The Complexity of Approximating a Trembling Hand Perfect Equilibrium of a Multi-player Game in Strategic Form

Kousha Etessami[1], Kristoffer Arnsfelt Hansen[2], Peter Bro Miltersen[2], and Troels Bjerre Sørensen[3]

[1] University of Edinburgh
kousha@inf.ed.ac.uk
[2] Aarhus University
{arnsfelt,bromille}@cs.au.dk
[3] IT-University of Copenhagen
trbj@itu.dk

Abstract. We consider the task of computing an approximation of a trembling hand perfect equilibrium for an n-player game in strategic form, $n \geq 3$. We show that this task is complete for the complexity class FIXP_a. In particular, the task is polynomial time equivalent to the task of computing an approximation of a Nash equilibrium in strategic form games with three (or more) players.

1 Introduction

Arguably [17], the most important *refinement* of Nash equilibrium for finite games in strategic form (a.k.a. games in normal form, i.e., games given by their tables of payoffs) is Reinhard Selten's [15] notion of *trembling hand perfection*. The set of trembling hand perfect equilibria of a game is a non-empty subset of the Nash equilibria of that game. Also, many "unreasonable" Nash equilibria of many games, e.g., those relying on "empty threats" in equivalent extensive forms of those games, are not trembling hand perfect, thus motivating and justifying the notion. The importance of the notion is illustrated by the fact that Selten received the Nobel prize in economics together with Nash (and Harsanyi), "for their pioneering analysis of equilibria in the theory of non-cooperative games". In this paper, we study the computational complexity of finding trembling hand perfect equilibria of games given in strategic form.

The computational complexity of finding a *Nash* equilibrium of a game in strategic form is well-studied. When studying this computational task, we assume that the game given as input is represented as a table of integer (or rational) payoffs, with each payoff given in binary notation. The output is a strategy profile, i.e., a family of probability distributions over the strategies of each player, with each probability being a rational number with numerator and denominator given in binary notation. The computational task is therefore discrete and we are interested in the Turing machine complexity of solving it. Papadimitriou [13]

R. Lavi (Ed.): SAGT 2014, LNCS 8768, pp. 231–243, 2014.

showed that for the case of two players, the problem of computing an *exact* Nash equilibrium is in PPAD, a natural complexity class introduced in that paper, as a consequence of the Lemke-Howson algorithm [10] for solving this task. For the case of three or more players, there are games where no Nash equilibrium which uses only rational probabilities exists [12], and hence considering some relaxation of the notion of "computing" a Nash equilibrium is necessary to stay within the discrete input/output framework outlined above. In particular, Papadimitriou showed that the problem of computing an ϵ-Nash equilibrium, with $\epsilon > 0$ given as part of the input in binary notation, is also in PPAD, as a consequence of Scarf's algorithm [14] for solving this task. Here, an ϵ-Nash equilibrium is a strategy profile where no player can increase its utility by more than ϵ by deviating. In breakthrough papers, Daskalakis *et al.* [5] and Chen and Deng [4] showed that both tasks are also hard for PPAD, hence settling their complexity: Both are PPAD-complete. Subsequently, Etessami and Yannakakis [6] pointed out that for some games, ϵ-Nash equilibria can be so remote from any exact Nash equilibrium (unless ϵ is so small that its binary notation has encoding size exponential in the size of the game), that the former tells us little or nothing about the latter. For such games, the ϵ-Nash relaxation is a bad proxy for Nash equilibrium, assuming the latter is what we are actually interested in computing. Motivated by this, they suggested a different relaxation: Compute a strategy profile with ℓ_∞-distance at most δ from an exact Nash equilibrium, with $\delta > 0$ again given as part of the input in binary notation. In other words, *compute an actual Nash equilibrium to a desired number of bits of accuracy.* They showed that this problem is complete for a natural complexity class FIXP_a that they introduced in the same paper. Informally, FIXP_a is the class of discrete search problems that can be reduced to approximating (within desired ℓ_∞-distance) any one of the Brouwer fixed points of a function given by an algebraic circuit using gates: $+, -, *, /, \max, \min$. (We will formally define FIXP_a later.)

In this paper, we want to similarly understand the case of trembling hand perfect equilibrium. For the case of two players, the problem of computing an exact trembling hand perfect equilibrium is PPAD-complete. This follows from a number of known exact pivoting algorithms for computing refinements of this notion [18,11,16]. For the case of three or more players, we are not aware of any natural analogue of the notion of ϵ-Nash equilibrium as an approximate proxy for a trembling hand perfect equilibrium.[1] Thus, we only discuss in this paper the approximation notion of Etessami and Yannakakis. The main result of the present paper is the following:

Theorem 1. *The following computational task is* FIXP_a*-complete for any* $n \geq 3$*: Given an integer payoff table for an* n*-player game* Γ*, and a rational* $\delta > 0$*,*

[1] The already studied notion of an ϵ-perfect equilibrium (ϵ-PE), which we discuss later, does *not* qualify as such an analogue: For some three-player games, every ϵ-PE uses irrational probabilities, and thus "computing" an (exact) ϵ-PE is just as problematic as computing an exact NE. Indeed, the notion of a ϵ-PE is used as a technical step towards the definition of trembling hand perfect equilibrium, rather than as a natural "numerical relaxation" of this notion.

with all numbers given in standard binary notation, compute (the binary representation of) a strategy profile x' with rational probabilities having ℓ_∞ distance at most δ to a trembling hand perfect equilibrium of Γ.

As an immediate corollary of our main theorem, and the results of Etessami and Yannakakis, we have that *approximating a Nash equilibrium and approximating a trembling hand perfect equilibrium are polynomial time equivalent tasks.* In particular, there is a polynomial time algorithm that finds an approximation to a trembling hand perfect equilibrium of a given game, using access to any oracle solving the corresponding approximation problem for the case of Nash equilibrium. To put this result in perspective, we note that Nash equilibrium and trembling hand perfect equilibrium are computationally quite different in other respects: if instead of *finding* an equilibrium, we want to *verify* that a given strategy profile is such an equilibrium, the case of Nash equilibrium is trivial, while the case of trembling hand perfect equilibrium for games with 3 (or more) players is NP-hard [8]. This might lead one to believe that approximating a trembling hand perfect equilibrium for games with 3 or more players is likely to be harder than approximating a Nash equilibrium, but we show that this is not the case.

1.1 About the Proof

Informally (for formal definitions, see below), FIXP (resp., FIXP_a) is defined as the complexity class of search problems that can be cast as exactly computing (resp., approximating) a *Brouwer fixed point* of functions represented by circuits over basis $\{+, *, -, /, \max, \min\}$ with rational constants. It was established in [6] that computing (resp., approximating) an actual Nash Equilibrium (NE) for a finite n-player game is FIXP-complete (resp., FIXP_a-complete), already for $n = 3$. Since trembling hand perfect equilibria constitute a refinement of Nash Equilibria, to show that approximating a trembling hand perfect equilibrium is FIXP_a-complete, we merely have to show that this task is in FIXP_a.

An ϵ-perfect equilibrium (ϵ-PE for short) is defined to be a fully mixed strategy profile, x, where every strategy j of every player i that is played with probability $x_{i,j} > \epsilon$ must be a best response to the other player's strategies x_{-i}. Then, a trembling hand perfect equilibrium (PE for short) is defined to be a limit point of a sequence of ϵ-PEs, for $\epsilon > 0$, $\epsilon \to 0$. Here, by limit point we mean, as usual, any point to which a subsequence of the sequence converges. Such a point must exist, by the Bolzano-Weierstrass theorem.

In rough outline, our proof that approximating a PE is in FIXP_a has the following structure:

1. We first define (in section 3) for any n-player game Γ, a map, F^ϵ_Γ, parameterized by a parameter $\epsilon > 0$, so that F^ϵ_Γ defines a map from D^ϵ_Γ to itself, where D^ϵ_Γ denotes the space of fully mixed strategy strategy profiles x such that every player plays each strategy with probability at least ϵ. Also, $F^\epsilon_\Gamma(x)$ is described by a $\{+, -, *, \min, \max\}$-circuit with ϵ as one of its inputs.

In particular, the Brouwer fixed point theorem applies to this map. We show that the circuit defining F_Γ^ϵ can be computed in polynomial time from the input game instance Γ, and that every Brouwer fixed point of F_Γ^ϵ is an ϵ-PE of the original game Γ, making crucial use of, and modifying, a new fixed point characterization of NEs that was defined and used in [6].

2. We then show (in section 4) that if $\epsilon^* > 0$ is made sufficiently small as a function of the encoding size $|\Gamma|$ of the game Γ, and of a parameter $\delta > 0$, specifically if $\epsilon^* \leq \delta^{2^{g(|\Gamma|)}}$, where g is some polynomial, then any ϵ^*-PE must be δ-close (in the l_∞-norm) to an actual PE. This part of the proof relies on real algebraic geometry.
3. We then observe (in section 5) that for any desired δ, we can encode such a sufficiently small $\epsilon^* > 0$ as a circuit that is polynomially large in the encoding size of Γ and δ, simply by *repeated squaring*. We think of this as constructing a *virtual infinitesimal* and believe that this technique will have many other applications in the context of proving FIXP_a membership using real algebraic geometry. Finally, plugging in the circuit for ϵ^* for the input ϵ in the circuit for F_Γ^ϵ, we obtain a Brouwer function $F_\Gamma^{\epsilon^*}(x)$, defined by a $\{+, -, *, \max, \min\}$-circuit, such that any fixed point of $F_\Gamma^{\epsilon^*}(x)$ is guaranteed to be a fully mixed strategy profile, $x^*_{\epsilon^*}$, that is also within l_∞ distance δ of a PE, x^*, of Γ. The triangle inequality completes the proof.

2 Definitions and Preliminaries

2.1 Game-Theoretic Notions

We use $\mathbb{Q}_+$ to denote the set of positive rational numbers. A finite n-player normal form game, $\Gamma = (N, \langle S_i \rangle_{i \in N}, \langle u_i \rangle_{i \in N})$, consists of a set $N = \{1, \ldots, n\}$ of n players indexed by their number, a set of n (disjoint) finite sets of *pure strategies*, S_i, one for each player $i \in N$, and n rational-valued *payoff functions* $u_i : S \to \mathbb{Q}$, from the product strategy space $S = \Pi_{i=1}^n S_i$ to $\mathbb{Q}$.

The elements of S, i.e., combinations of pure strategies, one for each player, are called *pure strategy profiles*. The assumption of rational values is for computational purposes. Each rational number r is represented as usual by its numerator and denominator in binary, and we use $size(r)$ to denote the number of bits in the representation. The size $|\Gamma|$ of the instance (game) Γ is the total number of bits needed to represent all the information in the game: the strategies of all the players and their payoffs for all $s \in S$.

A *mixed strategy*, x_i, for a player i is a probability distribution on its set S_i of pure strategies. Letting $m_i = |S_i|$, we view x_i as a real-valued vector $x_i = (x_{i,1}, \ldots, x_{i,m_i}) \in [0,1]^{m_i}$, where $x_{i,j}$ denotes the probability with which player i plays pure strategy j in the mixed strategy x_i. Note that we must have $x_i \geq 0$ and $\sum_{i=1}^{m_i} x_{i,m_i} = 1$. That is, a vector x_i is a mixed strategy of player i iff it belongs to the *unit simplex* $\Delta_{m_i} = \{y \in R^{m_i} | y \geq 0; \sum_{j=1}^{m_i} y_j = 1\}$. We use the notation $\pi_{i,j}$ to identify the pure strategy j of player i, as well as its representation as a mixed strategy that assigns probability 1 to strategy j and probability 0 to the other strategies of player i.

A *mixed strategy profile* $x = (x_1, \ldots, x_n)$ is a combination of mixed strategies for all the players. That is, vector x is a mixed strategy profile iff it belongs to the product of the n unit simplexes for the n players, $\{x \in R^m \mid x \geq 0; \sum_{j=1}^{m_i} x_{i,j} = 1$ for $i = 1, \ldots, k\}$. We let D_Γ denote the set of all mixed profiles for game Γ. The profile is *fully mixed* if all the pure strategies of all players have nonzero probability. We use the notation x_{-i} to denote the subvector of x induced by the pure strategies of all players except for player i. If y_i is a mixed strategy of player i, we use $(y_i; x_{-i})$ to denote the mixed profile where everyone plays the same strategy as x except for player i, who plays mixed strategy y_i.

The payoff function of each player can be extended from pure strategy profiles to mixed profiles, and we will use U_i to denote the expected payoff function for player i. Thus the (expected) payoff $U_i(x)$ of mixed profile x for player i is $\sum x_{1,j_1} \ldots x_{k,j_k} u_i(j_1, \ldots, j_k)$ where the sum is over all pure strategy profiles $(j_1, \ldots, j_k) \in S$.

A *Nash equilibrium* (NE) is a (mixed) strategy profile x^* such that all $i = 1, \ldots, n$ and every mixed strategy y_i for player i, $U_i(x^*) \geq U_i(y_i; x^*_{-i})$. It is sufficient to check switches to pure strategies only, i.e., x^* is a NE iff $U_i(x^*) \geq U_i(\pi_{i,j}; x^*_{-i})$ for every pure strategy $j \in S_i$, for each player $i = 1, \ldots, n$. Every finite game has at least one NE [12].

A mixed profile x is called a *ϵ-perfect equilibrium* (ϵ-PE) if it is (a) fully mixed, i.e., $x_{i,j} > 0$ for all i, and (b), for every player i and pure strategy j, if $x_{i,j} > \epsilon$, then the pure strategy $\pi_{i,j}$ is a best response for player i to x_{-i}. We call a mixed profile x^*, a *trembling hand perfect equilibrium* (PE) of Γ if it is a limit point of ϵ-PEs of the game Γ. In other words, we call x a PE if there exists a sequence $\epsilon_k > 0$, such that $\lim_{k\to\infty} \epsilon_k = 0$, and such that for all k there is a corresponding ϵ_k-PE, x^{ϵ_k} of Γ, such that $\lim_{k\to\infty} x^{\epsilon_k} = x^*$. Every finite game has at least one PE, and all PEs are NEs [15].

2.2 Complexity Theoretic Notions

A $\{+, -, *, \max, \min\}$-*circuit* is a circuit with inputs $x_1, x_2, \ldots, x_n$, as well as rational constants, and a finite number of (binary) computation gates taken from $\{+, -, *, \min, \max\}$, with a subset of the computation gates labeled $\{o_1, o_2, \ldots, o_m\}$ and called output gates.[2]

All circuits of this paper are $\{+, -, *, \min, \max\}$-circuits, so we shall often just write "circuit" for "$\{+, -, *, \min, \max\}$-circuit". A circuit computes a continuous function from $\mathbb{R}^n \to \mathbb{R}^m$ (and $\mathbb{Q}^n \to \mathbb{Q}^m$) in the natural way. Abusing notation slightly, we shall often identify the circuit with the function it computes.

By a (total) *multi-valued function*, f, with domain A and co-domain B, we mean a function that maps each $a \in A$ to a non-empty subset $f(a) \subseteq B$. We use $f : A \twoheadrightarrow B$ to denote such a function. Intuitively, when considering a multi-valued function as a computational problem, we are interested in producing just one of the elements of $f(a)$ on input a, so we refer to $f(a)$ as the set of *allowed*

[2] Note that the gates $\{+, -, *, \min, \max\}$ are of course redundant: gates $\{+, *, \max\}$ with rational constants are equally expressive.

outputs. A multi-valued function $f : \{0,1\}^* \twoheadrightarrow \mathbb{R}^*$ is said to be in FIXP if there is a polynomial time computable map, r, that maps each instance $I \in \{0,1\}^*$ of f to $r(I) = \langle 1^{k^I}, 1^{d^I}, P^I, C^I, a^I, b^I \rangle$, where

- k^I, d^I are positive integers and $a^I, b^I \in \mathbb{Q}^{d^I}$.
- P^I is a convex polytope in $\mathbb{R}^{k^I}$, given as a set of linear inequalities with rational coefficients.
- C^I is a circuit which maps P^I to itself.
- $\phi^I : \{1, \ldots, d^I\} \to \{1, \ldots, k^I\}$ is a finite function given by its table.
- $f(I) = \{(a_i^I y_{\phi^I(i)} + b_i^I)_{i=1}^{d^I} \mid y \in P^I \ \wedge \ C^I(y) = y\}$. Note that $f(I) \neq \emptyset$, by Brouwer's fixed point theorem.

The above is in fact one of many equivalent characterizations of FIXP [6]. Informally, FIXP are those real vector multi-valued functions, with discrete inputs, that can be cast as Brouwer fixed point computations. A multi-valued function $f : \{0,1\}^* \twoheadrightarrow \mathbb{R}^*$ is said to be FIXP-*complete* if:

1. $f \in \mathsf{FIXP}$, and
2. for all $g \in \mathsf{FIXP}$, there is a polynomial time computable map, mapping instances I of g to $\langle y^I, 1^{k^I}, 1^{d^I}, \phi^I, a^I, b^I \rangle$, where y^I is an instance of f, k^I and d^I are positive integers, ϕ^I maps $\{1, \ldots, d^I\}$ to $\{1, \ldots, k^I\}$, a^I and b^I are d^I-tuples with rational entries, so that $g(I) \supseteq \{(a_i^I z_{\phi^I(i)} + b_i^I)_{i=1}^{d^I} \mid z \in f(y^I)\}$. In other words, for any allowed output z of f on input y^I, the vector $(a_i^I z_{\phi^I(i)} + b_i^I)_{i=1}^{d^I}$ is an allowed output of g on input I.

Etessami and Yannakakis [6] showed that the multi-valued function which maps games in strategic form to their Nash equilibria is FIXP-complete.[3]

Since the output of a FIXP function consists of real-valued vectors, and as there are circuits whose fixed points are all irrational, a FIXP function is not directly computable by a Turing machine, and the class is therefore not directly comparable with standard complexity classes of total search problems (such as PPAD, PLS, or TFNP). This motivates the following definition of the discrete class FIXP_a, also from [6]. A multi-valued function $f : \{0,1\}^* \twoheadrightarrow \{0,1\}^*$ (a.k.a. a totally defined discrete search problem) is said to be in FIXP_a if there is a function $f' \in \mathsf{FIXP}$, and polynomial time computable maps $\delta : \{0,1\}^* \to \mathbb{Q}_+$ and $g : \{0,1\}^* \to \{0,1\}^*$, such that for all instances I,

$$f(I) \supseteq \{\, g(\langle I, y \rangle) \mid y \in \mathbb{Q}^* \ \wedge \ \exists y' \in f'(I) : \ \|y - y'\|_\infty \leq \delta(I) \,\}.$$

Informally, FIXP_a are those totally defined discrete search problems that reduce to approximating exact Brouwer fixed points. A multi-valued function $f : \{0,1\}^* \twoheadrightarrow \{0,1\}^*$ is said to be FIXP_a-*complete* if:

1. $f \in \mathsf{FIXP}_a$, and

[3] To view the Nash equilibrium problem as a total multi-valued function, $f_{\mathrm{Nash}} : \{0,1\}^* \twoheadrightarrow \mathbb{R}^*$, we can view all strings in $\{0,1\}^*$ as encoding some game, by viewing "ill-formed" input strings as encoding a fixed trivial game.

2. For all $g \in \mathsf{FIXP}_a$, there are polynomial time computable maps $r_1, r_2 : \{0,1\}^* \to \{0,1\}^*$, such that $g(I) \supseteq \{\, r_2(\langle I, z\rangle) \mid z \in f(r_1(I)) \,\}$.

Etessami and Yannakakis showed that the multi-valued function that maps pairs $\langle \Gamma, \delta\rangle$, where Γ is a strategic form game and $\delta > 0$, to the set of rational δ-approximations (in ℓ_∞-distance) of Nash equilibria of Γ, is FIXP_a-complete.

3 Computing ϵ-PEs in FIXP

Given a game Γ, let $m = \sum_{i\in N} m_i$ denote the total number of pure strategies of all players in Γ. For $\epsilon > 0$, let $D^\epsilon_\Gamma \subseteq D_\Gamma$ denote the polytope of fully mixed profiles of Γ such that furthermore every pure strategy is played with probability at least $\epsilon > 0$ (recall that D_Γ is the polytope of all strategy profiles). In this section, we show the following theorem.

Theorem 2. *There is a function, $F^\epsilon_\Gamma(x) : D_\Gamma \to D^\epsilon_\Gamma$, given by a circuit computable in polynomial time from Γ, with the circuit having both x and $\epsilon > 0$ as its inputs, such that for all fixed $0 < \epsilon < 1/m$, every Brouwer fixed point of the function $F^\epsilon_\Gamma(x)$ is an ϵ-PE of Γ. In particular, the problem of computing an ϵ-perfect equilibrium for a finite n-player normal form game is in* FIXP.

The rest of the section is devoted to the proof of Theorem 2. We will directly use, and somewhat modify, a construction developed and used in [6] (Lemmas 4.6 and 4.7, and definitions before them) which characterize the Nash Equilibria of a game as fixed points of a $\{+, -, *, \max, \min\}$-circuit. In particular, compared to Nash's original functions [12], the use of division is avoided. The construction defined in [6] that we modify amounts to a concrete algebraic realization of certain geometric characterizations of Nash Equilibria that were described by Gul, Pierce, and Stachetti in [7].

Concretely, suppose we are given $0 < \epsilon < 1/m$. For each mixed strategy profile x, let $v(x)$ be a vector which gives the expected payoff of each pure strategy of each player with respect to the profile x for the other players. That is, vector x is a vector of dimension m, whose entries are indexed by pairs $(i, j), i = 1, \ldots, n; j = 1, \ldots, m_i$, and $v(x)$ is also a vector of dimension m whose (i, j)-entry is $U_i(\pi_{i,j}; x_{-i})$. Let $h(x) = x + v(x)$. We can write $h(x)$ as $(h_1(x), \ldots, h_n(x))$ where $h_i(x)$ is the subvector corresponding to the strategies of player i. For each player i, consider the function $f_{i,x}(t) = \sum_{j\in S_i} \max(h_{ij}(x) - t, \epsilon)$. Clearly, this is a continuous, piecewise linear function of t. The function is strictly decreasing as t ranges from $-\infty$ (where $f_{i,x}(t) = +\infty$) up to $\max_j h_{ij}(x) - \epsilon$ (where $f_{i,x}(t) = m_i \cdot \epsilon$). Since we have $m_i \cdot \epsilon < 1$, there is a unique value of t, call it t_i, where $f_{i,x}(t_i) = 1$. The function F^ϵ_Γ is defined as follows:

$$F^\epsilon_\Gamma(x)_{ij} = \max(h_{ij}(x) - t_i, \epsilon)$$

for every $i = 1, \ldots, n$, and $j \in S_i$. From our choice of t_i, we have $\sum_{j\in S_i} F^\epsilon_\Gamma(x)_{ij} = 1$ for all $i = 1, \ldots, n$, thus for any mixed profile, x, we have $F^\epsilon_\Gamma(x) \in D^\epsilon_\Gamma$. So F^ϵ_Γ maps D_Γ to D^ϵ_Γ, and since it is clearly also continuous, it has fixed points, by Brouwer's theorem.

Lemma 1. *For $0 < \epsilon < 1/m$, every fixed point of the function F_Γ^ϵ is an ϵ-PE of Γ.*

Proof. If x is a fixed point of F_Γ^ϵ, then $x_{ij} = \max(x_{ij} + v(x)_{ij} - t_i, \epsilon)$ for all i, j. Recall that $v(x)_{ij} = U_i(\pi_{i,j}; x_{-i})$ is the expected payoff for player i of his j'th pure strategy $\pi_{i,j}$, with respect to strategies x_{-i} of the other players.

Note that the equation $x_{ij} = \max(x_{ij} + U_i(\pi_{i,j}; x_{-i}) - t_i, \epsilon)$ implies that $U_i(\pi_{i,j}; x_{-i}) = t_i$ for all i, j such that $x_{ij} > \epsilon$, and that $U_i(\pi_{i,j}; x_{-i}) \leq t_i$ for all i, j such that $x_{ij} = \epsilon$. Consequently, by definition, x constitutes an ϵ-PE. □

The following Lemma shows that we can implement the function $F_\Gamma^\epsilon(x)$ by a circuit which has x and ϵ as inputs. The proof exploits sorting networks.

Lemma 2. *Given Γ, we can construct in polynomial time a $\{+, -, *, \max, \min\}$-circuit that computes the function $F_\Gamma^\epsilon(x)$, where x and ϵ are inputs to the circuit.*

Proof. The circuit does the following.

Given a vector $x \in D_\Gamma$, first compute $y = h(x) = x + v(x)$. It is clear from the definition of $v(x)$ that y can be computed using $+, *$ gates. For each player i, let y_i be the corresponding subvector of y induced by the strategies of player i. Sort y_i in decreasing order, and let z_i be the resulting sorted vector, i.e. the components of $z_i = (z_{i1}, \ldots, z_{im_i})$ are the same as the components of y_i, but they are sorted: $z_{i1} \geq z_{i2} \geq \ldots \geq z_{im_i}$. To obtain z_i, the circuit uses a polynomial sized sorting network, W_i, for each i (see e.g. Knuth [9] for background on sorting networks). For each comparator gate of the sorting network we use a max and a min gate.

Using this, for each player i, we compute t_i as the following expression:

$$\max\{(1/l) \cdot ((\sum_{j=1}^{l} z_{ij}) + (m_i - l) \cdot \epsilon - 1) | l = 1, \cdots, m_i\}$$

We will show below that this expression does indeed give the correct value of t_i. Finally, we output $x'_{ij} = \max(y_{ij} - t_i, \epsilon)$ for each $i = 1, \ldots, d; j \in S_i$.

We now have to establish that $t_i = \max\{(1/l) \cdot ((\sum_{j=1}^{l} z_{ij}) + (m_i - l) \cdot \epsilon - 1) | l = 1, \cdots, m_i\}$. Consider the function $f_{i,x}(t) = \sum_{j \in S_i} \max(z_{ij} - t, \epsilon)$ as t decreases from $z_{i1} - \epsilon$ where the function value is at its minimum of $m_i \epsilon$, down until the function reaches the value 1. In the first interval from $z_{i1} - \epsilon$ to $z_{i2} - \epsilon$ the function is $f_{i,x}(t) = z_{i1} - t + (m_i - 1) \cdot \epsilon$; in the second interval from $z_{i2} - \epsilon$ to $z_{i3} - \epsilon$ it is $f_{i,x}(t) = z_{i1} + z_{i2} - 2t + (m_i - 2) \cdot \epsilon$, and so forth. In general, in the l-th interval, $f_{i,x}(t) = \sum_{j=1}^{l} (z_{ij} - t) + (m_i - l) \cdot \epsilon = \sum_{j=1}^{l} z_{ij} - lt + (m_i - l) \cdot \epsilon$. If the function reaches the value 1 in the l'th interval, then clearly $t_i = ((\sum_{j=1}^{l} z_{ij}) + (m_i - l) \cdot \epsilon - 1)/l$. In that case, furthermore for $k < l$, we have $\sum_{j=1}^{k} (z_{ij} - t_i) + (m_i - k) \cdot \epsilon \leq \sum_{j=1}^{l} (z_{ij} - t_i) + (m_i - l) \cdot \epsilon = 1$, because in that case we know $(z_{ij} - t_i) \geq \epsilon$ for every $j \in \{1, \ldots, l\}$. Therefore, in this case $((\sum_{j=1}^{k} z_{ij}) + (m_i - k) \cdot \epsilon - 1)/k \leq t_i$. On the other hand, if $l < m_i$, then for $k > l$ we have $t_i \geq z_{ik} - \epsilon$, i.e., $z_{ik} - t_i \leq \epsilon$, and thus for all $k > l$, $k \leq m_i$, we have $\sum_{j=1}^{k}(z_{ij} - t_i) + (m_i - k) \cdot \epsilon \leq \sum_{j=1}^{l}(z_{ij} - t_i) + (m_i - l) \cdot \epsilon = 1$. Thus again $((\sum_{j=1}^{k} z_{ij}) + (m_i - k) \cdot \epsilon - 1)/k \leq t_i$. Therefore, $t_i = \max\{(1/l) \cdot ((\sum_{j=1}^{l} z_{ij}) + (m_i - l) \cdot \epsilon - 1) | l = 1, \cdots, m_i\}$. □

Lemma 1 and Lemma 2 together immediately imply Theorem 2.

4 Almost Implies Near

As outlined in the introduction, in this section, we want to exploit the "uniform" function $F_\Gamma^\epsilon(x)$ devised in the previous section for ϵ-PEs, and construct a "small enough" $\epsilon^* > 0$ such that any fixed point of $F_\Gamma^{\epsilon^*}(x)$ is δ-close, for a given $\delta > 0$, to an actual PE.

The following is a special case of the simple but powerful "almost implies near" paradigm of Anderson [1].

Lemma 3 (Almost implies near). *For any fixed strategic form game Γ, and any $\delta > 0$, there is an $\epsilon > 0$, so that any ϵ-PE of Γ has ℓ_∞-distance at most δ to some PE of Γ.*

Proof. Assume to the contrary that there is a game Γ and a $\delta > 0$ so that for all $\epsilon > 0$, there is an ϵ-PE x_ϵ of Γ so that there is no PE in a δ-neighborhood (with respect to l_∞ norm) of x_ϵ. Consider the sequence $(x_{1/n})_{n \in \mathbf{N}}$. Since this is a sequence in a compact space (namely, the space of mixed strategy profiles of Γ), it has a limit point, x^*, which is a PE of Γ (since x_ϵ is a ϵ-PE). But this contradicts the statement that there is no PE in a δ-neighborhood of any of the profiles $x_{1/n}$. □

A priori, we have no bound on ϵ, but we next use the machinery of real algebraic geometry [2,3] to obtain a specific bound as a "free lunch", just from the fact that Lemma 3 is true.

Lemma 4. *There is a constant c, so that for all integers $n, m, k, B \in \mathbb{N}$ and $\delta \in \mathbb{Q}_+$, the following holds. Let $\epsilon \leq \min(\delta, 1/B)^{n^{cm^3}}$. For any n-player game Γ with at most m pure strategies for all player, and with integer payoffs of absolute value at most B, any ϵ-PE of Γ has ℓ_∞-distance at most δ to some PE of Γ.*

Proof. The proof involves constructing formulas in the first order theory of real numbers, which formalize the "almost implies near" statement of Lemma 3, with δ being "hardwired" as a constant and ϵ being the only free variable. Then, we apply *quantifier elimination* to these formulas. This leads to a quantifier free statement to which we can apply standard theorems bounding the size of an instantiation of the free variable ϵ making the formula true. We shall apply and refer to theorems in the monograph of Basu, Pollack and Roy [2,3]. Note that we specifically refer to theorems and page numbers of the online edition [3]; these are in general different from the printed edition [2].

First-order formula for an ϵ-perfect equilibrium: Define $R_i(x \setminus k)$ as the polynomial expressing $U_i(\pi_{i,k}; x_{-i})$, that is, the expected payoff to player i when it uses pure strategy k, and the other players play according to their mixed strategy in the profile x. Thus,

$$R_i(x \setminus k) := \sum_{a_{-i}} u_i(k; a_{-i}) \prod_{j \neq i} x_{j,a_j} .$$

Let EPS-PE(x, ϵ) be the quantifier-free first-order formula, with free variables $x \in \mathbb{R}^m$ and $\epsilon \in \mathbb{R}$, defined by the conjunction of the following formulas that together express that x is an ϵ-perfect equilibrium:

$$\begin{gathered} x_{i,j} > 0 \quad \text{for } i = 1\ldots, n, \text{ and } j = 1, \ldots, m_i \ , \\ x_{i,1} + \cdots + x_{i,m_i} = 1 \quad \text{for } i = 1\ldots, n \ , \\ (R_i(x \setminus k) \geq R_i(x \setminus l)) \vee (x_{i,k} \leq \epsilon) \quad \text{for } i = 1\ldots, n, \text{ and } k, l = 1, \ldots, m_i \ . \end{gathered}$$

First-order formula for perfect equilibrium: Let PE(x) denote the following first-order formula with free variables $x \in \mathbb{R}^m$, expressing that x is a perfect equilibrium:

$$\forall \epsilon > 0 \, \exists y \in \mathbb{R}^m : \text{EPS-PE}(y, \epsilon) \wedge \|x - y\|^2 < \epsilon \ .$$

First-order formula for "almost implies near" statement: Given a *fixed* $\delta > 0$ let PE-bound$_\delta(\epsilon)$ denote the following first-order formula with free variable $\epsilon \in \mathbb{R}$, denoting that any ϵ-perfect equilibrium of G is δ-close to a perfect equilibrium (in ℓ_2-distance, and therefore also in ℓ_∞-distance):

$$\forall x \in \mathbb{R}^m \, \exists y \in \mathbb{R}^m : (\epsilon > 0) \wedge \left(\neg \, \text{EPS-PE}(x, \epsilon) \vee \left(\text{PE}(y) \wedge \|x - y\|^2 < \delta^2\right)\right) \ .$$

Suppose $\delta^2 = 2^{-k}$ and the payoffs have absolute value at most $B = 2^\tau$. Then for this formula we have

- The total degree of all involved polynomials is at most $\max(2, n - 1)$.
- The bitsize of coefficients is at most $\max(k, \tau)$.
- The number of free variables is 1.
- Converting to prenex normal form, the formula has 4 blocks of quantifiers, of sizes m, m, 1, m, respectively.

We now apply quantifier elimination [3, Algorithm 14.6, page 555] to the formula PE-bound$_\delta(\epsilon)$, converting it into an equivalent quantifier free formula PE-bound$'_\delta(\epsilon)$ with a single free variable ϵ. This is simply a Boolean formula whose atoms are sign conditions on various polynomials in ϵ. The bounds given by Basu, Pollack and Roy in association to Algorithm 14.6 imply that for this formula:

- The degree of all involved polynomials (which are univariate polynomials in ϵ) is $\max(2, n - 1)^{O(m^3)} = n^{O(m^3)}$.

– The bitsize of all coefficients is at most $\max(k,\tau)\max(2,n-1)^{O(m^3)} = \max(k,\tau)n^{O(m^3)}$.

By Lemma 3, we know that there exists an $\epsilon > 0$ so that the formula PE-bound$'_\delta(\epsilon)$ is true. We now apply (as the involved polynomials are univariate, simpler tools would also suffice) Theorem 13.14 of Basu, Pollack and Roy [3, Page 521] to the set of polynomials that are atoms of PE-bound$'_\delta(\epsilon)$ and conclude that PE-bound$'_\delta(\epsilon^*)$ is true for some $\epsilon^* \geq 2^{-\max(k,\tau)n^{O(m^3)}}$. By the semantics of the formula PE-bound$_\delta(\epsilon)$, we also have that PE-bound$_\delta(\epsilon')$ is true for all $\epsilon' \leq \epsilon^*$, and the statement of the lemma follows. □

5 Proof of the Main Theorem

We now prove Theorem 1. Let Γ be the n-player game given as input. Let m be the total number of pure strategies for all player. Let $B \in \mathbb{N}$ be the largest absolute value of any payoff of Γ. By the definition of FIXP_a, our task is the following. Given a parameter $\delta > 0$, we must construct a polytope P, a circuit $C : P \to P$, and a number δ', so that δ'-approximations to fixed points of C can be efficiently transformed into δ-approximations of PEs of Γ. In fact, we shall let $\delta' = \delta/2$ and ensure that δ'-approximations to fixed points of C *are* δ-approximations of PEs of Γ. The polytope P is simply the polytope D_Γ of all strategy profiles of Γ; clearly we can output the inequalities defining this polytope in polynomial time. The circuit C is the following: We construct the circuit for the function F^ϵ_Γ of Section 3. Then, we construct a circuit for the number $\epsilon^* = \min(\delta/2, B^{-1})^{2^{\lceil cm^3 \lg n\rceil}} \leq \min(\delta/2, B^{-1})^{n^{cm^3}}$, where c is the constant of Lemma 4: The circuit simply repeatedly squares the number $\min(\delta/2, B^{-1})$ (which is a rational constant) and thereby consists of exactly $\lceil cm^3 \lg n\rceil$ multiplication gates, i.e., a polynomially bounded number. We then plug in the circuit for ϵ^* for the parameter ϵ in the circuit for F^ϵ_Γ, obtaining the circuit C, which is obviously a circuit for $F^{\epsilon^*}_\Gamma$. Now, by Theorem 2, any fixed point of C on P is an ϵ^*-PE of Γ. Therefore, by Lemma 4, any fixed point of C is a $\delta/2$-approximation in ℓ_∞-distance to a PE of Γ. Finally, by the triangle inequality, any $\delta' = \delta/2$-approximation to a fixed point of C on P is a $\delta/2 + \delta/2 = \delta$ approximation of a PE of Γ. This completes the proof.

6 Conclusion

We have showed that the problem of *approximating* a trembling hand perfect equilibrium for a finite strategic form game is in FIXP_a. We do not know if *exactly* computing a trembling hand perfect equilibrium is in FIXP, and we consider this an interesting open problem, although it should be noted that if one is interested exclusively in the Turing Machine complexity of the problem, FIXP_a membership of the approximation version is arguably "the real thing". We also note that this makes our proof interesting as a case where membership in FIXP_a is *not* established as a simple corollary of the exact problem being in the "abstract class" FIXP, as was the case for all examples in the original paper of Etessami and Yannakakis.

Acknowledgements. Hansen and Miltersen acknowledge support from the Danish National Research Foundation and The National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which their work was performed. They also acknowledge support from the Center for Research in Foundations of Electronic Markets (CFEM), supported by the Danish Strategic Research Council.

References

1. Anderson, R.M.: "Almost" Implies "Near". Transactions of the American Mathematical Society 296(1), 229–237 (1986)
2. Basu, S., Pollack, R., Roy, M.: Algorithms in Real Algebraic Geometry, 2nd edn. Springer (2008)
3. Basu, S., Pollack, R., Roy, M.: Algorithms in Real Algebraic Geometry, online edition (2011), `http://perso.univ-rennes1.fr/marie-francoise.roy/bpr-ed2-posted2.html`
4. Chen, X., Deng, X.: Settling the complexity of two-player Nash equilibrium. In: Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), pp. 261–272 (2006)
5. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. In: Procedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC 2006), pp. 71–78 (2006)
6. Etessami, K., Yannakakis, M.: On the complexity of Nash equilibria and other fixed points. SIAM J. Comput. 39(6), 2531–2597 (2010)
7. Gul, F., Pearce, D., Stacchetti, E.: A bound on the proportion of pure strategy equilibria in generic games. Math. of Oper. Res. 18, 548–552 (1993)
8. Hansen, K.A., Miltersen, P.B., Sørensen, T.B.: The computational complexity of trembling hand perfection and other equilibrium refinements. In: Kontogiannis, S., Koutsoupias, E., Spirakis, P.G. (eds.) SAGT 2010. LNCS, vol. 6386, pp. 198–209. Springer, Heidelberg (2010)
9. Knuth, D.E.: The Art of Computer Programming. Sorting and Searching, vol. III. Addison-Wesley (1973)
10. Lemke, C.E.: Bimatrix equilibrium points and mathematical programming. Management Science 11, 681–689 (1965)
11. Miltersen, P.B., Sørensen, T.B.: Computing a quasi-perfect equilibrium of a two-player game. Economic Theory 42(1), 175–192 (2010)
12. Nash, J.: Non-cooperative games. Annals of Mathematics 54, 289–295 (1951)
13. Papadimitriou, C.H.: On graph-theoretic lemmata and complexity classes. In: Proceedings of the 31st Annual Symposium on Foundations of Computer Science, St. Louis, MS, pp. 794–801. IEEE Computer Society Press (October 1990)
14. Scarf, H.E.: The approximation of fixed points of a continuous mapping. SIAM Journal of Applied Mathematics 15, 1328–1343 (1967)

15. Selten, R.: A reexamination of the perfectness concept for equilibrium points in extensive games. International Journal of Game Theory 4, 25–55 (1975)
16. Sørensen, T.B.: Computing a proper equilibrium of a bimatrix game. In: ACM Conference on Electronic Commerce, EC 2012, pp. 916–928 (2012)
17. van Damme, E.: Stability and Perfection of Nash Equlibria, 2nd edn. Springer (1991)
18. von Stengel, B., van den Elzen, A., Talman, D.: Computing normal form perfect equilibria for extensive two-person games. Econometrica 70(2), 693–715 (2002)