The Myth of the Folk Theorem

Christian Borgs[*] Jennifer Chayes[*] Nicole Immorlica[†] Adam Tauman Kalai[‡]
Vahab Mirrokni[*] Christos Papadimitriou[§]

November 19, 2007

Abstract

A well-known result in game theory known as "the Folk Theorem" suggests that finding Nash equilibria in repeated games should be easier than in one-shot games. In contrast, we show that the problem of finding any (approximate) Nash equilibrium for a three-player infinitely-repeated game is computationally intractable (even when all payoffs are in $\{-1, 0, -1\}$), unless all of PPAD can be solved in randomized polynomial time. This is done by showing that finding Nash equilibria of $(k + 1)$-player infinitely-repeated games is as hard as finding Nash equilibria of k-player one-shot games, for which PPAD-hardness is known (Daskalakis, Goldberg and Papadimitriou, 2006; Chen, Deng and Teng, 2006; Chen, Teng and Valiant, 2007). This also explains why no computationally-efficient learning dynamics, such as the "no regret" algorithms, can be *rational* (in general games with three or more players) in the sense that, when one's opponents use such a strategy, it is not in general a best reply to follow suit.

[*] Microsoft Research.

[†] Centrum voor Wiskunde en Informatica (CWI).

[‡] Microsoft Research and Georgia Institute of Technology. This research was supported in part by NSF SES-0734780.

[§] UC Berkeley. Research partly performed while visiting Microsoft Research, and supported by an NSF grant, a MICRO grant, and gifts from Microsoft Research and Yahoo! Research.

1 Introduction

Complexity theory provides compelling evidence for the difficulty of finding Nash Equilibria (NE) in one-shot games. It is NP-hard, for a two-player $n \times n$ game, to determine whether there exists a NE in which both players get non-negative payoffs [GZ89]. Recently it was shown that the problem of finding *any* NE is PPAD-hard [DGP06], even in the two-player $n \times n$ case [CD06], even for ϵ-equilibria for inverse polynomial ϵ [CDT06], and even when all payoffs are ± 1 [CTV07]. PPAD-hardness implies that a problem is at least as hard as discrete variations on finding Brouwer fixed-point, and thus presumably computationally intractable [P94].

Repeated games, ordinary games played by the same players a large — usually infinite — number of times, are believed to be a different story. Indeed, a cluster of results known as *the Folk Theorem*[1] (see, for example, [AS84, FM86, FLM94, R86]) predict that, in a repeated game with infinitely many repetitions and/or discounting of future payoffs, there are mixed NE (functions mapping histories of play by all players to a distribution over the next round strategies for each player) which achieve a rich set of payoff combinations called *the individually rational region* — essentially anything above what each player can absolutely guarantee for him/herself (see below for a more precise definition). In the case of prisoner's dilemma, for example, a NE leading to full collaboration (all players playing "mum" ad infinitum) is possible. In fact, repeated games and their Folk Theorem equilibria have been an arena of early interaction between Game Theory and the Theory of Computation, as play by resource-bounded automata was also considered [S80, R86, PY94, N85].

Now, there is one simple kind of mixed NE that is immediately inherited from the one-shot game: Just play a mixed NE each time. In view of what we now know about the complexity of computing a mixed NE, however, this is hardly attractive computationally. Fortunately, in repeated games the Folk Theorem seems to usher in a space of outcomes that is both much richer and computationally benign. In fact, it was recently pointed out that, using the Folk Theorem, a pure NE can indeed be found in polynomial time for any repeated game *with two players* [LS05].

The main result in this paper is that, for three or more players, finding a NE in a repeated game is PPAD-complete, under randomized reductions. This follows from a simple reduction from finding NE in k-player one-shot games to finding NE in $k+1$-player repeated games, for any k (the reverse reduction is immediate). In other words, for three or more players, playing the mixed NE each time is not as bad an idea in terms of computational complexity as it may seem at first. In fact, there is no general way that is computationally easier. Our results also hold for finding approximate NE, called ϵ-NE, for any inverse-polynomial ϵ and discounting parameter, and even in the case where the game has all payoffs in the set $\{-1, 0, 1\}$.

To understand our result and its implications, it is useful to explain the Folk Theorem. Looking at the one-shot game, there is a certain "bottom line" payoff that any player can guarantee for him/herself, namely the minmax payoff: The best payoff against a worst-case mixed strategy by everybody else. The vector of all minmax payoffs is called the *threat point* of the game, call it θ. Consider now the convex hull of all payoff combinations achievable by pure strategy plays (in other words, the convex hull of all the payoff data); obviously all mixed and pure NE are in this convex hull. The *individually rational region* consists of all points x in this convex hull such that such that $x \geq \theta$ coordinate-wise. It is clear that all Nash equilibria lie in this region. Now the Folk Theorem, in its simplest version, takes any payoff vector x in the individually rational region, and approximates it with a rational (no pun) point $\tilde{x} \geq x \geq \theta$ (such a rational payoff is guaranteed to exist if the payoff data are rational). The players then agree to play a periodic schedule of plays that achieve, in the limit, the payoff $\tilde{x}$ on the average. The agreement implicit in the NE further mandates that, if any player ever deviates from this schedule, everybody else will switch to the mixed strategy that achieves the player's minmax.

[1] Named this way because it was well-known to game theorists far before its first appearance in print.

It is not hard to verify that this is a mixed NE of the repeated game. Since every mixed NE can play the role of x, it appears that the Folk Theorem indeed creates a host of more general, and at first sight computationally attractive, equilibria.

To implement the Folk Theorem in a computationally feasible way, all one has to do is to compute the threat point and corresponding punishing strategies. The question thus arises: what is the complexity of computing the minmax payoff? For two players, it is easy to compute the minmax values (since in the two-player case this reduces to a two-player zero-sum game), and the Folk theorem can be converted to a computationally efficient strategy for playing a NE of any repeated game [LS05]. In contrast, we show that, *for three or more players, computing the threat point is NP-hard* in general (Theorem 1).

But a little reflection reveals that this complexity result is no real obstacle. Computing the threat point is not necessary for implementing the Folk Theorem. In fact our negative result is more general. Not only these two familiar approaches to NE in repeated games (playing each round the one-shot NE, and implementing the Folk Theorem) are both computationally difficult, but also *any algorithm for computing a mixed NE of a repeated game with three or more players can be used to compute a mixed NE of a two-person game*, and hence it cannot be done in polynomial time, unless there is a randomized polynomial-time algorithm for every problem in PPAD (Theorem 3). In other words, the Folk Theorem gives us hope that other points in the individually rational region will be easier to compute than the NE; well, they are not.

We feel that this result is conceptually important as it dispels a common belief in game theory, stemming from the folk theorem, that it is easy to play equilibria of repeated games. Our analysis has interesting negative game-theoretic implications regarding *learning dynamics.* An example is the elegant *no-regret* strategies, which have been shown to quickly converge to the set of correlated equilibria [FV97] of the *one-shot game*, even in games with many players (see [BM07] Chapter 4 for a survey). Our result implies that, for more than two players (under the same computational assumption), no computationally efficient general game-playing strategies are *rational* in the sense that if one's opponents all employ such strategies, it is not in general a best response *in the repeated game* to follow suit. Thus the strategic justification of no-regret algorithms is called into question.

Like all negative complexity results, those about computing NE have spawned a research effort focusing on approximation algorithms [CDT06, DMP07], as well as special cases [CTV07, DP07, DP07b]. As we have mentioned, our results already severely limit the possibility of approximating a NE in a repeated game; but the question remains, are there meaningful classes of games for which the threat point is easy to compute? In Section 4, we show that computing the threat point in congestion games (a much studied class of games of special interest in Algorithmic Game Theory) is NP-complete. This is true even in the case of network congestion games with linear latencies on a directed acyclic graph (DAG). (It is open whether the NE of repeated congestion games can be computed efficiently.) In contrast, applying the techniques of [DP07, DP07b] we show that the threat point can be approximated by a PTAS in the case of *anonymous games* with a fixed number of strategies — another broad class of great interest. This of course implies, via the Folk Theorem, that NE are easy to approximate in repeated anonymous games with few strategies; however, this comes as no surprise, since in such games even NE can be so approximated [DP07b]. In further contrast, for the even more restricted class of symmetric games (albeit with an unbounded number of strategies), we show that computing the threat point in repeated games are as hard as in the general case. It is an interesting open question whether there are natural classes of games (with three or more players) with an intractable NE problem, for which however the threat point is easy to compute or approximate; that is, classes of games for which the Folk Theorem is useful.

We give next some definitions and notation. In Section 2, we show that computing the threat point is NP-complete. In Section 3, we show that computing NE of a repeated game with

$k+1$-players is equivalent to computing NE of a k-player one-shot game, and extend this result to ϵ-NE. Finally, in Section 4, we show that our negative results hold even for symmetric games and congestion games.

1.1 Definitions and Notation

A **game** $G = (I, A, u)$ consists of a set $I = \{1, 2, \ldots, k\}$ of players, a set $A = \times_{i \in I} A_i$ of **action profiles** where A_i is the set of **pure actions**[2] for player i, and a **payoff function** $u : A \to \mathbb{R}^k$ that assigns a payoff to each player given an action for each player. We write $u_i : A \to \mathbb{R}$ for the payoff to player i, so $u(a) = \big(u_1(a), \ldots, u_k(a)\big)$. We use the standard notation $a_{-i} \in A_{-i} = \times_{j \in I \setminus \{i\}} A_j$ to denote the actions of all players except player i.

Let $\Delta_i = \Delta(A_i)$ denote the set of probability distributions over A_i and $\Delta = \times_{i \in I} \Delta_i$. A **mixed action** $\alpha_i \in \Delta_i$ for player i is a probability distribution over S_i. An k-tuple of mixed actions $\alpha = (\alpha_1, \ldots, \alpha_k)$ determines a product distribution over A where $\alpha(a) = \prod_{i \in I} \alpha_i(a_i)$. We extend the payoff functions to $\alpha \in \Delta$ by expectation: $u_i(\alpha) = \mathrm{E}_{a \sim \alpha}\left[u_i(a)\right]$, for each player i.

Definition 1 (Nash Equilibrium)**.** *Mixed action profile $\alpha \in \Delta$ is an ϵ-NE ($\epsilon \geq 0$) of G if,*

$$\forall\ i \in I\ \forall\ \bar{a}_i \in A_i \quad u_i(\alpha_{-i}, \bar{a}_i) \leq u_i(\alpha) + \epsilon.$$

A NE is an ϵ-NE for ϵ=0.

For any game $G = (I, A, u)$, we denote the **infinitely repeated game** by G^∞. In this context, G is called the **stage game**. In G^∞, each period $t = 0, 1, 2, \ldots$, each player chooses an action $a_i^t \in A_i$. A **history** $h^t = (a^0, a^1, \ldots, a^{t-1}) \in (A)^t$ is the choice of strategies in each of the first t periods, and $h^\infty = (a^0, a^1, \ldots)$ describes the infinite game play.

A **pure strategy** for player i in the repeated game is a function $s_i : A^* \to A_i$, where $A^* = \bigcup_{t=0}^{\infty}(A)^t$, and $s_i(h^t) \in A_i$ determines what player i will play after every possible history of length t. A mixed strategy for player i in the repeated game is a function $\sigma_i : A^* \to \Delta_i$, where $\sigma_i(h^t) \in \Delta_i$ similarly determines the probability distribution by which player i chooses its action after each history $h_t \in (A)^t$.

We use the standard discounting model to evaluate payoffs in such an infinitely repeated game. A mixed strategy profile $\sigma = (\sigma_1, \ldots, \sigma_k)$ induces a probability distribution over histories $h^t \in (A)^t$ in the natural way. The infinitely-repeated game with **discounting parameter** $\delta \in (0, 1)$ is denoted $G^\infty(\delta)$. The expected **discounted payoff** to player i of $\sigma = (\sigma_1, \ldots, \sigma_k)$ by,

$$p_i(\sigma) = \delta \sum_{t=0}^{\infty} (1-\delta)^t \mathrm{E}_{h^t}\left[\mathrm{E}_{a^t \sim \sigma(h^t)}\left[u_i(a^t)\right]\right],$$

where the expectation is over action profiles $a^t \in A$ drawn according to the independent mixed strategies of the players on the tth period based on $\sigma^0, \ldots, \sigma^t$. The δ multiplicative term ensures that, if the payoffs in G are bounded by some $M \in \mathbb{R}$, then the discounted payoffs will also be bounded by M. This follows directly from the fact that the discounted payoff is the weighted average of payoffs over the infinite horizon.

For $G = (I, A, u)$, $G^\infty(\delta) = (I, (A^*)^A, p)$ can be viewed as a game as above, where $(A^*)_i^A$ denotes the set of functions from A^* to A_i. In this spirit, an ϵ-NE of G^∞ is thus a vector of mixed strategies $\sigma = (\sigma_1, \ldots, \sigma_k)$ such that,

$$\forall\ i \in I\ \forall\ \bar{s}_i : A^* \to A_i \quad p_i(\sigma_{-i}, \bar{s}_i) \leq p_i(\sigma) + \epsilon.$$

This means that no player can increase its expected discounted payoff more than ϵ by unilaterally changing its mixed strategy (function). A NE of G^∞ is an ϵ-NE for $\epsilon = 0$.

[2] To avoid confusion, we use the word **action** in stage games (played once) and the word **strategy** for repeated games.

1.1.1 Computational Definitions

Placing game-playing problems in a computational framework is somewhat tricky, as a general game is most naturally represented with real-valued payoffs while most models of computing only allow finite precision. Fortunately, our results hold for a class of games where the payoffs all in $\{-1, 0, 1\}$, so we define our models in this case.

A **win-lose** game is a game in which the payoffs are in $\{-1, 1\}$, and we define a **win-lose-draw** game to be a game whose payoffs are in $\{-1, 0, 1\}$. We say a game is $n \times n$ if $A_1 = A_2 = [n]$ and similarly for $n \times n \times n$ games. We now state a recent result about computing Nash equilibria in two-player $n \times n$ win-lose games due to Chen, Teng, and Valiant. Such games are easy to represent in binary and their (approximate) equilibria can be represented by rational numbers.

Fact 1. *(From [CTV07]) For any constant $c > 0$, the problem of finding an n^{-c}-NE in a two-player $n \times n$ win-lose games is PPAD-complete.*

For sets $\mathcal{X}$ and $\mathcal{Y}$, a search problem $S : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ is the problem of, given $x \in \mathcal{X}$, finding any $y \in S(x)$. A search problem is *total* if $S(x) \neq \emptyset$ for all $x \in \mathcal{X}$. The class PPAD [P94] is a set of total search problems. We do not define that class here — a good definition may be found in [DGP06]. However, we do note that a (randomized) reduction from search problem $S_1 : \mathcal{X}_1 \to \mathcal{P}(\mathcal{Y}_1)$ to $S_2 : \mathcal{X}_2 \to \mathcal{P}(\mathcal{Y}_2)$ is a pair of (randomized) polynomial-time computable functions $f : \mathcal{X}_1 \to \mathcal{X}_2$ and $g : \mathcal{X}_1 \times \mathcal{Y}_2 \to \mathcal{Y}_1$ such that for any $x \in \mathcal{X}_1$ and any $y \in S_2(f(x))$, $g(x, y) \in S_1(x)$. To prove PPAD-hardness (under randomized reductions) for a search problem, it suffices to give a (randomized) reduction from that problem of finding an n^{-c}-NE for two-player $n \times n$ win-lose games.

We now define a **strategy machine** for playing repeated games. Following the game-theoretic definition, a strategy machine M_i for player i in G^∞ is a Turing machine that takes as input any history $h^t \in A^t$ (any $t \geq 0$), where actions are represented as binary integers, and outputs a probability distribution over A_i represented by a vector of fractions of binary integers that sum to 1.[3] A strategy machine is said to have *runtime* $R(t)$ if, for any $t \geq 0$ and history $h^t \in A^t$, its runtime is at most $R(t)$. With a slight abuse of notation, we also denote by $M_i : h^t \to \Delta_i$ the function computed by the machine. We are now ready to formally define the *repeated Nash Equilibrium Problem.*

Definition 2 (RNE)**.** *Let $\langle \epsilon, \delta, R \rangle_{n \geq 2}$ be a sequence of triplets, where $\epsilon_n > 0, \delta_n \in (0, 1)$, and $R_n : \mathbb{N} \to \mathbb{N}$. The $(\epsilon_n, \delta_n, R_n)$-RNE problem is the following: given a win-lose-draw game G of size $n \geq 2$), output three machines, each running in time $R_n(t)$, such that the strategies computed by these three machines are an ϵ_n-NE to $G^\infty(\delta_n)$.*

2 The Complexity of the Threat Point

The minmax value for player i of game G is defined to be,

$$\theta_i(G) = \min_{\alpha_{-i} \in \Delta_{-i}} \max_{\alpha_i \in \Delta_i} u_i(\alpha_i, \alpha_{-i}).$$

The threat point $(\theta_1, \ldots, \theta_k)$ is key to the definition of the standard folk theorem, as it represents the worst punishment that can be inflicted on each player if the player deviates from some coordinated behavior plan.

[3] We note that several alternative formulations are possible for the definition of strategy machines. For simplicity, we have chosen deterministic Turing machines. (The limitation to rational-number output is not crucial because our results apply to ϵ-NE as well.) A natural alternative formulation would be randomized machines that output pure actions. Our results could be extended to such a setting in a straightforward manner using sampling.

Theorem 1. *Given a three-player $n \times n \times n$ game with payoffs $\in \{0, 1\}$, it is NP-hard to approximate the minmax value for each of the players to within $3/n^2$.*

The above theorem also implies that it is hard for the players to find mixed actions that achieve the threat point within $< 3/n^2$. For, suppose that two players could find strategies to force the third down to its threat value. Then they could approximate the threat value easily by finding an approximate best response for the punished player and estimating its expected payoff, by sampling.

Proof. The proof is by a reduction from the NP-hard problem of 3-colorability. For notational ease, we will show that it is hard to distinguish between a minmax value of $\frac{1}{n}$ and $\frac{1}{n} + \frac{1}{3n^2}$ in $3n \times 3n \times 2n$ games. More formally, given an undirected graph (V, E) with $|V| = n \geq 4$, we form a 3-player game in which, if the graph is 3-colorable, the minmax value for the third player is $1/n$, while if the graph is not 3-colorable, the minmax value is $\geq 1/n + 1/(3n^2)$.

P1 and P2 each choose a node in V and a color for that node (ideally consistent with some valid 3-coloring). P3 tries to guess which node one of the players has chosen by picking a player (1 or 2) and a node in V. Formally, $A_1 = A_2 = V \times \{\text{r}, \text{g}, \text{b}\}$ and $A_3 = V \times \{0, 1\}$.

The payoff to P3 is 1 if either (a) P1 and P2 are exposed for not choosing a valid coloring or (b) P3 correctly guesses the node of the chosen player. Formally,

$$u_3((v_1, c_1), (v_2, c_2), (v_3, i)) = \begin{cases} 1 & \text{if } v_1 = v_2 \text{ and } c_1 \neq c_2 \\ 1 & \text{if } (v_1, v_2) \in E \text{ and } c_1 = c_2 \\ 1 & \text{if } v_i = v_3 \\ 0 & \text{otherwise} \end{cases}$$

In the case of either of the first two events above, we say that P1 and P2 are exposed. The payoffs to P1 and P2 are irrelevant for the purposes of the proof. Notice that if the graph is 3-colorable, then the threat point for player 3 is $1/n$. To achieve this, let $c : V \to \{\text{r}, \text{g}, \text{b}\}$ be a coloring. Then P1 and P2 can choose the same mixed strategy which picks $(v, c(v))$ for v uniformly random among the n nodes. They will never be exposed for choosing an invalid coloring, and P3 will guess which node they choose with probability $1/n$, hence P3 will achieve expected payoff exactly $1/n$. They cannot force player 3 to achieve less because there will always be some node that one player chooses with probability at least $1/n$.

It remains to show that if the graph is not 3-colorable, then for any $(\alpha_1, \alpha_2) \in \Delta_1 \times \Delta_2$, there is a (possibly mixed) action for P3 that achieves expected payoff at least $1/n + 1/(3n^2)$.

Case 1: there exists $i \in \{1, 2\}$ and $v \in V$ such that player i has probability at least $1/n + 1/(3n^2)$ of choosing v. Then we are done because P3 can simply choose (v, i) as his action.

Case 2: each player $i \in \{1, 2\}$ has probability at most $1/n + 1/(3n^2)$ of choosing any $v \in V$. We will have P3 pick action $(v, 2)$ for a uniformly random node $v \in V$. Hence, P3 will succeed with its guess with probability $1/n$, regardless of what P1 and P2 do, and independent of whether or not the two players are exposed.

It remains to show that this mixed action for P3 achieves payoff at least $1/n + 1/(3n^2)$ against any α_1, α_2 that assign probability at most $1/n + 1/(3n^2)$ to every node. To see this, a simple calculation shows that if α_i assigns probability at most $1/n + 1/(3n^2)$ to every node, this means that α_i also assign probability at least $2/(3n)$ to every node. Hence, the probability of the first two players being exposed is,

$$\begin{aligned} \Pr[\text{being exposed}] &\geq \sum_{v_1, v_2 \in V} \Pr[\text{P1 chooses } v_1] \Pr[\text{P2 chooses } v_2] \Pr[\text{being exposed} | v_1, v_2] \\ &\geq \frac{4}{9n^2} \sum_{v_1, v_2 \in V} \Pr[\text{being exposed} | v_1, v_2] \geq \frac{4}{9n^2}. \end{aligned}$$

Given: k-player game $G = (I = \{1, 2, \dots, k\}, A, u)$
$\widehat{G} = (\widehat{I} = \{1, 2, \dots, k+1\}, \widehat{A}, \widehat{u})$ is the player-kibitzer version of G:

$$r = k + 1 \quad \text{(notational convenience)}$$

$$\widehat{A}_i = \begin{cases} A_i & \text{if } i \neq r \\ \{(j, \bar{a}_j) | 1 \leq j \leq k, \bar{a}_j \in A_j\} & \text{if } i = r \end{cases}$$

$$\widehat{u}_i(a, (j, \bar{a}_j)) = \begin{cases} 0 & \text{if } i \notin \{j, r\} \\ u_j(a) - u_j(\bar{a}_j, a_{-j}) & \text{if } i = j \\ u_j(\bar{a}_j, a_{-j}) - u_j(a) & \text{if } i = r \end{cases}$$

Figure 1: The player-kibitzer version of k-player game G. Players $i = 1, 2, \dots, k$, are the **players**, player $r = k + 1$ is the **kibitzer**. At most one player may receive a nonzero payoff. The kibitzer singles out player j and suggests alternative action $\bar{a}_j$. The kibitzer and player j exchange the difference between how much j would have received had all the players played their chosen actions in G and how much they would have received had j played $\bar{a}_j$ instead. All other payoffs are 0.

The last step follows from the probabilistic method. To see this, note that the sum in the equality is the expected number of inconsistencies over all n^2 pairs of nodes, if one were to take two random colorings based on the two distributions of colors. If the expectation were less than 1, it would mean that there was some consistent coloring, which we know is impossible. Finally, the probability of P3 achieving a payoff of 1 is $\geq 1/n + (1 - 1/n)4/(9n^2)$, which is $\geq 1/n + 1/(3n^2)$ for $n \geq 4$. □

3 The Complexity of Playing Repeated Games

Take a k-player game $G = (I = \{1, 2, \dots, k\}, A, u)$. We will construct an *player-kibitzer* version of G, a simple $(k+1)$-player game $\widehat{G}$ such that, in the NE of the infinitely repeated $\widehat{G}^\infty$, the first k players must be playing a NE of G. The construction is given in Figure 3. A few observations about this construction are worth making now.

- It is not difficult to see that a best response by the kibitzer in $\widehat{G}$ gives the kibitzer payoff 0 if and only if the players mixed actions are a NE of G. Similarly, a best response gives the kibitzer ϵ if and only if the mixed actions of the players are an ϵ-NE of G but not an ϵ'-NE of G for all $\epsilon' < \epsilon$. Hence, the intuition is that in order to maximally punish the kibitzer, the players must play a NE of G.
- While we show that such games are difficult to "solve," the threat point and individually rational region of any player-kibitzer game are trivial. They are the origin and the singleton set containing the origin, respectively.[4]
- If G is an $n \times n$ game, then $\hat{G}$ is a $n \times n \times 2n$ game.
- If the payoffs in G are in $\{0, 1\}$, then the payoffs in $\widehat{G}$ are in $\{-1, 0, 1\}$. If the payoffs in G are in $[-B, B]$, then the payoffs in $\widehat{G}$ are in $[-2B, 2B]$.

[4] Considering that Folk theorems are sometimes stated in terms of the set of *strictly* individually rational payoffs (those which are strictly larger than the minmax counterparts), our example may seem less convincing because this set is empty. However, one can easily extend our example to make this set nonempty. By doubling the size of each player's action set, one can give each player i the option to reduce all of its opponents payoffs by $\rho > 0$, at no cost to player i, making the minmax value $-\rho k$ for each player. For $\rho < \epsilon/(2k)$, our analysis remains qualitatively unchanged.

Theorem 2. *For any k-player game G, let $\widehat{G}$ be the player-kibitzer version of G as defined in Figure 3. (a) At any NE of the infinitely repeated $\widehat{G}^\infty$, the mixed strategies played by the players at each period t, are a NE of G with probability 1. (b) For any $\epsilon > 0, \delta \in (0,1)$, at any ϵ-NE of $\widehat{G}^\infty(\delta)$, the mixed strategies played by the players in the first period are a $\left(\frac{k+1}{\delta}\epsilon\right)$-NE of G.*

Proof. We first observe that any player in $\widehat{G}$, fixing its opponents' actions, can guarantee itself expected payoff ≥ 0. Any player can do this simply by playing an action that is a best response, *in* G, to the other players' actions, as if they were actually playing G. In this case, the kibitzer cannot achieve expected positive payment from this player. On the other hand, the kibitzer can guarantee 0 expected payoff by mimicking, say, player 1 and choosing $\widehat{\alpha}_r(1, a_1) = \widehat{\alpha}_1(a_1)$ for all $a_1 \in A_1$.

Since each player can guarantee itself expected payoff ≥ 0 in $\widehat{G}$, and $\widehat{G}$ is a zero-sum game, then the payoffs at any NE of $\widehat{G}^\infty$ must be 0 for all players. Otherwise, there would be some player with negative expected discounted payoff, and that player could improve by guaranteeing itself 0 in each stage game.

Now, suppose that part (a) of the theorem was false. Let t be the first period in which the mixed actions of the players may not be a NE of G, with positive probability. The kibitzer may achieve a positive expected discounted payoff by changing its strategy as follows. On period t, the kibitzer plays a best response $(j, \bar{a}_j)$ where j is the player that can maximally improve its expected payoff on period t and $\bar{a}_j$ is player j's best response during that period. After period t, the kibitzer could simply mimic player 1's mixed actions, and achieve expected payoff 0. This would give the kibitzer a positive expected discounted payoff, which contradicts the fact that they were playing a NE of $\widehat{G}^\infty$.

For part (b), note that at any ϵ-NE of $\widehat{G}^\infty$, the kibitzers (expected) discounted payoff cannot be greater than $k\epsilon$, or else there would be some player whose discounted payoff would be $< -\epsilon$, contradicting the definition of ϵ-NE. Therefore, any change in kibitzer's strategy can give the kibitzer at most $(k+1)\epsilon$. Now, suppose on the first period, the kibitzer played the best response to the players' first-period actions, $\widehat{a}_r^0 = b(\widehat{\alpha}_{-r}^0)$, and on each subsequent period guaranteed expected payoff 0 by mimicking player 1's mixed action. Then this must give the kibitzer discounted payoff $\leq (k+1)\epsilon$, implying that the kibitzer's expected payoff on period 0 is at most $(k+1)\epsilon/\delta$, and that the player's mixed actions on period 0 are a $(k+1)\epsilon/\delta$-NE of G. □

The above theorem implies that given an algorithm for computing ϵ-NE for $(k+1)$-player repeated games, one immediately gets an algorithm for computing $(\frac{k+1}{\delta}\epsilon)$-NE for one-shot k-player games. Combined with the important point that the payoffs in $\widehat{G}$ are bounded by a factor of 2 with respect to the payoffs in G, this is already a meaningful reduction, but only when ϵ and δ are small. This is improved by our next theorem.

Lemma 1. *Let $k \geq 1, \epsilon > 0, \delta \in (0,1)$, $T = \lceil 1/\delta \rceil$, G be a k-player game, strategy machine profile $M = (M_1, \ldots, M_{k+1})$ be an $\frac{\epsilon}{8k}$-NE of $\widehat{G}^\infty(\delta)$, and $R \geq 1$ such that the runtime of M_i on any history of length $\leq T = \lceil 1/\delta \rceil$ is at most R. Then the algorithm of Figure 3 outputs an ϵ-NE of G in expected runtime that is poly$(1/\delta, \log(1/\epsilon), R, |G|)$.*

Proof. Let $b_r : \Delta_{-r} \to A_r$ be any best response function for the kibitzer in the stage game G. On each period t, if the kibitzer plays $b_r(\hat{\sigma}_{-r}(h^t))$, then let its expected payoff be denoted by z^t, where, $\rho^t = u_r\left(M_{-r}(h^t), b_r(M_{-r}(h^t))\right)$ and $z^t = \mathrm{E}_{h^t}[\rho^t] \geq 0$. Note that ρ^t is a random variable that depends on h^t. As observed before, $M_{-r}(h^t)$ is a ρ^t-NE of G. Note that we can easily verify if a mixed action profile is an ϵ-equilibrium in poly$(|G|)$, i.e., time polynomial in the size of the game. Hence, it suffices to show that the algorithm encounters M_{-r} which is an ϵ-NE of G in expected polynomial time. We next argue that any execution of Step 2 of the algorithm succeeds with probability $\geq 1/2$. This means that the expected number of executions of Step

```
Given: $k$-player $G$, $\epsilon > 0, T \geq 1$, strategy machines $M = (M_1, \ldots, M_{k+1})$ for $\widehat{G}^\infty$.

  1. Let $h^0 := ()$, $r = k + 1$.
  2. For $t = 0, 1, \ldots, T$:
       • If $\sigma = M_{-r}(h^t)$ is an $\epsilon$-NE of $G$, then stop and output $\sigma$.
       • Let $a^t_{k+1}$ be best response to $M_{-r}(h^t)$ in $\widehat{G}$ (break ties lexicographically).
       • Choose actions $a^t_{-(k+1)}$ independently according to $M_{-r}(h^t)$, respectively.
       • Let $h^{t+1} := (h^t, a^t)$.
```

Figure 2: Algorithm for extracting an approximate NE of G from an approximate NE of $\widehat{G}^\infty$.

2 is at most 2. Since each such execution is polynomial time, this will suffice to complete the proof.

Imagine that the algorithm were run for $t = 0, 1, 2, \ldots$ rather than stopping at $T = \lceil 1/\delta \rceil$. Also as observed before, the kibitzer's expected payoff is $< (k+1)\epsilon/(8k) \leq \epsilon/4$, or else there would be some player that could improve over M by more than $\epsilon/(8k)$. But the kibitzer's expected payoff, $\delta \sum_0^\infty (1-\delta)^t z^t$, is an average of z^t, weighted according to an exponential distribution with parameter $1-\delta$. This average, in turn, can be decomposed into $\lambda = (1-\delta)^{T+1}$ times the (weighted) average of z^t on periods $T+1, T+2, \ldots$ plus $1-\lambda$ times the (weighted) average of z^t on periods $1, 2, \ldots, T$. Since $z^t \geq 0$, the weighted average payoff of the kibitzer on the first T periods is at most $\epsilon/(4(1-\lambda))$. By Markov's inequality, this weighted average is at most $\epsilon/(2(1-\lambda)) \leq \epsilon$ (using $\lambda \leq (1-1/T)^T \leq 1/e$) with probability at least 1/2. This completes the proof. □

Theorem 3. *Let $c_1, c_2, c_3 > 0$ be any positive constants. The problem $\langle \epsilon, \delta, R \rangle_n$-RNE, for any $\epsilon_n = n^{-c_1}$, $\delta_n \geq n^{-c_2}$ and $R(t) = (nt)^{c_3}$ is PPAD-complete under randomized reductions.*

Proof. Let $c_1, c_2, c_3 > 0$ be arbitrary constants. Suppose that one had a randomized algorithm $\widehat{A}$ for the $\langle \epsilon, \delta, T \rangle_n$-RNE problem, for $\epsilon_n = n^{-c_1}$, $\delta_n \geq n^{-c_2}$ and $R(t) = (nt)^{c_3}$. Then we will show that there is a randomized polynomial-time algorithm for finding a n^{-c}-NE in two-player $n \times n$ win-lose games, for $c = c_1/2$, establishing Theorem 3 by way of the Fact 1.

In particular, suppose we are given an n-sized game G. If $n \leq 16^{2/c_1}$ is bounded by a constant, then we can brute-force search for an approximate equilibrium in constant time, since we have a constant bound on the magnitude of the denominators of the rational-number probabilities of some NE. Otherwise, we have $n^{-c} \geq 16n^{-c_1}$, so it suffices to find an $16n^{-c_1}$-NE of G by a randomized polynomial-time algorithm.

We run $\widehat{A}$ on $\widehat{G}$ ($\widehat{G}$ can easily be constructed in time polynomial in n). With constant probability, the algorithm is successful and outputs strategy machines $S_1, \ldots, S_{k+1}$ such that the strategies they compute are an n^{-c_1}-NE of $\widehat{G}^\infty(\delta)$. By Lemma 1, the extraction algorithm run on this input will give a $8kn^{-c_1} = 16n^{-c_1}$-NE of G. □

4 Special Cases of Interest

We already know that our negative complexity results (about computing the threat point and playing repeated games) hold even for the special case of win-lose-draw games. But what about the many other important restricted classes of games treated in the literature?

First, we consider *congestion games*, a class of games of central interest in algorithmic game theory, and show that computing the threat point of congestion games with many players is hard to compute. Briefly, in a congestion game the actions of each player are source-sink paths (sets of edges), and, once a set of paths are chosen, the nonpositive payoff of each player is the negation of the sum over all edges in the chosen path of the *delays* of those edges, where each edge has a delay function mapping its congestion (the number of chosen paths that go through it) to the nonnegative integers. Of special interest are congestion games in which the paths are not abstract sets of objects called "edges," but are instead actual paths from source to sink (where each player may have his/her own source and sink) in a particular graph (ideally, a DAG). These are called *network congestion games.*

Theorem 4. *Computing the threat point in a network congestion game on a DAG is NP-complete.*

Proof. We give a reduction from SAT. In fact, we shall start from a stylized NP-complete special case of SAT in which all clauses have either two literals ("short" clauses) or three literals ("long" clauses), and each variable appears three times, once positively in a short clause, once negatively in a short clause, and once in a long clause (positively or negatively). It is straightforward to show that such a problem is NP-complete, as any 3-SAT can be converted into such a form by replacing a variable x_i that occurs k times with new variables $x_{i1}, \ldots, x_{ik}$ and a cycle of clauses, $(x_{i1}, \bar{x}_{i2}), (x_{i2}, \bar{x}_{i3}), \ldots, (x_{ik}, \bar{x}_{i1})$ forcing them to be equal.

Given such an instance, we construct a network congestion game as follows. We consider two types of delay functions. The first is a *zero*-delay which is always 0. The second is a *step*-delay which is 0 if the congestion on the edge is ≤ 1 and is 1 if the congestion is ≥ 2. Say there are n variables, $x_1, \ldots, x_n$, in the formula. Then there are $n+1$ players, one for each variable, and another player ("the victim") whose threat level we are to compute. The players sources' are $s_1, \ldots, s_n$ and s (one for each variable x_i and one for the victim) and sinks $t_1, \ldots, t_n, t$, respectively. For each short clause c, there are two nodes a_c and b_c, with a step-delay edge from a_c to b_c. Each a_c has two incoming zero-delay edges, one from each source s_i where x_i or $\bar{x}_i$ belongs to c. Similarly, each b_c has two outgoing zero-delay edges (plus possible additional outgoing edges, discussed next), one to each t_i such that x_i or $\bar{x}_i$ belongs to c. For each long clause c, there are nodes u_c and v_c with a step-delay edge from u_c to v_c. Each u_c as three incoming edges, one from each $b_{c'}$ where short clause c' has a variable in common with c (with the same positivity/negation). Each v_c has three outgoing zero-delay edges, one to each t_i such that x_i or $\hat{x}_i$ belongs to c.

Finally, we add zero-delay edges from s to each a_c and each u_c, and zero-delay edges from each b_c and v_c to t. Now, each player $1, \ldots, n$, can saturate either one of the two edges corresponding to the positive/negative appearances it has in short clauses and possibly also the long clause to which the variable belongs. However, it saturates edges corresponding to both short and long clause, they must have the same positive/negative occurrence of the variable.

We claim that the minmax for the victim is -1 if the clauses are satisfiable, and at least $\frac{1-3n}{3n}$ otherwise. For the *if* part, suppose that there is a satisfying truth assignment, and each variable-player chooses the path that goes through the short clause corresponding to the value of its variable and then through the long clause if possible. Then it is easy to see that all paths available to the victim are blocked by at least one flow, and thus any strategy chosen will result in utility -1 or less. Conversely, if the formula is unsatisfiable, by choosing a path from s through a random edge (a_c, b_c) to t, uniformly at random over c, the victim will guarantee a utility of at least $\frac{1-3n}{3n}$, since at least one path must now be left unblocked. It is easy to see that, because of the step nature of the delay functions, this is guaranteed even if the opponents randomize. □

In many settings, repeated games and the Folk Theorem come up in the literature in the context of *symmetric games* such as the prisoner's dilemma. A game is symmetric if it is invariant under any permutation of the players; that is, the action sets are the same, and, for any permutation π of the set of players $[n]$, that the utility of player i when player j plays a_j for $j = 1, \ldots, n$ is the same as the utility of player $\pi(i)$ when player $\pi(j)$ plays a_j for $j = 1, \ldots, n$. Do the negative results in the previous two sections hold in the case of symmetric games? The next theorem partially answers this question.

Theorem 5. *Computing the threat point in symmetric games with three players is NP-hard.*

A proof sketch of the above theorem appears in the Appendix. We conjecture that it is PPAD-complete to play repeated symmetric games with three or more players, but the specialization of this result to the symmetric case is much more challenging. The obvious problem is the inherent asymmetry of the player-kibitzer game, but the real problem is the resistance of symmetry to many sophisticated attempts to circumvent it in repeated play.

Anonymous games [DP07, DP07b] comprise a much larger class of games than the symmetric ones. In such games players have the same action sets but different utilities; however, these utilities do not depend on the identity of the other players, but only on the *number* of other players taking each action. Naturally, computing threat points and playing repeated games are intractable problems for this class. However, when we further restrict the games to have an *action set of fixed size*, then the techniques of [DP07, DP07b] can be used to establish the following:

Theorem 6. *There is a PTAS for approximating the threat point in anonymous games with a fixed number of actions.*

A proof is given in an Appendix. Hence, the Folk Theorem applies to repeated anonymous games with few strategies, and therefore such games can be approximately played efficiently. However this follows from the result in [DP07b] that for these games an approximate NE of the one-shot game is also easy to compute.

5 Conclusions

We have shown that a k-player one shot game can easily be converted to a $(k+1)$-player repeated game, where the only NE of the repeated game are NE of the one-shot game. Since a one-shot game can be viewed as a repeated game with discounting parameter $\delta = 1$, our reduction generalizes recent PPAD-hardness results regarding NE for $\delta = 1$ to all $\delta \in (0, 1]$, showing that repeated games are not easy to play — the Folk Theorem notwithstanding. Note that our theorems are essentially independent of game representation. They just require the player-kibitzer version of a game to be easily representable. Moreover, our simple reduction should easily incorporate any new results about the complexity of one-shot games that may emerge.

References

[AS84] Aumann, R. and L. Shapley "Long-term competition: a game-theoretic analysis," mimeo, Hebrew University, 1976.

[BM07] Blum, A. and Y. Mansour (2007) Learning, Regret Minimization, and Equilibria. In *Algorithmic Game Theory* (eds. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani), Cambridge University Press.

[CD06] Chen, X., and X. Deng (2006) Settling the Complexity of 2-Player Nash-Equilibrium. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science* (FOCS), 261–272.

[CDT06] Chen, X., Deng, X., and Teng, S. (2006) Computing Nash Equilibria: Approximation and Smoothed Complexity. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science* (FOCS), 603–612.

[CTV07] Chen, X., S. Teng and P. Valiant (2007) The approximation complexity of win-lose games. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithm.*

[DGP06] Daskalakis, C., P. Goldberg, C. Papadimitriou (2006) The complexity of computing a Nash equilibrium. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing* (STOC), 71–78.

[DMP07] C. Daskalakis, A. Mehta, and C. H. Papadimitriou "Progress in approximate Nash equilibria," EC 2007.

[DP07] C. Daskalakis and C. H. Papadimitriou "Computing equilibria in anonymous games," 2007 FOCS.

[DP07b] C. Daskalakis and C. H. Papadimitriou "On the exhaustive algorithm for Nash equilibria," submitted to this conference.

[FM86] Fudenberg, D. and E. Maskin"The Folk Theorem in repeated games with discounting or with incomplete information," *Econometrica,* 1986.

[FLM94] Fudenberg, D., D. Levine and E. Maskin "The Folk Theorem with imperfect public information," *Econometrica* 1994.

[FV97] Foster, D. and R. Vohra (1997) Regret in the on-line decision problem. *Games and Economic Behavior*, 21:40-55.

[GZ89] Gilboa, I. and E. Zemel (1989) Nash and correlated equilibria: Some complexity considerations, *Games and Economic Behavior* 1:80–93.

[LMM03] Lipton, R., E. Markakis and A. Mehta (2003) Playing large games using simple strategies. In *Proceedings of the 4th ACM conference on Electronic commerce* (EC), 36–41.

[LS05] Littman, M., and P. Stone (2005) A polynomial-time Nash equilibrium algorithm for repeated games. *Decision Support Systems* 39(1): 55-66.

[N51] Nash, J. (1951) Noncooperative games. *Ann Math*, 54:286–295.

[N85] Neyman, A. (1985) Bounded Complexity Justifies Cooperation in the Finitely Repeated Prisoner's Dilemma. *Eco- nomic Letters*, 19: 227-229.

[P94] Papadimitriou, C. (1994) On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 498–532.

[PY94] Papadimitriou, C. and M. Yannakakis (1994) On complexity as bounded rationality. In *Proceedings of the Twenty- Sixth Annual ACM Symposium on Theory of Computing*, 726–733.

[R86] Rubenstein, A. (1986) Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, 39:83–96.

[S80] Smale, S. (1980) The prisoner's dilemma and dynamical systems associated to noncooperative games. *Econometrica*, 48:1617-1634.

6 Appendix

Proof of Theorem 5

Proof. Sketch. To make the construction in the NP-completeness proof of Theorem 1 symmetric, we assume that all three players have access to all actions (both actions of the type (v, i) where v is a vertex and $i \in \{1, 2\}$ identifies one of the other two players, namely player number $p+i \bmod 3$, where p is the player being considered) and actions of type (v, c) where v is a vertex and c is a color. If two of them choose an action of the type (v, c) and the third an action of the type (v, i), then the payoffs are as in that proof. If all three choose an action of the type (v, c) then they all lose 1, and if fewer than two do, then they all win 1. To achieve the maxmin, say of player P3, P1 and P2 should only play with nonzero probabilities actions of the form (v, c) — because the other actions are dominated by them — and hence P3 should only play with nonzero probability strategies of the form (v, i). Now the original argument applies to establish NP-completeness. □

Proof of Theorem 6 *With thanks to Costas Daskalakis for this much simplified proof.*

Proof. Let us reconsider the min-max problem faced by player i in an anonymous game of k players and some constant number of a of actions per player: $\min_{\alpha_{-i}\in\Delta_{-i}} \max_{s_i\in[a]} u_i(s_i, \alpha_{-i})$. Observe that, since the game is anonymous, the function $u_i(s_i, \cdot) : \times_{j\neq i} A_j \to \mathbb{R}$ is a symmetric function of its arguments. Hence, we can rewrite the above min-max problem as follows

$$\min_{\alpha_{-i}\in\Delta_{-i}} \max_{s_i\in[t]} E_{\mathcal{X}_j\sim\alpha_j, j\neq i}\left[u_i\left(s_i, \sum_{j\neq i}\mathcal{X}_j\right)\right],$$

where by $\mathcal{X}_j \sim \alpha_j$ we denote a random a-dimensional vector such that $Pr[\mathcal{X}_j = e_\ell] = \alpha_j(\ell)$, for all $\ell = 1, \ldots, a$. It is not hard to see that, if the maximum payoff of every player in the game is absolutely bounded by some constant U, then, for any sets of mixed strategies $\{\alpha_j\}_{j\neq i}$ and $\{\alpha'_j\}_{j\neq i}$,

$$\left|E_{\mathcal{X}_j\sim\alpha_j, j\neq i}\left[u_i\left(s_i, \sum_{j\neq i}\mathcal{X}_j\right)\right] - E_{\mathcal{Y}_j\sim\alpha'_j, j\neq i}\left[u_i\left(s_i, \sum_{j\neq i}\mathcal{Y}_j\right)\right]\right| \leq U \left|\left|\sum_{j\neq i}\mathcal{X}_j - \sum_{j\neq i}\mathcal{Y}_j\right|\right|_{TV},$$

where the right hand side represents the total variation distance between $\sum_{j\neq i}\mathcal{X}_j$ and $\sum_{j\neq i}\mathcal{Y}_j$, where, for every j, $\mathcal{X}_j$ is distributed according to α_j and $\mathcal{Y}_j$ according to α'_j. Suppose now that $\{\alpha^*_j\}_{j\neq i}$ is the set of mixed strategies for which the minimum value of the min-max problem is achieved. It follows from the above that if we perturb the α^*_j's to another set of mixed strategies $\{\alpha'^*_j\}_{j\neq i}$ the value of the minmax problem is only affected by an additive term $U \left|\left|\sum_{j\neq i}\mathcal{X}_j - \sum_{j\neq i}\mathcal{Y}_j\right|\right|_{TV}$, where $\mathcal{X}_j \sim \alpha^*_j$ and $\mathcal{Y}_j \sim \alpha'^*_j$, for all $j \neq i$. In [DP07b], it was shown that there exists some function $f(\epsilon) \approx \epsilon^{1/5}$ such that for any set of distributions $\{\alpha^*_j\}_{j\neq i}$ there exists another set of ϵ-"discretized" distributions $\{\alpha'^*_j\}_{j\neq i}$, that is distributions for which $\alpha'^*_j(\ell)$ is an integer multiple of ϵ for all $\ell \in [a]$, such that

$$\left|\left|\sum_{j\neq i}\mathcal{X}_j - \sum_{j\neq i}\mathcal{Y}_j\right|\right|_{TV} \leq a\ 2^a f(\epsilon).$$

Hence, we can restrict the minimization to ϵ-discretized probability distributions with an additive loss of $\approx Ua\ 2^a\epsilon^{1/5}$ in the value of the minmax problem. Even so, the search space is of size $\Omega\left(\left(\frac{1}{\epsilon^t}\right)^{k-1}\right)$ which is exponential in the input size $O(k^a)$. By exploiting the fact that the functions $\{u_i(s_i, \cdot)\}_{s_i \in [a]}$ are symmetric functions of their arguments we can prune the search space to $O((k-1)^{1/\epsilon^a})$ therefore achieving a PTAS. □