Computing Nash Equilibria: Approximation and Smoothed Complexity

Xi Chen
Tsinghua University
Beijing, P.R.China
xichen00@mails.thu.edu.cn

Xiaotie Deng
City University of Hong Kong
Hong Kong, P.R.China
deng@cs.cityu.edu.hk

Shang-Hua Teng
Boston University and Akamai
Boston, USA
steng@cs.bu.edu

Abstract

We advance significantly beyond the recent progress on the algorithmic complexity of Nash equilibria by solving two major open problems in the approximation of Nash equilibria and in the smoothed analysis of algorithms.

- *We show that no algorithm with complexity* $\mathrm{poly}(n, \frac{1}{\epsilon})$ *can compute an* ϵ*-approximate Nash equilibrium in a two-player game, in which each player has* n *pure strategies, unless* **PPAD** $\subseteq$ **P**. *In other words, the problem of computing a Nash equilibrium in a two-player game does not have a fully polynomial-time approximation scheme unless* **PPAD** $\subseteq$ **P**.
- *We prove that no algorithm for computing a Nash equilibrium in a two-player game can have smoothed complexity* $\mathrm{poly}(n, \frac{1}{\sigma})$ *under input perturbation of magnitude* σ*, unless* **PPAD** $\subseteq$ **RP**. *In particular, the smoothed complexity of the classic Lemke-Howson algorithm is not polynomial unless* **PPAD** $\subseteq$ **RP**.

Instrumental to our proof, we introduce a new discrete fixed-point problem on a high-dimensional hypergrid with constant side-length, and show that it can host the embedding of the proof structure of any **PPAD** *problem. We prove a key geometric lemma for finding a discrete fixed-point, a new concept defined on* $n+1$ *vertices of a unit hypercube. This lemma enables us to overcome the curse of dimensionality in reasoning about fixed-points in high dimensions.*

1 Introduction

Shortly after Spielman and Teng [24] proved that the smoothed complexity of the simplex algorithm with shadow-vertex pivoting rule is polynomial, a number of people [1] asked them whether their analysis can be extended to another classic algorithm, the Lemke-Howson algorithm [15]. Indeed, whether the smoothed complexity of the Lemke-Howson algorithm is polynomial has been the question most frequently raised during talks on smoothed analysis.

The Lemke-Howson algorithm is the most popular method for computing a Nash equilibrium in a two-player game. It can be viewed as an extension of the simplex algorithm for linear programming and its worst-case complexity is also exponential [23]. As the simplex algorithm can be used to solve *zero-sum two-player games* [18], a special subclass of two-player games, it is natural to conjecture that the Lemke-Howson algorithm has smoothed polynomial complexity like the simplex algorithm. Earlier and in the same spirit, it was asked [21] whether the problem of finding a Nash equilibrium in a two-player game is in **P** as two-player zero-sum games and linear programming [13].

Motivated by the result of Bárány, Vempala, and Vetta [2] that random two-player games can be solved in polynomial time, the following conjecture was included in a recent survey on smoothed analysis [25]:

Smoothed 2-NASH Conjecture: *The problem of finding a Nash equilibrium in a two-player game is in smoothed polynomial time.*

A positive answer to this conjecture would provide an encouraging result to the computation of Nash equilibria.

However, so far, no polynomial smoothed analysis of the Lemke-Howson algorithm or any other algorithm for computing Nash equilibria in two-player games has been found.

In the final installment of a series of recent exciting developments initiated by Daskalakis, Goldberg and Papadimitriou [9], Chen and Deng [4] proved that `2-NASH`, the problem of computing a Nash equilibrium in a two-player game, is **PPAD**-complete. These results, despite unknowns about the **PPAD** complexity class [22], provide strong evidence that this search problem might be hard for **P**.

These developments inspired us to attempt disproving the Smoothed 2-NASH Conjecture. In this regard, we formulated a competing conjecture that, `2-NASH` is not in smoothed polynomial time unless **PPAD** $\subseteq$ **RP**. A connection between the smoothed complexity and approximation complexity of Nash equilibria ([25], Proposition 9.12) then led us to the conjecture: `2-NASH` is **PPAD**-hard to approximate in fully polynomial time.

By proving these two conjectures, we advance significantly beyond the recent progress on the algorithmic complexity of Nash equilibria. Not only do we answer the question about the smoothed complexity of the Lemke-Howson algorithm, but our investigation also enables us to settle an important open question about Nash equilibria by proving `2-NASH` *does not have a fully polynomial-time approximation scheme, unless* **PPAD** $\subseteq$ **P**.

Consequently, it is unlikely that the $n^{O(\log n/\epsilon^2)}$-time algorithm of Lipton, Markakis, and Mehta [16], the fastest algorithm known today for finding an ϵ-approximate Nash equilibrium, can be improved to poly$(n, 1/\epsilon)$. Also, it is unlikely that the average-case polynomial time result of [2] is extendible to the smoothed model.

Our results on approximation and smoothed complexity of Nash equilibria, together with those of [4, 9] on "exact" Nash equilibria, demonstrate computational difference between two-player games and their zero-sum specializations. These results may further encourage the study of the difference between local search and fixed-point computation.

1.1 Nash Equilibria of Two-Player Games

A *two-player game* (or *bimatrix game*) [20, 14, 15] is a non-cooperative game between two players in which the players have m and n choices of actions or pure strategies, respectively. The games can be specified by two $m \times n$ matrices $\mathbf{A} = (a_{i,j})$ and $\mathbf{B} = (b_{i,j})$. If the first player chooses action i and the second player chooses action j, then their payoffs are $a_{i,j}$ and $b_{i,j}$, respectively. A mixed strategy of a player is a probability distribution over its choices. The Nash Equilibrium Theorem [20, 19] on non-cooperative games when specialized to bimatrix games states that there exists a profile of possibly mixed strategies, called a *Nash equilibrium*, such that neither player can gain by changing his or her (mixed) strategy alone.

Let $\mathbb{P}^n$ denote the set of all *probability vectors* in $\mathbb{R}^n$, i.e., non-negative vectors whose entries sum to 1.

Mathematically, a profile of mixed strategies can be expressed by two column vectors $(\mathbf{x}^* \in \mathbb{P}^m, \mathbf{y}^* \in \mathbb{P}^n)$. A *Nash equilibrium* is then a pair $(\mathbf{x}^*, \mathbf{y}^*)$ such that for all $\mathbf{x} \in \mathbb{P}^m$ and $\mathbf{y} \in \mathbb{P}^n$,

$$(\mathbf{x}^*)^T\mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T\mathbf{A}\mathbf{y}^* \text{ and } (\mathbf{x}^*)^T\mathbf{B}\mathbf{y}^* \geq (\mathbf{x}^*)^T\mathbf{B}\mathbf{y}. \quad (1)$$

Computationally, one might settle with an approximate Nash equilibrium. An *ϵ-relatively-approximate Nash equilibrium* is a pair $(\mathbf{x}^*, \mathbf{y}^*)$ that satisfies

$$(\mathbf{x}^*)^T\mathbf{A}\mathbf{y}^* \geq (1-\epsilon)\mathbf{x}^T\mathbf{A}\mathbf{y}^* \qquad \text{and}$$
$$(\mathbf{x}^*)^T\mathbf{B}\mathbf{y}^* \geq (1-\epsilon)(\mathbf{x}^*)^T\mathbf{B}\mathbf{y},$$

for all $\mathbf{x} \in \mathbb{P}^m, \mathbf{y} \in \mathbb{P}^n$, while an *$\epsilon$-approximate Nash equilibrium* is a pair $(\mathbf{x}^*, \mathbf{y}^*)$ that satisfies

$$(\mathbf{x}^*)^T\mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T\mathbf{A}\mathbf{y}^* - \epsilon \qquad \text{and}$$
$$(\mathbf{x}^*)^T\mathbf{B}\mathbf{y}^* \geq (\mathbf{x}^*)^T\mathbf{B}\mathbf{y} - \epsilon.$$

The zero-sum two-player game [18] is a two-player game $(\mathbf{A}, \mathbf{B})$ with $\mathbf{B} = -\mathbf{A}$. It can be formulated as a linear program and can be solved in (weakly) polynomial time [13].

1.2 Our Results and Approaches

Our main complexity-theoretic results are:

- It is **PPAD**-complete to compute a $1/n^c$-approximate Nash equilibrium of an $n \times n$ normalized[1] two-player game, for any fixed $c > 0$. Thus, no algorithm can compute an ϵ-approximate Nash equilibrium in time poly$(n, 1/\epsilon)$ for all ϵ, unless **PPAD** $\subseteq$ **P**.
- The smoothed complexity of the Lemke-Howson algorithm or in fact of any algorithm for bimatrix games is not poly$(n, 1/\sigma)$ under perturbations with magnitude σ, unless **PPAD** $\subseteq$ **RP**.

The first result implies that it is also **PPAD**-hard to find a $1/$poly(n)-relatively-approximate Nash equilibrium in an $n \times n$ two-player game. It also implies that for any fixed $c > 0$, it remains **PPAD**-complete to compute the first $(1 + c)\log n$-bits of an exact Nash equilibrium in a two-player game, even when the entries of the matrices that defined the game are integers of poly(n) magnitude. Thus, computing these leading bits is as hard as finding an exact Nash equilibrium of a rational two-player game.

We emphasize that the **PPAD**-hard games constructed in [9, 5, 10, 4] can not be used to demonstrate that the problem of computing a $1/$poly(n)-approximate Nash equilibrium in a $\{2, 3, 4\}$-player game is **PPAD**-hard. For example, we can compute in polynomial time a $1/n$-approximate Nash equilibrium in the **PPAD**-hard two-player games constructed by Chen and Deng [4], although they proved that finding a $2^{-\Theta(n)}$-approximate Nash equilibrium in these games is **PPAD**-complete. Thus, their proof does not apply to the search of $1/$poly(n)-approximate Nash equilibria.

This limitation of [9, 5, 10, 4] is inherent in their choice of the **PPAD**-complete fixed-point problems from which they build their $\{2, 3, 4\}$-player games. The exponential dependency is the result of the fact that the approximation precision needed increases geometrically in the number of bits required to represent the coordinate of a point in the search space. In order to prove our new result, we need a reduction from a **PPAD**-complete fixed-point problem on hypergrids of a constant or polynomial side-length, which seems inconceivable in previous approaches.

[1]As an ϵ-approximate Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game $(\mathbf{A}, \mathbf{B})$ becomes a $c \cdot \epsilon$-approximate Nash equilibrium in game $(c\mathbf{A}, c\mathbf{B})$ for $c > 0$. Following Lipton, Markakis, and Mehta [16], we normalize the matrices $\mathbf{A}$ and $\mathbf{B}$ so that all their entries are between 0 and 1, or between -1 and 1 when studying ϵ-approximate Nash equilibria.

As an instrumental step of our work, we introduce a family of high-dimensional discrete fixed-point problems, one associated with the $(8 \times 8 ... \times 8)$ n-dimensional hypergrid.

Fortunately and somewhat surprisingly, the fixed-point problem is still **PPAD**-complete in this hypergrid with a seemingly very small side-length. We show this hypergrid has enough flexibility and structure to host the embedding of the proof structure of any **PPAD** problem.

However, high dimensionality come with their own challenges, as the original approach defines a fixed point to be a unit hypercube with 2^n points, which creates new difficulties to the efficiency of the reduction. As a critical step of our reduction, we prove a geometric lemma for finding a fixed point defined on $n+1$ vertices of a unit hypercube. This lemma allows us to overcome the curse of dimensionality in reasoning about fixed points in high dimensions.

1.3 Notations

We will use bold lower-case Roman letters such as $\mathbf{a}$ to denote vectors and upper-case Roman letters such as $\mathbf{A}$ to denote matrices. We denote the i^{th}-entry of $\mathbf{a}$ by a_i and the $(i,j)^{th}$ entry of $\mathbf{A}$ by $a_{i,j}$. Further, (1) $\mathbb{Z}^d_+$ denotes the set of d-dimensional vectors with positive integer entries, (2) $\mathbb{Z}^d_{[a,b]} = \{\, \mathbf{q} \in \mathbb{Z}^d \mid a \leq q_i \leq b, \forall\, 1 \leq i \leq d \,\}$, (3) $\mathbb{R}^{m\times n}_{[a:b]}$ is the set of $m \times n$ matrices with real entries between a and b, (4) $\langle \mathbf{a}|\mathbf{b}\rangle$ is the dot-product of two vectors. (5) $\mathbf{e}_i$ stands for the unit vector whose i^{th} entry is equal to 1 and all other entries are zeros, (6) for $x, y \in \mathbb{R}$ and $\epsilon \in \mathbb{R}^+$, by $x = y \pm \epsilon$, we mean $y - \epsilon \leq x \leq y + \epsilon$, (7) for $\mathbf{p}, \mathbf{q} \in \mathbb{R}^n$ and $\epsilon \in \mathbb{R}^+$, by $\mathbf{p} = \mathbf{q} \pm \epsilon$, we mean $p_i = q_i \pm \epsilon$, for all i, and (8) a game $(\mathbf{A}, \mathbf{B})$ is *positively normalized* if $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}_{[0,1]}$.

2 PPAD and Discrete Fixed Points

A binary relation $R \subset \{0,1\}^* \times \{0,1\}^*$ is *polynomially balanced* if there exists a polynomial p such that for all pairs $(x,y) \in R$, $|y| \leq p(|x|)$. It is a *polynomial-time computable relation* if for each pair (x,y), one can decide whether or not $(x,y) \in R$ in time polynomial in $|x| + |y|$. One can define the **NP** search problem Q_R specified by R as: Given $x \in \{0,1\}^*$, if there exists y such that $(x,y) \in R$, return y, otherwise, return a special string "no".

Relation R is *total* if for every $x \in \{0,1\}^*$, there exists y such that $(x,y) \in R$. Following [17], let **TFNP** denote the class of all **NP** search problems specified by total relations. A search problem $Q_{R_1} \in$ **TFNP** is *polynomial-time reducible* to problem $Q_{R_2} \in$ **TFNP** if there exists a pair of polynomial-time computable functions (f,g) such that for every x of R_1, if y satisfies that $(f(x), y) \in R_2$, then $(x, g(y)) \in R_1$. Q_{R_1} and Q_{R_2} are polynomial-time equivalent if Q_{R_2} is also reducible to Q_{R_1}.

The complexity class **PPAD** [22] is sub-class of **TFNP**, containing problems polynomial-time reducible to:

Definition 2.1 (LEAFD). *The input instance of* LEAFD *is a pair $(M, 0^n)$ where M defines a polynomial-time Turing machine satisfying: (1). For every $v \in \{0,1\}^n$, $M(v)$ is an ordered pair (u_1, u_2) with $u_1, u_2 \in \{0,1\}^n \cup \{$"no"$\}$; (2). $M(0^n) = ($"no"$, 1^n)$ and the first component of $M(1^n)$ is 0^n. This instance defines a directed graph $G = (V, E)$ with $V = \{0,1\}^n$ and $(u,v) \in E$ iff v is the second component of $M(u)$ and u is the first component of $M(v)$. The output is a directed leaf of G other than 0^n. A vertex is a* directed leaf *if its out-degree plus in-degree equals one.*

2.1 Problem Brouwer^f

We say an integer function $f(n)$ is *well-behaved* if it is polynomial-time computable and there exists an integer constant n_0 such that $3 \leq f(n) \leq n/2$ for all $n \geq n_0$. 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 well-behaved functions.

Let $K_{\mathbf{p}} = \{ \mathbf{q} \in \mathbb{Z}^d \mid q_i \in \{p_i, p_i + 1\}, \forall i. \}$ for $\mathbf{p} \in \mathbb{Z}^d$. Let $A^d_{\mathbf{r}} = \{ \mathbf{q} \in \mathbb{Z}^d \mid 0 \leq q_i \leq r_i - 1, \forall i. \}$, for $d \in \mathbb{Z}^1_+$ and $\mathbf{r}$ in $\mathbb{Z}^d_+$, denote the *hypergrid* with side lengths specified by $\mathbf{r}$. The *boundary* of $A^d_{\mathbf{r}}$, $\partial(A^d_{\mathbf{r}})$, is the set of points $\mathbf{q} \in A^d_{\mathbf{r}}$ with $q_i \in \{0, r_i - 1\}$ for some i. For each $\mathbf{r} \in \mathbb{Z}^d_+$, let Size $[\mathbf{r}] = \sum_{1 \leq i \leq d} \lceil \log(r_i + 1) \rceil$.

Definition 2.2 (Brouwer-Mapping Circuit). *For $d \in \mathbb{Z}^1_+$, $\mathbf{r} \in \mathbb{Z}^d_+$, a Boolean circuit C is a* Brouwer-mapping circuit *with parameters d and $\mathbf{r}$ if it has Size $[\mathbf{r}]$ input bits and $2d$ output bits $\Delta^+_1, \Delta^-_1, ..., \Delta^+_d, \Delta^-_d$. C is* valid *if*

- *$\forall\ \mathbf{p} \in A^d_{\mathbf{r}}$, the $2d$ output bits evaluated at $\mathbf{p}$ satisfy: case i, $1 \leq i \leq d$: $\Delta^+_i = 1$ and all other $2d-1$ bits are 0; or case $(d+1)$: $\forall i$, $\Delta^+_i = 0$ and $\Delta^-_i = 1$.*
- *$\forall\ \mathbf{p} \in \partial(A^d_{\mathbf{r}})$, if there exists an i such that $p_i = 0$, letting $i_{\max} = \max\{\, i \mid p_i = 0 \,\}$, then the output bits satisfy the $i^{th}_{\max}$ case, otherwise ($\forall i$, $p_i \neq 0$ and some p_i is $r_i - 1$), the output bits satisfy the $d + 1^{st}$ case.*

Definition 2.3 (Brouwer Color Assignment and Panchromatic Simplex). *Let C be a valid Brouwer-mapping circuit with parameters d and $\mathbf{r}$. It defines a color assignment $Color_C : A^d_{\mathbf{r}} \to \{1, 2, ..., d+1\}$ as: $Color_C[\mathbf{p}] = i$ if the output bits of C evaluated at $\mathbf{p}$ satisfy the i^{th} case.*

A subset $P \subset A^d_{\mathbf{r}}$ is accommodated *if $P \subset K_{\mathbf{p}}$ for some point $\mathbf{p} \in A^d_{\mathbf{r}}$. $P \subset A^d_{\mathbf{r}}$ is a* panchromatic simplex *of C if it is accommodated and contains exactly $d+1$ points with distinct $d+1$ colors.*

Definition 2.4 (BROUWER^f). *For a well-behaved function f and an integer $n > 0$, let $d = \lceil n/f(n) \rceil$ and $m = f(n)$. An input instance of* BROUWER^f *is a pair $(C, 0^n)$, where C is a valid Brouwer-mapping circuit with parameters d*

and $\mathbf{r}$ *with* $r_i = 2^m$ *for all* $i : 1 \leq i \leq d$. *The output of the problem is then a panchromatic simplex of circuit* C.

Both BROUWER^{f_2} [3] and BROUWER^{f_3} [9] are **PPAD**-complete. In section 5, we will establish the following theorem. It states that the complexity of finding a fixed point is independent of the shape or dimension of the search space.

Theorem 2.5 (High-Dimensional Fixed Points)**.** *For each well-behaved function* f*,* BROUWER^f *is* **PPAD***-complete.*

3 Smoothed Analysis and Approximation

In the smoothed analysis [24] of two-player games, we consider perturbed games: For a pair of $m \times n$ matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ with $|\overline{a}_{i,j}|, |\overline{b}_{i,j}| \leq 1$, let $\mathbf{A}$ and $\mathbf{B}$ be matrices with $a_{i,j} = \overline{a}_{i,j} + r^A_{i,j}$, $b_{i,j} = \overline{b}_{i,j} + r^B_{i,j}$, while $r^A_{i,j}$ and $r^B_{i,j}$ are chosen independently and uniformly from $[-\sigma, \sigma]$ or from Gaussian distribution with variance σ^2. We refer to these perturbations as *σ-uniform* and *σ-Gaussian perturbations*. An algorithm for bimatrix games has *polynomial smoothed complexity* [25] if it finds a Nash equilibrium of $(\mathbf{A}, \mathbf{B})$ in expected time poly$(m, n, 1/\sigma)$, for all $(\overline{\mathbf{A}}, \overline{\mathbf{B}})$.

The following lemma shows that if the smoothed complexity of the bimatrix game, under uniform or Gaussian perturbations, is low, then one can quickly find an approximate Nash equilibrium. See the full version [7] for a proof.

Lemma 3.1 (Smoothed Nash vs Approximate Nash)**.** *If* `2-NASH` *can be solved in smoothed* poly$(m, n, 1/\sigma)$ *time under σ-uniform perturbations or σ-Gaussian perturbations, then for all $\epsilon > 0$, there exists a randomized algorithm to compute an ϵ-approximate Nash equilibrium in a two-player game with expected time* $O(\text{poly}(m, n, 1/\epsilon))$ *or* $O(\text{poly}(m, n, \sqrt{\log \max(m, n)}/\epsilon))$*, respectively.*

4 Approximating Nash Equilibria is Hard

In this section, we prove our main theorem:

Theorem 4.1 (Unlikely FPTAS for 2-NASH)**.** *For any constant $c > 0$, the problem of computing a $1/n^c$-approximate Nash equilibrium of a positively normalized $n \times n$ bimatrix game is* **PPAD***-complete.*

Setting $\epsilon = 1/\text{poly}(n)$, by Theorem 4.1 and Lemma 3.1:

Theorem 4.2 (Hardness of Smoothed 2-NASH)**.** `2-NASH` *is not in smoothed polynomial time, under uniform or Gaussian perturbations, unless* **PPAD** $\subseteq$ **RP**.

Consequently,

Theorem 4.3 (Simplex vs Lemke-Howson: Smoothed Perspective)**.** *The smoothed complexity of the Lemke-Howson algorithm is not polynomial under uniform or Gaussian perturbations, unless* **PPAD** $\subseteq$ **RP**.

4.1 Well-Supported Nash Equilibria

In our analysis, we will use an alternative notion of approximate Nash equilibria as introduced in [9]: For a game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$, let $\mathbf{a}_i$ and $\mathbf{b}_i$ denote the i^{th} row of $\mathbf{A}$ and the i^{th} column of $\mathbf{B}$, respectively. An *ϵ-well-supported Nash equilibrium* of $\mathcal{G}$ is a pair $(\mathbf{x}^*, \mathbf{y}^*)$, such that for all j, k,

$$\langle \mathbf{b}_j | \mathbf{x}^* \rangle > \langle \mathbf{b}_k | \mathbf{x}^* \rangle + \epsilon \;\Rightarrow\; y^*_k = 0 \quad \text{and}$$
$$\langle \mathbf{a}_j | \mathbf{y}^* \rangle > \langle \mathbf{a}_k | \mathbf{y}^* \rangle + \epsilon \;\Rightarrow\; x^*_k = 0.$$

We prove the following lemma in our full version [7].

Lemma 4.4 (Polynomial Equivalence)**.** *In a bimatrix game* $(\mathbf{A}, \mathbf{B})$ *with* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}_{[0:1]}$*, for any $0 \leq \epsilon \leq 1$, (1) each ϵ-well-supported Nash equilibrium is also an ϵ-approximate Nash equilibrium; and (2) from any $\epsilon^2/(8n)$-approximate Nash equilibrium* $(\mathbf{u}, \mathbf{v})$*, one can find in polynomial time an ϵ-well-supported Nash equilibrium* $(\mathbf{x}, \mathbf{y})$.

4.2 From Brouwer to Approximate Nash

We reduce the search of a panchromatic simplex in an instance of BROUWER^{f_1} (which will be simply referred to as BROUWER in this section) to the computation of a $1/n^c$-approximate Nash equilibrium in a positively normalized bimatrix game, for some constant $c > 0$. Recall $f_1(n) = 3$. Our reduction will use ideas and gadgets from [9, 4].

However, we will need to develop several new techniques to meet the geometric and combinatorial challenges in the consideration of high-dimensional fixed-points.

Let $U = (C, 0^{3n})$ be an input instance of BROUWER, which colors the hypergrid $B^n = \mathbb{Z}^n_{[0,7]}$ with colors from $\{1, ..., n, n+1\}$. Let m be the smallest integer such that $2^m \geq$ Size$[C]$ and $N = 2^{6m+1} = 2K$, where Size$[C]$ is the number of gates plus the number of input and output variables in a Boolean circuit C. We first, in polynomial time, construct an $N \times N$ bimatrix game $\mathcal{G}^U = (\mathbf{A}^U, \mathbf{B}^U)$. Our construction ensures for $\epsilon = 1/K^3$, $\epsilon' = \epsilon^2/(8N)$,

- Property $\mathbf{A}_1$: $|a^U_{i,j}|, |b^U_{i,j}| \leq N^3, \forall\, 1 \leq i, j \leq N$ and
- Property $\mathbf{A}_2$: From every ϵ'-approximate Nash equilibrium of game $\mathcal{G}^U$, we can compute a panchromatic simplex P of circuit C in polynomial time.

By Lemma 4.4 it suffices to show

- Property $\mathbf{A}'_2$: From each ϵ-well-supported Nash equilibrium of game $\mathcal{G}^U$, we can compute a panchromatic simplex P of circuit C in polynomial time.

We normalize $\mathcal{G}^U$ to obtain $\overline{\mathcal{G}^U} = (\overline{\mathbf{A}^U}, \overline{\mathbf{B}^U})$ by setting

$$\overline{a^U}_{i,j} = \frac{a^U_{i,j} + N^3}{2N^3} \quad \text{and} \quad \overline{b^U}_{i,j} = \frac{b^U_{i,j} + N^3}{2N^3}, \quad \forall i, j.$$

From Property $\mathbf{A}_2$, a panchromatic simplex P of C can be found in polynomial time from every N^{-10}-approximate Nash equilibrium of $\overline{\mathcal{G}^U}$. By the following lemma, (proof omitted), we can extend the constant 10 to any fixed $c > 0$.

Lemma 4.5. *If the problem of finding an n^{-c}-approximate Nash equilibrium in an $n \times n$ positively normalized game is* **PPAD***-complete for some constant $c = c_0 > 0$, then the problem is* **PPAD***-complete for every constant $c > 0$.*

4.3 Overcome the Curse of Dimensionality

We now prove a key geometric lemma for finding a panchromatic simplex in order to overcome the curse of dimensionality. Our reduction will then build on this lemma.

For $a \in \mathbb{R}^+$, let $\pi(a) = \max\{\, i \mid 0 \le i \le 7 \text{ and } i < a \,\}$ be the largest integer in $[0:7]$ that is smaller than a. Let $E^n = \{\, \mathbf{z}^1, \mathbf{z}^2 ... \mathbf{z}^n, \mathbf{z}^{n+1} \,\}$ where $\mathbf{z}^i = \mathbf{e}_i/K^2$ and $\mathbf{z}^{n+1} = -\sum_{1 \le i \le n} \mathbf{e}_i/K^2$. We encode the i^{th} color by vector $\mathbf{z}^i$.

For $\mathbf{p} \in \mathbb{R}^n_+$, let $\mathbf{q} = \pi(\mathbf{p})$ be the lattice point in B^n with $q_i = \pi(p_i), \forall i$. Let $\xi(\mathbf{p}) = \mathbf{z}^t$, where $t = \text{Color}_C\,[\pi(\mathbf{p})]$.

Definition 4.6 (Well-Positioned Points)**.** *A real $a \in \mathbb{R}^+$ is* poorly-positioned *if there is an integer $t : 0 \le t \le 7$ such that $|\,a - t\,| \le 80K\epsilon = 80/K^2$. A point $\mathbf{p} \in \mathbb{R}^n_+$ is* well-positioned *if none of its components is* poorly-positioned, *otherwise, it is* poorly-positioned.

Let $S = \{\mathbf{p}^1, \mathbf{p}^2 ..., \mathbf{p}^h\}$ be a set of h points in $\mathbb{R}^n_{[0,8]}$. Let $I_B(S) = \{\, k \mid \mathbf{p}^k \text{ is a poorly-positioned point} \,\}$, and $I_G(S) = \{\, k \mid \mathbf{p}^k \text{ is a well-positioned point} \,\}$.

Lemma 4.7 (Key Geometry: Equiangle Averaging Segment)**.** *Let $S = \{\mathbf{p}^i, 1 \le i \le n^3\}$ be n^3 points in $\mathbb{R}^n_{[0,8]}$ such that $\mathbf{p}^i = \mathbf{p}^{i-1} + \sum_{i=1}^n \mathbf{e}_i/K$, $\forall\, 1 < i \le n^3$. If there is a vector $\mathbf{r}^k \in \mathbb{R}^n_{[0,1/K^2]}$ for each k in $I_B(S)$, such that,*

$$\Big\|\, \textstyle\sum_{k \in I_G(S)} \xi(\mathbf{p}^k) \; + \sum_{k \in I_B(S)} \mathbf{r}^k \,\Big\|_\infty = O(\epsilon),$$

then $Q = \{\pi(\mathbf{p}^k), k \in I_G(S)\}$ is a panchromatic simplex of Boolean circuit C.

Proof. We first prove that $Q' = \{\mathbf{q}^k = \pi(\mathbf{p}^k), 1 \le k \le n^3\}$ is accommodated, and $|\,Q'\,| \le n + 1$. As sequence $\{\mathbf{p}^k\}$ is strictly increasing, $\{\mathbf{q}^k\}$ is non-decreasing. Since $n/K \ll 1$, there exists at most one k_i, for each i, such that $q_i^{k_i} = q_i^{k_i - 1} + 1$, which implies that Q' is accommodated. Moreover, as $\{\mathbf{q}^k\}$ is non-decreasing, $|\,Q'\,| \le n + 1$. As $Q \subset Q'$, Q is also accommodated and $|\,Q\,| \le n + 1$.

Because $1/K^2 \ll 1/K \ll 1$, there is at most one k_i, for each i, such that $p_i^{k_i}$ is poorly-positioned. Since every poorly-positioned point has at least one poorly-positioned component, $|\,I_B(S)\,| \le n$ and $|\,I_G(S)\,| \ge n^3 - n$.

For every $1 \le i \le n + 1$, let W_i denote the number points in $\big\{\, \mathbf{q}^k : k \in I_G(S) \,\big\}$ that is colored i by circuit C.

$$\begin{array}{ll}
G_+: & \mathcal{P}[T] = [\; \mathbf{x}[v] = \min(\mathbf{x}[v_1] + \mathbf{x}[v_2], \mathbf{x}_C[v]\,) \pm \epsilon \;] \\
G_\zeta: & \mathcal{P}[T] = [\; \mathbf{x}[v] = c \pm \epsilon \;] \\
G_{\times\zeta}: & \mathcal{P}[T] = [\; \mathbf{x}[v] = \min(c\mathbf{x}[v_1], \mathbf{x}_C[v]\,) \pm \epsilon \;] \\
G_=: & \mathcal{P}[T] = [\; \mathbf{x}[v] = \min(\mathbf{x}[v_1], \mathbf{x}_C[v]\,) \pm \epsilon \;] \\
G_<: & \mathcal{P}[T] = \left[\begin{array}{l} \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] < \mathbf{x}[v_2] - \epsilon \\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] > \mathbf{x}[v_2] + \epsilon \end{array}\right] \\
G_-: & \mathcal{P}[T] = \left[\begin{array}{c} \min(\mathbf{x}[v_1] - \mathbf{x}[v_2], \mathbf{x}_C[v]\,) - \epsilon \le \mathbf{x}[v] \\ \mathbf{x}[v] \le \max(\mathbf{x}[v_1] - \mathbf{x}[v_2], 0\,) + \epsilon \end{array}\right] \\
G_\vee: & \mathcal{P}[T] = \left[\begin{array}{l} \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1 \text{ or } \mathbf{x}[v_2] =_B 1 \\ \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0 \text{ and } \mathbf{x}[v_2] =_B 0 \end{array}\right] \\
G_\wedge: & \mathcal{P}[T] = \left[\begin{array}{l} \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 0 \text{ or } \mathbf{x}[v_2] =_B 0 \\ \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 1 \text{ and } \mathbf{x}[v_2] =_B 1 \end{array}\right] \\
G_\neg: & \mathcal{P}[T] = \left[\begin{array}{l} \mathbf{x}[v] =_B 0 \text{ if } \mathbf{x}[v_1] =_B 1 \\ \mathbf{x}[v] =_B 1 \text{ if } \mathbf{x}[v_1] =_B 0 \end{array}\right]
\end{array}$$

Figure 1. Constraint P[T]

It suffices to show that $W_i > 0$ for all i in order prove Q is a panchromatic simplex,

Let $\mathbf{r}^G = \sum_{k \in I_G(S)} \xi(\mathbf{p}^k)$ and $\mathbf{r}^B = \sum_{k \in I_B(S)} \mathbf{r}^k$. Since $|I_B(S)| \le n$ and $\|\,\mathbf{r}^k\,\|_\infty \le 1/K^2$, we have

$$\begin{aligned} &\|\,\mathbf{r}^B\,\|_\infty \le n/K^2, \quad \text{and} \\ &\|\,\mathbf{r}^G\,\|_\infty \le \|\,\mathbf{r}^B\,\|_\infty + O(\epsilon) \le n/K^2 + O(\epsilon). \end{aligned} \tag{2}$$

Assume by way of contradiction that one of W_i is zero:

(i) $W_{n+1} = 0$: Suppose $W_{i^*} = \max_{1 \le i \le n} W_i$, then $W_{i^*} \ge n^2 - 1$. But $r_{i^*}^G \ge (n^2 - 1)/K^2 \gg n/K^2 + O(\epsilon)$, since $\epsilon = 1/K^3$, which contradicts with (2) above.

(ii) $W_t = 0$ for some $1 \le t \le n$: We assert $W_{n+1} \le n^2/2$, for otherwise, $|r_t^G| > n^2/(2K^2) \gg n/K^2 + O(\epsilon)$, contradicting with (2). Suppose $W_{i^*} = \max_{1 \le i \le n+1} W_i$, then $W_{i^*} \ge n^2 - 1$ and $i^* \ne n + 1$. So $r_{i^*}^G \ge (n^2 - 1 - n^2/2)/K^2 \gg n/K^2 + O(\epsilon)$, contradicting with (2). □

4.4 Gadget Games and Constraints

To construct game $\mathcal{G}^U$, we transform a prototype game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$, a zero-sum game to be defined, by adding "gadget" games: We will build a collection of gadgets $\mathcal{S}^U = \{\, T_1, ..., T_l \,\}$ for some $l \le K$. Each $T \in \mathcal{S}^U$ defines two $N \times N$ matrices $(\mathbf{L}[T], \mathbf{R}[T])$, as shown in Figure 2.

We define a function named BUILDGAME that takes a valid collection (will be defined) of gadgets $\mathcal{S}^U$ and returns a bimatrix game $\mathcal{G}^U = (\mathbf{A}^U, \mathbf{B}^U) = \text{BUILDGAME}(\mathcal{S}^U)$ as: $\mathbf{A}^U = \mathbf{A}^* + \sum_{T \in \mathcal{S}^U} \mathbf{L}[T]$, $\mathbf{B}^U = \mathbf{B}^* + \sum_{T \in \mathcal{S}^U} \mathbf{R}[T]$.

4.4.1 Nodes, Values and Capacities

We choose two sets V_A and V_I with $|V_A| = |V_I| = K$. It will become clear what these elements are. So, don't worry too much about them now. We call elements in V_A *arithmetic nodes* and elements in V_I *internal nodes*.

Below, we will always use v to denote a node in V_A and w to denote a node in V_I. We also pick two one-to-one correspondences $\mathcal{C}_A$ and $\mathcal{C}_I$ to index these sets: $\mathcal{C}_A$ maps V_A to $[K] = \{1, 2, ..., K\}$ and $\mathcal{C}_I$ maps V_I to $[K]$.

Let $(\mathbf{x} \in \mathbb{P}^N, \mathbf{y} \in \mathbb{P}^N)$ be a pair of probability vectors. For each $v \in V_A$, let $\mathbf{x}[v] = x_{2k-1}$ and $\mathbf{x}_C[v] = x_{2k-1} + x_{2k}$, where $k = \mathcal{C}_A(v)$. We refer to $\mathbf{x}[v]$ and $\mathbf{x}_C[v]$ as the *value* and *capacity*, respectively, of v in $(\mathbf{x}, \mathbf{y})$. Similarly, the value and capacity of each $w \in V_I$ is $\mathbf{y}[w] = y_{2t-1}$ and $\mathbf{y}_C[w] = y_{2t-1} + y_{2t}$, respectively, where $t = \mathcal{C}_I(w)$.

4.4.2 The Prototype Game and its Properties

Our prototype $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$ is the game of *Matching Pennies* with parameter $M = 2^{18m+1} = 2K^3$: $\mathbf{A}^*$ is a $K \times K$ block-diagonal matrix where each diagonal block is a 2×2 matrix of all M's, and $\mathbf{B}^* = -\mathbf{A}^*$. All games we will consider below belong to the following class:

Definition 4.8 (Class $\mathcal{L}$). *A two-player game* $(\mathbf{A}, \mathbf{B})$ *belongs to class* $\mathcal{L}$ *if* $0 \leq \mathbf{A} - \mathbf{A}^*, \mathbf{B} - \mathbf{B}^* \leq 1$.

Every equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}^*$ enjoys a nice property: $\mathbf{x}_C[v] = \mathbf{y}_C[w] = 1/K$, for all $v \in V_A$ and $w \in V_I$.

Let $\mathcal{P}$ denote the following constraint on $(\mathbf{x}, \mathbf{y})$:

$$[\mathbf{x}_C[v] = 1/K \pm \epsilon, \mathbf{y}_C[w] = 1/K \pm \epsilon, \forall\, v \in V_A, w \in V_I].$$

Lemma 4.9 (Nearly Uniform Capacities). *If* $\mathcal{G} \in \mathcal{L}$*, then every* 1.0*-well-supported Nash equilibrium* $(\mathbf{x}, \mathbf{y})$ *of game* $\mathcal{G}$ *satisfies constraint* $\mathcal{P}$ *for* $\epsilon = 2^{-18m} = 1/K^3$.

4.4.3 Gadgets and their Constraints

We use all the nine types of gadgets designed in [4] for our construction: $\{G_\zeta, G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\wedge, G_\vee, G_\neg\}$. Among them, $G_\wedge, G_\vee$ and $G_\neg$ are logic gadgets. They will be used to simulate the logic gates in the Boolean circuit C. Associated with probability vectors $(\mathbf{x}, \mathbf{y})$, the value of $v \in V_A$ represents boolean 1 ($\mathbf{x}[v] =_B 1$) if $\mathbf{x}[v] = \mathbf{x}_C[v]$; it represents boolean 0 ($\mathbf{x}[v] =_B 0$) if $\mathbf{x}[v] = 0$. A gadget $T = (G, v_1, v_2, v, c, w)$ is a 6-tuple that implements an arithmetic or logic constraint $\mathcal{P}[T]$, which requires the values of nodes v, v_1 and v_2 to satisfy certain functional relationship. The requirements for logic gadgets will hold exactly and the requirements for arithmetic ones will hold approximately. By Figure 1, the logic constraints implemented by the logic gadgets are effective only when the values of their input nodes are representations of binary bits.

In gadget T, $v_1 \in V_A \cup \{nil\}$ and $v_2 \in V_A \cup \{nil\}$ are the first and second input nodes, respectively, $v \in V_A$ is the output node, and $w \in V_I$ is the internal node of the gadget. Parameter c is only used in G_ζ and $G_{\times\zeta}$ gadgets: when type $G = G_\zeta$, we have $c \in \mathbb{R}$ and $0 \leq c \leq 1/K - \epsilon$; when $G = G_{\times\zeta}$, $0 \leq c \leq 1$; otherwise, $c = nil$.

Definition 4.10 (Valid Collection). *A collection* $\mathcal{S}$ *of gadgets is* valid *if for each pair* $T = (G, v_1, v_2, v, w, c)$ *and* $T' = (G', v_1', v_2', v', c', w')$ *in* $\mathcal{S}$, $v \neq v'$ *and* $w \neq w'$.

For every valid collection $\mathcal{S}$, BUILDGAME($\mathcal{S}$) satisfies the following constraints:

Lemma 4.11 ($\mathcal{P}$). *Let* $\mathcal{G} =$ BUILDGAME($\mathcal{S}$). *If* $\mathcal{S}$ *is valid then* $\mathcal{G} \in \mathcal{L}$ *and* $|\mathcal{S}| \leq K$. *So, by Lemma 4.9, each* ϵ*-well-supported Nash equilibrium of* $\mathcal{G}$ *satisfies constraint* $\mathcal{P}$.

Proof. For each $T \in S$, $(\mathbf{L}[T], \mathbf{R}[T])$ of Figure 2 satisfy:

Property 4.12. *Let* $T = (G, v_1, v_2, v, c, w)$, $\mathbf{L}[T] = (L_{i,j})$ *and* $\mathbf{R}[T] = (R_{i,j})$. *Let* $k = \mathcal{C}_A(v)$ *and* $t = \mathcal{C}_I(v)$. *Then*

- $i \neq 2k$ *or* $2k-1 \;\Rightarrow\; L_{i,j} = 0, \forall\, 1 \leq j \leq 2K;$
- $j \neq 2t$ *or* $2t-1 \;\Rightarrow\; R_{i,j} = 0, \forall\, 1 \leq i \leq 2K;$
- $i = 2k$ *or* $2k-1 \;\Rightarrow\; 0 \leq L_{i,j} \leq 1, \forall\, 1 \leq j \leq 2K;$
- $j = 2t$ *or* $2t-1 \;\Rightarrow\; 0 \leq R_{i,j} \leq 1, \forall\, 1 \leq i \leq 2K.$

As $\mathcal{S}$ is valid, $\mathcal{G} \in \mathcal{L}$. As $|V_A| = |V_I| = K$, $|\mathcal{S}| \leq K$. ☐

Theorem 4.13 (Gadget Constraints). *If* $\mathcal{S}$ *is a valid collection of gadgets, and* $(\mathbf{x}, \mathbf{y})$ *is an* ϵ*-well-supported Nash equilibrium of* BUILDGAME($\mathcal{S}$)*, then for all* $T \in \mathcal{S}$*, constraint* $\mathcal{P}[T]$ *as defined in Figure 1 is satisfied by* $(\mathbf{x}, \mathbf{y})$.

Proof. A similar theorem is proved in [4], also see [7]. ☐

4.5 Construction of the Game $\mathbf{G^U}$

Our objective is to design a bimatrix game $\mathcal{G}^U$ to encode n^3 points in $\mathbb{R}^n_{[0,8]}$, to simulate the π function, and the boolean circuit C so that in every ϵ-well-supported equilibrium of $\mathcal{G}^U$, the sum of the n^3 vectors as given in Lemma 4.7 is close to zero, i.e., $O(\epsilon)$. To achieve this, We build a valid collection $\mathcal{S}^U$ of gadgets to define the bimatrix game. Then Q defined in Lemma 4.7 is a panchromatic simplex of C, which can be computed in polynomial time.

Let us define some notations that will be useful. Let $\mathcal{S}$ be a valid collection of gadgets. A node $v \in V_A$ (or node $w \in V_I$) is said to be *unused* if none of the gadgets in $\mathcal{S}$ uses v (or w) as its output node (or internal node). Suppose $T \notin \mathcal{S}$ is a gadget such that $\mathcal{S} \cup \{T\}$ is valid. We will use INSERT($\mathcal{S}, T$) to denote the insertion of T into $\mathcal{S}$.

To encode these n^3 points, let $\{v_i^k\}_{1 \leq k \leq n^3, 1 \leq i \leq n}$ be n^4 distinguished nodes in set V_A. We start with $\mathcal{S}^U = \emptyset$ and insert a number of gadgets into it so that, in every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}^U$, values of these nodes encode n^3 points $S = \{\mathbf{p}^k : 1 \leq k \leq n^3\}$ in $\mathbb{R}^n_{[0,8]}$ that *approximately* satisfy all the conditions of Lemma 4.7. In our encoding, we let $p_i^k = 8K\mathbf{x}[v_i^k]$.

$\mathbf{L}[T]$ and $\mathbf{R}[T]$, where $T = (G, v_1, v_2, v, c, w)$

Set $\mathbf{L}[T] = (L_{i,j}) = \mathbf{R}[T] = (R_{i,j}) = 0$

$k = \mathcal{C}_A(v)$, $k_1 = \mathcal{C}_A(v_1)$, $k_2 = \mathcal{C}_A(v_2)$, and $t = \mathcal{C}_I(w)$

$$G_+\colon \begin{cases} L_{2k-1,2t-1} = L_{2k,2t} = 1 \\ R_{2k_1-1,2t-1} = R_{2k_2-1,2t-1} = R_{2k-1,2t} = 1 \end{cases}$$

$$G_\delta\colon \begin{cases} L_{2k-1,2t} = L_{2k,2t-1}1 \\ R_{2k-1,2t-1} = 1,\ R_{i,2t} = c, \forall\, 1 \le i \le 2K \end{cases}$$

$$G_{\times\delta}\colon \begin{cases} L_{2k-1,2t-1} = L_{2k,2t} = 1 \\ R_{2k_1-1,2t-1} = c,\ R_{2k-1,2t} = 1 \end{cases}$$

$$G_=\colon \begin{cases} L_{2k-1,2t-1} = L_{2k,2t} = 1 \\ R_{2k_1-1,2t-1} = R_{2k-1,2t} = 1 \end{cases}$$

$$G_-\colon \begin{cases} L_{2k-1,2t-1} = L_{2k,2t} = 1 \\ R_{2k_1-1,2t-1} = R_{2k_2-1,2t} = R_{2k-1,2t} = 1 \end{cases}$$

$$G_<\colon \begin{cases} L_{2k-1,2t} = L_{2k,2t-1} = 1 \\ R_{2k_1-1,2t-1} = R_{2k_2-1,2t} = 1 \end{cases}$$

$$G_\vee\colon \begin{cases} L_{2k-1,2t-1} = L_{2k,2t} = R_{2k_1-1,2t-1} = 1 \\ R_{2k_2-1,2t-1} = 1,\ R_{i,2t} = 1/(2K), \forall\, 1 \le i \le 2K \end{cases}$$

$$G_\wedge\colon \begin{cases} L_{2k-1,2t-1} = L_{2k,2t} = R_{2k_1-1,2t-1} = 1 \\ R_{2k_2-1,2t-1} = 1,\ R_{i,2t} = 3/(2K), \forall\, 1 \le i \le 2K \end{cases}$$

$$G_\neg\colon \begin{cases} L_{2k-1,2t} = L_{2k,2t-1} = 1 \\ R_{2k_1-1,2t-1} = R_{2k_1,2t} = 1 \end{cases}$$

Figure 2. Matrices L[T] and R[T]

EXTRACTBITS$(\mathcal{S}, v, v^1, v^2, v^3)$

```
1: pick unused nodes $v_1, v_2, v_3, v_4 \in V_A$ and $w \in V_I$
2: INSERT$(\mathcal{S}, (G_=, v, nil, v_1, nil, w))$
3: for $j$ from 1 to 3 do
4:     pick unused $v_{j1}, v_{j2} \in V_A$ and $w_{j1}, w_{j2}, w_{j3}, w_{j4} \in V_I$
5:     INSERT$(\mathcal{S}, (G_\zeta, nil, nil, v_{j1}, 2^{-(6m+j)}, w_{j1}))$
6:     INSERT$(\mathcal{S}, (G_<, v_{j1}, v_j, v^j, nil, w_{j2}))$
7:     INSERT$(\mathcal{S}, (G_{\times\zeta}, v^j, nil, v_{j2}, 2^{-j}, w_{j3}))$
8:     INSERT$(\mathcal{S}, (G_-, v_j, v_{j2}, v_{j+1}, nil, w_{j4}))$
```

Figure 3. Function ExtractBits

We define two functions EXTRACTBITS and COLORINGSIMULATION. They will be used as the building blocks of our reduction. EXTRACTBITS given in Figure 3, enables us to realize the π function. We have

Lemma 4.14 (Encoding Binary with Games)**.** *Suppose $\mathcal{S}$ is a valid collection of gadgets. For each $v \in V_A$ and three unused nodes $v^1, v^2, v^3 \in V_A$, let $\mathcal{S}'$ be the set obtained after calling* EXTRACTBITS$(\mathcal{S}, v, v^1, v^2, v^3)$*. Then $\mathcal{S}'$ is also valid and in every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game $\mathcal{G}' =$* BUILDGAME$(\mathcal{S}')$*, if $a = 8K\mathbf{x}[v]$ is well-positioned, then $\mathbf{x}[v_i] =_B b_i$, where $b_1b_2b_3$ is the binary representation of integer $\pi(a) \in [0:7]$.*

For each $\{v_i\}_{i\in[1:n]} \subset V_A$ and $3n$ *unused* nodes $\{v_i^+, v_i^-\}_{i\in[1:n]} \subset V_A$, let $\mathbf{p} \in \mathbb{R}_+^n$ denote the point encoded by $\{v_i\}_{i\in[1:n]}$. Think $\mathbf{p}$ as a point in $S = \{\mathbf{p}^k : 1 \le i \le n^3\}$. COLORINGSIMULATION$(\mathcal{S}, \{v_i\}_{i\in[1:n]}, \{v_i^+, v_i^-\}_{i\in[1:n]})$ simulates circuit C on input $\pi(\mathbf{p})$, given a valid collection $\mathcal{S}$: (i) Pick $3n$ unused nodes $\{v_{i,j}\}_{i\in[1:n]j\in[1:3]}$ in V_A. Call EXTRACTBITS$(\mathcal{S}, v_t, v_{t,1}, v_{t,2}, v_{t,3})$, for each $1 \le t \le n$; (ii) View the values of $\{v_{i,j}\}$ as $3n$ input bits of C. Insert the corresponding logic gadgets from $\{G_\vee, G_\wedge, G_\neg\}$ into $\mathcal{S}$ to simulate the evaluation of C, one for each gate, and place the $2n$ output bits in $\{v_i^+, v_i^-\}$.

Lemma 4.15 (Point Coloring)**.** *Let $\mathcal{S}'$ be the set of gadgets after calling the above* COLORINGSIMULATION*. For a pair of probability vectors $(\mathbf{x}, \mathbf{y})$, let $\mathbf{p}$ be the point with $p_i = 8K\mathbf{x}[v_i]$, $\mathbf{q} = \pi(\mathbf{p})$, and $\Delta_i^+[\mathbf{q}]$ and $\Delta_i^-[\mathbf{q}]$, for all i, be the output bits of C evaluated at $\mathbf{q}$. Then, $\mathcal{S}'$ is valid. Moreover, if $(\mathbf{x}, \mathbf{y})$ is an ϵ-well-supported equilibrium of* BUILDGAME$(\mathcal{S}')$*, and $\mathbf{p}$ is a well-positioned point, then $\mathbf{x}[v_i^+] =_B \Delta_i^+[\mathbf{q}]$ and $\mathbf{x}[v_i^-] =_B \Delta_i^-[\mathbf{q}]$ for all $i \in [1:n]$.*

Note that if the point $\mathbf{p}$ in the lemma above is not well-positioned, the values of $\{v_i^+, v_i^-\}$ could be arbitrary. Fortunately, because $\mathcal{S}'$ is valid and thus $(\mathbf{x}, \mathbf{y})$ must satisfy $\mathcal{P}$, we know $0 \le \mathbf{x}[v_i^+], \mathbf{x}[v_i^-] \le 1/K + \epsilon$.

We build $\mathcal{S}^U$ and thus $\mathcal{G}^U$ with a four-step construction.

Part 1 [*Equiangle Sampling Segment*]:
$\mathcal{S}^U = \emptyset$. Insert properly-valued G_ζ gadgets and G_+ gadgets to ensure that in each ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of the resulting game,

$$\mathbf{x}[v_i^k] = \min(\mathbf{x}[v_i^1] + (k-1)/(8K^2), \mathbf{x}_C[v_i^k]) \pm O(\epsilon),$$

for all $k \in [1:n^3]$ and $i \in [1:n]$.

Part 2 [*Point Coloring*]:
Pick $2n^4$ unused nodes $\{v_i^{k+}, v_i^{k-}\}_{i\in[1:n],k\in[1:n^3]}$ from V_A. For each integer $k \in [1:n^3]$, call function COLORINGSIMULATION$(\mathcal{S}^U, \{v_i^k\}, \{v_i^{k+}, v_i^{k-}\}_{i\in[1:n]})$.

Part 3 [*Summing up the Coloring Vectors*]:
Pick $2n$ unused nodes $\{v_i^+, v_i^-\}_{i\in[1:n]}$ from V_A and insert properly-valued $G_{\times\zeta}$ gadgets and G_+ gadgets to ensure in the resulting game, every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ satisfies

$$\mathbf{x}[v_i^+] = \textstyle\sum_{1\le k\le n^3} \left(\frac{1}{K}\,\mathbf{x}[v_i^{k+}]\right) \pm O(n^3\epsilon) \quad \text{and}$$
$$\mathbf{x}[v_i^-] = \textstyle\sum_{1\le k\le n^3} \left(\frac{1}{K}\,\mathbf{x}[v_i^{k-}]\right) \pm O(n^3\epsilon).$$

Part 4. For each $i \in [1:n]$, pick unused nodes $v_i', v_i'' \in V_A$ and $w_i^1, w_i^2, w_i^3 \in V_I$. Insert the following gadgets:

$$\text{INSERT}\left(\mathcal{S}^U, (G_+, v_i^0, v_i^+, v_i', nil, w_i^1)\right),$$
$$\text{INSERT}\left(\mathcal{S}^U, (G_-, v_i', v_i^-, v_i'', nil, w_i^2)\right),$$
$$\text{INSERT}\left(\mathcal{S}^U, (G_=, v_i'', nil, v_i^0, nil, w_i^3)\right).$$

4.6 Analysis of the Reduction

We now prove the correctness of our construction.

Let $(\mathbf{x}, \mathbf{y})$ be an ϵ-well-supported Nash equilibrium of $\mathcal{G}^U$. Recall $S = \{\mathbf{p}^k, \text{ with } p_i^k = 8K\mathbf{x}[v_i^k], 1 \le k \le n^3\}$ is the set of n^3 points that we want to produce. Let $I_G = I_G(S)$ and $I_B = I_B(S)$. For each $t \in I_G$, let $c_t \in [1 : n+1]$ be the color of $\mathbf{q}^t = \pi(\mathbf{p}^t)$ assigned by C, and for each $i \in [1 : n+1]$, $W_i = |\{t \in I_G \mid c_t = i\}|$.

By **Part 1** of our construction, we have

Lemma 4.16 (Not Too Many Poorly-Positioned Points)**.** *$|I_B| \le n$, and hence $|I_G| \ge n^3 - n$.*

Lemma 4.17 (Accommodated)**.** *$Q = \{\pi(\mathbf{p}^k), k \in I_G\}$ is accommodated and $|Q| \le n+1$.*

For each $1 \le k \le n^3$, let $\mathbf{r}^k$ denote the vector that, after **Part 2**, satisfies $r_i^k = \mathbf{x}[v_i^{k+}] - \mathbf{x}[v_i^{k-}]$, for all i. The construction and Lemma 4.9 guarantees that

Lemma 4.18 (Correct Encoding of Colors)**.** *For each $t \in I_G$, $\mathbf{r}^t = K\mathbf{z}^{c_t} \pm \epsilon$; for each $t \in I_B$, $\|\mathbf{r}^t\|_\infty \le 1/K + \epsilon$.*

Recall that **Part 3** sums up these n^3 vectors $\{\mathbf{r}^k\}$. Let $\mathbf{r}$ denote the vector that can be defined after **Part 3**, with $r_i = \mathbf{x}[v_i^+] - \mathbf{x}[v_i^-]$, for all $i : 1 \le i \le n$.

Ideally, with the gadgets inserted in **Part 4**, we wish to establish $\|\mathbf{r}\|_\infty = O(\epsilon)$. However, whether or not this condition holds depends on the values of $\{v_i^1\}_{1 \le i \le n}$. For example, in the case when $\mathbf{x}[v_i^1] = 0$, the magnitude of $\mathbf{x}[v_i^-]$ could be much larger than that of $\mathbf{x}[v_i^+]$. We are able to establish the following lemma which is sufficient to carry out our correctness proof of the reduction.

Lemma 4.19 (Well-Conditioned Solution)**.** *For all $1 \le i \le n$, if $\mathbf{x}[v_i^1] > 4\epsilon$, then $r_i = \mathbf{x}[v_i^+] - \mathbf{x}[v_i^-] > -4\epsilon$; if $\mathbf{x}[v_i^1] < 1/K - 2n^3/K^2$, then $r_i = \mathbf{x}[v_i^+] - \mathbf{x}[v_i^-] < 4\epsilon$.*

Now we start to show that Q is a panchromatic simplex of C. By Lemma 4.17, it suffices to show that $W_i > 0, \forall i$.

By **Part 3** of the construction and Lemma 4.18,

$$\begin{aligned} \mathbf{r} &= \frac{1}{K}\sum_{i \in I_G} \mathbf{r}^i + \frac{1}{K}\sum_{i \in I_B} \mathbf{r}^i \pm O(n^3\epsilon) \\ &= \sum_{1 \le i \le n+1} W_i\, \mathbf{z}^i + \frac{1}{K}\sum_{i \in I_B} \mathbf{r}^i \pm O(n^3\epsilon) \\ &= \mathbf{r}^G + \mathbf{r}^B \pm O(n^3\epsilon). \end{aligned}$$

where $\mathbf{r}^G = \sum_{1 \le i \le n+1} W_i \mathbf{z}^i$ and $\mathbf{r}^B = \sum_{i \in I_B} \mathbf{r}^i/K$. Since $|I_B| \le n$ and $\|\mathbf{r}^i\|_\infty \le 1/K + \epsilon$ for each $i \in I_B$, following Lemma 4.9, $\|\mathbf{r}^B\|_\infty = O(n/K^2)$.

As $|I_G| \ge n^3 - n$, we have $\sum_{1 \le i \le n+1} W_i \ge n^3 - n$. The following lemma shows that, if one of W_i is equal to zero, then $\|\mathbf{r}^G\|_\infty \gg \|\mathbf{r}^B\|_\infty$.

Lemma 4.20 (Color Gap)**.** *If one of W_i equals zero, then $\|\mathbf{r}^G\|_\infty \ge n^2/(3K^2)$, and thus $\|\mathbf{r}\|_\infty \gg 4\epsilon$.*

Therefore, if Q is not a panchromatic simplex, then one of the W_i's is equal to 0, and hence $\|\mathbf{r}\|_\infty \gg 4\epsilon$. Had the **Part 4** of our construction guaranteed that $\|\mathbf{r}\|_\infty = O(\epsilon)$, we would have completed the proof. As it is not always the case, we prove the following lemma to complete the proof.

Lemma 4.21 (Well-Conditioness)**.** *For all $i : 1 \le i \le n$, $4\epsilon < \mathbf{x}[v_i^1] < 1/K - 2n^3/K^2$.*

Proof. In this proof, we will use Lemma 4.22 below about the boundary conditions of circuit C. Recall $1/K = 2^{-6m}$, $\epsilon = 2^{-18m} = 1/K^3$ and $2^m > n$.

First, if there exists an integer $k : 1 \le k \le n$ such that $\mathbf{x}[v_k^1] \le 4\epsilon$, then $q_k^t = 0$ for all $t \in I_G$, according to **Part 1**. By Lemma 4.22.1, $W_{n+1} = 0$. Let l be the integer such that $W_l = \max_{1 \le i \le n} W_i$. As $\sum_{i=1}^{n+1} W_i = |I_G| \ge n^3 - n$, we have $W_l \ge n^2 - 1$. So, $r_l \ge W_l/K^2 - O(n/K^2) - O(n^3\epsilon) \gg 4\epsilon$. Now consider the following cases: (1) If $\mathbf{x}[v_l^1] < 1/K - 2n^3/K^2$, then we get a contradiction in Lemma 4.19. (2) If $\mathbf{x}[v_l^1] \ge 1/K - 2n^3/K^2$, then for all $t \in I_G$, $p_l^t = 8K(\min(\mathbf{x}[v_l^0] + t/K^2, \mathbf{x}_C[v_l^t]) \pm O(\epsilon)) > 1$ and hence $q_l^t > 0$. By Lemma 4.22.2, we have $W_l = 0$, which contradicts with the assumption.

Second, if there exists an integer $1 \le k \le n$ such that $\mathbf{x}[v_k^1] \ge 1/K - 2n^3/K^2$, then $q_k^t = 7$ for all $t \in I_G$. By Lemma 4.22.3, $W_k = 0$. If $W_{n+1} \ge n^2/2$, then $r_k \le -W_{n+1}/K^2 + O(n/K^2) + O(n^3\epsilon) \ll -4\epsilon$, which contradicts with the assumption that $\mathbf{x}[v_k^1] \ge 1/K - 2n^3/K^2 > 4\epsilon$ (see Lemma 4.19.1). Below, we assume $W_{n+1} < n^2/2$.

Let l be the integer such that $W_l = \max_{1 \le i \le n+1} W_i$. Because $W_k = 0$, we have $W_l \ge n^2 - 1$ and $l \ne k$. As $W_{n+1} < n^2/2$, $W_l - W_{n+1} > n^2/2 - 1$ and thus, $r_l \ge (W_l - W_{n+1})/K^2 - O(n/K^2) - O(n^3\epsilon) \gg 4\epsilon$. We now consider the following two cases: (1) If $\mathbf{x}[v_l^1] < 1/K - 2n^3/K^2$, then we get a contradiction in Lemma 4.19.2; (2) If $\mathbf{x}[v_l^1] \ge 1/K - 2n^3/K^2$, then $p_l^t > 1$ and thus $q_l^t > 0$ for all $t \in I_G$. By Lemma 4.22.4, we have $W_l = 0$ which contradicts with the assumption. □

Lemma 4.22. *For each $\mathbf{q} \in B^n$ and $1 \le k \ne l \le n$, (1) if $q_k = 0$, then Color$_C$ $[\mathbf{q}] \ne n+1$, (2) if $q_k = 0$ and $q_l > 0$, then Color$_C$ $[\mathbf{q}] \ne l$, (3) if $q_k = 7$, then Color$_C$ $[\mathbf{q}] \ne k$, and (4) if $q_k = 7$ and Color$_C$ $[\mathbf{q}] = l \ne k$, then $q_l = 0$.*

5 PPAD-Completeness of Brouwer^f

Our reduction starts with the 2D problem BROUWER^{f_2}. To simplify the proof, we modify the definition of problem BROUWER^f as follows: In the original definition, each valid Brouwer-mapping circuit C defines a color assignment from the search space to $\{1..., d+1\}$. In this section,

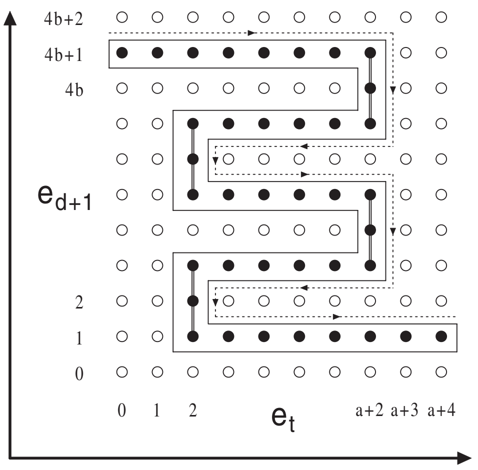

Figure 4. The 2-Dimensional View of Set W

```
Color_{C'} [p] in $A^{d+1}_{\mathbf{r}'}$ by $\mathbf{L}^3(T,t,a,b)$
 1: if $\mathbf{p} \in W$ then
 2:     $\text{Color}_{C'}[\mathbf{p}] = \text{Color}_C[\psi(\mathbf{p})]$
 3: else if $\mathbf{p} \in \partial\left(A^{d+1}_{\mathbf{r}'}\right)$ then
 4:     if there exists $i$ such that $p_i = 0$ then
 5:         $\text{Color}_{C'}[\mathbf{p}] = i_{\max} = \max\{\, i \mid p_i = 0 \,\}$
 6:     else $\text{Color}_{C'}[\mathbf{p}] = $ red
 7: else if $p_{d+1} \equiv 0 \pmod 4$ and $1 \le p_t \le a+1$
 8:     $\text{Color}_{C'}[\mathbf{p}] = d+1$
 9: else if $p_{d+1} \equiv 1, 2$ or $3 \pmod 4$ and $p_t = 1$
10:     $\text{Color}_{C'}[\mathbf{p}] = d+1$
11: else $\text{Color}_{C'}[\mathbf{p}] = $ red
```

Figure 5. $\mathbf{L}^3$(T,t,a,b)

we replace the color $d+1$ by a special color "red". In another word, if the output bits of C evaluated at $\mathbf{p}$ satisfy the $d+1$st case, then $\text{Color}_C[\mathbf{p}]=$"red".

The basic idea of our reduction is to iteratively embed an instance of BROUWER^{f_2} into one for a space one dimension higher. We use the following concept to describe such embedding processes. A triple $T=(C,d,\mathbf{r})$ is a *coloring triple* if $\mathbf{r}\in\mathbb{Z}^d$ with $r_i \ge 7$ for all $1\le i\le d$ and C is a valid Brouwer-mapping circuit with parameters d and $\mathbf{r}$. We let Size $[C]$ denote the number of gates plus the number of input and output variables in a circuit C.

Our embedding is defined by a sequence of three polynomial-time transformations: $\mathbf{L}^1(T,t,u)$, $\mathbf{L}^2(T,u)$, and $\mathbf{L}^3(T,t,a,b)$. They embed a coloring triple T into a larger T' *such that from every panchromatic simplex of T', one can find a panchromatic simplex of T efficiently.*

Both $\mathbf{L}^1$ and $\mathbf{L}^2$ are very simple operations. Given a $T=(C,d,\mathbf{r})$ and two integers $1\le t\le d$, $u>r_t$, $\mathbf{L}^1(T,t,u)$ pads dimension t to size u, i.e., it builds a new coloring triple $T'=(C',d,\mathbf{r}')$ with $r'_t=u$ and $r'_i=r_i, \forall\, i: 1\le i\ne t\le d$. For $u\ge 7$, $\mathbf{L}^2(T,u)$ adds a dimension to T by constructing $T'=(C',d+1,\mathbf{r}')$ such that $r'_{d+1}=u$ and $r'_i=r_i, \forall\, i: 1\le i\le d$.

$\mathbf{L}^3(T,t,a,b)$ is the one that does all the hard work.

Lemma 5.1 ($\mathbf{L}^3(T,t,a,b)$: Snake Embedding)**.** *Given a coloring triple $T=(C,d,\mathbf{r})$ and an integer $1\le t\le d$, if $r_t=a(2b+1)+5$ for some integers $a,b\ge 1$, then we can construct a new coloring triple $T'=(C',d+1,\mathbf{r}')$ that satisfies the following two conditions. (1) $r'_t=a+5$, $r'_{d+1}=4b+3$, and $r'_i=r_i$ for all $1\le i\ne t\le d$. Moreover, there exists a polynomial $g(n)$ such that Size $[C']=$ Size $[C]+O(g(\text{Size}\,[\mathbf{r}']))$ and T' can be computed in time polynomial in Size $[C']$. (2) From each panchromatic simplex P' of coloring triple T', one can compute a panchromatic simplex P of T in polynomial time.*

In the construction of C', we use a snake-pattern $W\subset A^{d+1}_{\mathbf{r}'}$ to realize the longer t^{th} dimension of $A^d_{\mathbf{r}}$ in the two-dimensional space defined by the shorter t^{th} and $d+1^{st}$ dimensions, see Figure 4 and 5. In Figure 5, ψ is a onto map (defined in the full version [7]) from W to $A^d_{\mathbf{r}}$. The proof of Lemma 5.1 can be found in [7].

By repeatedly applying the three operations $\mathbf{L}^{1-3}$, we can prove Theorem 2.5. The proof is relatively procedural and can be found in [7].

6 Extensions and Open Questions

As the fixed-points and Nash equilibria are fundamental to many other search and optimization problems, our results and techniques may have a broader scope of applications and implications.

Our hardness results can be naturally extended to both r-player games [20] and r-graphical games [12], for every fixed $r\ge 3$. Recently, there are a few exciting extensions of our work: Huang and Teng [11] applied our result to show that the problem of finding a market equilibrium in a Leontief exchange economy does not have a fully polynomial-time approximation scheme, unless **PPAD** $\subseteq$ **P**. They also show that the smoothed complexity of neither Scarf's general market equilibrium algorithm nor any other algorithm for Leontief market is likely polynomial. Chen, Teng, and Valiant [8] solved one of our conjectures by extending the **PPAD**-hardness result to the approximation of win-lose bimatrix games. In addition, we proved in [6] that the problem of computing a $1/\text{poly}(n)$-approximate Nash equilibrium remains **PPAD**-complete for a sparse bimatrix game $(\mathbf{A},\mathbf{B})$ in which each row and column of $\mathbf{A}$ and $\mathbf{B}$ contains at most a constant number of nonzero entries.

There remains a complexity gap on the approximation of Nash equilibria in two-player games: Lipton, Markakis and Mehta [16] show that an ϵ-approximate Nash equilibrium can be computed in $n^{O(\log n/\epsilon^2)}$-time, while this paper shows that no algorithm can find an ϵ-approximate Nash equilibrium in poly$(n, 1/\epsilon)$-time for ϵ of order $1/\text{poly}(n)$, unless **PPAD** $\subseteq$ **P**. However, our hardness result does not cover the case when ϵ is a constant between 0 and 1, or of order $1/\text{polylog}(n)$. Naturally, it is unlikely that finding an ϵ-approximate Nash equilibrium is **PPAD**-complete when ϵ is an absolute constant, for otherwise, all the search problems in **PPAD** would be solvable in $n^{O(\log n)}$-time, due to the result of [16].

Thinking optimistically, we would like to see the following conjectures to be true.

Conjecture 1 (PTAS for NASH). *There is an $O(n^{k+\epsilon^{-c}})$-time algorithm for finding an ϵ-approximate Nash equilibrium in a two-player game, for some constants c and k.*

Conjecture 2 (Smoothed 2-NASH: Constant σ). *There is an algorithm to compute a Nash equilibrium in a two-player game with smoothed complexity $O(n^{k+\sigma^{-c}})$ under perturbations with magnitude σ, for some constants c and k.*

We also conjecture that Theorem 4.3 remains true without any complexity assumption on **PPAD**. A positive answer would extend the result of Savani and von Stengel [23] to smoothed games.

7 Acknowledgements

In order to help readers who might be interested in both this paper and [4], the authors of both papers decided to use some common notations and terminologies in the presentation.

We would like to thank everyone who asked about the smoothed complexity of the Lemke-Howson algorithm. Part of Teng's work was done while visiting Tsinghua University and Microsoft Beijing Research Lab. His work is supported by the NSF grants CCR-0311430 and ITR CCR-0325630. Xiaotie Deng would like to thank the support of City University of Hong Kong for his research. Several critical ideas of this work were shaped when the authors were attending ISAAC2005 at Sanya, Hainan, China.

References

[1] John Reif, Nicole Immorlica, Steve Vavasis, Christos Papadimitriou, Mohammad Mahdian, Ding-Zhu Du, Santosh Vempala, Aram Harrow, Adam Kalai, Imre Bárány, Adrian Vetta, Jonathan Kelner and a number of other people asked whether the smoothed complexity of the Lemke-Howson algorithm or Nash Equilibria is polynomial, 2001–2005.

[2] I. Bárány, S. Vempala, and A. Vetta. Nash equilibria in random games. In *FOCS 2005*, pages 123–131.

[3] X. Chen and X. Deng. On the complexity of 2d discrete fixed point problem. In *ICALP 2006*, pages 489–500.

[4] X. Chen and X. Deng. Settling the complexity of 2-player nash-equilibrium. In *FOCS 2006*.

[5] X. Chen and X. Deng. 3-nash is ppad-complete. *ECCC, TR05-134*, 2005.

[6] X. Chen, X. Deng, and S.-H. Teng. Sparse games are hard. In *WINE 2006*.

[7] X. Chen, X. Deng, and S.-H. Teng. Computing nash equilibria: Approximation and smoothed complexity. *ECCC TR06-023*, 2006.

[8] X. Chen, S.-H. Teng, and P. A. Valiant. The approximation complexity of win-lose games. Manuscript: Tsinghua-BU-MIT, 2006.

[9] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a nash equilibrium. In *STOC 2006*.

[10] C. Daskalakis and C. Papadimitriou. Three-player games are hard. *ECCC, TR05-139*, 2005.

[11] L.-S. Huang and S.-H. Teng. On the approximation and smoothed complexity of leontief market equilibria. *ECCC, TR06-031*, 2006.

[12] M. J. Kearns, M. L. Littman, and S. P. Singh. Graphical models for game theory. In *UAI 2001*, pages 253–260.

[13] L. Khachian. A Polynomial Algorithm in Linear Programming. *Dokl. Akad. Nauk*, SSSR 244:1093–1096, *English translation in Soviet Math. Dokl. 20, 191–194*, 1979.

[14] C. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11:681–689, 1965.

[15] C. Lemke and J. J.T. Howson. Equilibrium points of bimatrix games. *J. Soc. Indust. Appl. Math.*, 12:413–423, 1964.

[16] R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *EC 2003*, pages 36–41.

[17] N. Megiddo and C. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoret. Comput. Sci.*, 81:317–324, 1991.

[18] O. Morgenstern and J. von Neumann. *The Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[19] J. Nash. Equilibrium point in n-person games. *Porceedings of the National Academy of the USA*, 36(1):48–49, 1950.

[20] J. Nash. Noncooperative games. *Annals of Mathematics*, 54:289–295, 1951,.

[21] C. Papadimitriou. Algorithms, games, and the internet. In *STOC 2001*, pages 749–753.

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

[23] R. Savani and B. von Stengel. Exponentially many steps for finding a nash equilibrium in a bimatrix game. In *FOCS 2004*, pages 258–267.

[24] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004, also STOC 2001.

[25] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms and heuristics: Progress and open questions. In L. M. Pardo, A. Pinkus, E. Süli and M. J. Todd, editor, *Foundations of Computational Mathematics*, pages 274–342. Cambridge University Press, 2006.