Overview: Computing Nash Equilibria: Approximation and Smoothed Complexity.

Jinchuan Sun

Beijing Institute of Technology
1120212368@bit.edu.cn

Abstract. This paper [1] is situated within the field of the approximation of Nash equilibria in bimatrix games. The main focus is on the PPAD-completeness of computing a $1/n^{\Theta(1)}$-approximate Nash equilibrium, suggesting a fully polynomial-time approximation scheme may not exist unless PPAD $\subseteq$ P. I will present the primary results with proofs, and recast the author's concepts based on my comprehension of certain sections of the paper.

Keywords: Nash Equilibrium Approximation · PPAD-completeness.

1 Introduction

The complexity of the Nash equilibrium and its approximations have long been a topic of intense study in algorithmic game theory. This complexity pertains particularly to the bimatrix game, where two players select strategies simultaneously without knowledge of the other's selection, each with their own payoffs.

In this paper [1], Chen Xi et al. delved into the complexity of computing Nash equilibria in two-person games, specifically focusing on normalized $n \times n$ bimatrix games. They proved that calculating a $1/n^6$-approximate Nash equilibrium is a PPAD-complete problem, a complexity class pertaining to certain problems of finding pure Nash equilibria.

Their research also identified a complexity gap in the approximation of Nash equilibria of bimatrix games. This gap was established between a previous result by Lipton, Markakis, and Mehta [2] that shows an ϵ -approximate Nash equilibrium can be found in $n^{O\left(\log n/\epsilon^2\right)}$-time. Chen Xi et al. found that calculating an $O\left(1/n^{\Theta(1)}\right)$-approximate Nash equilibrium is PPAD-complete. However, the findings do not provide a complete understanding of this complexity gap, especially when ϵ is an absolute constant between 0 and 1. They further explored this by considering the possibility of a fully polynomial-time approximation scheme for the bimatrix game, which seems unlikely given the results.

This report is organized into three sections. Section 2 provides a fundamental understanding, explaining essential definitions and discussing the primary findings from [1]. Section 3 illuminates the pivotal ideas underlying the authors' proof. Given the complexity of the complete proof, only a few brilliant aspects are presented, which implies potential inaccuracies.

2 Preliminaries

In this segment, I will initially introduce the fundamental understanding of the paper within a specific context. Additionally, I'll provide some universal definitions. Following this, I'll demonstrate the primary findings related to the approximation of Nash equilibria.

2.1 Basic Concepts

The paper argues that finding a fully polynomial-time approximation scheme for this problem is likely infeasible due to the complexity class of the problem (PPAD-complete). The significance of the result lies in the relationship between PPAD and P, two complexity classes. The paper's proof relies heavily on the introduction of a new discrete fixed-point problem on a high-dimensional cube with a constant side-length, such as an n-dimensional cube with side-length 7.

In the context of the Lemke-Howson algorithm, or any other algorithm for computing a Nash equilibrium of a bimatrix game, the paper tackles the problem of smoothed complexity under perturbations. The result states that it's improbable for the smoothed complexity to be polynomial in n and $1/\sigma$ under perturbations with magnitude σ, unless PPAD is a subset of RP.

PPAD-completeness: The full name of PPAD is Polynomial Parity Arguments on Directed graphs, which is a complexity class, a subset of TFNP (Total Function Nondeterministic Polynomial), that contains computational problems for which it is known that a solution exists and such solution can be found or verified in polynomial time, but no efficient algorithm is known for finding a solution.

PPAD-Completeness is a term used to describe a problem that is in the PPAD class, and for which it is believed that if an efficient (polynomial time) algorithm can solve it, then an efficient algorithm can solve all problems in the PPAD class. One of the most well-known PPAD-complete problems is the problem of computing a Nash equilibrium in a game. The complexity of finding such an equilibrium has been a topic of much research and the result that it is PPAD-complete provides evidence that it is unlikely that there is a polynomial time algorithm for this problem, even though a solution is guaranteed to exist by Nash's Theorem.

Smoothed Analysis: Smoothed analysis is an innovative method used to measure the efficiency of algorithms when dealing with inputs subject to small random perturbations. In a worst-case analysis, we evaluate the performance of an algorithm based on the most difficult inputs of a given size. In contrast, in an average-case analysis, we consider the performance over all possible inputs, each of which is equally likely. Smoothed analysis lies between these two extremes: it evaluates the expected performance of an algorithm over small random perturbations of worst-case inputs.

Problem BROUWER^f: One of the most important PPAD problems concerns the task of search for a discrete Brouwer's fixed point. In this segment, we define this class of search problems in various dimensions. A discrete version of the Brouwer's fixed point theorem provides key to obtain our main results. In order to handle the exponential exploration of vertices in high dimensional cube, we need to define an efficient discrete version. We will introduce some notations - some elementary and some geometric - to facilitate our discussion.

To define our high dimensional Brouwer's fixed point problems, we need a notion of wellbehaved functions (please note that this is not the function for the fixed point problem) to parameterize the shape of the search space. We say an integer function $f(n)$ is well-behaved if it is polynomial-time computable and there exists a constant n_0 such that $3 \leq f(n) \leq n/2$ for every integer $n \geq n_0$.

For example, $f_1(n) = 3, f_2(n) = \lfloor n/2 \rfloor, f_3(n) = \lfloor n/3 \rfloor$ and $f_4(n) = \lfloor \log n \rfloor$ are all wellbehaved functions. For each $\mathbf{p} \in \mathbb{Z}^d$, let $K_{\mathbf{p}}$ denote the following unit hypercube incident to $\mathbf{p}$, that is,

$$K_{\mathbf{p}} = \left\{ \mathbf{q} \in \mathbb{Z}^d \mid q_i = p_i \text{ or } p_i + 1, \forall 1 \leq i \leq d. \right\}$$

For a positive integer d and a vector $\mathbf{r} \in \mathbb{Z}_+^d$, let

$$A_{\mathbf{r}}^d = \left\{ \mathbf{p} \in \mathbb{Z}^d \mid 0 \leq p_i \leq r_i - 1, \forall 1 \leq i \leq d \right.$$

be the hyper-grid with side length given by $\mathbf{r}$. The boundary of $A_{\mathbf{r}}^d, \partial\left(A_{\mathbf{r}}^d\right)$, is then the set of integer lattice points $\mathbf{p} \in A_{\mathbf{r}}^d$ with $p_i \in \{0, r_i - 1\}$ for some $i : 1 \leq i \leq d$.

For each $\mathbf{r} \in \mathbb{Z}_+^d$, let $\text{Size}[\mathbf{r}] = \sum_{1 \leq i \leq d} \lceil \log(r_i + 1) \rceil$ denote the number of bits necessary and sufficient to represent $\mathbf{r}$.

2.2 Definitions

Definition 1 (Polynomial Reduction). *A search problem $Q_{R_1} \in$ **TFNP** is* polynomial-time reducible *to another $Q_{R_2} \in$ **TFNP** if there exists a pair of polynomial-time computable functions (f, g) such that, for every input x of R_1, if y satisfies $(f(x), y) \in R_2$, then $(x, g(y)) \in R_1$. Q_{R_1} and Q_{R_2} are then polynomial-time equivalent (or simply, equivalent) if Q_{R_2} is also reducible to Q_{R_1}.*

A search problem Q_{R_1} is said to be polynomial-time reducible to another search problem Q_{R_2} if there exist two polynomial-time computable functions, f and g, which transform an input x from Q_{R_1}'s domain into a form suitable for Q_{R_2}, and the solution y from Q_{R_2} back into a form suitable for Q_{R_1}, respectively. More specifically, if we can find a solution y for Q_{R_2} from an input f(x), then there's a corresponding solution g(y) for the original problem Q_{R_1} with input x. If this reduction is possible in both directions, then the two problems Q_{R_1} and Q_{R_2} are considered to be polynomial-time equivalent.

Definition 2 (LEAFD)**.** *The input instance of* LEAFD *is a pair $(M, 0^n)$ where M is the description of a polynomial-time Turing machine satisfying the following two conditions:*

1. *for every* $v \in \{0,1\}^n$, $M(v)$ *is an ordered pair* (u_1, u_2) *where* $u_1, u_2 \in \{0,1\}^n \cup \{\text{"no"}\}$.

2. $M(0^n) = (\text{"no"}, 1^n)$ *and the first component of* $M(1^n)$ *is* 0^n.

This instance defines a directed graph $G = (V, E)$ *where* $V = \{0,1\}^n$ *and a pair* $(u, v) \in E$ *if and only if* v *is the second component of* $M(u)$ *and* u *is the first component of* $M(v)$.

The output of this problem is a directed leaf of G *other than* 0^n*, where a vertex of* V *is a* directed leaf *if its out-degree plus in-degree equals one.*

The output of this problem is a directed leaf of G other than 0^n, where a vertex of V is a directed leaf if its out-degree plus in-degree equals one. Simply from its definition, LEAFD is complete for PPAD.

Definition 3 (Brouwer-Mapping Circuit). *For a positive integer* d *and* $\mathbf{r} \in \mathbb{Z}_+^d$*, a Boolean circuit* C *is a Brouwer-mapping circuit with parameters* d *and* $\mathbf{r}$ *if it has exactly Size* $[\mathbf{r}]$ *input bits and* $2d$ *output bits* $\Delta_1^+, \Delta_1^- \ldots \Delta_d^+, \Delta_d^-$.

Moreover, C *is a valid Brouwer-mapping circuit if*

- *for every* $\mathbf{p} \in A_{\mathbf{r}}^d$*, the set of* $2d$ *output bits evaluated at* $\mathbf{p}$ *falls into one of the following cases:*
  - *case* $1 : \Delta_1^+ = 1$ *and all other bits are 0;*
  - *...*
  - *case* $d : \Delta_d^+ = 1$ *and all other bits are 0;*
  - *case* $d+1$: *for every* $i : 1 \le i \le d, \Delta_i^+ = 0$ *and* $\Delta_i^- = 1$.
- *for every* $\mathbf{p} \in \partial\left(A_{\mathbf{r}}^d\right)$*, if there exists an* $i : 1 \le i \le d$ *such that* $p_i = 0$*, letting* $i_{\max} = \max\{i \mid p_i = 0\}$*, then the output bits satisfy the* $i_{\max}^{th}$ *case, otherwise, (when none of the* p_i *is 0 and some are* $r_i - 1$*), the output bits satisfy the* $d + 1^{st}$ *case.*

The validity of this type of circuit is contingent upon certain conditions about the arrangement of $1s$ and $0s$ in its output, based on the input from either a set A_r^d or its boundary.

Definition 4 (Brouwer Color Assignment and Panchromatic Simplex). *Suppose* C *is a valid Brouwer-mapping circuit with parameter* d *and* $\mathbf{r}$*. Circuit* C *defines a color assignment:* $Color_C : A_{\mathrm{r}}^d \to \{1, 2, \ldots d, \text{"red"}\}$*, where "red" is a special color, according to the following rule: For each point* $\mathbf{p} \in A_{\mathbf{r}}^d$*, if the set of output bits of* C *evaluated at* $\mathbf{p}$ *satisfies the* i^{th} *case where* $1 \le i \le d$*, then* $Color_C[\mathbf{p}] = i$*; otherwise, the output bits of* C *satisfy the* $d + 1^{th}$ *case and* $Color_C[\mathbf{p}] = \text{"red"}$.

A subset $P \subset A_{\mathbf{r}}^d$ *is accommodated if there exists a point* $\mathbf{p} \in A_{\mathbf{r}}^d$ *such that* $P \subset K_{\mathbf{p}}$*.A set* $P \subset A_{\mathrm{r}}^d$ *is a panchromatic set or a panchromatic simplex of* C *if it is an accommodated set of* $d + 1$ *points assigned with all* $d + 1$ *colors.*

A set $P \subset A_{\mathbf{r}}^d$ is referred to as a panchromatic set or a panchromatic simplex of C if it's an accommodated set of $d + 1$ points, and each of these points is assigned with a unique color from the $d + 1$ colors, including d colors from 1 to

d and the special color "red".

Definition 5 (BROUWER^f). *For a well-behaved function f and a parameter n, let $m = f(n)$ and $d = \lceil n/f(n) \rceil$. An input instance of* BROUWER^f *is a pair $(C, 0^n)$ where C is a valid Brouwer-mapping circuit*[5] *with parameter d and $\mathbf{r}$ where $r_i = 2^m, \forall i : 1 \leq i \leq d$. The output of this search problem is then a panchromatic simplex of C.*

Definition 6 (ϵ-well-supported Nash Equilibria). An ϵ-well-supported Nash equilibrium of a bimatrix game $(\mathbf{A}, \mathbf{B})$ is a profile of mixed strategies $(\mathbf{x}^*, \mathbf{y}^*)$, such that for all $1 \leq i, j \leq n$,

$$\langle \mathbf{b}_i \mid \mathbf{x}^* \rangle > \langle \mathbf{b}_j \mid \mathbf{x}^* \rangle + \epsilon \Rightarrow y_j^* = 0 \quad \textit{and} \quad \langle \mathbf{a}_i \mid \mathbf{y}^* \rangle > \langle \mathbf{a}_j \mid \mathbf{y}^* \rangle + \epsilon \Rightarrow x_j^* = 0.$$

Definition 7 (ϵ-approximate Nash equilibria). An ϵ-approximate Nash equilibrium of game $(\mathbf{A}, \mathbf{B})$ is a profile of mixed strategies $(\mathbf{x}^*, \mathbf{y}^*)$, such that for all probability vectors $\mathbf{x}, \mathbf{y} \in \mathbb{P}^n$

$$(\mathbf{x}^*)^T \mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T \mathbf{A}\mathbf{y}^* - \epsilon \quad \textit{and} \quad (\mathbf{x}^*)^T \mathbf{B}\mathbf{y}^* \geq (\mathbf{x}^*)^T \mathbf{B}\mathbf{y} - \epsilon$$

The authors showed that these two notions are polynomially related. This polynomial relation allows us to prove the PPAD result with a pair-wise approximation condition. Thus, we can locally argue certain properties of the bimatrix game that built from the fixed-point problem.

2.3 Main Results

The key findings of the paper include five theorems that address the approximation and smoothed complexity in computing Nash equilibria. Here's an overview of these theorems:

Theorem 1 (High Dimensional Brouwer's Fixed Points). *For any well-behaved function f, search problem* BROUWER^f *is* **PPAD**-*complete.*

Theorem 2 (Main). *The problem of computing a $1/n^6$-well-supported Nash equilibrium of a positively normalized $n \times n$ bimatrix game is* **PPAD**-*complete.*

Theorem 3 (Unlikely Fully Polynomial-Time Approximation). *The problem of computing a $1/n^{\Theta(1)}$-approximate Nash equilibrium of a positively normalized $n \times n$ bimatrix game is* **PPAD**-*complete.*

Theorem 4 (Hardness of Smoothed Bimatrix Games). *It is unlikely that the problem of computing a Nash equilibrium of a bimatrix game is in smoothed polynomial time, under uniform or Gaussian perturbations, unless* **PPAD** $\subset$ **RP**.

Theorem 5 (Smoothed Complexity of Lemke-Howson). *It is unlikely that the LemkeHowson algorithm has a polynomial smoothed complexity (in n and $1/\sigma$) under σ-uniform or σ-Gaussian perturbations, unless* $\mathbf{PPAD} \subset \mathbf{RP}$.

Here are the main corollary and conjectures:

Corollary 1. *For any constant $\delta > 0$, the problem of computing a $1/n^{\delta}$-approximate Nash equilibrium of a normalized $n \times n$ bimatrix game is* **PPAD-** *complete.*

Conjecture 1 (Smoothed 2-NASH Conjecture). *The problem of finding a Nash equilibrium of a bimatrix game can be solved in smoothed time polynomial in n and $1/\sigma$, under uniform perturbations and Gaussian perturbations with magnitude σ for all $0 < \sigma < 1$.*

Conjecture 2 (PTAS Approximate NASH). *There is an algorithm to find an ϵ-approximate Nash equilibrium of an $n \times n$ bimatrix game in time $O\left(n^{k+\epsilon^{-c}}\right)$ for some positive constants c and k.*

Conjecture 3 (Smoothed 2-NASH: Constant Perturbations). *There is an algorithm to find a Nash equilibrium of an $n \times n$ bimatrix game with smoothed time complexity $O\left(n^{k+\sigma^{-c}}\right)$ under perturbations with magnitude σ, for some positive constants c and k.*

3 Lemmas and Proofs

Lemma 1. *For any well-behaved function f, search problem* $\mathrm{BROUWER}^{f}$ *is in* **PPAD**.

Lemma 2 $\left(\mathbf{L}^{1}(T, t, u)\right.$**: Padding a Dimension).** *Given a coloring triple $T = (C, d, \mathbf{r})$ and integers $1 \leq t \leq d$ and $u > r_t$, we can construct a new coloring triple $T' = (C', d, \mathbf{r}')$ that satisfies the following two conditions.*

A. *For all $i : 1 \leq i \neq t \leq d, r'_i = r_i$, and $r'_t = u$. In addition, there exists a polynomial $g_1(n)$ such that Size $[C'] =$ Size $[C] + O\left(g_1\left(\text{ Size}\left[\mathbf{r}'\right]\right)\right)$ and T' can be computed in time polynomial in Size $[C']$. We write $T' = \mathbf{L}^{1}(T, t, u)$.*
B. *From each panchromatic simplex P' of coloring triple T', we can compute a panchromatic simplex P of T in polynomial time.*

Proof. We define circuit C' by its color assignment in Figure 1 Property **A** is true according to this definition.

To show Property **B**, let $P' \subset K_{\mathrm{p}}$ be a panchromatic simplex of T'. We first note that $p_t \leq r_t - 1$, because had $p_t > r_t - 1, K_{\mathbf{p}}$ would not contain color t according to the color assignment, Thus, it follows from Color ${}_{C'}[\mathbf{q}] = \text{Color}_{C}[\mathbf{q}]$

Color$_{C'}$ [p] of a point $\mathbf{p} \in A^d_{\mathbf{r}'}$ assigned by $\mathbf{L}^1(T, t, u)$

```
if p ∈ ∂(A^d_r') then
    if there exists i such that p_i = 0 then
        Color_C' [p] = i_max = max{ i | p_i = 0 }
    else
        Color_C' [p] = red
else if p_t ≤ r_t then
    Color_C' [p] = Color_C [p]
else
    Color_C' [p] = red
```

Fig. 1. How $\mathbf{L}^1(T, t, u)$ extends the color assignment

for each $\mathbf{q} \in A^d_{\mathbf{r}}$ that P' is also be a panchromatic simplex of the coloring triple T.

Lemma 3 $\left(\mathbf{L}^2(T, u)\right.$ **: Adding a Dimension).** *Given a coloring triple $T = (C, d, \mathbf{r})$ and an integer $u \geq 7$, we can construct a new coloring triple $T' = (C', d+1, \mathbf{r}')$ that satisfies the following conditions.*

A. *For all $i : 1 \leq i \leq d, r'_i = r_i$, and $r'_{d+1} = u$. Moreover, there exists a polynomial $g_2(n)$ such that Size $[C'] =$ Size $[C] + O\left(g_2\left(\text{ Size}\left[\mathbf{r}'\right]\right)\right)$ and T' can be computed in time polynomial in Size $[C']$. We write $T' = \mathbf{L}^2(T, u)$.*
B. *From each panchromatic simplex P' of coloring triple T', we can compute a panchromatic simplex P of T in polynomial time.*

Color$_{C'}$ [p] of a point $\mathbf{p} \in A^{d+1}_{\mathbf{r}'}$ assigned $\mathbf{L}^2(T, u)$

```
if p ∈ ∂(A^d_r') then
    if there exists i such that p_i = 0 then
        Color_C' [p] = i_max = max{ i | p_i = 0 }
    else
        Color_C' [p] = red
else if p_{d+1} = 1 then
    Color_C' [p] = Color_C [p̄]
else
    Color_C' [p] = red
```

Fig. 2. How $\mathbf{L}^2(T, u)$ extends the color assignment

Proof. For each point $\mathbf{p} \in A^{d+1}_{\mathbf{r}'}$, we use $\overline{\mathbf{p}}$ to denote the point $\mathbf{q} \in A^d_{\mathbf{r}}$ with $q_i = p_i, \forall i : 1 \leq i \leq d$. The color assignment of circuit C' is given in Figure 2.

Clearly, Property A is true. To prove property **B**, let $P' \subset K_{\mathbf{p}}$ be a panchromatic simplex of T'. We note that $p_{d+1} = 0$, for otherwise, $K_{\mathbf{p}}$ does not contain color $d+1$. Note also Color $C_{C'}[\mathbf{q}'] = d+1$ for every $\mathbf{q} \in A_{\mathbf{r}'}^{d+1}$ with $q_{d+1} = 0$. Thus, for every $\mathbf{q} \in P'$ with Color $C_{C'}[\mathbf{q}] \neq d+1$, wehave $q_{d+1} = 1$. So, because $\mathrm{Color}_{C'}[\mathbf{q}] = \mathrm{Color}_{C}[\overline{\mathbf{q}}]$ for each $\mathbf{q} \in A_{\mathbf{r}'}^{d+1}$ with $q_{d+1} = 1$, $P = \{\overline{\mathbf{q}} \mid \mathbf{q} \in P'$ and $\mathrm{Color}_{C'}[\mathbf{p}] \neq d+1\}$ is a panchromatic simplex of coloring triple T.

Lemma 5 (Polynomially equivalence of the two notions of approximate Nash Equilibria).

1. *From any $\epsilon^2/(8n)$-approximate Nash equilibrium $(\mathbf{u}, \mathbf{v})$ of game $(\mathbf{A}, \mathbf{B})$, we can compute in polynomial time an ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $(\mathbf{A}, \mathbf{B})$.*
2. *For any $0 \leq \epsilon \leq 1$ and for any bimatrix game $(\mathbf{A}, \mathbf{B})$ where $\mathbf{A}, \mathbf{B} \in \mathbb{R}_{[0:1]}^{n \times n}$, if $(\mathbf{x}, \mathbf{y})$ is an ϵ-well-supported Nash equilibrium of $(\mathbf{A}, \mathbf{B})$, then $(\mathbf{x}, \mathbf{y})$ is also an ϵ-approximate Nash equilibrium of $(\mathbf{A}, \mathbf{B})$.*

Proof. By the definition of approximate Nash equilibria, we have

$$\begin{aligned}
\forall \mathbf{u}' \in \mathbb{P}^n,\ (\mathbf{u}')^T \mathbf{A}\mathbf{v} &\leq \mathbf{u}^T \mathbf{A}\mathbf{v} + \epsilon^2/(8n), \\
\forall \mathbf{v}' \in \mathbb{P}^n,\quad \mathbf{u}^T \mathbf{B}\mathbf{v}' &\leq \mathbf{u}^T \mathbf{B}\mathbf{v} + \epsilon^2/(8n).
\end{aligned}$$

Consider some j with some i such that $\langle \mathbf{a}_i \mid \mathbf{v} \rangle \geq \langle \mathbf{a}_j \mid \mathbf{v} \rangle + \epsilon/2$, where $\mathbf{a}_i$ is the i^{th} row of matrix A. By changing u_j to 0 and u_i to $u_i + u_j$ we can increase the first-player's profit by $u_j(\epsilon/2)$, implying $u_j < \epsilon/(4n)$. Similarly, all such j have $v_j < \epsilon/(4n)$.

We now set all these u_j and v_j to 0 and uniformly increase the probability of other strategies to obtain a new pair of mixed strategies, $(\mathbf{x}, \mathbf{y})$.

Note for all i, $|\langle \mathbf{a}_i \mid \mathbf{x} \rangle - \langle \mathbf{a}_i \mid \mathbf{u} \rangle| \leq \epsilon/4$, because we assume the absolute value of each entry in $\mathbf{a}_i$ is less then 1. Thus, the relative change between $\langle \mathbf{a}_i \mid \mathbf{x} \rangle$ and $\langle \mathbf{a}_j \mid \mathbf{x} \rangle$ is no more than $\epsilon/2$. Thus, any j that is beaten some i by a gap of ϵ is set to zero in $(\mathbf{x}, \mathbf{y})$.

References

1. Chen Xi, Xiaotie Deng, and Shang-Hua Teng. Computing Nash Equilibria: Approximation and Smoothed Complexity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 603-612, 2006.
2. Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *EC'03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 36–41, 2003.