, however, that the optimal punishments may not be required for all equilibria; e.g., the repetition of stage game Nash equilibrium has no profitable deviations and thus requires no punishment at all.

It should be noted that the minimum SPE payoff may be smaller in randomized strategies. For example, the set of equilibria in the matching pennies game is empty in pure strategies, whereas there is a single mixed-strategy subgame-perfect equilibrium outcome. Moreover, it is possible to obtain a lower SPE payoff in the battle-of-the-sexes game with randomized strategies.

### 2.1 Monotonicity of equilibria

The following results illustrate the difference between the monotonicity of equilibrium paths and payoffs, and how they depend on the convexity assumptions and the minimum payoffs. The proofs are from Berg (2013); Berg and Kärki (2014).

Let $V(\delta)$ denote the payoff set when the players have the discount factors $\delta = (\delta_1, \ldots, \delta_n)$. By $\delta^2 \geq \delta^1$ we mean that $\delta_i^2 \geq \delta_i^1$ for all $i \in N$. We say that a pair $(a, w)$ of an action profile $a \in A$ and a continuation payoff $w \in W$ is admissible with respect to $W$ if it satisfies the incentive compatibility conditions:

$$(1-\delta_i)u_i(a) + \delta_i w_i \geq (1-\delta_i)d_i(a) + \delta_i v_i^-(W),$$

for all $i \in N$. Let us define a mapping $B^\delta : \mathbb{R}^n \mapsto \mathbb{R}^n$

$$B^\delta(W) = \bigcup_{(a,w)\in A\times W} (I-T)u(a) + Tw, \quad (5)$$

where $(a, w)$ is admissible with respect to $W$, $I$ is an $n \times n$ identity matrix, and $T$ is a diagonal matrix with $\delta_1, \ldots, \delta_n$ on the diagonal. The following proposition is proven in Abreu et al. (1990); Mailath and Samuelson (2006).

**Proposition 1** *If a bounded set $W$ is self-generating, i.e., $W \subseteq B^\delta(W)$, then $B^\delta(W) \subseteq V(\delta)$.*

**Theorem 1** *Suppose $V(\delta^1)$ is convex then $V(\delta^1) \subseteq V(\delta^2)$ for $\delta^2 \geq \delta^1$.*

*Proof* By Proposition 1, it is enough to show that for all $v \in V(\delta_1)$ it holds that $v \in B^{\delta_2}(V(\delta_1))$. It is enough to show that there is an admissible pair $(a, w^2)$ of an action profile $a$ and a continuation payoff $w^2 \in V(\delta_1)$ such that $v = (1-\delta_2)u(a) + \delta_2 w^2$ for every $v \in V(\delta_1)$, i.e., $(a, w^1)$ is admissible for a continuation payoff $w^1 \in V(\delta_1)$ and $v = (1-\delta_1)u(a) + \delta_1 w^1$. By denoting $\delta_2 = \delta_1 + \epsilon$, we can solve

$$(\delta_1 + \epsilon)w^2 = \delta_1 w^1 + \epsilon u(a).$$

This means that $w^2$ is a convex combination of $w^1$ and $u(a)$. Combining with the result that $v$ is between $u(a)$ and both $w^1$ and $w^2$, we get that $w^2$ is between $w^1$ and $v$. Thus, it follows that $w^2 \in V(\delta_1)$ by convexity and $v, w^1 \in V(\delta_1)$. Finally, we need to check the admissibility with $\delta_2$. The only remaining thing to check is that the punishment payoff is not increasing and it is not since $v_i^-(V(\delta_1)) \in V_i(\delta_1)$ for all $i \in N$ and together with the above result we have $v^-(V(\delta_2)) \leq v^-(V(\delta_1))$.

**Theorem 2** *Suppose a path $p \in A^\infty$ is an SPE path for $\delta^1$ and $v^-(\delta^1) \geq v^-(\delta^2)$, then $p$ is an SPE path for $\delta^2 \geq \delta^1$.*

*Proof* Let $u^k$ denote the payoffs at stage $k$ on path $p$, i.e., $u^k = u(a^k)$ when the action profile $a^k$ is played at stage $k$. Also, let $d^k = d(a^k)$ denote the deviation payoffs, and $T_1$ and $T_2$ are the diagonal matrices corresponding the discount factors $\delta_1$ and $\delta_2$, respectively. Now, we can rewrite the incentive compatibility conditions for $\delta_1$:

$$(I - T_1)u^k + T_1 \left[ (I - T_1) \sum_{j=0}^{\infty} T_1^j u^{k+j+1} \right] \geq (I - T_1)d^k + T_1 v^-(V(\delta_1)),$$

for all $k \geq 0$. Let us rearrange the equation and multiply from left by $(I - T_1)^{-1}$:

$$S_1^k \doteq u^k - d^k + T_1 \sum_{j=0}^{\infty} T_1^j \left( u^{k+j+1} - v^-(V(\delta_1)) \right) \geq \mathbf{0} \text{ for all } k = 0, 1, \ldots.$$

Similar expression can be derived for $S_2^k$ with $T_2$, and the purpose of the proof is to show that $S_2^k \geq \mathbf{0}$ for all $k \geq 0$, which means that the incentive compatibility conditions hold for $T_2$ along the path $p$.

We can solve the recursion which $S_i^k$ satisfies:

$$S_i^k = u^k - d^k + T_i \left( d^{k+1} - v^-(V(\delta_i)) + S_i^{k+1} \right), \text{ for all } k \geq 0 \text{ and } i = 1, 2.$$

Note that the first part $u^k - d^k$ is a vector with non-positive components, which implies that the components of the second part $d^{k+1} - v^-(V(\delta_1)) + S_1^{k+1}$, $k \geq 0$, must be non-negative, since $S_1^k \geq \mathbf{0}$ due to incentive compatibility.

Let $\delta_2 = \delta_1 + \epsilon$ and $E$ is the diagonal matrix corresponding $\epsilon \geq \mathbf{0}$. We can simplify the expression for $\delta_2$:

$$\begin{aligned} S_2^k &\geq u^k - d^k + (T_1 + E)\left(d^{k+1} - v^-(V(\delta_1)) + S_2^{k+1}\right) \\ &= S_1^k + E\left(d^{k+1} - v^-(V(\delta_1)) + S_1^{k+1}\right) + (T_1 + E)\left(S_2^{k+1} - S_1^{k+1}\right), \end{aligned}$$

where the first inequality follows from the fact that $v^-(V(\delta_1)) \geq v^-(V(\delta_2))$. Now, we can write

$$\begin{aligned} S_2^k - S_1^k &\geq E\left(d^{k+1} - v^-(V(\delta_1)) + S_1^{k+1}\right) + (T_1 + E)\left(S_2^{k+1} - S_1^{k+1}\right) \\ &\geq (T_1 + E)\left(S_2^{k+1} - S_1^{k+1}\right), \end{aligned}$$

where the inequality follows from the earlier observed non-negativity of $d^{k+1} - v^-(V(\delta_1)) + S_1^{k+1}$. Now, we can use this recursion:

$$S_2^k - S_1^k \geq E \sum_{j=0}^{\infty} (T_1 + E)^k Z_k,$$

where $Z_k = d^{k+1} - v^-(V(\delta_1)) + S_1^{k+1} \geq \mathbf{0}$. Thus, $S_2^k \geq S_1^k \geq \mathbf{0}$. □

Note that the path monotonicity is a more robust property since the convexity assumption is not required and the punishment payoffs are monotone in many games for all discount factors. For example, the equilibrium paths are monotone in all prisoner's dilemma games where the punishment payoffs are constants, but the payoff sets are not monotone in general in these games (Berg and Kärki 2014).

## 3 Branch-and-bound algorithm

We now present the main algorithm and its steps are explained in more detail in the subsections. The algorithm is based on systematically examining all the finite-length paths and discarding the ones that either have some profitable deviations or do not provide small enough payoff to the punished player. Since there can be a huge number of paths, we start with the ones that provide small payoffs to the players. This heuristic provides good upper bounds fast, which helps reducing the paths that need to be examined. However, proving optimality can be slow as it requires going through all the remaining paths. This is a typical feature in integer optimization, where heuristics provide good solutions fast but proving optimality is slow.

Algorithm 1 consists of three steps and it maintains two lists: a set $B$ contains the finite sequences how the punishment paths start, and a set $Q$ contains infinitely-long paths. The set $B$ is initialized by generating all the one-length paths. If all of them have a profitable deviation for some player,

**Algorithm 1:** Compute the minimum paths and payoffs

**Input**: Stage game payoffs, discount factors $\delta_i$, error tolerance $\epsilon$, maximum path length *maxlen*, parameter *num* for number of paths
**Result**: Bounds for $v^-(\delta)$, best found feasible paths
Initialize the set of paths $B$;
Set $ub \leftarrow \overline{v}$ and $lb \leftarrow \underline{v}$;
**while** $ub_i - lb_i > \epsilon$, $\forall i$ **do**
  1. Select *num* paths for each active player from $B$;
  2. For each selected path, we do
  a) Check that the punished player does not deviate at any stage;
  **if** *there is a profitable deviation* **then** Move to next path;
  b) Compute the minimum continuation payoff requirements for all players;
  **if** *lowest payoff from the path is higher than* $ub_i$ **then** Move to next path;
  **else if** *any cont. payoff requirement is too high* **then** Move to next path;
  c) Form and add the possible punishment paths to $Q$;
  3. Find the minimum feasible paths from $Q$ and update $B$;
**end**

then the stage game does not have a pure-strategy Nash equilibrium and the algorithm is terminated as the set of repeated game equilibria is empty as well.

In Step 1, we choose a number of paths given by the parameter *num* for the active players, i.e., for each player $i$ for which $ub_i - lb_i > \epsilon$, where $lb$ and $ub$ are the players' lower and upper bounds for the minimum payoffs. The parameter *num* affects how large integer program needs to be solved in Step 3. The chosen paths are required to be shorter than the parameter *maxlen* in order to bound the required computation time. The paths can, e.g., be chosen based on the length, the payoff or the total payoff that takes into account the required continuation payoffs. In the algorithm, we select the paths based on total payoff, which are equal to the lower bound estimates in Eq. (9).

### 3.1 Step 2a: Checking deviations

In Step 2, we go through the selected paths and generate the possible punishment paths from them. Step 2a checks that the punished player does not deviate from the examined path. For example, if the path starts with *abca* and the player could get better payoff by deviating from $c$ to $a$, then the outcome path would be $(aba)^\infty$, and the path starting with *abca* cannot be the punishment path.

Let $M(k)$ be the payoff of the punished player if he always deviates from the $k$-th action profile

$$M(k) = \frac{1-\delta_i}{1-\delta_i^k}\left[U_i(p^{k-1}) + \delta_i^{k-1} v_i^*(f(p_{k-1}))\right], \tag{6}$$

where $v_i^*(f(p))$ is the best possible deviation as in Eq. (2) and $f(p)$ the first action profile of path $p$. The deviation is not profitable if $M(k)$ is smaller than or equal to the payoff that the player receives when path $p$ is followed. Now, if

any $M(k) \geq ub_i$ then also path $p$ gives a higher payoff than $ub_i$ and the path cannot be the punishment path. Thus, if the following condition does not hold

$$ub_i \geq \frac{1-\delta_i}{1-\delta_i^k}\left[U_i(pi^{k-1}) + \delta_i^{k-1} v_i^*(f(pi_{k-1}))\right], \quad \forall k, \tag{7}$$

then the path can be discarded.

### 3.2 Step 2b: Computing the minimum continuation payoffs

Step 2b computes the minimum continuation payoffs that are required after the examined path for all players. These values can be used in discarding paths that are not feasible or provide too high payoffs compared to the current upper bounds.

Let $con_i^j$ denote the minimum continuation payoff of player $i$ that is required after the punishment path of player $j$, $pj$, which can be solved from Eq. (4)

$$con_i^j = \max_{k=1,\ldots,|pj|} \left[(1-\delta_i)v_i^*(f(pj_{k-1})) + \delta_i lb_i(par(b)) - U_i(pj_{k-1})\right] / \delta_i^{|pj|-k+1}, \tag{8}$$

where $par(b)$ is the parent node of path $b$. The index $k$ goes through all the possible stages where the player can deviate, and the deviation is followed by the punishment payoff of the parent node $lb_i(par(b))$. Now, the lower bound of path $b$ can be computed

$$lb_i(b) = (1-\delta_i)U_i(pi) + \delta_i^{|pi|} con_i^i, \tag{9}$$

where $U_i(pi)$ is the payoff of player $i$ from path $pi$. This assumes that the minimum continuation payoff after $pi$, $con_i^i$, can be achieved after $pi$ is played. If the lower bound of the path is higher than the global upper bound, i.e., $lb_i(b) \geq ub_i$, then the path can be discarded. Thus, the path should be examined only if

$$con_i^i < \delta_i^{-|pi|}(ub_i - (1-\delta_i)U_i(p_i)). \tag{10}$$

The computation of $con$ also helps cutting the infeasible paths. If a path requires too high continuation payoff that cannot be achieved in the game, then the path can by discarded. The path can be an equilibrium only if

$$con_i^j \leq \overline{v}_i, \quad \forall i, j. \tag{11}$$

The value $\overline{v}_i$ can be replaced by a better bound if one is known for the game. If either of the conditions (10) and (11) does not hold, the path is discarded.

### 3.3 Steps 2c and 3: Compute the bounds and update $B$

For each path $p$, we form all the possible loops: $qk = p^{k-1}(p_{k-1})^{\infty}$ for $k = 1, \ldots, |p|$. For each loop, we compute the players' payoffs and the minimum punishment payoffs that are required in order for the path to be feasible. If the payoff is smaller than the current upper bound, we add the infinitely-long path to the set $Q$. For example, for a path $bcd$ we form paths $(bcd)^{\infty}$, $b(cd)^{\infty}$ and $bcd^{\infty}$. We also generate all the children of the path; e.g., for a path $bd$ we add paths $bda, bdb, bdc, bdd$ to $B$ if the game has four action profiles.

In Step 3, we have a list $Q$ of infinitely-long paths with payoffs that they provide and payoffs that are required for them to be feasible. We solve a binary linear program that finds for each player the feasible path from $Q$ that gives the minimum payoff. The binary variables select one path for each player and the objective is to minimize the sum of the payoffs of the selected paths. The constraints make sure that the selected paths are feasible, i.e., the selected paths need to yield smaller payoffs than the required continuation payoffs. If the upper bounds are improved, the paths that now give too high payoffs are discarded from the set $B$.

### 3.4 Maximum and Pareto efficient payoffs

A similar method can be used in computing the maximum payoffs once the minimum payoffs are known. This is much easier task since these paths do not depend on each other and the integer program in Step 3 is not needed any more. We can discard the paths for three reasons: 1) if it is infeasible, 2) if it provides too low payoff compared to the best found paths, or 3) if the required continuation payoff cannot be achieved in the game. These correspond to the conditions in Step 2. Moreover, this extension allows us to find those Pareto efficient payoffs that can be represented as a weighted sum of the players payoffs. Thus, the method can be used in computing the extreme payoffs of the supergame.

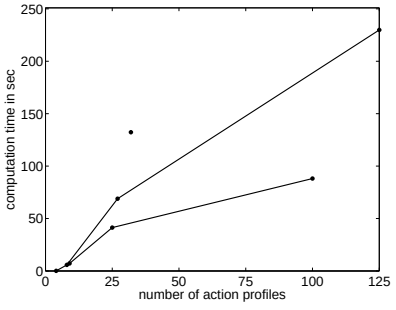
## 4 Numerical results

### 4.1 Random games

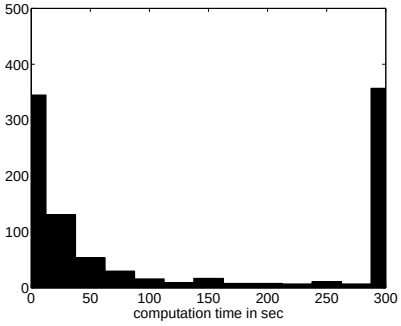
We test the algorithm with randomly generated normal-form games. The payoffs and the discount factors are randomly drawn from the uniform distribution between zero and one, and the game is discarded if it does not have a pure-strategy Nash equilibrium. On average, around 1400 games were generated in order to obtain 1000 games with at least one pure-strategy equilibrium; i.e., 30% of the games had an empty set of equilibria. We set $maxlen = 6$, $\epsilon = 0.01$, repeat for 1000 games and stop the computation after 300 seconds if it takes longer than that. Moreover, we set $num = 50$ for two-player games

Table 1: The results for the random games.

| n | Acts | Time | #OT | OT $\epsilon$ | #OL | OL $\epsilon$ | Len1 | Len |
|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 0.09 | 0 | - | 261 | 0.063 | 87% | 4.4 |
| 2 | 3 | 7.16 | 18 | 0.075 | 191 | 0.034 | 73% | 3.9 |
| 2 | 5 | 41.3 | 112 | 0.066 | 127 | 0.025 | 57% | 3.6 |
| 2 | 10 | 88.2 | 233 | 0.049 | 77 | 0.021 | 51% | 3.1 |
| 3 | 2 | 5.81 | 10 | 0.149 | 235 | 0.041 | 67% | 4.4 |
| 3 | 3 | 68.9 | 189 | 0.087 | 225 | 0.043 | 47% | 4.1 |
| 3 | 5 | 230 | 686 | 0.090 | 59 | 0.047 | 39% | 3.1 |
| 5 | 2 | 132 | 353 | 0.092 | 440 | 0.044 | 42% | 4.1 |




(a) Average computation times




(b) Histogram of times in 5-player games

Fig. 1: Computation times in random games.

and $num = 25$ for the others. The results are given in Table 1 and Figure 1. Time column gives the average computation time of the games. #OT column shows the number of games that hit the time limit, and OT $\epsilon$ column gives the average gap between lower and upper bounds for the overtime games. #OL column gives the number of games that reached the point where all paths up to *maxlen* are examined, and OL $\epsilon$ column gives the error gap for these games. Len1 column gives the percentage of paths that are of length one and Len shows the average length of paths longer than one. In the figure, the times of the two-player and three-player games are connected with a line. The runs were conducted on Intel Core i5-520M at 2.40 GHz with 3 GB of RAM under 32-bit Windows 7. The algorithms were implemented in Matlab 7.10.0.499 (R2010a) and CPLEX 12.6.0 (cplexbilp).

We can see from the results that the computation times grow reasonably when the number of action profiles increase and it is fast to solve games up to 100 action profiles. Figure 1b shows a typical distribution over the repetitions: some games hit the 300 second limit and the other times follow a geometric distribution. We did not solve larger games as the number of action profiles grow fast in the normal-form games. For example, a ten-player game with only two actions has $2^{10} = 1024$ action profiles. The errors in the overtime games

Table 2: The results for the maximum payoffs.

| $n$ | *Acts* | Time | #OT | OT $\epsilon$ | #OL | OL $\epsilon$ | *Len*1 | Len | Single | Range |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 0.12 | 0 | - | 68 | 0.054 | 91% | 4.4 | 583 | 0.29 |
| 2 | 3 | 7.75 | 20 | 0.090 | 23 | 0.053 | 83% | 4.1 | 364 | 0.29 |
| 2 | 5 | 25.1 | 74 | 0.054 | 4 | 0.028 | 76% | 3.4 | 174 | 0.26 |
| 2 | 10 | 33.0 | 88 | 0.036 | 0 | - | 78% | 2.6 | 177 | 0.19 |
| 3 | 2 | 5.99 | 13 | 0.036 | 92 | 0.044 | 71% | 4.1 | 254 | 0.37 |
| 3 | 3 | 26.7 | 72 | 0.044 | 22 | 0.037 | 62% | 3.6 | 112 | 0.43 |
| 3 | 5 | 53.1 | 138 | 0.036 | 0 | - | 68% | 2.7 | 92 | 0.38 |
| 5 | 2 | 27.0 | 61 | 0.040 | 79 | 0.027 | 50% | 3.4 | 68 | 0.58 |

are small, which means that it is possible to find good approximations even for the most difficult games in the sample. Many of the games also hit the *maxlen* limit and the average error is larger than $\epsilon$ in these games. The average errors are, however, smaller than the errors in the games that hit the time limit.

The real errors are much smaller than the found error bounds. We have solved the $2 \times 2$ OL games with a $maxlen = 12$ value and the average error falls from 0.063 to 0.028. A half of these games are solved to $\epsilon = 0.01$ limit and the other half hit the *maxlen* limit again. The upper bounds decrease on average by 0.0019, which means that mainly the lower bounds are increased. Thus, the found solutions seem to be close to the optimal ones and the problem with the method is proving the optimality, i.e., updating the lower bounds and eliminating the infeasible paths.

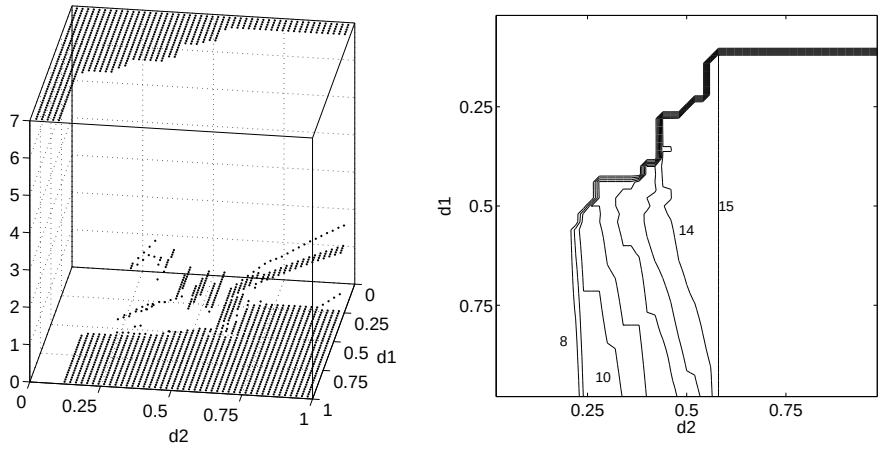
The results for the maximum payoffs are given in Table 2. *Single* column shows the number of games where the payoff set is a single point and *Range* gives the average between the maximum and minimum payoff for cases where these payoffs do not coincide. The computation times are much smaller, especially for bigger games. There are much less games that hit the time cap or the *maxlen* limit. Moreover, the paths with maximum payoffs are on average shorter than the punishment paths.

### 4.2 Duopoly game of Abreu

Oligopoly models are one of the most studied applications of repeated games. Here, we examine the duopoly model of Abreu (1988), where the punishment strategies and the payoff sets for different discount factors have been a mystery until now.

| | L | M | H |
|---|---|---|---|
| L | $10, 10$ $(a)$ | $3, 15$ $(b)$ | $0,\ \ 7$ $(c)$ |
| M | $15,\ \ 3$ $(d)$ | $7,\ \ 7$ $(e)$ | $-4,\ \ 5$ $(f)$ |
| H | $7,\ \ 0$ $(g)$ | $5, -4$ $(h)$ | $-15, -15$ $(i)$ |

The firms have three output levels: low (L), medium (M) and high (H). The nine action profiles are denoted by letters $a$ to $i$, and the stage game's Nash




(a) Minimum payoff

(b) Maximum payoff

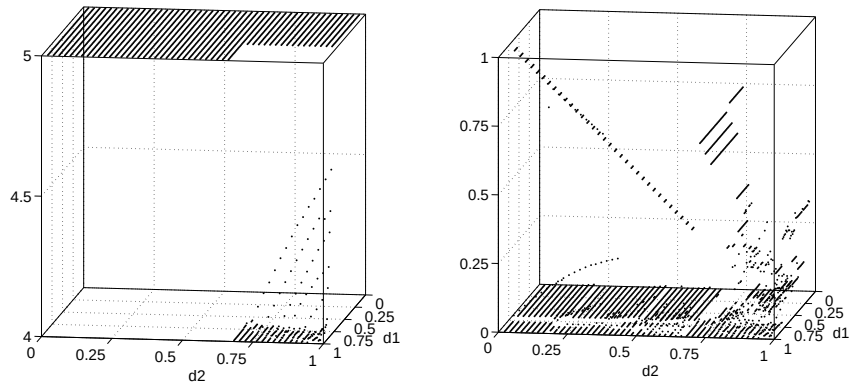
Fig. 2: Player 1's payoffs for different discount factors in duopoly.

equilibrium is $e$, i.e., $(M, M)$, giving payoff 7. The minimax payoff is $\underline{v}_i = 0$, and thus for all discount factors it holds that $0 \leq v_i^-(\delta) \leq 7$, $i = 1, 2$.

We run the algorithm with the same parameters as before, except we allow the punishment paths to be longer by setting $maxlen = 12$. The upper bounds and the maximum payoffs of player 1 are presented in Figure 2, where $d1$ and $d2$ refer to the players' discount factors. For low discount factors the punishment strategy is to play the stage game's Nash equilibrium and the punishment payoff for high discount factors is the minimax value. These are obtained by playing very simple paths $e^\infty$ and $c^\infty$. In between, there is a region where the punishment strategies are more complicated and the punishment payoff is roughly between 0 and 2. For example, the optimal paths found for $(\delta_1, \delta_2) = (0.4, 0.66)$ are $p1 = c^\infty$ and $p2 = h(hbhd)^\infty$ giving payoffs 0 and 0.03.

Note that the punishment payoffs are not monotone in this game as the punishment payoffs may increase when one of the players becomes more patient. Thus, this example suggests that the equilibria may not be monotone in real-world applications, which complements the results presented in Berg (2013).

The errors are large only in few isolated points and two bigger regions (the errors are between 3-7). The two regions are when one player has a discount factor over 0.7 and the other under 0.1. In these regions the punishment payoff is probably 7 but the method has problems updating the lower bounds. In other regions, the errors are basically zero. 211 games out of 2401 took over 300 seconds to solve and for others the average time is 6.1 seconds. 275 games hit the $maxlen$ limit and the average error for these games is 0.23. The average error for the overtime instances is 3.91, which is quite high.




(a) Best found

(b) Errors

Fig. 3: Player 1's punishment payoffs in anti-no conflict game.

The regions with high maximum payoff coincide with the low minimum payoffs. The maximum payoffs depend on both players' punishment payoffs, and we can see that the maximum payoff of player 1 increases as the punishment payoff of player 2 decreases. Playing $d$ with payoff $(15, 3)$ requires that player 2 does not want to switch to action profile $e$ with payoff 7. This is achieved when the punishment payoff after the deviation is close to zero. Note also that the maximum payoff is not monotone with respect to the player's own discount factor.

### 4.3 Anti-no conflict game

There are only few types of $2 \times 2$ games that have nontrivial punishment strategies and here we examine one of them, which is called the anti-no conflict game. The game itself is artificial as it has a dominant Nash equilibrium. However, it is important to find the minimum payoffs, since this type of payoffs can be a part of some larger game and finding a lower payoff than 5 may support some other action profiles that give higher payoff than 5 in the larger game.

| | $L$ | $R$ |
|---|---|---|
| $T$ | $5, 5$ | $4, 3$ |
| $B$ | $3, 4$ | $2, 2$ |

The minimax value is 4, but the corresponding action profiles $(T, R)$ and $(B, L)$ cannot be played repeatedly since they give a smaller payoff than 4 to the punishing player. The upper bounds and the errors are shown in Figure 3.

We can see from the figures that it is possible to punish the other player if the discount factors are high enough. The minimum payoffs are close to the minimax values, and the punishment paths are long and complicated for

high discount factor values. For example, it is possible to punish with a path $daac(baaaadaa)^\infty$ with a payoff of $4+2.9\cdot 10^{-7}$, when $\delta_1=\delta_2=0.8$. However, it should be noted that many times it is possible to obtain a low payoff with a short path. For example, the path $dca^\infty$ gives a payoff of 4.08 with the given discount factors. Thus, it may be that simple punishments are enough in order to support good outcomes and the difficult-to-find optimal punishments are not required. Moreover, we note again that the algorithm has some problems in updating the lower bounds when one discount factor is low and the other one is high.

## 5 Approximations

Discounting makes the payoffs far ahead in the future less important and the first action profiles on the path determine mainly the players' payoffs. This makes it possible to bound the length of the paths that need to be examined if we restrict to certain approximations.

It has been shown in Berg and Kitti (2012) that all the equilibrium paths consist of fragments called elementary subpaths. The following result tells us that the length of the elementary subpaths is finite for paths whose payoff is strictly above the deviation payoffs; see Proposition 3 in Berg and Kitti (2012).

**Proposition 2** *For any $\varepsilon>0$ there is $k$ such that $p^l$ is an $l$-length elementary subpath for some $l\leq k$ when $p\in A^\infty(a)$ is an SPE path, $a\in A$, and*

$$v_i(p)\geq(1-\delta_i)v_i^*(a)+\delta_i v_i^-(\delta)+\varepsilon,\ \textit{for all } i\in N. \tag{12}$$

This means that if a path is such that no player is near to deviating at any stage then that path can be composed of elementary subpaths which are of finite length. Morever, the more the payoffs are above the deviation payoffs, the shorter the required elementary subpaths are.

Now, if we restict to paths where the players are not near to deviating, we can bound the length of elementary subpaths that need to be examined. Let a strategy profile $\sigma$ be an $\varepsilon$-strict incentive compatible equilibrium if all one-shot deviations from $\sigma$ for any player lead to payoffs that are worse than the deviating player's original payoff by at least $\varepsilon$. Note that the more common notion of $\varepsilon$-strict equilibrium means that a strategy profile $\sigma$ satisfies

$$U_i(\sigma|p)\geq U_i(\sigma_i',\sigma_{-i}|p)+\varepsilon \text{ for all } i\in N,\ p\in A^k,\ 0\leq k<\infty, \text{ and } \sigma_i'\in\Sigma_i.$$

These strategies would allow for more general deviations than only one-shot deviations provided that they lead to payoffs that are worse than the original payoff by at least $\varepsilon$. Any strategy that is not $\varepsilon$-strict incentive compatible equilibrium cannot be $\epsilon$-strict equilibrium either. Hence, $\varepsilon$-strict equilibria are a subset of $\varepsilon$-strict incentive compatible equilibria.

An $\varepsilon$-strict incentive compatible equilibrium path is a path of action profiles induced by an $\varepsilon$-strict incentive compatible equilibrium strategy. If all elementary subpaths are found up to length $k(\varepsilon)$, then all $\varepsilon$-strict equilibria are obtained from these subpaths; see Proposition 9 in Berg and Kitti (2012).

**Proposition 3** *For any $\varepsilon > 0$ there is $k$ such that all $\varepsilon$-strict incentive compatible equilibrium paths are obtained from the up to $k$-length elementary subpaths.*

It is also possible to relax the feasibility and get all 0-equilibria. Let a strategy profile $\sigma$ be an $\varepsilon$-incentive compatible equilibrium if no player has a one-shot deviation from $\sigma$ that would benefit the deviating player by at most $\varepsilon$. Moreover, a strategy $\sigma$ is an $\varepsilon$-equilibrium if

$$U_i(\sigma|p) \geq U_i(\sigma_i', \sigma_{-i}|p) - \varepsilon \text{ for all } i \in N,\ p \in A^k,\ 0 \leq k < \infty, \text{ and } \sigma_i' \in \Sigma_i.$$

Hence, $\varepsilon$-incentive compatible equilibria are a subset of $\varepsilon$-equilibria. A path that is induced by an $\varepsilon$-incentive compatible equilibrium strategy profile is an $\varepsilon$-incentive compatible equilibrium path. Proposition 8 in Berg and Kitti (2012) shows that all 0-equilibria (and some $\varepsilon$-equilibria) can be formed from elementary subpaths up to length $k$.

These results help in finding the punishment strategies if we restrict the search to either $\epsilon$-strict equilibria or relax the feasibility to $\varepsilon$-equilibrium. It is possible to combine these results with the method presented in Berg (2013). In that method the punishment values are set below the optimal values and then all equilibria is computed. The punishment values are increased if the payoffs from the found paths are higher than the current values, i.e., there are no paths that give that low payoffs to the players. Since the punishment values are lower than the optimal punishments, the equilibrium paths are subset of the paths found in the algorithm. By restricting to the above approximations, this algorithm would terminate in a finite number of iterations.

These are theoretical results and they can also be implemented with Algorithm 1. By limiting the *maxlen* parameter, we get the punishment paths for $\epsilon$-strict equilibria, and by relaxing the equations related to the feasibility we find punishment paths that give lower payoffs compared to 0-equilibria with paths that are $\varepsilon$-equilibria.

## 6 Conclusions and future research

In this paper, we have presented an algorithm for computing the minimum pure-strategy subgame-perfect equilibrium paths and payoffs. The algorithm applies the idea of branch-and-bound, and it systematically considers all the possible finite paths for each player. The method tries to find paths that give a low payoff to one of the players, but so that none of the players want to deviate from the path. These paths provide upper bounds to the minimum payoffs. Once good upper bounds are found, many of the paths can be discarded as they give too high payoffs. Based on the numerical experiments, the method works well when the number of action profiles is small enough. For larger games, approximations can be computed by relaxing either optimality or feasibility of the paths.

The method makes it possible to study the punishment strategies as well as the Pareto efficient solutions, which has only been possible before for small

games (Berg 2013). For example, we have demonstrated the method in a duopoly game and observed that the punishment paths can be complicated for certain discount factors and the minimum payoffs are not monotone. It is left for future research how to make the algorithm faster. It finds good solutions fast but takes long time to prove the optimality. Thus, there may be some better way to discard the infeasible paths and update the lower bounds faster. One idea is to combine the search with finding the maximum payoffs or the payoff set as a whole, which helps cut some of the infeasible paths. For example, if it is known that certain high payoffs cannot be achieved in the game, then the paths that require such continuation payoffs can be discarded. Moreover, it would be interesting to examine how the results change if the players are allowed to randomize between the pure actions.

## Acknowledgements

Kimmo Berg acknowledges funding from Emil Aaltosen Säätiö through Post doc -pooli.

## References

Abreu, D. (1988). On the theory of infinitely repeated games with discounting. *Econometrica*, 56 (2), 383–396.

Abreu, D., Pearce, D., & Stacchetti, E. (1986). Optimal cartel equilibria with imperfect monitoring. *Journal of Economic Theory*, 39 (1), 251–269.

Abreu, D., Pearce, D., & Stacchetti, E. (1990). Toward a theory of discounted repeated games with imperfect monitoring. *Econometrica*, 58 (5), 1041–1063.

Abreu, D., & Rubinstein, A. (1988). The Structure of Nash Equilibrium in Repeated Games with Finite Automata. *Econometrica*, 56 6, 1259–1281.

Abreu, D., & Sannikov, Y. (2013). An algorithm for two-player games with perfect monitoring. *Theoretical Economics*, in press.

Berg, K. (2012). Characterization of equilibrium paths in discounted stochastic games. Working paper.

Berg, K. (2013). Extremal Pure Strategies and Monotonicity in Repeated Games. Working paper.

Berg, K., & Kitti, M. (2012). Equilibrium paths in discounted supergames. Working paper. http://sal.aalto.fi/publications/pdf-files/mber09b.pdf

Berg, K., & Kitti, M. (2014). Fractal geometry of equilibrium payoffs in discounted supergames. *Fractals*, 22 (4). http://sal.aalto.fi/publications/pdf-files/pber14.pdf

Berg, K., & Kitti, M. (2013). Computing equilibria in discounted $2 \times 2$ supergames. *Computational Economics*, 41, 71–78.

Berg, K., & Kärki, M. (2014). How patient the players need to be to get all the relevant payoffs in the symmetric $2 \times 2$ supergames? Working paper.

Berg, K., & Schoenmakers, G. (2014). Construction of randomized subgame-perfect equilibria in repeated games. Working paper.

Burkov, A., & Chaib-draa, B. (2010). An Approximate Subgame-Perfect Equilibrium Computation Technique for Repeated Games. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 729–736.

Cronshaw, M. B. (1997). Algorithms for finding repeated game equilibria. *Computational Economics*, 10, 139–168.

Cronshaw, M. B., & Luenberger, D. G. (1994). Strongly symmetric subgame perfect equilibria in infinitely repeated games with perfect monitoring. *Games and Economic Behavior*, 6, 220–237.

Daskalakis, C., Goldberg, P. W., & Papadimitriou, C. H. (2006). The Complexity of Computing Nash Equilibrium. Proc. 38th Ann. ACM Symp. Theory of Computing (STOC), 71–78.

Gossner, O., & Hörner, J. (2010). When is the lowest equilibrium payoff in a repeated game equal to the minmax payoff? *Journal of Economic Theory*, 145 (1), 63–84.

Herings, P.J.-J., & Peeters, R. (2010). Homotopy methods to compute equilibria in game theory. *Economic Theory*, 42 (1), 119–156.

Hörner, J., Sugaya, T., Takahashi, S., & Vieille, N. (2011). Recursive Methods in Discounted Stochastic Games: An Algorithm for $\delta \mapsto 1$ and a Folk Theorem. *Econometrica*, 79 (4), 1277-1318.

Judd, K., Yeltekin, Ş., & Conklin, J. (2003). Computing supergame equilibria. *Econometrica*, 71, 1239–1254.

Judd, K., & Yeltekin, Ş. (2011). Computing Equilibria of Dynamic Games. Working paper.

Kalai, E., & Stanford, W. (1988). Finite Rationality and Interpersonal Complexity in Repeated Games. *Econometrica*, 56 (2), 397–410.

Littman, M. L., & Stone, P. (2005). A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems*, 39: 55–66.

Mailath, G. J., & Samuelson, L. (2006). *Repeated games and reputations: long-run relationships*. Oxford University Press.

Porter, R., & Nudelman, E., & Shoham, Y. (2008). Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63 (2), 642–662.

Sandholm T. (2007). Perspectives on multiagent learning. *Artificial Intelligence*, 171, 382–391.

Sandholm T. (2012). The state of solving large incomplete-information games, and application to poker. *AI Magazine*, 31 (4), 13–32.

Sandholm, T., & Gilpin, A., & Conitrzer, V. (2005). Mixed-integer programming methods for finding Nash equilibria. Proceedings of the 20th National Conference on Artificial Intelligence (AAAI). 495–501.

Salcedo, B., & Sultanum, B. (2010). Computation of subgame-perfect equilibria of repeated games with perfect monitoring and public randomization. Working paper.

Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press.