2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Settling the complexity of computing approximate two-player Nash equilibria

Aviad Rubistein
UC Berkeley
aviad@eecs.berkeley.edu

***Abstract*—We prove that there exists a constant $\epsilon > 0$ such that, assuming the Exponential Time Hypothesis for PPAD, computing an ϵ-approximate Nash equilibrium in a two-player $n \times n$ game requires time $n^{\log^{1-o(1)} n}$. This matches (up to the $o(1)$ term) the algorithm of Lipton, Markakis, and Mehta [54].**

Our proof relies on a variety of techniques from the study of probabilistically checkable proofs (PCP); this is the first time that such ideas are used for a reduction between problems inside PPAD.

En route, we also prove new hardness results for computing Nash equilibria in games with many players. In particular, we show that computing an ϵ-approximate Nash equilibrium in a game with n players requires $2^{\Omega(n)}$ *oracle queries* to the payoff tensors. This resolves an open problem posed by Hart and Nisan [43], Babichenko [13], and Chen et al. [28]. In fact, our results for n-player games are stronger: they hold with respect to the (ϵ, δ)-WeakNash relaxation recently introduced by Babichenko et al. [15].

***Keywords*-Computational complexity**

I. Introduction

For the past decade, the central open problem in equilibrium computation has been whether two-player Nash equilibrium admits a PTAS. We had good reasons to be hopeful: there was a series of improved approximation ratios [51], [34], [33], [23], [67] and several approximation schemes for special cases [48], [35], [3], [17]. Yet most interesting are two inefficient algorithms for two-player Nash:

- the classic Lemke-Howson algorithm [53] finds an exact Nash equilibrium in exponential time; and
- a simple algorithm by Lipton, Markakis, and Mehta [54] finds an ϵ-Approximate Nash Equilibrium in time $n^{O(\log n)}$.

Although the Lemke-Howson algorithm takes exponential time, it has a special structure which places the problem inside the complexity class PPAD [57]; i.e. it has a polynomial time reduction to the canonical problem EndOfALine[1]:

Definition I.1 (EndOfALine [32])**.** Given two circuits S and P, with m input bits and m output bits each, such that $P(0^m) = 0^m \neq S(0^m)$, find an input $x \in \{0,1\}^m$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^m$.

Proving hardness for problems in PPAD is notoriously challenging because they are *total*, i.e. they always have a solution, so the standard techniques from NP-hardness do not apply. By now, however, we know that exponential and polynomial approximations for two-player Nash are PPAD-complete [32], [29], and so is ϵ-approximation for games with n players [62].

However, ϵ-approximation for two-player Nash is unlikely to have the same fate: otherwise, the quasi-polynomial algorithm of [54] would refute the Exponential Time Hypothesis for PPAD:

Hypothesis 1 (ETH for PPAD [15])**.** *Solving* EndOfALine *requires time* $2^{\tilde{\Omega}(n)}$.[2]

Thus the strongest hardness result we can hope to prove (given our current understanding of complexity[3]) is a quasi-polynomial hardness that sits inside PPAD:

Theorem I.2 (Main Theorem)**.** *There exists a constant $\epsilon > 0$ such that, assuming ETH for* PPAD*, finding an ϵ-Approximate Nash Equilibrium in a two-player $n \times n$ game requires time $T(n) = n^{\log^{1-o(1)} n}$.*

A. Techniques

Given an EndOfALine instance of size n, we construct a two-player $N \times N$ game for $N = 2^{n^{1/2+o(1)}}$ whose approximate equilibria correspond to solutions to the EndOfALine instance. Thus, assuming the "ETH for PPAD", finding an approximate equilibrium requires time $2^n = N^{\log^{1-o(1)} N}$.

The main steps of the final construction are: (i) reducing EndOfALine to a new discrete problem which we call LocalEndOfALine; (ii) reducing LocalEndOfALine to a problem of finding an approximate Brouwer fixed point; (iii) reducing from Brouwer fixed point to finding an approximate Nash equilibrium in a multiplayer game over $n^{1/2+o(1)}$ players with $2^{n^{1/2+o(1)}}$ actions each; and (iv) reducing to the two-player game.

The main novelty in the reduction is the use of techniques such as error correcting codes and probabilistically checkable proofs (PCPs) inside PPAD. In particular, the way we use PCPs in our proof is very unusual.

[1]In the literature the problem has been called EndOfTheLine; we believe that the name EndOfALine is a more accurate description.

[2]As usual, n is the size of the description of the instance, i.e. the size of the circuits S and P.

[3]Given our current understanding of complexity, refuting ETH for PPAD seems unlikely: there are matching black-box lower bounds [45], [19]. Recall that the NP-analogue ETH [47] is widely used (e.g. [49], [55], [1], [25], [30]), often in stronger variants such as SETH [46], [26] and NSETH [27].

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.35

*Constructing the first gap: showing hardness of ϵ-*SUCCINCTBROUWER$_2$

The first step in all known PPAD-hardness results for (approximate) Nash equilibrium is reducing ENDOFALINE to the problem of finding an (approximate) Brouwer fixed point of a continuous, Lipschitz function $f\colon [0,1]^n \to [0,1]^n$. Let $\epsilon > 0$ be an arbitrarily small constant. Previously, the state of the art for computational hardness of approximation of Brouwer fixed points was:

Theorem I.3 ([62], informal)**.** *It is* PPAD*-hard to find an* $\mathbf{x} \in [0,1]^n$ *such that* $\|f(\mathbf{x}) - \mathbf{x}\|_\infty \le \epsilon$.

Here and for the rest of the paper, all distances are relative; in particular, for $\mathbf{x} \in [0,1]^n$ and $p < q$, we have $\|\mathbf{x}\|_p \le \|\mathbf{x}\|_q$.

Theorem I.3 implied that it is hard to find an $\mathbf{x}$ such that $f(\mathbf{x})$ is approximately equal to $\mathbf{x}$ *on every coordinate*. The first step in our proof is to strengthen this result to obtain hardness of approximation with respect to 2-norm:

Theorem I.4 (Informal)**.** *It is* PPAD*-hard to find an* $\mathbf{x} \in [0,1]^n$ *such that* $\|f(\mathbf{x}) - \mathbf{x}\|_2 \le \epsilon$.

Now, even finding an $\mathbf{x}$ such that $f(\mathbf{x})$ is approximately equal to $\mathbf{x}$ on *most of the coordinates* is already PPAD-hard.

Theorem I.3 was obtained by adapting a construction due to Hirsch, Papadimitriou, and Vavasis [45]. The main idea is to partition the $[0,1]^n$ into 2^n subcube, and consider the grid formed by the subcube-centers; then embed a path (in fact, many paths and cycles when reducing from ENDOFALINE) along an arbitrary sequence of neighboring grid-points/subcube-centers. The function is carefully defined along the embedded path, guaranteeing both Lipschitz continuity and that approximate endpoints occur only near subcube-centers corresponding to ends of paths.

Here we observe that if we want a larger displacement (in particular, constant relative 2-norm) we actually want the consecutive vertices on the path to be as far as possible from each other. We thus replace the neighboring grid-points with their encoding by an *error correcting code*.

The first obstacle to using PCP-like techniques for problems in PPAD is their totality (i.e. a solution always exists). For NP-hard problems, the PCP verifier expects the proof to be encoded in some error correcting code. If the proof is far from any codeword, the verifier detects that (with high probability), and immediately rejects. For problems in PPAD (more generally, in TFNP) this is always tricky because it is not clear what does it mean "to reject". Hirsch et al.'s construction has the following useful property: for the vast majority of $\mathbf{x}$'s (in particular, all $\mathbf{x}$'s far from the embedding of the paths) the displacement $f(\mathbf{x}) - \mathbf{x}$ is the same default displacement. Thus, when an $\mathbf{x}$ is too far from any codeword to faithfully decode it, we can simply apply the default displacement.

We note that Theorem I.4 is already significant enough to obtain new results for many-player games (see discussion in Subsection I-B). Furthermore, its proof is relatively simple, and in particular "PCP-free". (See full version for details.)

The main challenge: locality.

Our ultimate goal is to construct a two-player game that simulates the Brouwer function from Theorem I.4. This is done via an *imitation gadget*: Alice's mixed strategy induces a point $\mathbf{x}^{(\mathcal{A})} \in [0,1]^n$; Bob's strategy induces $\mathbf{x}^{(\mathcal{B})} \in [0,1]^n$; Alice wants to minimize $\left\|\mathbf{x}^{(\mathcal{A})} - \mathbf{x}^{(\mathcal{B})}\right\|_2$, whereas Bob wants to minimize $\left\|f\left(\mathbf{x}^{(\mathcal{A})}\right) - \mathbf{x}^{(\mathcal{B})}\right\|_2$. Alice and Bob are both satisfied at a fixed point, where $\mathbf{x}^{(\mathcal{A})} = \mathbf{x}^{(\mathcal{B})} = f\left(\mathbf{x}^{(\mathcal{A})}\right)$.

The main obstacle is that we want to incentivize Bob to minimize $\left\|f\left(\mathbf{x}^{(\mathcal{A})}\right) - \mathbf{x}^{(\mathcal{B})}\right\|_2$ via *local constraints* (payoffs - each depends on one pure strategy), while $f\left(\mathbf{x}^{(\mathcal{A})}\right)$ has a *global dependency* on Alice's entire mixed strategy.

Our goal is thus to construct a hard Brouwer function that can be *locally computed*. How local does the computation need to be? In a game of size $2^{\sqrt{n}} \times 2^{\sqrt{n}}$, each strategy can faithfully store information about $\sqrt{n}$ bits. Specifically, our construction will be $n^{1/2+o(1)}$-local.

We haven't yet defined exactly what it means for our construction to be "$n^{1/2+o(1)}$-local"; the exact formulation is quite cumbersome as the query access needs to be partly adaptive, robust to noise, etc. Eventually we formalize the "locality" of our Brouwer function via a statement about multiplayer games. On a high level, however, our goal is to show that for any $j \in \{1,\ldots,n\}$, the j-th output $f_j(\mathbf{x})$ can be approximately computed, with high probability, by accessing $\mathbf{x}$ at only $n^{1/2+o(1)}$ coordinates.

This is a good place to note that achieving any sense of "local computation" in our setting is surprising, even if we consider just the error correcting encoding for our Brouwer function: in order to maintain constant relative distance, an average bit of the output must depend on a constant fraction of the input bits!

LOCALENDOFALINE

In order to introduce locality, we go back to the ENDOFALINE problem. "Wishful thinking": imagine that we could replace the arbitrary predecessor and successor circuits in ENDOFALINE with NC^0 (constant depth and constant fan-in) circuits $S^{\text{LOCAL}}, P^{\text{LOCAL}} : \{0,1\}^n \to \{0,1\}^n$, so that each output bit only depends on a constant number of input bits. Imagine further that we had the guarantee that for each input, the outputs of $S^{\text{LOCAL}}, P^{\text{LOCAL}}$ differ from the input on just a constant number of bits. Additionally, it would be really nice if we had a succinct pointer that immediately told us which bits are about to be replaced. (We later call this succinct pointer the *counter*, because it also cycles through its possible values in a fixed order.)

Suppose all our wishes came true, and furthermore the hard Brouwer function from Theorem I.4 used a *linear* error correcting code. Then, we could use the encoding of the counter, henceforth $C(u)$, to read only the bits that are about to be replaced, and the inputs that determine the new values of those bits. Thus, using only local access to a tiny fraction of the bits ($|C(u)| + O(1)$), we can construct a difference vector $u - S^{\text{LOCAL}}(u)$ (which is 0 almost everywhere). As we discussed above, the encodings $E(u), E(S^{\text{LOCAL}}(u))$ must differ on a constant fraction of the bits - but because the code is linear, we can also locally construct the difference vector $E(u) - E(S^{\text{LOCAL}}(u)) = E(u - (S^{\text{LOCAL}}(u)))$. Given $E(u) - E(S^{\text{LOCAL}}(u))$, we can locally compute any bit of $E(S^{\text{LOCAL}}(u))$ by accessing only the corresponding bit of $E(u)$.

Back to reality: unfortunately we do not know of a reduction to such a restricted variant of ENDOFALINE. Surprisingly, we can almost do that. The problem LOCALENDOFALINE (formally defined in the full version) satisfies all the guarantees defined above, is linear-time reducible from ENDOFALINE, but has one caveat: it is only defined on a strict subset V^{LOCAL} of the discrete hypercube ($V^{\text{LOCAL}} \subsetneq \{0,1\}^n$). Verifying that a vertex belongs to V is quite easy - it can be done in AC^0. Let us take a brief break to acknowledge this new insight about the canonical problem of PPAD:

Theorem I.5. *The predecessor and successor circuits of* ENDOFALINE *are, wlog,* AC^0 *circuits.*

The class AC^0 is quite restricted, but the outputs of its circuits are not local functions of the inputs. Now, we want to represent u in a way that will make it possible to locally determine whether $u \in V^{\text{LOCAL}}$ or not. To this end we augment the linear error correcting encoding $E(u)$ with a *probabilistically checkable proof* (PCP) $\pi(u)$ of the statement $(u \in V^{\text{LOCAL}})$.

Our holographic proof system

Some authors distinguish between PCPs and holographic proofs[4]: a PCP verifier has unrestricted access to the instance, and queries the proof locally; whereas the holographic proof verifier has restricted, local access to both the proof and (an error correcting encoding of) the instance. In this sense, what we actually want is a holographic proof.

We construct a holographic proof system with some very unusual properties. We are able to achieve them thanks to our modest locality desideratum: $n^{1/2+o(1)}$, as opposed to the typical $\mathsf{polylog}(n)$ or $O(1)$. We highlight here a few of those properties; full version for details.

- **(Local proof construction)** The most surprising property of our holographic proof system is that the proof $\pi(u)$ can be constructed from local access to the encoding $E(u)$. In particular, note that we can locally compute $E(S^{\text{LOCAL}}(u))$ because $E(\cdot)$ is linear - but $\pi(\cdot)$ is not. Once we obtain $E(S^{\text{LOCAL}}(u))$, we can use local proof construction to compute $\pi(S^{\text{LOCAL}}(u))$ locally.
- **(Very low random-bit complexity)** Our verifier is only allowed to use $(1/2 + o(1)) \log_2 n$ random bits - this is much lower even than the $\log_2 n$ bits necessary to choose one entry at random. In related works, similar random-bit complexity was achieved by bundling the entries together via "birthday repetition". To some extent, something similar happens here, but our locality is already $n^{1/2+o(1)}$ so no bundling (or repetition) is necessary. To achieve nearly optimal random-bit complexity, we use λ-biased sets over large finite fields together with the Sampling Lemma of Ben-Sasson et al. [21].
- **(Tolerant verifier)** Typically, a verifier must reject (with high probability) whenever the input is far from valid, but it is allowed to reject even if the input is off by only one bit. Our verifier, however, is *required* to accept (with high probability) inputs that are close to valid proofs. (This is related to the notion of "tolerant testing", which was defined in [58] and discussed in [42] for locally testable codes.)
- **(Local decoding)** We make explicit use of the property that our holographic proof system is also a locally decodable code. While the relations between PCPs and locally testable codes have been heavily explored (see e.g. Goldreich's survey [41]), the connection to locally decodable codes is not as immediate. Nevertheless, related ideas of Locally Decode/Reject Codes [56] and decodable PCP [38] have been used before in order to facilitate composition of tests (our holographic proof system, in contrast, is essentially composition-free). Fortunately, as noted by [38] many constructions of PCPs are already implicitly locally decodable.
- **(Robust everything)** Ben-Sasson et al. [20] introduce a notion of *robust soundness*, where on an invalid proof, the string read by the verifier must be far from any acceptable string. (Originally the requirement is far in expectation, but we want far with high probability.) Another way of looking at the same requirement, is that even if a malicious prover *adaptively* changes a small fraction of the bits queried by the verifier, the test is still sound. In this sense, we require that all our guarantees, not just soundness, continue to hold (with high probability) even if a malicious entity adaptively changes a small fraction of the bits queried by the verifier.

How is local proof construction possible? At a high level, our holographic proof system expects an encoding of u as a

[4]In a nutshell, PCPs or holographic proofs are proofs that can be verified "locally" (with high probability) by reading only a small (random) portion of the proof; see e.g. [7, Chapter 18] for many more details.

low-degree t-variate polynomial, and a few more low-degree t-variate polynomials, that encode the proof of $u \in V^{\mathrm{LOCAL}}$. (This is essentially the standard "arithmetization", dating back at least to [11], [63], although our construction is most directly inspired by [59], [65].) In our actual proof, t is a small super-constant, e.g. $t \triangleq \sqrt{\log n}$; but for our exposition here, let us consider $t = 2$, i.e. we have bivariate polynomials.

The most interesting part of the proof verification is testing that a certain low-degree polynomial $\Psi \colon \mathcal{G}^2 \to \mathcal{G}$, for some finite field $\mathcal{G}$ of size $|\mathcal{G}| = \Theta\left(n^{1/2+o(1)}\right)$, is identically zero over all of $\mathcal{F}^2$, for some subset $\mathcal{F} \subsetneq \mathcal{G}$ of cardinality $|\mathcal{F}| = |\mathcal{G}| / \mathsf{polylog}\,(n)$. This can be done by expecting the prover to provide the following low-degree polynomials:

$$\begin{aligned}\Psi'(x,y) &\triangleq \sum_{f_i \in \mathcal{F}} \Psi(x, f_i)\, y^i \\ \Psi''(x,y) &\triangleq \sum_{f_j \in \mathcal{F}} \Psi'(f_j, y)\, x^j.\end{aligned}$$

Then, $\Psi''(x,y) = \sum_{f_i, f_j \in \mathcal{F}} \Psi(f_j, f_i)\, x^j y^i$ is the zero polynomial if and only if Ψ is indeed identically zero over all of $\mathcal{F}^2$. $\Psi(x,y)$ can be computed by accessing $E(u)$ on just a constant number of entries. Thus, computing $\Psi'(x,y)$ requires $\Psi(x, f_i)$ for all $f_i \in \mathcal{F}$, so a total of $\Theta\left(n^{1/2+o(1)}\right)$ queries to $E(u)$. However, computing even one entry of $\Psi''(\cdot)$ requires $\Omega(n)$ queries to $E(u)$. The crucial observation is that we don't actually need the prover to provide Ψ''. Instead, it suffices that the prover provide Ψ', and the verifier checks that $\sum_{f_j \in \mathcal{F}} \Psi'(f_j, y)\, x^j = 0$ for sufficiently many (x, y).

Putting it all together via polymatrix games

The above arguments suffice to construct a hard Brouwer function (in the sense of Theorem I.4) that can be computed "$n^{1/2+o(1)}$-locally". We formalize this statement in terms of approximate Nash equilibria in a polymatrix game.

Definition I.6 (Polymatrix games)**.** In a polymatrix game, each pair of players simultaneously plays a separate two-player subgame. Every player has to play the same strategy in every two-player subgame, and her utility is the sum of her subgame utilities. The game is given in the form of the payoff matrix for each two-player subgame.

We construct a bipartite polymatrix game between $n^{1/2+o(1)}$ players with $2^{n^{1/2+o(1)}}$ actions each. By "bipartite", we mean that each player on Alice's side only interacts with players on Bob's side and vice versa. The important term here is "polymatrix": it means that when we compute the payoffs in each subgame, they can only depend on the $n^{1/2+o(1)}$ coordinates described by the two players' strategies. It is in this sense that we guarantee "local computation".

The mixed strategy profile $\mathcal{A}$ of all the players on Alice's side of the bipartite game induces a vector $\mathbf{x}^{(\mathcal{A})} \in [0,1]^m$, for some $m = n^{1+o(1)}$. The mixed strategy profile $\mathcal{B}$ of all the players on Bob's side induces a vector $\mathbf{x}^{(\mathcal{B})} \in [0,1]^m$. Our main technical result is:

Proposition I.7 (Informal)**.** *If all but an ϵ-fraction of the players play ϵ-optimally, then $\left\|\mathbf{x}^{(\mathcal{A})} - \mathbf{x}^{(\mathcal{B})}\right\|_2^2 = O(\epsilon)$ and $\left\|f\left(\mathbf{x}^{(\mathcal{A})}\right) - \mathbf{x}^{(\mathcal{B})}\right\|_2^2 = O(\epsilon)$.*

Each player on Alice's side corresponds to one of the PCP verifier's random string. Her strategy corresponds to an assignment to the bits queried by the verifier given this random string. On Bob's side, we consider a partition of $\{1, \ldots, m\}$ into $n^{1/2+o(1)}$ tuples of $n^{1/2+o(1)}$ indices each. Each player on Bob's side assigns values to one such tuple.

On each two-player subgame, the player on Alice's side is incentivized to imitate the assignment of the player on Bob's side on the few coordinates where they intersect. The player on Bob's side, uses Alice's strategy to locally compute $f_j\left(\mathbf{x}^{(\mathcal{A})}\right)$ on a few j's in his $\left(n^{1/2+o(1)}\right)$-tuple of coordinates. This computation may be inaccurate, but we can guarantee that for most coordinates it is approximately correct most of the time.

From polymatrix to bimatrix

The final reduction from the polymatrix game to two-player game follows more or less from known techniques for hardness of Nash equilibria [4], [32], [15]. We let each of Alice and Bob control one side of the bipartite polymatrix game. In particular, each strategy in the two-player game corresponds to picking a player of the polymatrix game, and a strategy for that player. We add a gadget due to Althofer [4] to guarantee that Alice and Bob mix approximately uniformly across all their players. See full version for details.

B. Results for multiplayer relaxations of Nash equilibrium

Our hardness for norm-2 approximate Brouwer fixed point (Theorem I.4) has some important consequences for multiplayer games. All our results in this regime (as well as much of the existing literature) are inspired by a paper of Babichenko [13] and a blog post of Shmaya [64].

For multiplayer games, there are several interesting questions one can ask. First, note that the normal form representation of the game is exponential in the number of players, so it is difficult to talk about computational complexity. To alleviate this, different restricted classes of multiplayer games have been studied. We have already seen *polymatrix games* (Definition I.6), which have a succinct description in terms of the normal forms of the $\binom{n}{2}$ bimatrix subgames. Another interesting class is *graphical games* where we are given a (low-degree) graph over the players, and each player's utility is only affected by the actions of its neighbors. The graph of the game constructed in Proposition I.7 has a high degree, but it is important that it is bipartite, as are all the games described below. Most generally, we can talk about the class of *succinct games*, which are described via a circuit that computes any entry of the payoff tensors (this includes polymatrix and graphical games). Finally, there has been recent significant progress on the *query complexity* of finding approximate Nash equilibria in arbitrary n-player games where the payoff tensors are given via a black-box oracle [14], [43], [13], [28].

There are also a few different notions of approximation of Nash equilibrium. The strictest notion, *ϵ-Well-Supported Nash Equilibrium*, requires that for *every action* in the support of *every player*, the expected utility, given other players' mixed strategies, is within (additive) ϵ of the optimal strategy for that player. For this notion, Babichenko [13] showed a $2^{\Omega(n)}$ lower bound on query complexity for any (possibly randomized) algorithm. In followup work, [60] showed PPAD-completeness for succinct games, and soon after [62] extended this PPAD-completeness to games that are both polymatrix degree-3-graphical.

The most central model in the literature, *ϵ-Approximate Nash Equilibrium*, requires that *every player's* expected utility from her mixed strategy is within ϵ of the optimum she can achieve (given other players' strategies). I.e., any player is allowed to assign a small probability to poor strategies, as long as in expectation she does well. The last PPAD-completeness result extends immediately to this model (the two notions of approximation are equivalent, up to constant factors, for constant degree graphical games). For query complexity, Hart and Nisan [43] and Babichenko [13] asked whether the latter's exponential lower bound can be extended to ϵ-Approximate Nash Equilibrium. Very recently Chen et al. [28] solved it almost entirely, showing a lower bound of $2^{\Omega(n/\log n)}$; and they asked whether the $\Theta(\log n)$ gap in the exponent can be resolved. Here, we obtain a tight $2^{\Omega(n)}$ lower bound on the query complexity, as well as stronger inapproximability guarantees.

Finally, the most lenient notion is that of *(ϵ, δ)-WeakNash*: it only requires that a $(1-\delta)$-fraction of the players play ϵ-optimally ("Can almost everybody be almost happy?"). This notion was recently defined in [15] who conjectured that it is also PPAD-complete for polymatrix, graphical games. Their conjecture remains an interesting open problem (see Subsection I-C). Here, as a consequence of Theorem I.4, we prove that (ϵ, δ)-WeakNash is PPAD-complete for the more general class of succinct games.

Corollary I.8. *There exist constants $\epsilon, \delta > 0$, such that finding an (ϵ, δ)-WeakNash is* PPAD*-hard for succinct multiplayer games where each player has two actions.*

Furthermore, as we hinted earlier, our proof also extends to giving truly exponential lower bounds on the query complexity:

Corollary I.9. *There exist constants $\epsilon, \delta > 0$, such that any (potentially randomized) algorithm for finding an (ϵ, δ)-WeakNash for multiplayer games where each player has two actions requires $2^{\Omega(n)}$ queries to the players' payoffs tensors.*

C. The PCP Conjecture for PPAD

Rather than posing new open problems, let us restate the following conjecture due to [15]:

Conjecture I.10 (PCP for PPAD; [15])**.** *There exist constants $\epsilon, \delta > 0$ such that finding an (ϵ, δ)-WeakNash in a bipartite, degree three polymatrix game with two actions per player is* PPAD*-complete.*

The main original motivation was an approach to prove our main theorem given this conjecture. As pointed out by [15], it turns out that resolving this conjecture would also have interesting consequences for relative approximations of two-player Nash equilibrium, as well as applications to inapproximability of market equilibrium.

More importantly, this question is interesting in its own right: how far can we extend the ideas from the PCP Theorem (for NP) to the wold of PPAD? The $\mathsf{PCP}[r(n), q(n)]$ characterization [8] is mainly concerned with two parameters: $r(n)$, the number of random bits, and $q(n)$, the number of bits read from the proof. A major tool in all proofs of the PCP Theorem is *verifier composition*: in the work of Polishchuk and Spielman [59], [65], for example, it is first shown that $\mathsf{NP} \subseteq \mathsf{PCP}\left[O(\log n), n^{1/2+o(1)}\right]$, and then via composition it is eventually obtained that $\mathsf{NP} = \mathsf{PCP}[O(\log n), O(1)]$. In some **informal sense**, one may think of our main technical result as something analogous[5] to $\mathsf{PPAD} \subseteq \mathsf{PCP}\left[(1/2+o(1))\log_2 n, n^{1/2+o(1)}\right]$. Furthermore, our techniques build on many existing ideas from the PCP literature [12], [59], [65], [21], [20] that have been used to show similar statements for NP. It is thus natural to ask: is there a sense in which our "verifier" can be composed? can such composition eventually resolve the PCP Conjecture for PPAD?

More generally, some of the tools we use here, even as simple as error correcting codes, have been the basic building blocks in hardness of approximation for decades, yet to the best of our knowledge have not been used before for any problem in PPAD. We hope to see other applications of similar ideas in this regime[6].

D. Additional related work

Aside from [15], previous attempts to show lower bounds for approximate Nash in two player games have mostly focused on limited models of computation [35] and lower bounding the support required for obtaining approximate equilibria [4], [39], [6], [5] (in contrast, [54]'s algorithm runs

[5]We stress that our analogy is very loose. For example, we are not aware of any formal extension of PCP to function problems, and it is well known that $\mathsf{NP} \subseteq \mathsf{PCP}\left[(1/2+o(1))\log_2 n, n^{1/2+o(1)}\right]$.

[6]In fact, in the few months since our paper first appeared, our techniques already found applications for lower bounds on the *communication complexity* of Nash equilibrium [16].

in quasi-polynomial time because there exist approximate equilibria with support size at most $O\left(\frac{\log n}{\epsilon^2}\right)$).

Birthday repetition and related quasi-polynomial lower bounds: Hazan and Krauthgamer [44] showed that finding an ϵ-Approximate Nash Equilibrium with ϵ-optimal welfare is as hard as the PLANTED-CLIQUE problem; Austrin et al. [9] later showed that the optimal-welfare constraint can be replaced by other decision problems. Braverman et al. [25] recently showed that the hardness PLANTED-CLIQUE can be replaced by the Exponential Time Hypothesis, the NP-analog of the ETH for PPAD we use here. The work of Braverman et al, together with an earlier paper by Aaronson et al. [1] inspired a line of works on quasi-polynomial hardness results via the technique of "birthday repetition" [15], [24], [61], [22]. In particular [15] investigated whether birthday repetition can give quasi-polynomial hardness for finding any ϵ-Approximate Nash Equilibrium (our main theorem). As we discussed in Subsection I-C the main obstacle is that we don't have a PPAD-analogue for the PCP Theorem.

Multiplicative hardness of approximation: Daskalakis [31] and our recent work [60] show that finding an ϵ-relative Well-Supported Nash Equilibrium in two-player games is PPAD-hard. The case of ϵ-relative Approximate Nash Equilibrium is still open: our main theorem implies that it requires at least quasi-polynomial time, but it is not known whether it is PPAD-hard, or even if it requires a large support (see also discussion in [15]).

Approximation algorithms: The state of the art for games with arbitrary payoffs is ≈ 0.339 for two-player games due to Tsaknakis and Spirakis [67] and $0.5 + \epsilon$ for polymatrix games due to Deligkas et al. [37]. For two-player games, PTAS have been given for the special cases of constant rank games by Kannan and Theobald [48], small-probability games by Daskalakis and Papadimitriou [35], positive semi-definite games by Alon et al. [3], and sparse games by Barman [17]. For games with many players and a constant number of strategies, PTAS were given for the special cases of anonymous games by Daskalakis and Papadimitriou [36] and polymatrix games on a tree by Barman et al. [18]. Finally, let us return to the more general class of succinct n-player games, and mention an approximation algorithm due to Goldberg and Roth [40]; their algorithm runs in exponential time, but uses only a polynomial number of oracle queries.

Communication complexity: The hard instance of Brouwer we construct here (Theorem I.4) has already been useful in followup work [16], for proving lower bounds on the communication complexity of approximate Nash equilibrium in $N \times N$ two-player games, as well as binary-action n-player games.

ACKNOWLEDGMENT

I am grateful to Yakov Babichenko, Jonah Brown-Cohen, Karthik C.S., Alessandro Chiesa, Elad Haramaty, Christos Papadimitriou, Muli Safra, Luca Trevisan, Michael Viderman, and anonymous reviewers for inspiring discussions and suggestions. This research was supported by Microsoft Research PhD Fellowship. It was also supported in part by NSF grant CCF1408635 and by Templeton Foundation grant 3966. This work was done in part at the Simons Institute for the Theory of Computing.

REFERENCES

[1] Scott Aaronson, Russell Impagliazzo, and Dana Moshkovitz. Am with multiple merlins. In *Computational Complexity (CCC), 2014 IEEE 29th Conference on*, pages 44–55. IEEE, 2014.

[2] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[3] Noga Alon, Troy Lee, Adi Shraibman, and Santosh Vempala. The approximate rank of a matrix and its algorithmic applications: approximate rank. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 675–684, 2013.

[4] Ingo Althofer. On sparse approximations to randomized strategies and convex combinations. 1993.

[5] Yogesh Anbalagan, Hao Huang, Shachar Lovett, Sergey Norin, Adrian Vetta, and Hehui Wu. Large supports are required for well-supported nash equilibria. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, pages 78–84, 2015.

[6] Yogesh Anbalagan, Sergey Norin, Rahul Savani, and Adrian Vetta. Polylogarithmic supports are required for approximate well-supported nash equilibria below 2/3. In *Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings*, pages 15–23, 2013.

[7] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[8] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[9] Per Austrin, Mark Braverman, and Eden Chlamtac. Inapproximability of np-complete variants of nash equilibrium. *Theory of Computing*, 9:117–142, 2013.

[10] Yossi Azar, Rajeev Motwani, and Joseph Naor. Approximating probability distributions using small sample spaces. *Combinatorica*, 18(2):151–171, 1998.

[11] László Babai and Lance Fortnow. Arithmetization: A new method in structural complexity theory. *Computational Complexity*, 1:41–66, 1991.

[12] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 21–31, 1991.

[13] Yakov Babichenko. Query complexity of approximate nash equilibria. In *STOC*, pages 535–544, 2014.

[14] Yakov Babichenko and Siddharth Barman. Query complexity of correlated equilibrium. *ACM Trans. Economics and Comput.*, 3(4):22, 2015.

[15] Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinstein. Can almost everybody be almost happy? In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 1–9, 2016.

[16] Yakov Babichenko and Aviad Rubinstein. Communication complexity of approximate Nash equilibria. In preparation.

[17] Siddharth Barman. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory's theorem. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 361–369, 2015.

[18] Siddharth Barman, Katrina Ligett, and Georgios Piliouras. Approximating nash equilibria in tree polymatrix games. In *Algorithmic Game Theory - 8th International Symposium, SAGT 2015, Saarbrücken, Germany, September 28-30, 2015, Proceedings*, pages 285–296, 2015.

[19] Paul Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *J. Comput. Syst. Sci.*, 57(1):3–19, 1998.

[20] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006.

[21] Eli Ben-Sasson, Madhu Sudan, Salil P. Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 612–621, 2003.

[22] Umang Bhaskar, Yu Cheng, Young Kun Ko, and Chaitanya Swamy. Near-optimal hardness results for signaling in bayesian games. *CoRR*, abs/1512.03543, 2015.

[23] Hartwig Bosse, Jaroslaw Byrka, and Evangelos Markakis. New algorithms for approximate nash equilibria in bimatrix games. *Theor. Comput. Sci.*, 411(1):164–173, 2010.

[24] Mark Braverman, Young Kun-Ko, Aviad Rubinstein, and Omri Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. *CoRR*, abs/1504.08352, 2015.

[25] Mark Braverman, Young Kun-Ko, and Omri Weinstein. Approximating the best nash equilibrium in $n^{o(\log n)}$-time breaks the exponential time hypothesis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 970–982, 2015.

[26] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, pages 75–85, 2009.

[27] Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270, 2016.

[28] Xi Chen, Yu Cheng, and Bo Tang. Well-supported versus approximate nash equilibria: Query complexity of large games. *CoRR*, abs/1511.00785, 2015.

[29] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *J. ACM*, 56(3), 2009.

[30] Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. Known algorithms for edge clique cover are probably optimal. *SIAM J. Comput.*, 45(1):67–83, 2016.

[31] Constantinos Daskalakis. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms*, 9(3):23, 2013.

[32] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *Commun. ACM*, 52(2):89–97, 2009.

[33] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. Progress in approximate nash equilibria. In *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*, pages 355–358, 2007.

[34] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. A note on approximate nash equilibria. *Theor. Comput. Sci.*, 410(17):1581–1588, 2009.

[35] Constantinos Daskalakis and Christos H. Papadimitriou. On oblivious ptas's for nash equilibrium. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 75–84, 2009. Full version available at http://arxiv.org/abs/1102.2280.

[36] Constantinos Daskalakis and Christos H. Papadimitriou. Approximate nash equilibria in anonymous games. *J. Economic Theory*, 156:207–245, 2015.

[37] Argyrios Deligkas, John Fearnley, Rahul Savani, and Paul G. Spirakis. Computing Approximate Nash Equilibria in Polymatrix Games. In *Web and Internet Economics - 10th International Conference, WINE 2014, Beijing, China, December 14-17, 2014. Proceedings*, pages 58–71, 2014.

[38] Irit Dinur and Prahladh Harsha. Composition of low-error 2-query pcps using decodable pcps. *SIAM J. Comput.*, 42(6):2452–2486, 2013.

[39] Tomás Feder, Hamid Nazerzadeh, and Amin Saberi. Approximating nash equilibria using small-support strategies. In *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*, pages 352–354, 2007.

[40] Paul W. Goldberg and Aaron Roth. Bounds for the query complexity of approximate equilibria. In *EC*, pages 639–656, 2014.

[41] Oded Goldreich, editor. *Property Testing - Current Research and Surveys [outgrow of a workshop at the Institute for Computer Science (ITCS) at Tsinghua University, January 2010]*, volume 6390 of *Lecture Notes in Computer Science*. Springer, 2010.

[42] Venkatesan Guruswami and Atri Rudra. Tolerant locally testable codes. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 306–317, 2005.

[43] Sergiu Hart and Noam Nisan. The query complexity of correlated equilibria. *CoRR*, abs/1305.4874, 2013.

[44] Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best nash equilibrium? *SIAM J. Comput.*, 40(1):79–91, 2011.

[45] Michael D. Hirsch, Christos H. Papadimitriou, and Stephen A. Vavasis. Exponential lower bounds for finding brouwer fix points. *J. Complexity*, 5(4):379–416, 1989.

[46] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[47] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[48] Ravi Kannan and Thorsten Theobald. Games of fixed rank: a hierarchy of bimatrix games. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1124–1132, 2007.

[49] Marek Karpinski and Warren Schudy. Faster algorithms for feedback arc set tournament, kemeny rank aggregation and betweenness tournament. In *Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*, pages 3–14, 2010.

[50] Michael Kearns. *Graphical games*, chapter 7, pages 159–180. Cambridge University Press, 2007.

[51] Spyros C. Kontogiannis, Panagiota N. Panagopoulou, and Paul G. Spirakis. Polynomial algorithms for approximating nash equilibria of bimatrix games. *Theor. Comput. Sci.*, 410(17):1599–1606, 2009.

[52] F. Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Array, Trees, Hypercubes*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[53] C. E. Lemke and J. T. Howson. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423, 1964.

[54] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *EC*, pages 36–41, 2003.

[55] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

[56] Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010.

[57] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994.

[58] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006.

[59] Alexander Polishchuk and Daniel A. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 194–203, 1994.

[60] Aviad Rubinstein. Computational complexity of approximate nash equilibrium in large games. *CoRR*, abs/1405.0524, 2014.

[61] Aviad Rubinstein. Eth-hardness for signaling in symmetric zero-sum games. *CoRR*, abs/1510.04991, 2015.

[62] Aviad Rubinstein. Inapproximability of nash equilibrium. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 409–418, 2015.

[63] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[64] Eran Shmaya. Brouwer Implies Nash Implies Brouwer. http://theoryclass.wordpress.com/2012/01/05/brouwer-implies-nash-implies-brouwer/, 2012.

[65] Daniel A. Spielman. *Computationally Efficient Error-Correcting Codes and Holographic Proofs*. PhD thesis, Massachusetts Institute of Technology, 1995.

[66] Daniel A. Spielman. The complexity of error-correcting codes. In *Fundamentals of Computation Theory, 11th International Symposium, FCT '97, Kraków, Poland, September 1-3, 1997, Proceedings*, pages 67–84, 1997.

[67] Haralampos Tsaknakis and Paul G. Spirakis. An optimization approach for approximate nash equilibria. *Internet Mathematics*, 5(4):365–382, 2008.

[68] Michael Viderman. A combination of testability and decodability by tensor products. *Random Struct. Algorithms*, 46(3):572–598, 2015.