Settling the Complexity of Computing Two-Player Nash Equilibria

XI CHEN

Tsinghua University, Beijing, China

XIAOTIE DENG

City University of Hong Kong, Hong Kong, China

AND

SHANG-HUA TENG

Boston University and Akamai Technologies Inc., Boston, Massachusetts

14

This article combines and expands on material that appeared in two conference papers under the titles "Settling the Complexity of 2-Player Nash-Equilibrium," by Xi Chen and Xiaotie Deng, and "Computing Nash Equilibria: Approximation and Smoothed Complexity," by the three authors of this article. Both conference papers appeared in the *Proceedings of the 47th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, Los Alamitos, CA, 261–272 and 603–612, respectively. The result that BIMATRIX is **PPAD**-complete is from the first article. We also include the main result from the conference paper "Sparse Games are Hard," by the three authors of this article, presented at the *2nd International Workshop on Internet and Network Economics*, 262–273, 2006.

X. Chen's work was supported by the Chinese National Key Foundation Plan (2003CB317807 and 2004CB318108), the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant (2007CB807900 and 2007CB807901). X. Chen's work was also partially supported by NSF Grant DMS-0635607, CCF-0832797, and the Princeton Center for Theoretical Computer Science. X. Deng's work was supported by a grant from Research Grants Council of the Hong Kong Special Administrative Region (Project No. CityU 112707) and by City University of Hong Kong. S.-H. Teng's work was supported by the NSF grants CCR-0311430, CCR-0635102, and ITR CCR-0325630.

Part of X. Chen's work was done while visiting City University of Hong Kong. Part of S.-H. Teng's work was done while visiting Tsinghua University and Microsoft Beijing Research Lab.

Authors' addresses: X. Chen, Department of Computer Science, Princeton University, Princeton, NJ 08540, e-mail: csxichen@gmail.com; X. Deng, Department of Computer Science, City University of Hong Kong, Hong Kong, China, e-mail: csdeng@cityu.edu.hk; S.-H. Teng, Department of Computer Science, Boston University, Boston, MA, and Akamai Technologies Inc., Cambridge, MA, e-mail: steng@cs.bu.edu (Affiliation after the summer of 2009: Department of Computer Science, University of Southern California, Los Angeles, CA).

DOI 10.1145/1516512.1516516 http://doi.acm.org/10.1145/1516512.1516516

Abstract. We prove that BIMATRIX, the problem of finding a Nash equilibrium in a two-player game, is complete for the complexity class **PPAD** (Polynomial Parity Argument, Directed version) introduced by Papadimitriou in 1991.

Our result, building upon the work of Daskalakis et al. [2006a] on the complexity of four-player Nash equilibria, settles a long standing open problem in algorithmic game theory. It also serves as a starting point for a series of results concerning the complexity of two-player Nash equilibria. In particular, we prove the following theorems:

—BIMATRIX does not have a fully polynomial-time approximation scheme unless every problem in **PPAD** is solvable in polynomial time.

—The smoothed complexity of the classic Lemke-Howson algorithm and, in fact, of any algorithm for BIMATRIX is not polynomial unless every problem in **PPAD** is solvable in randomized polynomial time.

Our results also have a complexity implication in mathematical economics:

—Arrow-Debreu market equilibria are **PPAD**-hard to compute.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures; geometrical problems and computations*

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Two-player game, Nash equilibrium, Brouwer's fixed point, PPAD-completeness, smoothed analysis, Lemke-Howson algorithm, Sperner's lemma, Arrow-Debreu market

ACM Reference Format:

Chen, X., Deng, X., and Teng, S.-H. 2009. Settling the complexity of computing two-player Nash equilibria. J. ACM 56, 3, Article 14 (May 2009), 57 pages. DOI = 10.1145/1516512.1516516 http://doi.acm.org/10.1145/1516512.1516516

1. *Introduction*

In 1944, Morgenstern and von Neumann [1947] initiated the study of game theory and its applications to economic behavior. At the center of their study was von Neumann's minimax equilibrium solution for two-player zero-sum games [von Neumann 1928]. In a two-player zero-sum game, one player's gain is equal to the loss of the other. They observed that any general n-player (non-zero-sum) game can be reduced to an $(n+1)$-player zero-sum game. Their work went on to introduce the notion of cooperative games and proposed the concept of stable sets as the rational outcomes for games of multiple players.

In 1950, following the original spirit of Morgenstern and von Neumann's work on two-player zero-sum games, Nash [1951, 1950] formulated an equilibrium concept for non-cooperative games among multiple players. This concept is now commonly referred to as the *Nash equilibrium*. It uses the fixed point for individual optimal strategies introduced in von Neumann [1928] to capture the notion of individual rationality: Each player's strategy is a best response to the other players' strategies. Nash proved that every n-player game has an equilibrium point [Nash 1951; Leonard 1994]. His original proof was based on Brouwer's Fixed Point Theorem [Brouwer 1910]. David Gale suggested the use of Kakutani's Fixed Point Theorem [Kakutani 1941] to simplify the proof. While von Neumann's Minimax Theorem for two-player zero-sum games can be proved by linear programming duality, the fixed point approach to Nash's Equilibrium Theorem seems to be necessary; even for the two-player case, linear programming duality alone does not seem to be sufficient to derive Nash's theorem.

The concept of Nash equilibrium has had a tremendous influence on economics, as well as on other social and natural science disciplines [Holt and Roth 2004]. Nash's approach to noncooperative games has played an essential role in shaping mathematical economics, which considers agents with competing individual interests; Nash's fixed-point based proof technique also enabled Arrow and Debreu [1954] to establish a general existence theorem for market equilibria.

However, the existence proofs based on fixed point theorems do not usually lead to efficient algorithms for finding equilibria. In fact, in spite of many remarkable breakthroughs in algorithmic game theory and mathematical programming, answers to several fundamental questions about the computation of Nash and Arrow-Debreu equilibria remain elusive. The most notable open problem is that of deciding whether there is a polynomial-time algorithm for finding an equilibrium point in a two-player game.

In this article, building on a recent work of Daskalakis et al. [2006a] on the complexity of four-player Nash equilibria, we settle the complexity of computing a two-player Nash equilibrium. We also extend our result to the approximation of Nash equilibrium and settle the smoothed complexity of computing a two-player Nash equilibrium. In the rest of this section, we review previous results on the computation of Nash equilibria, state our main results, and discuss their extension to the computation of market equilibria.

1.1. FINITE-STEP EQUILIBRIUM ALGORITHMS. Since Nash and Arrow-Debreu's pioneering work, great progress has been made on finding constructive and algorithmic proofs for equilibrium theorems. The advances in equilibrium computation can be chronologically classified into the following two periods:

—*Computability Period*. In this period, the main objective was to design equilibrium algorithms that terminate in a finite number of steps and to determine which equilibrium problems do not allow finite step algorithms.

—*Complexity Period*. In this period, the main objective has been to develop polynomial-time algorithms for computing equilibria and to characterize the complexity of equilibrium computation.

We will discuss the first period in this subsection and the second period in the next three subsections.

Von Neumann's proof of the minimax theorem using duality leads to a linear programming formulation of the set of equilibria in a two-player zero-sum game. One can apply the simplex algorithm to compute, in a finite number of steps,[1] an equilibrium in a two-player zero-sum game with rational payoffs. More than a decade after Nash's seminal work, Lemke and Howson [1964] developed a path-following, simplex-like algorithm for finding a Nash equilibrium in general two-player games. Their algorithm terminates in a finite number of steps for all two-player games with rational payoffs.

[1]The simplex algorithm terminates in a finite number of steps in the Turing model as well as in various computational models involving real numbers, such as the model defined by Ko [1991] and the model defined by Blum et al. [1989].

The Lemke-Howson algorithm has been extended to games with more than two players [Wilson 1971]. However, due to Nash's observation that there are rational three-player games all of whose equilibria are irrational, finite-step algorithms become harder to obtain for games with three or more players. Similarly, some exchange economies do not have any rational Arrow-Debreu equilibrium. The absence of a rational equilibrium underscores the continuous nature of equilibrium computation. Brouwer's Fixed Point Theorem—any continuous function f from a convex compact set, such as a simplex or a hypercube, to itself has a fixed point—is inherently continuous.

Due to this continuity and irrationality, one has to be careful when defining search problems for finding an equilibrium and a fixed point in the classical Turing model. There are two known approaches to ensure the existence of a solution with a finite description. The first approach uses a symbolic representation of an equilibrium or a fixed point. For example, one can represents equilibrium by a number of irreducible integer polynomials whose roots are entries of the equilibrium [Lipton and Markakis 2004]. The second approach uses approximation: one introduces imprecision and looks for an approximate equilibrium or an approximate fixed point [Scarf 1967a, 1967b; Papadimitriou 1991; Hirsch et al. 1989; Deng et al. 2003]. For example, one standard definition of an approximate fixed point of a continuous function f is a point $\mathbf{x}$ such that $\|f(\mathbf{x}) - \mathbf{x}\| \leq \epsilon$ for a given $\epsilon > 0$ [Scarf 1967a]. In this article, we only focus on the latter approach.

Sperner [1928] discovered a discrete fixed point theorem that led to one of the most elegant proofs of Brouwer's Fixed Point Theorem. Suppose that Ω is a d-dimensional simplex with vertices $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d+1}$, and that $\mathcal{S}$ is a simplicial decomposition of Ω. Recall that a *simplicial decomposition* $\mathcal{S}$ of Ω is a finite collection of d-dimensional simplices, whose union is Ω, and for all simplices S_1 and S_2 in $\mathcal{S}$, $S_1 \cap S_2$ is either empty or a face of both S_1 and S_2 [Edelsbrunner 2006]. We use $V(\mathcal{S})$ to denote the union of the vertices of the simplices in $\mathcal{S}$. Suppose Π assigns to each vertex in $V(\mathcal{S})$ a color from $\{1, 2, \ldots, d+1\}$ such that, for every vertex $\mathbf{v}$ in $V(\mathcal{S})$, $\Pi(\mathbf{v}) \neq i$ if the ith component of the barycentric coordinates[2] of $\mathbf{v}$, with respect to $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{d+1}$, is 0. Then, Sperner's Lemma asserts that there exists a simplex in $\mathcal{S}$ that contains all colors. This fully colored simplex is often referred to as a *panchromatic simplex* or a *Sperner simplex* of $(\mathcal{S}, \Pi)$. Consider a Brouwer function f with Lipschitz constant L over the simplex Ω. Suppose further that the diameter of each simplex in $\mathcal{S}$ is at most ϵ/L. One can define a color assignment Π_f as follows: For each $\mathbf{v} \in V(\mathcal{S})$, we view $f(\mathbf{v}) - \mathbf{v}$ as a vector from $\mathbf{v}$ to $f(\mathbf{v})$; Extend this vector until it reaches a face of Ω; Suppose the face is spanned by $\{\mathbf{v}^1, \ldots, \mathbf{v}^{d+1}\} - \{\mathbf{v}^i\}$ for some i, then we set $\Pi_f(\mathbf{v}) = i$. One can show that each panchromatic simplex in $(\mathcal{S}, \Pi_f)$ must have a vertex $\mathbf{v}$ satisfying $\|f(\mathbf{v}) - \mathbf{v}\| \leq \Theta(\epsilon)$. Thus, a panchromatic simplex of $(\mathcal{S}, \Pi_f)$ can be viewed as an approximate, discrete fixed point of f.

Inspired by the Lemke-Howson algorithm, Scarf developed a path-following algorithm, using simplicial subdivision, for computing approximate fixed points [Scarf 1967a] and competitive equilibrium prices [Scarf 1967b]. The path-following

[2] Let Ω be a d-dimensional simplex in $\mathbb{R}^d$ with vertices $\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}$. Every point $\mathbf{v} \in \mathbb{R}^d$ can be uniquely expressed as $\mathbf{v} = \sum_{1 \leq i \leq d+1} \lambda_i \mathbf{v}_i$, where $\sum_{1 \leq i \leq d+1} \lambda_i = 1$. The *barycentric coordinates* of $\mathbf{v}$ with respect to $\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}$ is $(\lambda_1, \ldots, \lambda_{d+1})$.

method has also had extensive applications in mathematical programming and has since grown into an algorithm-design paradigm in optimization and equilibrium analysis.

1.2. COMPUTER SCIENCE VIEW OF NASH EQUILIBRIA. Since the 1960's, the theory of computation has shifted its focus from whether problems can be solved on a computer to how efficiently problems can be solved on a computer. The field has gained maturity with rapid advances in algorithm design, algorithm analysis, and complexity theory. Problems are categorized into complexity classes capturing the difficulty of decision, search, and optimization problems. The complexity classes **P**, **RP**, and **BPP**, and their search counterparts such as **FP**, have become the standard classes for characterizing tractable computational problems.[3]

The rise of the internet has greatly enhanced the desire to find fast algorithms for computing equilibria [Papadimitriou 2001]. Furthermore, the internet has created a surge of human activities that make the computation, communication and optimization of participating agents accessible at microeconomic levels. Efficient computation is instrumental in supporting basic operations, such as pricing, in this large scale on-line market [Sandholm 2000]. Many new problems in game theory and economics have been introduced. In the meantime, classical problems in game theory and economics have been studied actively by complexity theorists [Papadimitriou 2001]. Algorithmic game theory has grown into a highly interdisciplinary field intersecting economics, mathematics, operations research, numerical analysis, and theoretical computer science.

Khachiyan [1979] showed that the ellipsoid algorithm can solve a linear program in polynomial time. Shortly thereafter, Karmarkar [1984] improved the complexity for solving linear programming with his path-following, interior-point algorithm. His work initiated the implementation of linear programming algorithms that are both theoretically sound and practically efficient. It had been a challenge for some time (see Condon et al. [1999]) to explain why some algorithms, most notably the simplex algorithm for linear programming, though known to take exponential time in the worst case [Klee and Minty 1972], were nevertheless very fast in practice. Spielman and Teng [2004] introduced a new algorithm analysis framework, called *smoothed analysis*, based on perturbation theory, to provide rigorous complexity-theoretic justification for the good practical performance of the simplex algorithm. They proved that the smoothed complexity of the simplex algorithm with the *shadow-vertex pivoting rule* is polynomial. As a result of these developments in linear programming, one can compute an equilibrium solution of a two-player zero-sum game in polynomial time using the ellipsoid or interior-point algorithms or in smoothed polynomial time using the simplex algorithm.

However, no polynomial-time algorithm has been found for computing discrete fixed points or approximate fixed points, rendering the equilibrium proofs based on fixed point theorems nonconstructive in the view of polynomial-time computability.

The difficulty of discrete fixed point computation is partially justified in the query model. Hirsch et al. [1989] provided an exponential lower bound on the

[3] **FP** stands for Function Polynomial-Time. In this article, as we only consider search problems, we will (ab)use **P** and **RP** to denote the classes of search problems that can be solved in polynomial time and in randomized polynomial time, respectively.

number of function evaluations necessary to find a discrete fixed point, even in two dimensions, assuming algorithms only have a black-box access to the fixed point function. Their bound has been made tight [Chen and Deng 2005b], and extended to the randomized query model [Chen and Teng 2007] as well as to the quantum query model [Friedl et al. 2005; Chen et al. 2008].

1.3. COMPUTATIONAL COMPLEXITY OF NASH EQUILIBRIA: PPAD. Motivated by the pivoting structure used in the Lemke-Howson algorithm, Papadimitriou [1991] introduced the complexity class **PPAD**. **PPAD** is an abbreviation for *Polynomial Parity Argument, Directed version*. He introduced several search problems concerning the computation of discrete fixed points. For example, he defined the problem SPERNER to be the search problem of finding a Sperner simplex given a Boolean circuit that assigns colors to a particular simplicial decomposition of a hypercube. Extending the model of Hirsch et al. [1989], he also defined a search problem for computing approximate Brouwer fixed points. He proved that in three dimensions, these fixed point problems are complete for the **PPAD** class. Chen and Deng [2006] proved that the two-dimensional discrete fixed point problem is also **PPAD**-complete. A partial solution, showing that the problem is **PPAD**-complete for the locally two-dimensional case, was obtained by Friedl et al. [2006] at about the same time.

Papadimitriou [1991] also proved that BIMATRIX, the problem of finding a Nash equilibrium in a two-player game with rational payoffs, is a member of **PPAD**. His proof can be extended to show that finding a (properly defined) approximate equilibrium in a non-cooperative game among three or more players is also in **PPAD**. Thus, if these problems are **PPAD**-complete, then the search problem of finding an equilibrium is polynomial-time equivalent to the search problem of finding a discrete fixed point.

It is conceivable that Nash equilibria might be easier to compute than discrete fixed points. In fact, by taking advantage of the special structure of normal form games, Lipton et al. [2004] developed a subexponential time algorithm for finding an approximate Nash equilibrium in these games. In their notion of an ϵ-approximate Nash equilibrium, for a positive parameter ϵ, each player's strategy is at most an additive ϵ worse than the best response to other players' strategies. They proved that if all payoffs are in [0, 1], then an ϵ-approximate Nash equilibrium can be found in $n^{O(\log n/\epsilon^2)}$ time.

In a recent work, Daskalakis et al. [2006a] proved that the problem of computing a Nash equilibrium in a game among four or more players is complete for **PPAD**. To cope with the fact that equilibria may not be rational, they considered an approximation version of the problem by allowing exponentially small errors. The complexity result was soon extended to three-player games [Chen and Deng 2005a; Daskalakis and Papadimitriou 2005].

The results of Daskalakis et al. [2006a], Chen and Deng [2005a], and Daskalakis and Papadimitriou [2005] characterize the complexity of computing k-player Nash equilibria for $k \geq 3$. These latest complexity advances left open the two-player case.

1.4. COMPUTING TWO-PLAYER NASH EQUILIBRIA AND SMOOTHED COMPLEXITY. There have been amazing parallels between the progress concerning the two-player zero-sum game and the progress concerning the general two-player game. First, von Neumann proved the existence of an equilibrium for the zero-sum game, then Nash did the same for the general game. Both classes of games have rational equilibria when payoffs are rational. Second, more than a decade after von Neumann's Minimax Theorem, Dantzig developed the simplex algorithm, which can find a solution of a two-player zero-sum game in a finite number of steps. Again, more than a decade after Nash's work, Lemke and Howson developed their finite-step algorithm for BIMATRIX. Then, about a quarter century after their respective developments, both the simplex algorithm [Klee and Minty 1972] and the Lemke-Howson algorithm [Savani and von Stengel 2004] were shown to have exponential worst-case complexity.

A half century after von Neumann's Minimax Theorem, Khachiyan proved that the ellipsoid algorithm can solve a linear program and hence can find a solution of a two-player zero-sum game with rational payoffs in polynomial time. Shortly after that, Borgwardt [1982] showed that the simplex algorithm has polynomial average-case complexity. Then, Spielman and Teng [2004] proved that the smoothed complexity of the simplex algorithm is polynomial. If history is of any guide, then a half century after Nash's Equilibrium Theorem, one could be hopeful of proving the following two natural conjectures:

—*Polynomial 2*-NASH *Conjecture*: There exists a (weakly) polynomial-time algorithm for BIMATRIX.
—*Smoothed Lemke-Howson Conjecture*: The smoothed complexity of the Lemke-Howson algorithm for BIMATRIX is polynomial.

An upbeat attitude toward the first conjecture has been encouraged by the following two facts. First, unlike three-player games, every rational two-player game has a rational equilibrium. Second, a key technical step in the **PPAD**-hardness proofs for three/four-player games fails to extend to two-player games [Daskalakis et al. 2006a; Chen and Deng 2005a; Daskalakis and Papadimitriou 2005]. The Smoothed Lemke-Howson Conjecture was asked by a number of people[4] (John Reif and others 2001–2005). This conjecture is a special case of the following one, which was posted by Spielman and Teng [2006] in a survey of smoothed analysis of algorithms and inspired by the result of Bárány et al. [2005] that an equilibrium of a random two-player game can be found in polynomial time:

—*Smoothed 2*-NASH *Conjecture*: The smoothed complexity of BIMATRIX is polynomial.

1.5. OUR CONTRIBUTIONS. Despite much effort in the last half century, it remains an open problem for characterizing the algorithmic complexity of two-player Nash equilibria. Thus, BIMATRIX, the most studied computational problem about Nash equilibria, stood out as the last open problem in equilibrium computation for normal form games. Papadimitriou [2001] named it, along with FACTORING, as one of the two "most important concrete open questions" at the boundary of **P**. In fact, ever since Khachiyan's discovery [Khachian 1979], BIMATRIX has been on the frontier of natural problems possibly solvable in polynomial time. Now, it is also on the frontier of the hard problems, assuming **PPAD** is not contained in **P**.

[4] The question that has come up most frequently during presentations and talks on smoothed analysis is the following: Does the smoothed analysis of the simplex algorithm extend to the Lemke-Howson algorithm?

In this article, building on the result of Daskalakis, Goldberg, and Papadimitriou [Goldberg and Papadimitriou 2006; Daskalakis et al. 2006a] (see Section 2 for a detailed discussion), we settle the computational complexity of the two-player Nash equilibrium problem. In later sections, we prove:

THEOREM 1.1. BIMATRIX *is* **PPAD**-*complete.*

Theorem 1.1 demonstrates that, even in this simplest form of noncooperative games, equilibrium computation is polynomial-time equivalent to discrete fixed point computation. In particular, we show that from each discrete Brouwer function f, we can build a two-player game $\mathcal{G}$ and a polynomial-time map Π from the Nash equilibria of $\mathcal{G}$ to the fixed points of f. Our proof complements Nash's proof that for each two-player game $\mathcal{G}$, there is a Brouwer function f and a map Φ from the fixed points of f to the equilibrium points of $\mathcal{G}$.

The success in proving the **PPAD**-completeness of BIMATRIX inspired us to attempt to disprove the Smoothed 2-NASH Conjecture. A connection between the smoothed complexity and the approximation complexity of Nash equilibria [Spielman and Teng 2006, Proposition 9.12], then led us to prove the following result:

THEOREM 1.2. *For any $c > 0$, the problem of computing an n^{-c}-approximate Nash equilibrium of a two-player game is* **PPAD**-*complete.*

This result enables us to establish the following theorem about the approximation complexity of Nash equilibria. It also enables us to answer the question about the smoothed complexity of the Lemke-Howson algorithm and disprove the Smoothed 2-NASH Conjecture assuming **PPAD** is not contained in **RP**.

THEOREM 1.3. BIMATRIX *does not have a fully polynomial-time approximation scheme unless* **PPAD** is contained in **P**.

THEOREM 1.4. BIMATRIX *is not in smoothed polynomial time unless* **PPAD** *is contained in* **RP**.

Consequently, it is unlikely that the $n^{O(\log n/\epsilon^2)}$-time algorithm of Lipton et al. [2004], 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 Bárány et al. [2005] can be extended to the smoothed model.

1.6. IMPLICATIONS. Because two-player Nash equilibria enjoy several structural properties that Nash equilibria with three or more players do not have, our result enables us to answer additional long-standing open questions in mathematical economics. In particular, we derive the following important corollary.

COROLLARY 1.5. *Arrow-Debreu market equilibria are* **PPAD**-*hard to compute.*

To prove the corollary, we use a recent discovery of Ye [2005] (see also Codenotti et al. [2006]) on the connection between two-player Nash equilibria and Arrow-Debreu equilibria in two-group Leontief exchange economies.

We further refine our reduction to show that a Nash equilibrium in a *sparse* two-player game is **PPAD**-complete to compute and **PPAD**-hard to approximate in fully polynomial time [Chen et al. 2006] (see Section 10.1 for details).

Applying a recent reduction of Abbott et al. [2005], our result also implies the following corollary.

COROLLARY 1.6. WIN-LOSE BIMATRIX[5] *is* **PPAD**-*complete.*

In a subsequent work, Chen et al. [2007] extended the result to the approximation complexity of WIN-LOSE BIMATRIX; Huang and Teng [2007] extended both the smoothed complexity and the approximation results to the computation of Arrow-Debreu equilibria. Using the connection between Nash equilibria and Arrow-Debreu equilibria, our complexity result on sparse games can be extended to market equilibria in economies with sparse exchange structures [Chen et al. 2006b].

2. *Overview with Proof Sketches*

In this section, we discuss previous work that our results build upon as well as the new techniques and ideas that we introduce. As this article is somewhat long, this section also serves as a shorter, high-level description of the proofs. In the longer and more complete sections to follow, we will present the technical details of our results.

2.1. THE DGP FRAMEWORK. Technically, we use the proof framework developed by Daskalakis, Goldberg, and Papadimitriou [Goldberg and Papadimitriou 2006; Daskalakis et al. 2006a] in their work for characterizing the complexity of four-player Nash equilibria. In our proofs, we also introduce several ideas that are crucial for us to resolve the complexity of two-player Nash equilibria.

The framework of Daskalakis, Goldberg, and Papadimitriou, which we will refer to as the *DGP framework*, uses the following steps to establish that the problem of computing an exponentially accurate approximate Nash equilibrium in a game among four or more players is complete for **PPAD**.

(1) It defines a *3-dimensional discrete fixed point problem*, 3-DIMENSIONAL BROUWER, and proves that it is complete for **PPAD**.
(2) It establishes a geometric lemma (see Section 8.1), which introduces a *sampling* and *averaging* technique, to characterize discrete fixed points. This lemma provides a computationally efficient way to express the conditions of discrete fixed points and is the basis of the reduction in the next step.
(3) It reduces 3-DIMENSIONAL BROUWER to degree-3 graphical games, a class of games proposed in Kearns et al. [2001]. For this purpose, it constructs a set of gadgets, that is, a set of small graphical games for which the entries of every exponentially accurate approximate Nash equilibrium satisfy certain relations. These relations include the arithmetic relations ("addition", "subtraction", and "multiplication"), the logic relations ("and" and "or"), and several other relations ("brittle comparator" and "assignment"). It then systematically connects and combines these gadgets to simulate the input Boolean circuit of 3-DIMENSIONAL BROUWER and to encode the geometric lemma. The reduction

[5] A win-lose game is a game whose payoff entries are either 0 or 1. Here, WIN-LOSE BIMATRIX denotes the problem of finding a Nash equilibrium in a two-player win-lose game.

scheme in this framework creatively encodes fixed points by exponentially accurate approximate Nash equilibria.

(4) Finally, it reduces the graphical game to a four-player game. This step introduces the idea of using the *matching pennies game* (see Section 7.3) to enforce the players to play the strategies (almost) uniformly. One of the pivoting elements of this framework is a new concept of approximate Nash equilibria which measures the pairwise stability of pure strategies (see Section 3 for formal definition). This notion of approximate Nash equilibrium is different from the ϵ-approximate Nash equilibrium used in Lipton et al. [2004].

With further refinements, as shown in Chen and Deng [2005a] and Daskalakis and Papadimitriou [2005], the **PPAD**-completeness result can be extended to the computation of an exponentially accurate approximate Nash equilibrium in a three-player game.

2.2. A HIGH-LEVEL DISCUSSION OF OUR PROOFS. Below, we outline the important steps we take in proving the main theorems of this article—Theorems 1.1, 1.2, 1.3, and 1.4.

2.2.1. ***PPAD**-Completeness of* BIMATRIX. To prove Theorem 1.1, in principle, we follow the first two steps of the DGP framework and make some modifications to the last two steps. The reason why we only need two players instead of four is due to the following observations:

(1) We observe that the multiplication operation is unnecessary in the reduction from 3-DIMENSIONAL BROUWER to graphical games [Daskalakis et al. 2006a], and come up with an approach to utilize this simple yet important observation.

(2) We realize that Step 3 in the DGP framework can be conceptually divided into two steps, which we will refer to as Steps 3.1 and 3.2:

—In Step 3.1, it builds a constraint system from the input Boolean circuit of 3-DIMENSIONAL BROUWER. The system consists of a collection of relations (arithmetic, logic, and others) among a set of real variables. Every exponentially accurate solution to the system—that is, an assignment to the variables that approximately satisfies all the relations—can be transformed in polynomial time back to a discrete fixed point of the original 3-DIMENSIONAL BROUWER problem.

—In Step 3.2, it simulates this constraint system with a degree-3 graphical game (by simulating each relation with an appropriate gadget).

(3) We develop a method to directly reduce a multiplication-free constraint system to a two-player game, without using graphical games as an intermediate step.

In order to better express the multiplication-free constraint system, we introduce a concept called the *generalized circuit* (see Section 5.2), which might be interesting on its own. On the one hand, the generalized circuit is a direct analog of the graphical games used in Goldberg and Papadimitriou [2006] and Daskalakis et al. [2006a]. On the other hand, the generalized circuit is a natural extension of the classical algebraic circuit—the pivotal difference is that the underlying directed graph of a generalized circuit may contain cycles, which is necessary for expressing fixed points. Using this intermediate structure, we follow Step 3.1 of the DGP framework to show that 3-DIMENSIONAL BROUWER can be reduced to the computation of an exponentially accurate solution in a generalized circuit.

As an instrumental step in our proof, we give a polynomial-time reduction from the problem of finding an exponentially accurate solution in a generalized circuit to BIMATRIX, hence proving that BIMATRIX is **PPAD**-complete. A subtle but critical point of this reduction is that it may connect some exact Nash equilibria of the obtained two-player game with only approximate solutions to the original generalized circuit. In contrast, in Step 4 of the reduction of Daskalakis et al. [2006a] every exact Nash equilibrium of the obtained four-player game can be transformed back into an exact Nash equilibrium of the original graphical game. The loss of exactness in our reduction is especially necessary because every rational two-player game always has a rational Nash equilibrium. However, like the graphical games and three-player games, there exist rational generalized circuits that only have irrational solutions.

In the recent journal version of Daskalakis et al. [2009], a short alternative description of our proof of Theorem 1.1, using the language of graphical games, can be found. Daskalakis et al. [2009] show that Theorem 1.1 follows from their proof of the result for three players (with two extra pages), by reducing their degree-3 graphical games directly to two-player games.

2.2.2. *Fully Polynomial-Time Approximation of Nash Equilibria.* There is a fundamental reason why the DGP framework and our approach of the previous subsection do not immediately prove Theorems 1.2 and 1.3: The underlying grid of the **PPAD**-complete 3-DIMENSIONAL BROUWER must have an exponential number of points in some dimension. Thus, in order to specify a point in the grid, one needs $\Theta(n)$-bits for that dimension. Then, in order to encode the discrete fixed points of 3-DIMENSIONAL BROUWER directly with approximate Nash equilibria, the latter must be exponentially accurate. In order to establish Theorems 1.2 and 1.3, however, we need to encode the discrete fixed points of a **PPAD**-complete fixed point problem with polynomially-accurate approximate Nash equilibria!

We consider a natural high-dimensional extension of 3-DIMENSIONAL BROUWER. The observation is the following. The underlying grid for 3-DIMENSIONAL BROUWER is $\{0, 1, \dots, 2^n\}^3$. It has 2^{3n} cells, each of which is a cube and can be identified by three n-bit integers. Note that the n-dimensional grid $\{0, 1, \dots, 8\}^n$ also has 2^{3n} cells, each of which is an n-dimensional hypercube. Each hypercube in this high-dimensional grid can be identified by n three-bit integers. Thus, we need much less precision in each dimension.

The high-dimensional discrete fixed point problem comes with its own challenges. In 3-DIMENSIONAL BROUWER of Daskalakis et al. [2006a] each vertex of the 3D grid is colored with one of the 4 colors from $\{1, 2, 3, 4\}$, specified by a Boolean circuit that guarantees some boundary conditions. As a search problem, we are given this circuit and are asked to find a *panchromatic* cube whose vertices contain all four colors. Similarly, in the high-dimensional discrete fixed point problem, which we will refer to as BROUWER, each vertex of the n-dimensional grid is colored with one of the $n + 1$ colors from $\{1, \dots, n, n + 1\}$, also specified by a Boolean circuit. However, computationally, we can no longer define a discrete fixed point as a *panchromatic* hypercube. In n dimensions, a hypercube has 2^n vertices, which is exponential in n—too many for verifying the panchromatic condition in polynomial time. Following Sperner [1928] and the intuition of

3-DIMENSIONAL BROUWER of Daskalakis et al. [2006a] we define a discrete fixed point as a panchromatic simplex inside a hypercube. We then prove that BROUWER is also **PPAD**-complete.

The exponential curse of dimensionality, often referred to by computational geometers, goes beyond the definition of discrete fixed points: the original sampling-and-averaging technique used in Step 3.1 of the DGP framework does not seem to provide a computationally efficient way to express the conditions of fixed points in high dimensions. We develop a new geometric sampling method (see Lemma 8.2) for overcoming this curse of dimensionality.

Now, if we follow the original DGP framework with BROUWER as the starting point and make use of our new sampling method in Step 3.1, we can prove that the problem of computing a Nash equilibrium in a four-player game does not have a fully-polynomial-time approximation scheme, unless **PPAD** is in **P**. To prove Theorems 1.2 and 1.3, we follow our modification to the DGP framework. In particular, we use the new sampling method to reduce BROUWER to the computation of a polynomially accurate solution to a generalized circuit, and then further to the computation of a polynomially approximate Nash equilibrium in a two-player game. In the first reduction, we only need polynomial accuracy because the side length of BROUWER is a constant (in contrast to 3-DIMENSIONAL BROUWER, in which the side length is exponential).

Finally, to establish the approximation result for the commonly accepted ϵ-approximate Nash equilibrium, we derive an equivalence relation (see Lemma 3.2) between the ϵ-approximate Nash equilibrium and the new approximation notion used in the DGP framework.

2.2.3. *The Smoothed Complexity of Nash Equilibria*. The proof of Theorem 1.4 is then the simplest part of the article. It follows directly from Theorem 1.2 and an observation of Spielman and Teng (see Proposition 9.12 of Spielman and Teng [2006]) on the connection between the smoothed complexity and approximation complexity of Nash equilibria.

2.3. ORGANIZATION OF THE ARTICLE. In the rest of the article, we will prove Theorem 1.2, which implies Theorem 1.1, and derive Theorem 1.4. We organized the article as follows.

In Section 3, we review concepts in equilibrium theory. We also prove an important equivalence between various notions of approximate Nash equilibria. In Section 4, we recall the complexity class **PPAD**, the concept of polynomial-time reduction among search problems, and the smoothed analysis framework. In Section 5, we introduce two concepts: high-dimensional discrete Brouwer fixed points and *generalized circuits*, followed by the definitions of two search problems based on these concepts. In Section 6, we state our main results and provide an outline of the proofs. In Section 7, we show that one can simulate generalized circuits with two-player games. In Section 8, we show that discrete fixed points can be modeled by generalized circuits. In Section 9, we prove a **PPAD**-completeness result for a large family of high-dimensional discrete fixed point problems. In Section 10, we discuss extensions of our work and present several open questions and conjectures motivated by this research. In particular, we show that sparse BIMATRIX does not have a fully polynomial-time approximation scheme unless **PPAD** is in **P**. Finally, we thank many wonderful people who helped us in this work.

2.4. NOTATION. We use bold lower-case Roman letters such as $\mathbf{x}$, $\mathbf{a}$, $\mathbf{b}_j$ to denote vectors. Whenever a vector such as $\mathbf{a} \in \mathbb{R}^n$ is present, its components will be denoted by lower-case Roman letters with subscripts as $a_1, \ldots, a_n$. Matrices are denoted by bold upper-case Roman letters such as $\mathbf{A}$ and scalars are usually denoted by lower-case Roman letters, but sometimes by upper-case Roman letters such as M, N, and K. The (i, j)th entry of a matrix $\mathbf{A}$ is denoted by $a_{i,j}$. Depending on the context, we may use $\mathbf{a}_i$ to denote the ith row or the ith column of $\mathbf{A}$.

We now enumerate some other notation that are used in this article. For positive integer n, we let $[n]$ denote the set $\{1, 2, \ldots, n\} \subset \mathbb{Z}$; we let $\mathbb{Z}_+^d$ denote the set of d-dimensional vectors with positive integer entries; let $\langle \mathbf{a}|\mathbf{b} \rangle$ denote the dot-product of two vectors in the same dimension; let $\mathbf{e}_i$ denote the unit vector whose ith entry is equal to 1 and other entries are 0; and let $\| \cdot \|_1$ and $\| \cdot \|_\infty$ denote the L^1-norm and the infinity norm, respectively: $\|\mathbf{x}\|_1 = \sum_{i=1}^d |x_i|$ and $\|\mathbf{x}\|_\infty = \max_{1 \le i \le d} |x_i|$, for $\mathbf{x} \in \mathbb{R}^d$. Finally, for $a, b \in \mathbb{R}$, by $a = b \pm \epsilon$, we mean $b - \epsilon \le a \le b + \epsilon$.

3. *Two-Player Nash Equilibria*

A *two-player game* [Nash 1951; Lemke 1965; Lemke and Howson, Jr. 1964] is a noncooperative game between two players, where both players simultaneously choose an action from their respective action spaces, and then receive a payoff that is a function of both choices. When the first player has m choices of actions and the second player has n choices of actions, the game, in its normal form, can be specified by two $m \times n$ matrices $\mathbf{A} = \left(a_{i,j}\right)$ and $\mathbf{B} = \left(b_{i,j}\right)$. 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. Thus, a two-player game is also often referred to as a *bimatrix game*. A mixed strategy of a player is a probability distribution over his or her choices. Nash's Equilibrium Theorem [Nash 1951, 1950], when specialized to bimatrix games, asserts that every two-player game has an equilibrium point, that is, a pair of mixed strategies, such that neither player can gain by changing his or her strategy unilaterally. The zero-sum two-player game [Morgenstern and von Neumann 1947] is a special case of the bimatrix game that satisfies $\mathbf{B} = -\mathbf{A}$.

Let $\mathbb{P}^n$ denote the set of all *probability vectors* in $\mathbb{R}^n$, that is, nonnegative, length n vectors whose entries sum to 1. Then, a pair of mixed strategies can be expressed by two column vectors $(\mathbf{x} \in \mathbb{P}^m, \mathbf{y} \in \mathbb{P}^n)$. Let $\mathbf{a}_i$ and $\mathbf{b}_j$ denote the ith row of $\mathbf{A}$ and the jth column of $\mathbf{B}$, respectively. In a pair of mixed strategies $(\mathbf{x}, \mathbf{y})$, the expected payoff of the first player when choosing the ith row is $\mathbf{a}_i \mathbf{y}$, and the expected payoff of the second player when choosing the ith column is $\mathbf{x}^T \mathbf{b}_i$; the expected payoff of the first player is $\mathbf{x}^T \mathbf{A} \mathbf{y}$, and the expected payoff of the second player is $\mathbf{x}^T \mathbf{B} \mathbf{y}$.

Mathematically, a *Nash equilibrium* of a bimatrix game $(\mathbf{A}, \mathbf{B})$ is a pair $(\mathbf{x}^* \in \mathbb{P}^m, \mathbf{y}^* \in \mathbb{P}^n)$ such that for all $\mathbf{x} \in \mathbb{P}^m$ and $\mathbf{y} \in \mathbb{P}^n$,

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

Computationally, one might settle with an approximate Nash equilibrium. Several notions of approximate Nash equilibria have been defined in the literature. The following are two most popular ones: Lipton et al. [2004] and Kannan and Theobald [2007]. However, in the rest of the article, we use a third notion of approximate Nash equilibria, which was introduced in Daskalakis et al. [2006a] for the study of the complexity of equilibrium approximation. We will define it later in this section.

For a positive parameter ϵ, an *ϵ-approximate Nash equilibrium* of a bimatrix game $(\mathbf{A}, \mathbf{B})$ is a pair $(\mathbf{x}^* \in \mathbb{P}^m, \mathbf{y}^* \in \mathbb{P}^n)$ such that for all $\mathbf{x} \in \mathbb{P}^m$ and $\mathbf{y} \in \mathbb{P}^n$,

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

For two nonnegative matrices $\mathbf{A}$ and $\mathbf{B}$, an *ϵ-relatively-approximate Nash equilibrium* of $(\mathbf{A}, \mathbf{B})$ is a pair $(\mathbf{x}^*, \mathbf{y}^*)$ such that for all $\mathbf{x} \in \mathbb{P}^m$ and $\mathbf{y} \in \mathbb{P}^n$,

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

Nash equilibria of a bimatrix game $(\mathbf{A}, \mathbf{B})$ are invariant under positive scalings, meaning, the bimatrix game $(c_1\mathbf{A}, c_2\mathbf{B})$ has the same set of Nash equilibria as $(\mathbf{A}, \mathbf{B})$, when $c_1, c_2 > 0$. They are also invariant under shifting: For any constants c_1 and c_2, the bimatrix game $(c_1 + \mathbf{A}, c_2 + \mathbf{B})$ has the same set of Nash equilibria as $(\mathbf{A}, \mathbf{B})$. It is easy to verify that ϵ-approximate Nash equilibria are also invariant under shifting. However, each ϵ-approximate Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $(\mathbf{A}, \mathbf{B})$ becomes a $(c \cdot \epsilon)$-approximate Nash equilibrium of the bimatrix game $(c\mathbf{A}, c\mathbf{B})$ for $c > 0$. Meanwhile, ϵ-relatively-approximate Nash equilibria are invariant under positive scaling, but may not be invariant under shifting.

Because the ϵ-approximate Nash equilibrium is sensitive to scaling of $\mathbf{A}$ and $\mathbf{B}$, when studying its computational complexity, it is important to consider bimatrix games with *normalized* matrices, in which the absolute value of each entry of $\mathbf{A}$ and $\mathbf{B}$ is bounded, for example, by 1. Earlier work on this subject by Lipton et al. [2004] used a similar normalization. Let $\mathbb{R}^{m\times n}_{[a,b]}$ denote the set of $m \times n$ matrices with real entries between a and b. In this article, we say a bimatrix game $(\mathbf{A}, \mathbf{B})$ is *normalized* if $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}_{[-1,1]}$ and is *positively normalized* if $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}_{[0,1]}$.

For positively normalized bimatrix games, one can prove the following relation between the two notions:

PROPOSITION 3.1. *In a positively normalized bimatrix game* $(\mathbf{A}, \mathbf{B})$*, every ϵ-relatively-approximate Nash equilibrium is also an ϵ-approximate Nash equilibrium.*

We first define the input models in order to define our main search problems of computing and approximating a two-player Nash equilibrium. The most general input model is the *real model* in which a bimatrix game is specified by two real matrices $(\mathbf{A}, \mathbf{B})$. In the *rational model*, each entry of the payoff matrices is given by the ratio of two integers. The *input size* is then the total number of bits describing the payoff matrices. Using the fact that two-player Nash equilibria are invariant under positive scaling, we can transform a rational bimatrix game into an *integer bimatrix game* by multiplying the common denominators in a payoff matrix. Moreover, the total number of bits in this game with integer payoffs is within a factor of poly(m, n) of the input size of its rational counterpart. In fact, Abbott et al. [2005] made it much simpler: They show that for every bimatrix game with integer payoffs, one can construct a "homomorphic" bimatrix game with 0-1 payoffs whose size is within a polynomial factor of the input size of the original game.

We recall the proof of the well-known fact that each rational bimatrix game has a rational Nash equilibrium. Suppose $(\mathbf{A}, \mathbf{B})$ is a rational bimatrix game and $(\mathbf{u}, \mathbf{v})$

is one of its Nash equilibria. Let

$$\text{row-support} = \{i \mid u_i > 0\} \quad \text{and} \quad \text{column-support} = \{i \mid v_i > 0\}.$$

We let $\mathbf{a}_i$ and $\mathbf{b}_j$ denote the ith row vector of $\mathbf{A}$ and the jth column vector of $\mathbf{B}$, respectively. Then, by the condition of the Nash equilibrium, $(\mathbf{u}, \mathbf{v})$ is a feasible solution to the following linear program:

$$\begin{array}{ll}
\sum_i x_i = 1 \text{ and } \sum_i y_i = 1 & \\
x_i = 0, & \forall i \notin \text{row-support} \\
y_i = 0, & \forall i \notin \text{column-support} \\
x_i \geq 0, & \forall i \in \text{row-support} \\
y_i \geq 0, & \forall i \in \text{column-support} \\
\mathbf{a}_i \mathbf{y} = \mathbf{a}_j \mathbf{y}, & \forall i, j \in \text{row-support} \\
\mathbf{x}^T \mathbf{b}_i = \mathbf{x}^T \mathbf{b}_j, & \forall i, j \in \text{column-support} \\
\mathbf{a}_i \mathbf{y} \leq \mathbf{a}_j \mathbf{y}, & \forall i \notin \text{row-support}, \ j \in \text{row-support} \\
\mathbf{x}^T \mathbf{b}_i \leq \mathbf{x}^T \mathbf{b}_j, & \forall i \notin \text{column-support}, \ j \in \text{column-support}.
\end{array}$$

In fact, any solution to this linear program is a Nash equilibrium of $(\mathbf{A}, \mathbf{B})$. Therefore, $(\mathbf{A}, \mathbf{B})$ has at least one rational equilibrium point such that the total number of bits describing this equilibrium is within a polynomial factor of the input size of $(\mathbf{A}, \mathbf{B})$. By enumerating all possible row supports and column supports and solving the linear program above, we can find a Nash equilibrium of game $(\mathbf{A}, \mathbf{B})$. This exhaustive-search algorithm takes $2^{m+n}\text{poly}(L)$ time where L is the input size of the game, and m and n are, respectively, the number of rows and the number of columns.

In this article, we use BIMATRIX to denote the problem of finding a Nash equilibrium in a rational bimatrix game. Without loss of generality, we make two assumptions about BIMATRIX: all input games are positively normalized and both players have the same number of choices of actions. Two important parameters associated with each instance of BIMATRIX are: n, the number of actions, and L, the input size of the game. Thus, BIMATRIX is in **P** if there exists an algorithm for BIMATRIX with running time poly(L). We note as an aside that, in the two-player games we construct in this article, parameter L is a polynomial of n.

We also consider two families of approximation problems for two-player Nash equilibria. For a positive constant c,

—let EXP^c-BIMATRIX denote the following search problem: given a rational and positively normalized $n \times n$ bimatrix game $(\mathbf{A}, \mathbf{B})$, compute a 2^{-cn}-approximate Nash equilibrium of $(\mathbf{A}, \mathbf{B})$;

—let POLY^c-BIMATRIX denote the following search problem: given a rational and positively normalized $n \times n$ bimatrix game $(\mathbf{A}, \mathbf{B})$, compute an n^{-c}-approximate Nash equilibrium of $(\mathbf{A}, \mathbf{B})$.

In our analysis, we will use an alternative notion of approximate Nash equilibria as introduced in Daskalakis et al. [2006a], originally called ϵ-*Nash equilibria*, which measures the pairwise stability of pure strategies. To emphasize this pairwise stability and distinguish it from the more commonly used ϵ-approximate Nash equilibrium, we refer to this type of equilibria as *well-supported approximate Nash equilibria*.[6]

For a positive parameter ϵ, a pair of strategies ($\mathbf{x}^* \in \mathbb{P}^n, \mathbf{y}^* \in \mathbb{P}^n$) is an ϵ-*well supported Nash equilibrium* of $(\mathbf{A}, \mathbf{B})$ if for all j and k (recall that $\mathbf{a}_i$ and $\mathbf{b}_i$ denote the ith row of $\mathbf{A}$ and the ith column of $\mathbf{B}$, respectively),

$$(\mathbf{x}^*)^T \mathbf{b}_j > (\mathbf{x}^*)^T \mathbf{b}_k + \epsilon \;\Rightarrow\; y_k^* = 0, \quad \text{and}$$
$$\mathbf{a}_j \mathbf{y}^* > \mathbf{a}_k \mathbf{y}^* + \epsilon \;\Rightarrow\; x_k^* = 0.$$

A Nash equilibrium is a 0-well-supported Nash equilibrium as well as a 0-approximate Nash equilibrium. The following lemma, a key lemma in the study of the complexity of equilibrium approximation, shows that approximate Nash equilibria and well-supported Nash equilibria are polynomially related. This relation allows us to focus on pairwise comparisons, between any two pure strategies, in approximation conditions.

LEMMA 3.2 (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* $0 \le \epsilon \le 1$,

(1) *each ϵ-well-supported Nash equilibrium is also an ϵ-approximate Nash equilibrium; and*
(2) *from any $\epsilon^2/8$-approximate Nash equilibrium* $(\mathbf{u}, \mathbf{v})$, *one can find in polynomial time an ϵ-well-supported Nash equilibrium* $(\mathbf{x}, \mathbf{y})$.

PROOF. The first statement follows from the definitions.

Because $(\mathbf{u}, \mathbf{v})$ is an $\epsilon^2/8$-approximate Nash equilibrium, we have

$$\forall \mathbf{u}' \in \mathbb{P}^n,\; (\mathbf{u}')^T \mathbf{A}\mathbf{v} \le \mathbf{u}^T \mathbf{A}\mathbf{v} + \epsilon^2/8, \quad \text{and}$$
$$\forall \mathbf{v}' \in \mathbb{P}^n,\; \mathbf{u}^T \mathbf{B}\mathbf{v}' \le \mathbf{u}^T \mathbf{B}\mathbf{v} + \epsilon^2/8.$$

Recall that $\mathbf{a}_i$ denotes the ith row of $\mathbf{A}$ and $\mathbf{b}_i$ denotes the ith column of $\mathbf{B}$. Let i^* be an index such that $\mathbf{a}_{i^*}\mathbf{v} = \max_{1\le i\le n} \mathbf{a}_i \mathbf{v}$. We use J_1 to denote the set of indices $j : 1 \le j \le n$ such that $\mathbf{a}_{i^*}\mathbf{v} \ge \mathbf{a}_j \mathbf{v} + \epsilon/2$. Now by changing u_j to 0 for all $j \in J_1$, and changing u_{i^*} to $u_{i^*} + \sum_{j\in J_1} u_j$, we can increase the first-player's profit by at least $(\epsilon/2)\sum_{j\in J_1} u_j$, implying that $\sum_{j\in J_1} u_j \le \epsilon/4$. Similarly, we define

$$J_2 = \{\, j \,\big|\, 1 \le j \le n \text{ and } \exists i,\; \mathbf{u}^T \mathbf{b}_i \ge \mathbf{u}^T \mathbf{b}_j + \epsilon/2 \},$$

and have $\sum_{j\in J_2} v_j \le \epsilon/4$.

Let $(\mathbf{x}, \mathbf{y})$ be the vectors obtained by modifying $\mathbf{u}$ and $\mathbf{v}$ in the following manner: set all the $\{u_j \mid j \in J_1\}$ and $\{v_j \mid j \in J_2\}$ to zero; uniformly increase the probabilities of other strategies so that $\mathbf{x}$ and $\mathbf{y}$ are mixed strategies.

Note that for all $i \in [n]$, $|\mathbf{a}_i \mathbf{y} - \mathbf{a}_i \mathbf{v}| \le \epsilon/4$, because we assume that the value of each entry in $\mathbf{a}_i$ is between 0 and 1. Therefore, for every pair $i, j : 1 \le i, j \le n$, the relative change between $\mathbf{a}_i\mathbf{y} - \mathbf{a}_j\mathbf{y}$ and $\mathbf{a}_i\mathbf{v} - \mathbf{a}_j\mathbf{v}$ is no more than $\epsilon/2$. Thus, any j that is beaten by some i by a gap of ϵ is already set to zero in $(\mathbf{x}, \mathbf{y})$. As a result, $(\mathbf{x}, \mathbf{y})$ is an ϵ-well-supported Nash equilibrium, and the second statement follows. □

[6] We are honored that, in their journal version, Daskalakis, Goldberg, and Papadimitriou also adopted the name "well-supported approximate Nash equilibria."

We conclude this section by pointing out that there are other natural notions of approximation for equilibrium points. In addition to the rational representation of a rational equilibrium, one can use binary representations to define entries in an equilibrium. As each entry p in an equilibrium is a number between 0 and 1, we can specify it using its binary representation $(0.c_1 \cdots c_P \cdots)$, where $c_i \in \{0, 1\}$ and $p = \lim_{i\to\infty} \sum_{j=1}^{i} c_j/2^j$. Some rational numbers may not have a finite binary representation. Usually, we round off the numbers to store their finite approximations. The first P bits $c_1, \ldots, c_P$ give us a *P-bit approximation* $\tilde{p}$ of p: $\tilde{p} = \sum_{i=1}^{P} c_i/2^i$.

For a positive integer P, we use P-BIT-BIMATRIX to denote the search problem of computing the first P bits of the entries of a Nash equilibrium in a rational bimatrix game. The following proposition relates P-BIT-BIMATRIX with POLY^c-BIMATRIX. A similar proposition is stated and proved in Chen et al. [2007].

PROPOSITION 3.3. *Let $(\mathbf{x}, \mathbf{y})$ be a Nash equilibrium of a positively normalized $n \times n$ bimatrix game $(\mathbf{A}, \mathbf{B})$. For a positive integer P, let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ be the P-bit approximation of $(\mathbf{x}, \mathbf{y})$. Let $\bar{\mathbf{x}} = \tilde{\mathbf{x}}/\|\tilde{\mathbf{x}}\|_1$ and $\bar{\mathbf{y}} = \tilde{\mathbf{y}}/\|\tilde{\mathbf{y}}\|_1$. Then, $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is a $(3n2^{-P})$-approximate Nash equilibrium of $(\mathbf{A}, \mathbf{B})$.*

PROOF. Let $a = 2^{-P}$. Consider any $\mathbf{x}' \in \mathbb{P}^n$. We have

$$\begin{aligned}(\mathbf{x}')^T \mathbf{A}\bar{\mathbf{y}} &\le (\mathbf{x}')^T \mathbf{A}\tilde{\mathbf{y}} + na \le (\mathbf{x}')^T \mathbf{A}\mathbf{y} + na \le \mathbf{x}^T \mathbf{A}\mathbf{y} + na \\ &\le \tilde{\mathbf{x}}^T \mathbf{A}\mathbf{y} + 2na \le \tilde{\mathbf{x}}^T \mathbf{A}\tilde{\mathbf{y}} + 3na \le \bar{\mathbf{x}}^T \mathbf{A}\tilde{\mathbf{y}} + 3na \le \bar{\mathbf{x}}^T \mathbf{A}\bar{\mathbf{y}} + 3na.\end{aligned}$$

To see the first inequality, note that since the game is positively normalized, every component in $(\mathbf{x}')^T\mathbf{A}$ is between 0 and 1. The inequality follows from the fact that $\bar{y}_i \ge \tilde{y}_i$ for all $i \in [n]$, and $\|\tilde{\mathbf{y}}\|_1 \ge 1 - na$. The other inequalities can be proved similarly. □

4. *Complexity and Algorithm Analysis*

In this section, we review the complexity class **PPAD** and the concept of polynomial-time reductions among search problems. We define the perturbation models in the smoothed analysis of BIMATRIX. We then show that if the smoothed complexity of BIMATRIX is polynomial, then we can compute an ϵ-approximate Nash equilibrium of a bimatrix game in randomized polynomial time.

4.1. **PPAD** AND POLYNOMIAL-TIME REDUCTION AMONG SEARCH PROBLEMS. A binary relation $R \subset \{0, 1\}^* \times \{0, 1\}^*$ is *polynomially balanced* if there exist constants c and k such that for all pairs $(x, y) \in R$, we have $|y| \le c\,|x|^k$, where $|x|$ denotes the length of string x. It is *polynomial-time computable* if for each pair (x, y), one can decide whether or not $(x, y) \in R$ in time polynomial in $|x| + |y|$. A relation R is *total* if for every string $x \in \{0, 1\}^*$, there exists y such that $(x, y) \in R$.

For a binary relation R that is both polynomially balanced and polynomial-time computable, one can define the **NP** search problem SEARCH^R specified by R as: Given $x \in \{0, 1\}^*$, return a y satisfying $(x, y) \in R$ if such y exists, otherwise, return a special string "no". Following Megiddo and Papadimitriou [1991], we use **TFNP** to denote the class of all **NP** search problems specified by total relations.

A search problem SEARCH$^{R_1} \in$ **TFNP** is *polynomial-time reducible* to another search problem SEARCH$^{R_2} \in$ **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 $(x, g(y)) \in R_1$. In other words, one can use f to transform any input instance x of SEARCH^{R_1} into an input instance $f(x)$ of SEARCH^{R_2}, and use g to transform any solution y to $f(x)$ back into a solution $g(y)$ to x. Search problems SEARCH^{R_1} and SEARCH^{R_2} are polynomial-time equivalent if SEARCH^{R_2} is also reducible to SEARCH^{R_1}.

The complexity class **PPAD** [Papadimitriou 1994] is a subclass of **TFNP**. It contains all search problems that are polynomial-time reducible to the following problem called END-OF-LINE:

Definition 4.1 (*End-of-Line*). The input instance of END-OF-LINE is a pair $(0^n, \mathcal{M})$, where 0^n is a binary string of n 0's, and $\mathcal{M}$ is a Boolean circuit with n input bits. $\mathcal{M}$ defines a function M, over $\{0, 1\}^n$, satisfying:

—$\forall v \in \{0, 1\}^n$, $M(v)$ is an ordered pair (u_1, u_2) where $u_1, u_2 \in \{0, 1\}^n \cup \{\text{"}no\text{"}\}$;

—$M(0^n) = (\text{"}no\text{"}, u)$ for some $u \in \{0, 1\}^n$ and the first component of $M(u)$ is 0^n.

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

A vertex $v \in V$ is called an end vertex of G_M if the summation of its in-degree and out-degree is equal to one. The output of the problem is an end vertex of G_M other than 0^n.

Note that in graph G_M, both the in-degree and the out-degree of each vertex are at most 1. Thus, edges of G_M form a collection of directed paths and directed cycles. Because 0^n has in-degree 0 and out-degree 1, it is an end vertex of G_M. Thus, G_M has at least one directed path. As a result, it has another end vertex and END-OF-LINE is a member of **TFNP**. In fact, G_M has an odd number of end vertices other than 0^n. By evaluating the Boolean circuit $\mathcal{M}$ on an input $v \in \{0, 1\}^n$, we can access the predecessor and the successor of v, if they exist.

Many important problems, including the search versions of Brouwer's Fixed Point Theorem, Kakutani's Fixed Point Theorem, Smith's Theorem, and Borsuk-Ulam Theorem, are members of **PPAD** [Papadimitriou 1991]. BIMATRIX is also in **PPAD** [Papadimitriou 1991]. As a corollary, for all $c > 0$, POLY^c-BIMATRIX and EXP^c-BIMATRIX are in **PPAD**.

However, it is not clear whether P-BIT-BIMATRIX, for a positive integer P, is in **PPAD** or not, though obviously it is easier than BIMATRIX. The reason is that we do not know whether P-BIT-BIMATRIX is in **TFNP** (recall **PPAD** is a subclass of **TFNP**). In particular, given a pair of vectors $(\mathbf{x}, \mathbf{y})$ in which all x_i, y_i have the form $0.c_1 \ldots c_P$ where $c_j \in \{0, 1\}$, we do not know how to check in polynomial time whether $(\mathbf{x}, \mathbf{y})$ is the P-bit approximation of an equilibrium or not.

4.2. SMOOTHED MODELS OF BIMATRIX GAMES. In the smoothed analysis of bimatrix games, we consider perturbed games in which each entry of the payoff matrices is subject to a small and independent random perturbation. For a pair of $n \times n$ positively normalized matrices $\overline{\mathbf{A}} = (\overline{a}_{i,j})$ and $\overline{\mathbf{B}} = (\overline{b}_{i,j})$, in the smoothed model, the input instance[7] is defined by $(\mathbf{A}, \mathbf{B})$ where $a_{i,j}$ and $b_{i,j}$ are, respectively, independent perturbations of $\bar{a}_{i,j}$ and $\bar{b}_{i,j}$ with magnitude σ (see below). There are several models of perturbations for $a_{i,j}$ and $b_{i,j}$ with magnitude σ [Spielman and Teng 2006]. The two common ones are the uniform perturbation and the Gaussian perturbation.

In a *uniform perturbation* with magnitude σ, $a_{i,j}$ and $b_{i,j}$ are chosen uniformly from the intervals $[\bar{a}_{i,j} - \sigma, \bar{a}_{i,j} + \sigma]$ and $[\bar{b}_{i,j} - \sigma, \bar{b}_{i,j} + \sigma]$, respectively. In a *Gaussian perturbation* with magnitude σ, $a_{i,j}$ and $b_{i,j}$ are obtained from perturbations of $\bar{a}_{i,j}$ and $\bar{b}_{i,j}$, respectively, by adding independent random variables distributed as Gaussians with mean 0 and standard deviation σ. We refer to these perturbations as σ*-uniform* and σ*-Gaussian perturbations*, respectively.

The smoothed time complexity of an algorithm J for BIMATRIX is defined as follows: Let $T_J(\mathbf{A}, \mathbf{B})$ be the complexity of J for finding a Nash equilibrium in a bimatrix game $(\mathbf{A}, \mathbf{B})$. Then, the *smoothed complexity* of J under perturbations $N_\sigma()$ of magnitude σ is (We use $\mathbf{A} \leftarrow N_\sigma(\bar{\mathbf{A}})$ to denote that $\mathbf{A}$ is a perturbation of $\bar{\mathbf{A}}$ according to $N_\sigma()$)

$$\text{Smoothed}_J\,[n, \sigma] = \max_{\bar{\mathbf{A}}, \bar{\mathbf{B}} \in \mathbb{R}^{n \times n}_{[0,1]}} \mathrm{E}_{\mathbf{A} \leftarrow N_\sigma(\bar{\mathbf{A}}), \mathbf{B} \leftarrow N_\sigma(\bar{\mathbf{B}})} \left[T_J(\mathbf{A}, \mathbf{B}) \right].$$

An algorithm J has a *polynomial smoothed time complexity* [Spielman and Teng 2006] if for all $0 < \sigma < 1$ and for all positive integers n, there exist positive constants c, k_1 and k_2 such that

$$\text{Smoothed}_J\,[n, \sigma] \le c \cdot n^{k_1} \sigma^{-k_2}.$$

BIMATRIX is in *smoothed polynomial time* if there exists an algorithm J with polynomial smoothed time complexity for computing a two-player Nash equilibrium.

The following lemma shows that if the smoothed complexity of BIMATRIX is low, under uniform or Gaussian perturbations, then one can quickly find an approximate Nash equilibrium.

LEMMA 4.2 (SMOOTHED NASH VS APPROXIMATE NASH). *If problem* BIMATRIX *is in smoothed polynomial time under uniform or 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* $O(\text{poly}(m, n, 1/\epsilon))$.

PROOF. Informally argued in Spielman and Teng [2006]. See Appendix A for a proof of the uniform case. □

5. *Two Search Problems*

In this section, we consider two search problems that are essential to our main results. First, we define a class of high-dimensional discrete fixed point problems,

[7] For the simplicity of presentation, in this section, we model entries of payoff matrices and perturbations by real numbers. Of course, to connect with the complexity result of the previous section, where entries of matrices are in finite representations, we are mindful that some readers may prefer that we state our result and write the proof more explicitly using the finite representations. Using Eqs. (20) and (21) in the proof of Lemma 4.2 (see Appendix A), we can define a discrete version of the uniform and Gaussian perturbations and state and prove the same result.

which is a generalization of the 3-DIMENSIONAL BROUWER proposed in [Daskalakis et al. 2006a]. Then, we introduce the concept of generalized circuits, a structure used implicitly in Step 3 of the DGP framework (see Section 2).

5.1. DISCRETE BROUWER FIXED POINTS. The following is an obvious fact: Suppose we color the endpoints of an interval $[0, n]$ by two distinct colors, say red and blue, insert $n - 1$ points evenly into this interval to subdivide it into n unit subintervals, and color these new points arbitrarily with the two colors. Then, there must be a *bichromatic subinterval*, that is, a unit subinterval whose two endpoints have distinct colors.

Our first search problem is built on a high-dimensional extension of this fact. Instead of coloring points in a subdivision of an interval, we color the vertices in a hypergrid. When the dimension is d, we use $d + 1$ colors.

For positive integer d and $\mathbf{r} \in \mathbb{Z}_+^d$, let $A_\mathbf{r}^d = \{\mathbf{q} \in \mathbb{Z}^d \mid 0 \le q_i \le r_i - 1, \forall\, i \in [d]\}$ denote the vertices of the *hypergrid* with side lengths specified by $\mathbf{r}$. The *boundary* of $A_\mathbf{r}^d$, denoted by $\partial(A_\mathbf{r}^d)$, is the set of points $\mathbf{q} \in A_\mathbf{r}^d$ with $q_i \in \{0, r_i - 1\}$ for some i. Let $\text{Size}\,[\,\mathbf{r}] = \sum_{1 \le i \le d} \lceil \log r_i \rceil$, that is, the number of bits needed to encode a point in $A_\mathbf{r}^d$.

In one dimension, the interval $[0, n]$ is the union of n unit subintervals. In d dimensions, the hypergrid $A_\mathbf{r}^d$ can be viewed as the union of a collection of unit hypercubes. For a point $\mathbf{p} \in \mathbb{Z}^d$, let $K_\mathbf{p} = \{\mathbf{q} \in \mathbb{Z}^d \mid q_i \in \{p_i, p_i + 1\}, \forall\, i \in [d]\}$ be the vertices of the unit hypercube with $\mathbf{p}$ as its lowest-coordinate corner.

As a natural extension of 3-DIMENSIONAL BROUWER of Daskalakis et al. [2006a], we can color the vertices of a hypergrid with the $(d + 1)$ colors $\{1, 2, \ldots, d + 1\}$. As in one dimension, the coloring of the boundary vertices needs to meet certain requirements in the context of the discrete Brouwer fixed point problem. A color assignment ϕ of $A_\mathbf{r}^d$ is *valid* if $\phi(\mathbf{p})$ satisfies the following condition: For $\mathbf{p} \in \partial(A_\mathbf{r}^d)$, if there exists an $i \in [d]$ such that $p_i = 0$ then $\phi(\mathbf{p}) = \max\{i \mid p_i = 0\}$; for other boundary points, let $\phi(\mathbf{p}) = d + 1$. In the latter case, $\forall\, i,\ p_i \neq 0$ and $\exists\, i,\ p_i = r_i - 1$.

The following theorem is a high-dimensional extension of the one-dimensional fact mentioned above. It is also an extension of Sperner's Lemma.

THEOREM 5.1 (HIGH-DIMENSIONAL DISCRETE BROUWER FIXED POINTS). *For any valid coloring ϕ of $A_\mathbf{r}^d$, where d is a positive integer and $\mathbf{r} \in \mathbb{Z}_+^d$, there is a unit hypercube in $A_\mathbf{r}^d$ whose vertices have all $d + 1$ colors.*

In other words, Theorem 5.1 asserts that there exists a $\mathbf{p} \in A_\mathbf{r}^d$ such that ϕ assigns all $(d + 1)$ colors to $K_\mathbf{p}$. We call $K_\mathbf{p}$ a *panchromatic cube*. However, in d-dimensions, a panchromatic cube contains 2^d vertices. This exponential dependency in the dimension makes it inefficient to check whether a hypercube is panchromatic. We introduce the following notion of discrete fixed points.

Definition 5.2 (*Panchromatic Simplex*). A subset $P \subset A_\mathbf{r}^d$ is *accommodated* if $P \subset K_\mathbf{p}$ for some point $\mathbf{p} \in A_\mathbf{r}^d$. $P \subset A_\mathbf{r}^d$ is a *panchromatic simplex* of a color assignment ϕ if it is accommodated and contains exactly $d + 1$ points with $d + 1$ distinct colors.

COROLLARY 5.3 (EXISTENCE OF A PANCHROMATIC SIMPLEX). *For any valid coloring ϕ of $A_\mathbf{r}^d$, where d is a positive integer and $\mathbf{r} \in \mathbb{Z}_+^d$, there exists a panchromatic simplex in $A_\mathbf{r}^d$.*

We can define a search problem based on Corollary 5.3. An input instance is a hypergrid together with a Boolean circuit for coloring the vertices of the hypergrid.

Definition 5.4 (*Brouwer-Mapping Circuit and Color Assignment*). For positive integer d and $\mathbf{r} \in \mathbb{Z}_+^d$, a Boolean circuit C with *Size* $[\mathbf{r}]$ input bits and $2d$ output bits $\Delta_1^+, \Delta_1^-, \ldots, \Delta_d^+, \Delta_d^-$ is a valid *Brouwer-mapping circuit* (with parameters d and $\mathbf{r}$) if the following is true:

(1) For every $\mathbf{p} \in A_{\mathbf{r}}^d$, the $2d$ output bits of C evaluated at $\mathbf{p}$ satisfy one of the following $d+1$ cases:
 —Case i, $1 \le i \le d$: $\Delta_i^+ = 1$ and all other $2d-1$ bits are 0;
 —Case $(d+1)$: $\forall i$, $\Delta_i^+ = 0$ and $\Delta_i^- = 1$.

(2) For every $\mathbf{p} \in \partial(A_{\mathbf{r}}^d)$, if there exists an $i \in [d]$ such that $p_i = 0$, letting $i_{\max} = \max\{i \mid p_i = 0\}$, then the output bits satisfy Case $i_{\max}$, otherwise ($\forall i$, $p_i \neq 0$ and $\exists i$, $p_i = r_i - 1$), the output bits satisfy Case $d+1$.

Such a circuit C defines a valid color assignment $\text{Color}_C : A_{\mathbf{r}}^d \to \{1, 2, \ldots, d, d+1\}$ by setting $\text{Color}_C\,[\mathbf{p}] = i$, if the output bits of C evaluated at $\mathbf{p}$ satisfy Case i.

To define high-dimensional Brouwer's fixed point problems, we need a notion of *well-behaved* functions[8] to parameterize the shape of the search space. An integer function $f(n)$ is called *well behaved* if it is polynomial-time computable and there exists an integer constant n_0 such that $3 \le f(n) \le \lceil n/2 \rceil$ for all $n \ge n_0$. For example, let f_1, f_2, f_3 and f_4 denote the following functions:

$$f_1(n) = 3, \quad f_2(n) = \lceil n/2 \rceil, \quad f_3(n) = \lceil n/3 \rceil, \quad \text{and} \quad f_4(n) = \lceil \log n \rceil.$$

It is easy to check that they are all well behaved. Besides, since $f(n) \le \lceil n/2 \rceil$ for large enough n, we have $\lceil n/f(n) \rceil \ge 2$.

Definition 5.5. For each well behaved function f, the problem Brouwer^f is defined as follows: Given a pair $(C, 0^n)$, where C is a valid Brouwer-mapping circuit with parameters $d = \lceil n/f(n) \rceil$ and $\mathbf{r} \in \mathbb{Z}_+^d$, where $r_i = 2^{f(n)}$ for all $i \in [d]$, find a panchromatic simplex of C.

The *input size* of problem Brouwer^f is the sum of n and the size of the circuit C. When n is large enough, Brouwer^{f_2} is a two-dimensional search problem over grid $\{0, 1, \ldots, 2^{\lceil n/2 \rceil} - 1\}^2$, Brouwer^{f_3} is a three-dimensional search problem over $\{0, 1, \ldots, 2^{\lceil n/3 \rceil} - 1\}^3$, and Brouwer^{f_1} is an $\lceil n/3 \rceil$-dimensional search problem over $\{0, 1, \ldots, 7\}^{\lceil n/3 \rceil}$. Each of these three grids contains roughly 2^n hypercubes. Both Brouwer^{f_2} [Chen and Deng 2006] and Brouwer^{f_3} [Daskalakis et al. 2006a] are known to be complete in **PPAD**. In Section 9, we will prove for every well-behaved function f, Brouwer^f is **PPAD**-complete. Therefore, the complexity of finding a panchromatic simplex is essentially independent of the shape or dimension of the search space. In particular, the theorem implies that Brouwer^{f_1} is **PPAD**-complete.

5.2. Generalized Circuits and Their Assignment Problem. To effectively connect discrete Brouwer fixed points with two-player Nash equilibria, we

[8] Please note that this is not the coloring function in the fixed point problem.

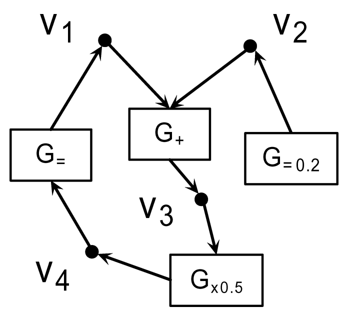

FIG. 1. An example of a generalized circuit.

use an intermediate structure called the generalized circuit. This family of circuits, used implicitly in Daskalakis et al. [2006a], extends the standard classes of Boolean or arithmetic circuits in several ways. (See Figure 1.)

Syntactically, a *generalized circuit* $\mathcal{S}$ is a pair $(V, \mathcal{T})$, where V is a set of nodes and $\mathcal{T}$ is a collection of gates. Every gate $T \in \mathcal{T}$ is a 5-tuple $T = (G, v_1, v_2, v, \alpha)$ in which

—$G \in \{G_\zeta, G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\wedge, G_\vee, G_\neg\}$ is the type of the gate. Among the nine types of gates, G_ζ, $G_{\times\zeta}$, $G_=$, G_+ and G_- are arithmetic gates implementing arithmetic constraints like addition, subtraction and constant multiplication. $G_<$ is a *brittle* comparator: it only distinguishes two values that are properly separated. Finally, $G_\wedge$, $G_\vee$ and $G_\neg$ are logic gates.

—$v_1, v_2 \in V \cup \{nil\}$ are the first and second input nodes of the gate;

—$v \in V$ is the output node, and $\alpha \in \mathbb{R} \cup \{nil\}$.

The collection $\mathcal{T}$ of gates must satisfy the following important property:

$$\text{For all gates } T = (G, v_1, v_2, v, \alpha) \neq T' = (G', v_1', v_2', v', \alpha') \text{ in } \mathcal{T},\ v \neq v'. \quad (1)$$

Suppose $T = (G, v_1, v_2, v, \alpha)$ in $\mathcal{T}$. If $G = G_\zeta$, then the gate has no input node and $v_1 = v_2 = nil$. If $G \in \{G_{\times\zeta}, G_=, G_\neg\}$, then $v_1 \in V$ and $v_2 = nil$. If $G \in \{G_+, G_-, G_<, G_\wedge, G_\vee\}$, then $v_1, v_2 \in V$ and $v_1 \neq v_2$. Parameter α is only used in G_ζ and $G_{\times\zeta}$ gates. If $G = G_\zeta$, then $\alpha \in [0, 1/|V|]$. If $G = G_{\times\zeta}$, then $\alpha \in [0, 1]$. For other types of gates, $\alpha = nil$.

The *input size* of a generalized circuit $\mathcal{S} = (V, \mathcal{T})$ is the sum of $|V|$ and the total number of bits needed to describe the gates $T \in \mathcal{T}$ (the type, the vertices v_1, v_2, v, and the parameter α of T). As an important point that will become clear later, we make the following remark: the input size of the generalized circuits $\mathcal{S}$ that we will construct is upper-bounded by poly$(|V|)$.

In addition to its expanded list of gate types, the generalized circuit differs crucially from the standard circuit in that it does not require the circuit to be acyclic. In other words, in a generalized circuit, the directed graph defined by connecting input nodes of all gates to their output counterparts may have cycles. We shall show later that the presence of cycles is necessary and sufficient to express fixed point computations with generalized circuits.

Semantically, we associate every node $v \in V$ with a real variable $\mathbf{x}[v]$. Each gate $T \in \mathcal{T}$ requires that the variables of its input and output nodes satisfy certain constraints, either logical or arithmetic, depending on the type of the gate (see Figure 2

$G = G_\zeta:$	$\mathcal{P}[T,\epsilon] = \Big[\ \mathbf{x}[v] = \alpha \pm \epsilon\ \Big]$
$G = G_{\times\zeta}:$	$\mathcal{P}[T,\epsilon] = \Big[\ \mathbf{x}[v] = \min\big(\alpha\,\mathbf{x}[v_1], 1/K\big) \pm \epsilon\ \Big]$
$G = G_=:$	$\mathcal{P}[T,\epsilon] = \Big[\ \mathbf{x}[v] = \min\big(\mathbf{x}[v_1], 1/K\big) \pm \epsilon\ \Big]$
$G = G_+:$	$\mathcal{P}[T,\epsilon] = \Big[\ \mathbf{x}[v] = \min\big(\mathbf{x}[v_1] + \mathbf{x}[v_2], 1/K\big) \pm \epsilon\ \Big]$
$G = G_-:$	$\mathcal{P}[T,\epsilon] = \Big[\ \min\big(\mathbf{x}[v_1] - \mathbf{x}[v_2], 1/K\big) - \epsilon \le \mathbf{x}[v] \le \max\big(\mathbf{x}[v_1] - \mathbf{x}[v_2], 0\big) + \epsilon\ \Big]$
$G = G_<:$	$\mathcal{P}[T,\epsilon] = \Big[\ \mathbf{x}[v] =_B^\epsilon 1 \text{ if } \mathbf{x}[v_1] < \mathbf{x}[v_2] - \epsilon;\ \mathbf{x}[v] =_B^\epsilon 0 \text{ if } \mathbf{x}[v_1] > \mathbf{x}[v_2] + \epsilon\ \Big]$
$G = G_\vee:$	$\mathcal{P}[T,\epsilon] = \begin{bmatrix} \mathbf{x}[v] =_B^\epsilon 1 \text{ if } \mathbf{x}[v_1] =_B^\epsilon 1 \text{ or } \mathbf{x}[v_2] =_B^\epsilon 1 \\ \mathbf{x}[v] =_B^\epsilon 0 \text{ if } \mathbf{x}[v_1] =_B^\epsilon 0 \text{ and } \mathbf{x}[v_2] =_B^\epsilon 0 \end{bmatrix}$
$G = G_\wedge:$	$\mathcal{P}[T,\epsilon] = \begin{bmatrix} \mathbf{x}[v] =_B^\epsilon 0 \text{ if } \mathbf{x}[v_1] =_B^\epsilon 0 \text{ or } \mathbf{x}[v_2] =_B^\epsilon 0 \\ \mathbf{x}[v] =_B^\epsilon 1 \text{ if } \mathbf{x}[v_1] =_B^\epsilon 1 \text{ and } \mathbf{x}[v_2] =_B^\epsilon 1 \end{bmatrix}$
$G = G_\neg:$	$\mathcal{P}[T,\epsilon] = \Big[\ \mathbf{x}[v] =_B^\epsilon 0 \text{ if } \mathbf{x}[v_1] =_B^\epsilon 1;\ \mathbf{x}[v] =_B^\epsilon 1 \text{ if } \mathbf{x}[v_1] =_B^\epsilon 0\ \Big]$

FIG. 2. Constraints $\mathcal{P}[T,\epsilon]$, where $T = (G, v_1, v_2, v, \alpha)$ and $K = |V|$.

for the details of the constraints). The notation $=_B^\epsilon$ will be defined shortly. A generalized circuit defines a set of constraints, which may be regarded a mathematical program over the set of variables $\{\mathbf{x}[v] \mid v \in V\}$.

Definition 5.6. Suppose $\mathcal{S} = (V, \mathcal{T})$ is a generalized circuit and $K = |V|$. For every $\epsilon \ge 0$, an ϵ-*approximate solution* to $\mathcal{S}$ is an assignment $\{\mathbf{x}[v] \mid v \in V\}$ to the variables such that

—the values of $\mathbf{x}$ satisfy the constraint

$$\mathcal{P}[\epsilon] \equiv [0 \le \mathbf{x}[v] \le 1/K + \epsilon, \forall\, v \in V]; \quad \text{and}$$

—for each gate $T = (G, v_1, v_2, v, \alpha) \in \mathcal{T}$, the values of $\mathbf{x}[v_1], \mathbf{x}[v_2]$ and $\mathbf{x}[v]$ satisfy the constraint $\mathcal{P}[T,\epsilon]$, defined in *Figure 2*.

The notation $=_B^\epsilon$ in Figure 2 is defined as follows. Given an assignment $\mathbf{x}$ to the variables, we say the value of $\mathbf{x}[v]$ represents Boolean 1 with precision ϵ, denoted by $\mathbf{x}[v] =_B^\epsilon 1$, if $1/K - \epsilon \le \mathbf{x}[v] \le 1/K + \epsilon$; it represents Boolean 0 with precision ϵ, denoted by $\mathbf{x}[v] =_B^\epsilon 0$, if $0 \le \mathbf{x}[v] \le \epsilon$. One can see that the logic constraints implemented by the three logic gates $G_\wedge, G_\vee, G_\neg$ are defined similarly to the classical ones.

Using the reduction in Section 7, we can prove the following theorem. A proof can be found in Appendix B.

THEOREM 5.7. *For any constant $c \geq 3$, every generalized circuit $\mathcal{S} = (V, \mathcal{T})$ has a $1/|V|^c$-approximate solution.*

For any positive constant $c \geq 3$, we let POLY^c-GCIRCUIT denote the problem of finding a K^{-c}-approximate solution of a given generalized circuit with K nodes.

6. *Main Results and Proof Outline*

As the main technical result of this article, we prove the following theorem:

THEOREM 6.1. *For any constant $c > 0$,* POLY^c-BIMATRIX *is* **PPAD**-*complete.*

This theorem immediately implies the following statement about the complexity of computing and approximating two-player Nash equilibria.

THEOREM 6.2 (COMPLEXITY OF BIMATRIX). BIMATRIX *is* **PPAD**-*complete. Moreover, it does not have a fully-polynomial-time approximation scheme, unless* **PPAD** *is contained in* **P**.

By Proposition 3.1, BIMATRIX does not have a fully polynomial-time approximation scheme for finding a relatively-approximate Nash equilibrium.

Setting $\epsilon = 1/\text{poly}(n)$, by Theorem 6.1 and Lemma 4.2, we obtain the following theorem on the smoothed complexity of two-player Nash equilibria:

THEOREM 6.3 (SMOOTHED COMPLEXITY OF BIMATRIX). BIMATRIX *is not in smoothed polynomial time under uniform or Gaussian perturbations, unless* **PPAD** *is contained in* **RP**.

COROLLARY 6.4 (SMOOTHED COMPLEXITY OF LEMKE-HOWSON). *If* **PPAD** is not contained in **RP**, *then the smoothed complexity of the Lemke-Howson algorithm is not polynomial.*

By Proposition 3.3, we obtain the following corollary from Theorem 6.1 about the complexity of BIT-BIMATRIX.

COROLLARY 6.5 (BIT-BIMATRIX). *For any $c > 0$, $(1+c)\log n$-*BIT-BIMATRIX *is* **PPAD**-*hard.*

To prove Theorem 6.1, we start with the discrete fixed point problem BROUWER^{f_1} (recall that $f_1(n) = 3$ for all n). In Section 9, we will prove the following theorem:

THEOREM 6.6 (HIGH-DIMENSIONAL DISCRETE FIXED POINTS). *For every well-behaved function f, search problem* BROUWER^f *is* **PPAD**-*complete.*

In particular, as f_1 is a well-behaved function, Theorem 6.6 implies that BROUWER^{f_1} is complete in **PPAD**. We then apply the following three lemmas to obtain a reduction from BROUWER^{f_1} to POLY^c-BIMATRIX:

LEMMA 6.7 (BROUWER TO GCIRCUIT). BROUWER^{f_1} *is polynomial-time reducible to* POLY^3-GCIRCUIT.

LEMMA 6.8 (GCIRCUIT TO BIMATRIX). POLY^3-GCIRCUIT *is polynomial-time reducible to* POLY^{12}-BIMATRIX.

LEMMA 6.9 (PADDING BIMATRIX GAMES). *If* POLY^c-BIMATRIX *is* **PPAD**-*complete for some constant $c > 0$, then* POLY$^{c'}$-BIMATRIX *is* **PPAD**-*complete for every constant $c' > 0$.*

We will prove Lemma 6.7 and Lemma 6.8, respectively, in Section 8 and Section 7. A proof of Lemma 6.9 can be found in Appendix C.

7. *Simulating Generalized Circuits with Nash Equilibria*

In this section, we reduce POLY^3-GCIRCUIT to POLY^{12}-BIMATRIX and prove Lemma 6.8. Since every two-player game has a Nash equilibrium, this reduction also implies that every generalized circuit with K nodes has a $1/K^3$-approximate solution.

In the construction, we use the game of matching pennies, initially introduced in Step 4 of the DGP framework, to enforce the two players to play the strategies (almost) uniformly. A set of gadgets are then used to simulate the nine types of gates, inspired by the gadget designs for graphical games developed in Goldberg and Papadimitriou [2006] and Daskalakis et al. [2006a].

7.1. OUTLINE OF THE REDUCTION. Suppose $\mathcal{S} = (V, \mathcal{T})$ is a generalized circuit. Let $K = |V|$ and $N = 2K$. Let $\mathcal{C}$ be a bijection from V to $\{1, 3, \ldots, 2K - 3, 2K - 1\}$. From every vector $\mathbf{x} \in \mathbb{R}^N$, we define two maps $\overline{\mathbf{x}}$, $\overline{\mathbf{x}}_{\mathcal{C}} : V \to \mathbb{R}$: For every node $v \in V$, we set

$$\overline{\mathbf{x}}[v] = x_{2k-1} \quad \text{and} \quad \overline{\mathbf{x}}_{\mathcal{C}}[v] = x_{2k-1} + x_{2k},$$

if $\mathcal{C}(v) = 2k - 1$.

In the reduction, we build an $N \times N$ bimatrix game $\mathcal{G}^{\mathcal{S}} = (\mathbf{A}^{\mathcal{S}}, \mathbf{B}^{\mathcal{S}})$ from $\mathcal{S}$. The construction of $\mathcal{G}^{\mathcal{S}}$ takes polynomial time and ensures the following two properties for $\epsilon = 1/K^3 = 8/N^3$:

—Property A_1: $|a^{\mathcal{S}}_{i,j}|, |b^{\mathcal{S}}_{i,j}| \le N^3$, for all $i, j : 1 \le i, j \le N$; and

—Property A_2: for every ϵ-well-supported Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game $\mathcal{G}^{\mathcal{S}}$, $\overline{\mathbf{x}}$ is an ϵ-approximate solution to $\mathcal{S}$.

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

$$\overline{a^{\mathcal{S}}}_{i,j} = \frac{a^{\mathcal{S}}_{i,j} + N^3}{2N^3} \quad \text{and} \quad \overline{b^{\mathcal{S}}}_{i,j} = \frac{b^{\mathcal{S}}_{i,j} + N^3}{2N^3}, \quad \text{for all } i, j : 1 \le i, j \le N.$$

By Lemma 3.2, from any $2/N^{12}$-approximate Nash equilibrium of $\overline{\mathcal{G}^{\mathcal{S}}}$, we can compute a $4/N^6$-well-supported Nash equilibrium of $\overline{\mathcal{G}^{\mathcal{S}}}$ in polynomial time. Since $4/N^6 = \epsilon/(2N^3)$, this is also an ϵ-well-supported Nash equilibrium of $\mathcal{G}^{\mathcal{S}}$. By Property A_2, we can thus compute an ϵ-approximate solution to $\mathcal{S}$, as desired.

In the remainder of this section, we assume $\epsilon = 1/K^3$.

7.2. CONSTRUCTION OF GAME $\mathcal{G}^{\mathcal{S}}$. To construct $\mathcal{G}^{\mathcal{S}}$ we transform a prototype game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$, an $N \times N$ zero-sum game to be defined in Section 7.3, by adding $|\mathcal{T}|$ carefully designed "gadget" games: For each gate $T \in \mathcal{T}$, we define a pair of $N \times N$ matrices $(\mathbf{L}[T], \mathbf{R}[T])$, in accordance with Figure 3. Then, we set

$$\mathcal{G}^{\mathcal{S}} = (\mathbf{A}^{\mathcal{S}}, \mathbf{B}^{\mathcal{S}}), \text{ where } \mathbf{A}^{\mathcal{S}} = \mathbf{A}^* + \sum_{T \in \mathcal{T}} \mathbf{L}[T] \text{ and } \mathbf{B}^{\mathcal{S}} = \mathbf{B}^* + \sum_{T \in \mathcal{T}} \mathbf{R}[T]. \quad (2)$$

$\mathbf{L}[T]$ **and** $\mathbf{R}[T]$**, where gate** $T = (G, v_1, v_2, v, \alpha)$	
Set $\mathbf{L}[T] = (L_{i,j}) = \mathbf{R}[T] = (R_{i,j}) = 0$, $k = \mathcal{C}(v)$, $k_1 = \mathcal{C}(v_1)$ and $k_2 = \mathcal{C}(v_2)$	
G_ζ :	$L_{2k-1,2k} = L_{2k,2k-1} = R_{2k-1,2k-1} = 1,\ R_{i,2k} = \alpha, \forall\, i : 1 \le i \le 2K.$
$G_{\times\zeta}$:	$L_{2k-1,2k-1} = L_{2k,2k} = R_{2k-1,2k} = 1,\ R_{2k_1-1,2k-1} = \alpha.$
$G_=$:	$L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k-1,2k} = 1.$
G_+ :	$L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k-1} = R_{2k-1,2k} = 1.$
G_- :	$L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k} = R_{2k-1,2k} = 1.$
$G_<$:	$L_{2k-1,2k} = L_{2k,2k-1} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k} = 1.$
$G_\vee$:	$L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k-1} = 1,\ R_{i,2k} = 1/(2K),\ \forall\, i \in [2K].$
$G_\wedge$:	$L_{2k-1,2k-1} = L_{2k,2k} = R_{2k_1-1,2k-1} = R_{2k_2-1,2k-1} = 1,\ R_{i,2k} = 3/(2K),\ \forall\, i \in [2K].$
$G_\neg$:	$L_{2k-1,2k} = L_{2k,2k-1} = R_{2k_1-1,2k-1} = R_{2k_1,2k} = 1.$

FIG. 3. Matrices $\mathbf{L}[T]$ and $\mathbf{R}[T]$.

For each $T \in \mathcal{T}$, $\mathbf{L}[T]$ and $\mathbf{R}[T]$ defined in Figure 3, satisfy the following property.

PROPERTY 1. *Let* $T = (G, v_1, v_2, v, \alpha)$, $\mathbf{L}[T] = (L_{i,j})$ *and* $\mathbf{R}[T] = (R_{i,j})$. *Suppose* $\mathcal{C}(v) = 2k - 1$. *Then,*

$$\begin{aligned}
i \notin \{2k, 2k-1\} &\Rightarrow L_{i,j} = 0, \quad \forall\, j \in [2K];\\
j \notin \{2k, 2k-1\} &\Rightarrow R_{i,j} = 0, \quad \forall i \in [2K];\\
i \in \{2k, 2k-1\} &\Rightarrow 0 \le L_{i,j} \le 1, \quad \forall\, j \in [2K];\\
j \in \{2k, 2k-1\} &\Rightarrow 0 \le R_{i,j} \le 1, \quad \forall i \in [2K].
\end{aligned}$$

7.3. THE PROTOTYPE GAME AND ITS PROPERTIES. The prototype game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$ is the bimatrix game called *Generalized Matching Pennies* with parameter $M = 2K^3$. It was used in Goldberg and Papadimitriou [2006] and Daskalakis et al. [2006a] for reducing degree-3 graphical games to four-player games. In $\mathcal{G}^*$, $\mathbf{A}^*$ is an $N \times N$ matrix:

$$\mathbf{A}^* = \begin{pmatrix}
M & M & 0 & 0 & \cdots & 0 & 0\\
M & M & 0 & 0 & \cdots & 0 & 0\\
0 & 0 & M & M & \cdots & 0 & 0\\
0 & 0 & M & M & \cdots & 0 & 0\\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\
0 & 0 & 0 & 0 & \cdots & M & M\\
0 & 0 & 0 & 0 & \cdots & M & M
\end{pmatrix},$$

and $\mathbf{B}^* = -\mathbf{A}^*$. As one can see, $\mathbf{A}^*$ is a $K \times K$ block-diagonal matrix where each diagonal block is a 2×2 matrix of all M's. All games we will consider below belong to the following class:

Definition 7.1 (*Class* $\mathcal{L}$). A bimatrix game $(\mathbf{A}, \mathbf{B})$ is a member of $\mathcal{L}$ if the entries in $\mathbf{A} - \mathbf{A}^*$ and $\mathbf{B} - \mathbf{B}^*$ are in [0, 1].

Note that every Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}^*$ enjoys the following property: For all $v \in V$, $\overline{\mathbf{x}}_C[v] = \overline{\mathbf{y}}_C[v] = 1/K$. We first prove an extension of this property for bimatrix games in $\mathcal{L}$. Recall $\epsilon = 1/K^3$.

LEMMA 7.2 (NEARLY UNIFORM CAPACITIES). *For every bimatrix game* $(\mathbf{A}, \mathbf{B})$ *in* $\mathcal{L}$, *if* $(\mathbf{x}, \mathbf{y})$ *is a* 1*-well-supported Nash equilibrium of* $(\mathbf{A}, \mathbf{B})$*, then*

$$1/K - 1/K^3 \ \le\ \overline{\mathbf{x}}_C[v],\ \overline{\mathbf{y}}_C[v] \ \le\ 1/K + 1/K^3, \quad \textit{for all } v \in V.$$

PROOF. Recall that $\langle \mathbf{a}|\mathbf{b}\rangle$ denotes the inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$ of the same length. By the definition of class $\mathcal{L}$, for each k, the $2k-1$st and $2k$th entries of rows $\mathbf{a}_{2k-1}$ and $\mathbf{a}_{2k}$ in $\mathbf{A}$ are in $[M, M+1]$ and all other entries in these two rows are in $[0, 1]$. Thus, for any probability vector $\mathbf{y} \in \mathbb{P}^N$ and for each node $v \in V$, supposing $\mathcal{C}(v) = 2k-1$, we have

$$M\overline{\mathbf{y}}_C[v] \ \le\ \langle \mathbf{a}_{2k-1}|\mathbf{y}\rangle\,,\ \langle \mathbf{a}_{2k}|\mathbf{y}\rangle \ \le\ M\overline{\mathbf{y}}_C[v] + 1. \tag{3}$$

Similarly, the $(2l-1)$th and $2l$th entries of columns vectors $\mathbf{b}_{2l-1}$ and $\mathbf{b}_{2l}$ in $\mathbf{B}$ are in $[-M, -M+1]$ and all other entries in these two columns are in $[0, 1]$. Thus, for any $\mathbf{x} \in \mathbb{P}^N$ and for each node $v \in V$, supposing $\mathcal{C}(v) = 2l-1$, we have

$$-M\overline{\mathbf{x}}_C[v] \ \le\ \langle \mathbf{b}_{2l-1}|\mathbf{x}\rangle\,,\ \langle \mathbf{b}_{2l}|\mathbf{x}\rangle \ \le\ -M\overline{\mathbf{x}}_C[v] + 1. \tag{4}$$

Now suppose $(\mathbf{x}, \mathbf{y})$ is a t-well-supported Nash equilibrium of $(\mathbf{A}, \mathbf{B})$ where $t \le 1$. We first prove that for each node $v \in V$, if $\overline{\mathbf{y}}_C[v] = 0$ then $\overline{\mathbf{x}}_C[v] = 0$. Note that $\overline{\mathbf{y}}_C[v] = 0$ implies that there exists $v' \in V$ such that $\overline{\mathbf{y}}_C[v'] > 1/K$. Suppose $\mathcal{C}(v) = 2l-1$ and $\mathcal{C}(v') = 2k-1$. By Inequality (3),

$$\langle \mathbf{a}_{2k}|\mathbf{y}\rangle - \max\big(\langle \mathbf{a}_{2l}|\mathbf{y}\rangle\,,\langle \mathbf{a}_{2l-1}|\mathbf{y}\rangle\big) \ge M\overline{\mathbf{y}}_C[v'] - \big(M\overline{\mathbf{y}}_C[v] + 1\big) \ge M/K - 1 > 1.$$

In other words, the payoff of the first player when choosing the $2k$th row is more than 1 plus the payoff when choosing the $2l$th or the $(2l-1)$th row. Because $(\mathbf{x}, \mathbf{y})$ is a t-well-supported Nash equilibrium with $t \le 1$, we have $\overline{\mathbf{x}}_C[v] = 0$.

Next, we prove $|\overline{\mathbf{x}}_C[v] - 1/K| < 1/K^3$ for all $v \in V$. To derive a contradiction, we assume that this statement is not true. Then, there exist v and $v' \in V$ such that $\overline{\mathbf{x}}_C[v] - \overline{\mathbf{x}}_C[v'] > 1/K^3$. Suppose $\mathcal{C}(v) = 2l-1$ and $\mathcal{C}(v') = 2k-1$. By (4),

$$\langle \mathbf{b}_{2k}|\mathbf{x}\rangle - \max\big(\langle \mathbf{b}_{2l}|\mathbf{x}\rangle\,,\langle \mathbf{b}_{2l-1}|\mathbf{x}\rangle\big) \ge -M\overline{\mathbf{x}}_C[v'] - \big(-M\overline{\mathbf{x}}_C[v] + 1\big) > 1,$$

since $M = 2K^3$. This implies $\overline{\mathbf{y}}_C[v] = 0$, which as shown above implies $\overline{\mathbf{x}}_C[v] = 0$, contradicting our assumption that $\overline{\mathbf{x}}_C[v] > \overline{\mathbf{x}}_C[v'] + 1/K^3 > 0$.

We can similarly show that $|\overline{\mathbf{y}}_C[v] - 1/K| < 1/K^3$ for all $v \in V$, and the lemma follows. $\square$

7.4. CORRECTNESS OF THE REDUCTION. We now prove that, for every ϵ-well supported equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}^{\mathcal{S}}$, $\overline{\mathbf{x}}$ must be an ϵ-approximate solution to $\mathcal{S} = (V, \mathcal{T})$. It suffices to show, as we do in the next two lemmas, that $\overline{\mathbf{x}}$ satisfies the following collection of $1 + |\mathcal{T}|$ constraints.

$$\{\mathcal{P}[\epsilon],\ \text{and } \mathcal{P}[T, \epsilon] : T \in \mathcal{T}\}.$$

LEMMA 7.3 (CONSTRAINT $\mathcal{P}[\epsilon]$). *Bimatrix game* $\mathcal{G}^{\mathcal{S}}$ *is in* $\mathcal{L}$, *and for every* ϵ*-well-supported Nash equilibrium* $(\mathbf{x}, \mathbf{y})$ *of* $\mathcal{G}^{\mathcal{S}}$, $\overline{\mathbf{x}}$ *satisfies constraint*

$$\mathcal{P}[\epsilon] = [\ 0 \le \overline{\mathbf{x}}[v] \le 1/K + \epsilon, \forall\, v \in V\ ].$$

PROOF. We only need to show that $\mathcal{G}^{\mathcal{S}} \in \mathcal{L}$. The second statement of the lemma then follows directly from Lemma 7.2.

Let $\mathcal{G}^{\mathcal{S}} = (\mathbf{A}^{\mathcal{S}}, \mathbf{B}^{\mathcal{S}})$, $\mathbf{A}^{\mathcal{S}} = (A^{\mathcal{S}}_{i,j})$, $\mathbf{B}^{\mathcal{S}} = (B^{\mathcal{S}}_{i,j})$, $\mathbf{A}^* = (A^*_{i,j})$ and $\mathbf{B}^* = (B^*_{i,j})$. By Eq.(2), we have

$$A^{\mathcal{S}}_{i,j} - A^*_{i,j} = \sum_{T \in \mathcal{T}} L_{i,j}[T] \quad \text{and} \quad B^{\mathcal{S}}_{i,j} - B^*_{i,j} = \sum_{T \in \mathcal{T}} R_{i,j}[T]$$

for all $i, j : 1 \le i, j \le 2K$. Here, we let $L_{i,j}[T]$ and $R_{i,j}[T]$ denote the (i, j)th entry of $\mathbf{L}[T]$ and $\mathbf{R}[T]$, respectively.

Now consider a pair $i, j : 1 \le i, j \le 2K$. By Property 1, $L_{i,j}[T]$ is nonzero only when the output node (or the fourth component) v of T satisfies $\mathcal{C}(v) = 2k - 1$ and $i \in \{2k, 2k - 1\}$. It then follows from the definition[9] of generalized circuits that there is at most one gate $T \in \mathcal{T}$ such that $L_{i,j}[T] \neq 0$. By Property 1, this nonzero $L_{i,j}[T]$ is between 0 and 1. As a result, we have $0 \le A^{\mathcal{S}}_{i,j} - A^*_{i,j} \le 1$.

It can be proved similarly that $0 \le B^{\mathcal{S}}_{i,j} - B^*_{i,j} \le 1$ for all i, j. Therefore, $\mathcal{G}^{\mathcal{S}}$ is in $\mathcal{L}$ and the lemma is proven. □

LEMMA 7.4 (CONSTRAINTS $P[T, \epsilon]$). *Let $(\mathbf{x}, \mathbf{y})$ be an ϵ-well-supported Nash equilibrium of $\mathcal{G}^{\mathcal{S}}$. Then, for each gate $T \in \mathcal{T}$, $\overline{\mathbf{x}}$ satisfies constraint $\mathcal{P}[T, \epsilon]$.*

PROOF. Recall $\mathcal{P}[T, \epsilon]$ is a constraint defined in Figure 2. By Lemma 7.3, we have

$$1/K - \epsilon \le \overline{\mathbf{x}}_C[v],\ \overline{\mathbf{y}}_C[v] \le 1/K + \epsilon, \quad \text{for all } v \in V.$$

Let $T = (G, v_1, v_2, v, \alpha)$ be a gate in $\mathcal{T}$, and $\mathcal{C}(v) = 2k - 1$. Let $\mathbf{a}^*_i$ and $\mathbf{l}_i$ denote the ith row vectors of $\mathbf{A}^*$ and $\mathbf{L}[T]$, respectively; let $\mathbf{b}^*_j$ and $\mathbf{r}_j$ denote the jth column vectors of $\mathbf{B}^*$ and $\mathbf{R}[T]$, respectively.

From Property 1, $\mathbf{L}[T]$ and $\mathbf{R}[T]$ are the only two gadget matrices that modify the entries in rows $\mathbf{a}^*_{2k-1}, \mathbf{a}^*_{2k}$ or columns $\mathbf{b}^*_{2k-1}, \mathbf{b}^*_{2k}$ in the transformation from the prototype $\mathcal{G}^*$ to $\mathcal{G}^{\mathcal{S}}$. Thus,

$$\mathbf{a}^{\mathcal{S}}_{2k-1} = \mathbf{a}^*_{2k-1} + \mathbf{l}_{2k-1}, \quad \mathbf{a}^{\mathcal{S}}_{2k} = \mathbf{a}^*_{2k} + \mathbf{l}_{2k}; \quad \text{and} \tag{5}$$
$$\mathbf{b}^{\mathcal{S}}_{2k-1} = \mathbf{b}^*_{2k-1} + \mathbf{r}_{2k-1}, \quad \mathbf{b}^{\mathcal{S}}_{2k} = \mathbf{b}^*_{2k} + \mathbf{r}_{2k}. \tag{6}$$

Now, we prove $\overline{\mathbf{x}}$ satisfies constraint $\mathcal{P}[T, \epsilon]$. Here we only consider the case when $G = G_+$. In this case, we need to prove $\overline{\mathbf{x}}[v] = \min(\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2], 1/K) \pm \epsilon$. Proofs for other types of gates are similar and can be found in Appendix D.

Since $\mathbf{a}^*_{2k-1} = \mathbf{a}^*_{2k}$ and $\mathbf{b}^*_{2k-1} = \mathbf{b}^*_{2k}$, from (5), (6) and Figure 3, we have

$$\langle \mathbf{x} | \mathbf{b}^{\mathcal{S}}_{2k-1} \rangle - \langle \mathbf{x} | \mathbf{b}^{\mathcal{S}}_{2k} \rangle = \overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] - \overline{\mathbf{x}}[v], \quad \text{and} \tag{7}$$
$$\langle \mathbf{a}^{\mathcal{S}}_{2k-1} | \mathbf{y} \rangle - \langle \mathbf{a}^{\mathcal{S}}_{2k} | \mathbf{y} \rangle = \overline{\mathbf{y}}[v] - \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big). \tag{8}$$

In a proof by contradiction, we consider two cases. First, we assume

$$\overline{\mathbf{x}}[v] > \min(\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2], 1/K) + \epsilon.$$

As $\overline{\mathbf{x}}[v] \le 1/K + \epsilon$, the assumption would imply $\overline{\mathbf{x}}[v] > \overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] + \epsilon$. By (7) and the definition of ϵ-well-supported Nash equilibria, we have $\overline{\mathbf{y}}[v] = y_{2k-1} = 0$. On the other hand, since $\overline{\mathbf{y}}_C[v] = 1/K \pm \epsilon \gg \epsilon$, by Eq. (8), we have $\overline{\mathbf{x}}[v] = x_{2k-1} = 0$, contradicting our assumption that $\overline{\mathbf{x}}[v] > \overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] + \epsilon > 0$.

[9] See (1): gates in $\mathcal{T}$ have distinct output nodes.

Next, we assume

$$\overline{\mathbf{x}}[v] < \min(\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2], 1/K) - \epsilon \leq \overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] - \epsilon.$$

Then, Eq. (7) implies $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$. By Eq. (8), we have $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v]$, and thus, $\overline{\mathbf{x}}[v] \geq 1/K - \epsilon$, which contradicts our assumption that $\overline{\mathbf{x}}[v] < \min(\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2], 1/K) - \epsilon \leq 1/K - \epsilon$. □

We have now completed the proof of Lemma 6.8.

8. *Computing Fixed Points with Generalized Circuits*

In this section, we show that fixed points can be modeled by generalized circuits. In particular, we reduce the search for a panchromatic simplex in an instance of BROUWER^{f_1} to POLY^3-GCIRCUIT. Recall that $f_1(n) = 3$ and POLY^3-GCIRCUIT is the search problem of finding a $1/K^3$-approximate solution to a generalized circuit with K nodes. In this section, we will simply refer to BROUWER^{f_1} as BROUWER.

At a high level, our reduction will follow Step 3.1 of the DGP framework. However, because our reduction will start with an instance of BROUWER instead of an instance of 3-DIMENSIONAL BROUWER as in the DGP framework, we will need to develop a new sampling method (see Section 8.1) to overcome the curse of dimensionality.

Suppose $U = (C, 0^{3n})$ is an input instance of BROUWER which colors the hypergrid $B^n = \{0, 1, \ldots, 7\}^n$ with colors from $\{1, \ldots, n, n+1\}$. Let m be the smallest integer such that $2^m \geq \text{Size}[C] > n$ and $K = 2^{6m}$. As above, Size$[C]$ denotes the number of gates plus the number of input and output variables in the boolean circuit C. Note that $m = O(\log(\text{Size}[C]))$. Hence, $2^{\Theta(m)}$ is polynomial in the input size of U.

We will construct a generalized circuit $\mathcal{S}^U = (V, \mathcal{T}^U)$ with $|V| = K$ in polynomial time. Our construction ensures that,

—Property **R**: From every $(1/K^3)$-approximate solution to $\mathcal{S}^U$, we can compute a panchromatic simplex P of circuit C in polynomial time.

Lemma 6.7 then follows directly. In the rest of the section, we assume $\epsilon = 1/K^3$.

8.1. OVERCOMING THE CURSE OF DIMENSIONALITY.

In Step 3.1 of the DGP framework, they developed a beautiful *sampling* and *averaging* technique to characterize the discrete fixed points (i.e., the panchromatic cubes) of 3-DIMENSIONAL BROUWER. This lemma provides a computationally efficient way to express the conditions of discrete fixed points.

In this section, we first briefly describe their sampling lemma and explain why it is no longer computationally efficient in high dimensions. Then, we overcome the curse of dimensionality with a new sampling method for characterizing the high-dimensional fixed points of BROUWER (see Lemma 8.2). In the rest of the section, we will translate the conditions of discrete fixed points, as expressed in this new lemma, into the language of generalized circuits, and build $\mathcal{S}^U$ from U.

Suppose C is a Boolean circuit that generates a valid 4-color assignment Color$_C$ from $\{0, 1 \ldots, 2^n - 1\}^3$ to $\{1, 2, 3, 4\}$. Let

$$S = \{\mathbf{p}^{\mathbf{t}} : \mathbf{t} \in \mathbb{Z}^3 \text{ and } |t_i| \leq 20 \text{ for all } i \in [3]\} \subset \mathbb{R}^3_+$$

be a $41 \times 41 \times 41$ grid such that

$$\mathbf{p^t} = \mathbf{p^0} + \sum_{i=1,2,3} t_i \cdot (\alpha \mathbf{e}_i), \quad \text{for all } \mathbf{t},$$

where α is a constant much smaller than 1. Here we let $\mathbb{R}_+$ denote the set of non-negative real numbers. The points in S are called sampling points, which sample Color_C in the following way: For each $\mathbf{p^t} \in S$, we use $\mathbf{q^t}$ to denote the point in $\{0, \ldots, 2^n - 1\}^3$ such that

$$q_i^{\mathbf{t}} = \max \left\{ j \,\middle|\, j \in \{0, \ldots, 2^n - 1\} \text{ and } j \le p_i^{\mathbf{t}} \right\}, \quad \text{for all } i \in [3].$$

Then, a vector $\mathbf{r^t} \in \mathbb{R}^3$ is assigned to each point $\mathbf{p^t}$ according to the color of $\mathbf{q^t}$ in Color_C: If $\text{Color}_C(\mathbf{q^t}) = i \in \{1, 2, 3\}$, then $\mathbf{r^t} = \mathbf{e}_i$; Otherwise, $\mathbf{r^t} = (-1, -1, -1)$. The geometric lemma of Daskalakis et al. [2006a] states that if $\| \sum_{\mathbf{p^t} \in S} \mathbf{r^t} \|_\infty$ is small, then there must exist a panchromatic cube around S.

However, the high-dimensional version of this lemma fails to provide a computationally efficient way to characterize high-dimensional fixed points. The number of points in the sampling grid is 41^n, which is exponential in dimension n. Consequently, one can not encode the high-dimensional version of this lemma by a polynomial-size generalized circuit. Below, we present a new lemma with an efficient sampling structure that uses only a polynomial number of sampling points. We begin with some notation.

For $a \in \mathbb{R}_+$, let

$$\pi(a) = \max \left\{ i \,\middle|\, i \in \{0, 1, \ldots, 7\} \text{ and } i \le a \right\}.$$

For any $\mathbf{p} \in \mathbb{R}^n_+$, let $\mathbf{q} = \pi(\mathbf{p})$ be the point in $B^n = \{0, 1, \ldots, 7\}^n$ with $q_i = \pi(p_i)$.

For $1 \le i \le n$, we let $\mathbf{e}_i$ denote the unit vector in $\mathbb{R}^n$ whose ith entry is equal to 1 and other entries are 0. Let $\mathbf{z}^i = \mathbf{e}_i / K^2 \in \mathbb{R}^n$ for all $1 \le i \le n$, and

$$\mathbf{z}^{n+1} = - \sum_{1 \le i \le n} \mathbf{e}_i / K^2.$$

We use E^n to denote the set of these $n+1$ vectors: $E^n = \{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n, \mathbf{z}^{n+1}\}$.

For every point $\mathbf{p} \in \mathbb{R}^n_+$, we assign a vector in E^n according to the color of point $\pi(\mathbf{p}) \in B^n$ in Color_C: Let ξ be a map from $\mathbb{R}^n_+$ to E^n, where

$$\xi(\mathbf{p}) = \mathbf{z}^{\text{Color}_C[\pi(\mathbf{p})]}, \quad \text{for all } \mathbf{p} \in \mathbb{R}^n_+.$$

Definition 8.1 (*Well-Positioned Points*). A real number $a \in \mathbb{R}_+$ is *poorly-positioned* if there is an integer $t \in \{0, 1, \ldots, 7\}$ 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 $S = \{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^{|S|}\}$ be a set of points in $\mathbb{R}^n_+$. We define

$$I_{\mathrm{P}}(S) = \left\{ k \,\middle|\, \mathbf{p}^k \text{ is poorly positioned} \right\}, \text{ and}$$
$$I_{\mathrm{W}}(S) = \left\{ k \,\middle|\, \mathbf{p}^k \text{ is well-positioned} \right\}.$$

The subscripts "P" and "W" stand for "poorly positioned" and "well-positioned", respectively.

LEMMA 8.2 (KEY GEOMETRY: EQUIANGLE AVERAGING). *Suppose* $U = (C, 0^{3n})$ *is an instance of* BROUWER. *Let* $S = \{\mathbf{p}^i : 1 \le i \le n^3\}$ *be* n^3 *points in* $\mathbb{R}^n_+$

that satisfy

$$\mathbf{p}^i = \mathbf{p}^1 + (i-1)\sum_{j\in[n]} \mathbf{e}_j/K, \quad \textit{for all } i : 2 \le i \le n^3. \tag{9}$$

For each k in $I_{\mathrm{P}}(S)$, let $\mathbf{r}^k \in \mathbb{R}^n$ be an arbitrary vector such that $\|\mathbf{r}^k\|_\infty \le 1/K^2$.
Then, $Q = \{\pi(\mathbf{p}^k) : k \in I_{\mathrm{W}}(S)\}$ is a panchromatic simplex of C if

$$\Big\| \textstyle\sum_{k\in I_{\mathrm{W}}(S)} \xi(\mathbf{p}^k) + \sum_{k\in I_{\mathrm{P}}(S)} \mathbf{r}^k \Big\|_\infty \le \epsilon.$$

PROOF. We first prove that set $Q' = \{\mathbf{q}^k = \pi(\mathbf{p}^k) : 1 \le k \le n^3\}$ is accommodated, and satisfies $|Q'| \le n+1$. As sequence $\{\mathbf{p}^k\}_{1\le k\le n^3}$ is strictly increasing, $\{\mathbf{q}^k\}_{1\le k\le n^3}$ is nondecreasing. Since $n^3/K \ll 1$, there exists at most one k_i for each $i \in [n]$, such that $q_i^{k_i} = q_i^{k_i-1} + 1$, which implies that Q' is accommodated. Since $\{\mathbf{q}^k\}$ is nondecreasing, $|Q'| \le n+1$. Because $Q \subset Q'$, Q is also accommodated and $|Q| \le n+1$.

Next, we give an upper bound for $|I_{\mathrm{P}}(S)|$. 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, we have $|I_{\mathrm{P}}(S)| \le n$ and $|I_{\mathrm{W}}(S)| \ge n^3 - n$.

Let W_i denote the number of points in $\{\mathbf{q}^k : k \in I_{\mathrm{W}}(S)\}$ that are colored i by circuit C. To prove Q is a panchromatic simplex, it suffices to show that $W_i > 0$ for all $i \in [n+1]$.

Let

$$\mathbf{r}^G = \sum_{k\in I_{\mathrm{W}}(S)} \xi(\mathbf{p}^k) \quad \text{and} \quad \mathbf{r}^B = \sum_{k\in I_{\mathrm{P}}(S)} \mathbf{r}^k.$$

Since $|I_{\mathrm{P}}(S)| \le n$ and $\|\mathbf{r}^k\|_\infty \le 1/K^2$, we have

$$\|\mathbf{r}^B\|_\infty \le n/K^2 \quad \text{and} \quad \|\mathbf{r}^G\|_\infty \le \|\mathbf{r}^B\|_\infty + \epsilon \le n/K^2 + \epsilon. \tag{10}$$

Assume for the sake of contradiction that one of W_i is zero:

—*Case* 1. $W_{n+1} = 0$. Let $W_{i^*} = \max_{1\le i\le n} W_i$ then we have $W_{i^*} \ge n^2 - 1$ because $|I_{\mathrm{W}}(S)| \ge n^3 - n$. But

$$r_{i^*}^G \ge (n^2-1)/K^2 \gg n/K^2 + \epsilon,$$

which contradicts (10) above, since $\epsilon = 1/K^3$.

—*Case* 2. $W_t = 0$, for some $t \in [n]$. We assert that $W_{n+1} \le n^2/2$, for otherwise, $|r_t^G| > n^2/(2K^2) \gg n/K^2 + \epsilon$, contradicting (10). Let $W_{i^*} = \max_{1\le i\le n+1} W_i$. Then, $W_{i^*} \ge n^2 - 1$ and $i^* \ne n+1$. So

$$r_{i^*}^G \ge (n^2 - 1 - n^2/2)/K^2 \gg n/K^2 + \epsilon,$$

contradicting (10).

As a result, $W_i > 0$ for all $i \in [n+1]$, and the lemma follows. □

8.2. CONSTRUCTION OF THE GENERALIZED CIRCUIT $\mathcal{S}^{\mathbf{U}}$. We now show how to perform the sampling operations in Lemma 8.2 using a generalized circuit. The construction of $\mathcal{S}^U$ is almost the same as Step 3.1 of the DGP framework, except that we make critical use of the new sampling lemma. Given an input

```
ExtractBits (S, v, v1, v2, v3)
pick four unused nodes u1, u2, u3, u4 ∈ V
Insert(S, (G=, v, nil, u1, nil))
for j from 1 to 3 do
    pick two unused nodes wj1, wj2 ∈ V
    Insert (S, (Gζ, nil, nil, wj1, 2^-(6m+j))), Insert (S, (G<, wj1, uj, vj, nil))
    Insert (S, (G×ζ, vj, nil, wj2, 2^-j)), Insert (S, (G−, uj, wj2, uj+1, nil))
```

FIG. 4. Function EXTRACTBITS.

$U = (C, 0^{3n})$ of BROUWER, our objective is to design a generalized circuit $\mathcal{S}^U = (V, \mathcal{T}^U)$ with $|V| = K$, such that, from any ϵ-approximate solution to $\mathcal{S}^U$, one can find a panchromatic simplex of C in polynomial time (recall $\epsilon = 1/K^3$).

The construction of $\mathcal{S}^U$ goes as follows. There are n^4 distinguished nodes in $\mathcal{S}^U$. We first insert appropriate gates to connect these nodes, so that in any ϵ-approximate solution, the values of these nodes encode a set S of n^3 points $\mathbf{p}^1, \ldots, \mathbf{p}^{n^3} \in \mathbb{R}^n_+$ that approximately satisfy Eq. (9). Starting from these n^4 nodes, we insert a number of gates to simulate the π function, the boolean circuit C, and finally, the map ξ for each $\mathbf{p}^i$. This means for each $1 \le i \le n^3$, there are n nodes in $\mathcal{S}^U$ such that in any ϵ-approximate solution to $\mathcal{S}^U$, the values of these n nodes are very close to $\xi(\mathbf{p}^i)$, provided that $\mathbf{p}^i$ is well-positioned. Then, following Lemma 8.2, we compute the sum of these $\xi(\mathbf{p}^i)$ vectors. Finally, more gates are inserted and cycles are formed in the underlying directed graph of $\mathcal{S}^U$ to enforce that in every ϵ-approximate solution, the sum of $\xi(\mathbf{p}^i)$ is very close to zero.

Now suppose we are given an ϵ-approximate solution to $\mathcal{S}^U$. We can extract the n^3 points $\mathbf{p}^i$ encoded by the values of the n^4 nodes, and compute the set Q (as defined in Lemma 8.2) efficiently. By similar but more complicated arguments used in proving Lemma 8.2, we prove in Section 8.3 that Q must be a panchromatic simplex of Color_C, and complete the reduction.

Let us define some notation that will be useful. Suppose $\mathcal{S} = (V, \mathcal{T})$ is a generalized circuit with $|V| = K$. A node $v \in V$ is said to be *unused* in $\mathcal{S}$ if none of the gates $T \in \mathcal{T}$ uses v as its output node. Now, suppose $T \notin \mathcal{T}$ is a gate such that the output node of T is unused in $\mathcal{S}$. We use INSERT($\mathcal{S}$, T) to denote the insertion of T into $\mathcal{S}$. After calling INSERT($\mathcal{S}$, T), $\mathcal{S}$ becomes $(V, \mathcal{T} \cup \{T\})$.

To encode n^3 points in $\mathbb{R}^n_+$, let $\{v_i^k\}_{1\le k\le n^3, 1\le i\le n}$ be n^4 distinguished nodes in V. We start with $\mathcal{S}^U = (V, \emptyset)$ and insert a number of gates into it so that, in any ϵ-approximate solution $\mathbf{x}$, the values of these nodes encode n^3 points $S = \{\mathbf{p}^k : 1 \le k \le n^3\}$ that approximately satisfy all the conditions of Lemma 8.2. In the encoding, we represent p_i^k as $p_i^k = 8K\,\mathbf{x}[v_i^k]$ for all k, i. Recall that $\mathbf{x}[v_i^k]$ is the value of node v_i^k in $\mathbf{x}$.

We define two functions EXTRACTBITS and COLORINGSIMULATION. They are the building blocks in the construction. EXTRACTBITS implements the π function, and is given in Figure 4. It has the following property (recall the $=_B^\epsilon$ notation: $\mathbf{x}[v] =_B^\epsilon 1$, if $1/K - \epsilon \le \mathbf{x}[v] \le 1/K + \epsilon$; and $\mathbf{x}[v] =_B^\epsilon 0$, if $0 \le \mathbf{x}[v] \le \epsilon$):

LEMMA 8.3 (ENCODING BINARY). *Let $\mathcal{S} = (V, \mathcal{T})$ be a generalized circuit with $|V| = K$. For each node $v \in V$ and three unused nodes $v_1, v_2, v_3 \in V$,* *we use $\mathcal{S}'$ to denote the generalized circuit obtained after calling* ExtractBits $(\mathcal{S}, v, v_1, v_2, v_3)$. *Then, in every ϵ-approximate solution $\mathbf{x}$ of $\mathcal{S}'$, if $a = 8K\mathbf{x}[v]$ is well positioned, then $\mathbf{x}[v_i] =^{\epsilon}_{B} b_i$ for all $1 \le i \le 3$, where $b_1b_2b_3$ is the binary representation of the integer $\pi(a) \in \{0, 1, \ldots, 7\}$.*

Proof. First, we consider the case when $\pi(a) = 7$. As $a \ge 7 + 80K\epsilon$, we have $\mathbf{x}[v] \ge 1/(2K) + 1/(4K) + 1/(8K) + 10\epsilon$. Solving the constraints in Figure 4, we find $\mathbf{x}[u_1] \ge \mathbf{x}[v] - 2\epsilon$, $\mathbf{x}[v_1] =^{\epsilon}_{B} 1$ in the first loop, and

$$\begin{aligned}\mathbf{x}[u_2] \;&\ge\; \mathbf{x}[u_1] - \mathbf{x}[w_{12}] - \epsilon \;\ge\; \mathbf{x}[v] - 2\epsilon - (2^{-1}\mathbf{x}[v_1] + \epsilon) - \epsilon\\ &\ge\; \mathbf{x}[v] - 2^{-1}(1/K + \epsilon) - 4\epsilon \;\ge\; 1/(4K) + 1/(8K) + 5\epsilon.\end{aligned}$$

Since $\mathbf{x}[w_{21}] \le 1/(4K) + \epsilon$ and $\mathbf{x}[u_2] - \mathbf{x}[w_{21}] > \epsilon$, we have $\mathbf{x}[v_2] =^{\epsilon}_{B} 1$ and

$$\mathbf{x}[u_3] \;\ge\; \mathbf{x}[u_2] - \mathbf{x}[w_{22}] - \epsilon \;>\; 1/(8K) + 2\epsilon.$$

As a result, $\mathbf{x}[u_3] - \mathbf{x}[w_{31}] > \epsilon$ and $\mathbf{x}[v_3] =^{\epsilon}_{B} 1$.

Next, we consider the general case that $t < \pi(a) < t + 1$ for some $0 \le t \le 6$. Let $b_1b_2b_3$ be the binary representation of t. As a is well positioned, we have

$$\begin{aligned}b_1/(2K) \;+\; b_2/(4K) + b_3/(8K) + 10\epsilon \le \mathbf{x}[v] \le b_1/(2K)\\ +\; b_2/(4K) + (b_3 + 1)/(8K) - 10\epsilon.\end{aligned}$$

With similar arguments, after the first loop one can show that $\mathbf{x}[v_1] =^{\epsilon}_{B} b_1$ and

$$b_2/(4K) + b_3/(8K) + 5\epsilon \;\le\; \mathbf{x}[u_2] \;\le\; b_2/(4K) + (b_3 + 1)/(8K) - 5\epsilon.$$

After the second loop, we have $\mathbf{x}[v_2] =^{\epsilon}_{B} b_2$ and

$$b_3/(8K) + 2\epsilon \;\le\; \mathbf{x}[u_3] \;\le\; (b_3 + 1)/(8K) - 2\epsilon.$$

Thus, $\mathbf{x}[v_3] =^{\epsilon}_{B} b_3$. □

Next, we introduce ColoringSimulation. Suppose $\mathcal{S} = (V, \mathcal{T})$ is a generalized circuit with $|V| = K$. Let $\{v_i\}_{i\in[n]}$ be n nodes in V, and $\{v_i^+, v_i^-\}_{i\in[n]} \subset V$ be $2n$ *unused* nodes. We use $\mathbf{p} \in \mathbb{R}^n_+$ to denote the point encoded by nodes $\{v_i\}_{i\in[n]}$, that is, $p_i = 8K\mathbf{x}[v_i]$. Imagine that $\mathbf{p}$ is a point in

$$S = \{\mathbf{p}^k : 1 \le i \le n^3\}.$$

ColoringSimulation$(\mathcal{S}, \{v_i\}_{i\in[n]}, \{v_i^+, v_i^-\}_{i\in[n]})$ simulates the boolean circuit C on input $\pi(\mathbf{p})$, by inserting gates into $\mathcal{S}$ as follows:

(1) Pick $3n$ *unused* nodes $\{v_{i,j}\}_{i\in[n], j\in[3]}$ in V.
Call ExtractBits$(\mathcal{S}, v_t, v_{t,1}, v_{t,2}, v_{t,3})$, for each $1 \le t \le n$;

(2) View the values of $\{v_{i,j}\}$ as the $3n$ input bits of C.
Insert the corresponding logic gates from $\{G_\vee, G_\wedge, G_\neg\}$ into $\mathcal{S}$ to simulate the evaluation of C, one for each gate in C, and place the $2n$ output bits in $\{v_i^+, v_i^-\}$.

We obtain the following lemma for ColoringSimulation as a direct consequence of Lemma 8.3, and the definitions in Figure 2. Let $\mathcal{S}'$ be the generalized circuit obtained after calling the above ColoringSimulation and $\mathbf{x}$ be an ϵ-approximate solution to $\mathcal{S}'$. We let $\mathbf{p} \in \mathbb{R}^n_+$ denote the point with $p_i = 8K\mathbf{x}[v_i]$ for all $i \in [n]$, and $\mathbf{q} = \pi(\mathbf{p})$. We let $\{\Delta_i^+[\mathbf{q}], \Delta_i^-[\mathbf{q}]\}_{i\in[n]}$ denote the $2n$ output bits of C evaluated at $\mathbf{q}$. Then

LEMMA 8.4 (POINT COLORING). *If* $\mathbf{p}$ *is a well-positioned point, then*

$$\mathbf{x}[v_i^+] =_B^\epsilon \Delta_i^+[\mathbf{q}] \quad \textit{and} \quad \mathbf{x}[v_i^-] =_B^\epsilon \Delta_i^-[\mathbf{q}], \quad \textit{for all } i \in [n].$$

Note that the equations ($=_B^\epsilon$) in Lemma 8.4 hold only when $\mathbf{p}$ is well positioned. Also note that no matter whether $\mathbf{p}$ is well positioned or not, we always have

$$0 \le \mathbf{x}[v_i^+], \mathbf{x}[v_i^-] \le 1/K + \epsilon$$

for all $i \in [n]$, according to the definition of approximate solutions.

Finally, we build the promised generalized circuit $\mathcal{S}^U$ with a four-step construction. We analyze it in the next section. Initially, set $\mathcal{S}^U = (V, \emptyset)$ and $|V| = K$.

Part 1 (*Equiangle Sampling Segment*). Let $\{v_i^k\}_{1\le k\le n^3, 1\le i\le n}$ be n^4 nodes in V. We insert G_ζ gates, with properly chosen parameters, and G_+ gates into $\mathcal{S}^U$ to ensure that every ϵ-approximate solution $\mathbf{x}$ of $\mathcal{S}^U$ satisfies

$$\mathbf{x}[v_i^k] = \min\left(\mathbf{x}[v_i^1] + (k-1)/(8K^2), 1/K\right) \pm O(\epsilon), \tag{11}$$

for all $2 \le k \le n^3$ and $1 \le i \le n$.

Part 2 (*Point Coloring*). Pick $2n^4$ unused nodes $\{v_i^{k+}, v_i^{k-}\}_{i\in[n],k\in[n^3]}$ from V. For every $k \in [n^3]$, we call

$$\text{COLORINGSIMULATION}\left(\mathcal{S}^U, \{v_i^k\}, \{v_i^{k+}, v_i^{k-}\}_{i\in[n]}\right).$$

Part 3 (*Summing Up the Coloring Vectors*). Pick $2n$ unused nodes $\{v_i^+, v_i^-\}_{i\in[n]} \subset V$. Insert properly valued $G_{\times\zeta}$ gates and G_+ gates to ensure that in the resulting generalized circuit $\mathcal{S}^U$ each ϵ-approximate solution $\mathbf{x}$ satisfies

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

Part 4 (*Closing the Loop*). For each $i \in [n]$, pick unused nodes $v_i', v_i'' \in V$ and insert the following gates:

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

8.3. ANALYSIS OF THE REDUCTION. We now prove the correctness of the construction.

Let $\mathbf{x}$ be an ϵ-approximate solution to $\mathcal{S}^U$. Let

$$S = \{\mathbf{p}^k \,\big|\, \text{with } p_i^k = 8K\,\mathbf{x}[v_i^k], 1 \le k \le n^3\}$$

be the set of n^3 points that are extracted from $\mathbf{x}$. Let $I_W = I_W(S)$ and $I_P = I_P(S)$.

We note that $Q = \{\pi(\mathbf{p}^k) : k \in I_W\}$ can be computed in polynomial time, and complete the reduction by showing that Q is a panchromatic simplex of Color_C. The line of the proof is very similar to the one for Lemma 8.2. First, we use the constraints introduced by the gates in Part 1 to prove the following two lemmas:

LEMMA 8.5 (NOT TOO MANY POORLY POSITIONED POINTS). $|I_P| \le n$. *Thus,* $|I_W| \ge n^3 - n$.

PROOF. For each $t \in I_{\mathrm{P}}$, in accordance with the definition of poorly positioned points, there exists an integer $1 \le l \le n$ such that p_l^t is a poorly positioned number. We will prove that, for every integer $1 \le l \le n$, there exists at most one $t \in [n^3]$ such that $p_l^t = 8K\mathbf{x}[v_l^t]$ is poorly positioned, which implies $|I_{\mathrm{P}}| \le n$ immediately.

Assume p_l^t and $p_l^{t'}$ are both poorly positioned, for a pair of integers $1 \le t < t' \le n^3$. Then, from the definition of poorly positioned points, there exists a pair of integers $0 \le k, k' \le 7$,

$$\left|\mathbf{x}[v_l^t] - k/(8K)\right| \le 10\epsilon \quad \text{and} \quad \left|\mathbf{x}[v_l^{t'}] - k'/(8K)\right| \le 10\epsilon. \tag{12}$$

Because (12) implies that $\mathbf{x}[v_l^t] < 1/K - \epsilon$ and $\mathbf{x}[v_l^{t'}] < 1/K - \epsilon$, by (11) of Part 1,

$$\mathbf{x}[v_l^t] = \mathbf{x}[v_l^1] + (t-1)/(8K^2) \pm O(\epsilon), \quad \text{and}$$
$$\mathbf{x}[v_l^{t'}] = \mathbf{x}[v_l^1] + (t'-1)/(8K^2) \pm O(\epsilon).$$

Hence, $\mathbf{x}[v_l^t] < \mathbf{x}[v_l^{t'}]$, $k \le k'$ and

$$\mathbf{x}[v_l^{t'}] - \mathbf{x}[v_l^t] = (t'-t)/(8K^2) \pm O(\epsilon). \tag{13}$$

Note that when $k = k'$, Eq. (12) implies that $\mathbf{x}[v_l^{t'}] - \mathbf{x}[v_l^t] \le 20\epsilon$, while when $k < k'$, it implies that

$$\mathbf{x}[v_l^{t'}] - \mathbf{x}[v_l^t] \ge (k'-k)/(8K) - 20\epsilon \ge 1/(8K) - 20\epsilon.$$

In either case, the derived inequality contradicts (13). Thus, only one of p_l^t or $p_l^{t'}$ can be poorly positioned. □

LEMMA 8.6 (ACCOMMODATED). *$Q = \{\pi(\mathbf{p}^k) : k \in I_{\mathrm{W}}\}$ is accommodated and $|Q| \le n+1$.*

PROOF. To show that Q is accommodated, it suffices to prove the following monotonicity property:

$$q_l^t \le q_l^{t'} \le q_l^t + 1, \quad \text{for all } l \in [n] \text{ and } t, t' \in I_{\mathrm{W}} \text{ such that } t < t'. \tag{14}$$

For the sake of contradiction, we assume that (14) is not true. We need to consider the following two cases.

First, assume $q_l^t > q_l^{t'}$ for some $t, t' \in I_{\mathrm{W}}$ with $t < t'$. Because $q_l^{t'} < q_l^t \le 7$, we have $p_l^{t'} < 7$ and thus, $\mathbf{x}[v_l^{t'}] < 7/(8K)$. As a result, the first component of the min operator in (11) is the smallest for both t and t', implying that $\mathbf{x}[v_l^t] < \mathbf{x}[v_l^{t'}]$ and $p_l^t < p_l^{t'}$. This contradicts the assumption that $q_l^t > q_l^{t'}$.

Otherwise, $q_l^{t'} - q_l^t \ge 2$ for some $t, t' \in I_{\mathrm{W}}$ with $t < t'$. From the definition of π, we have $p_l^{t'} - p_l^t > 1$ and thus, $\mathbf{x}[v_l^{t'}] - \mathbf{x}[v_l^t] > 1/(8K)$. But from (11), we have

$$\mathbf{x}[v_l^{t'}] - \mathbf{x}[v_l^t] \le (t'-t)/(8K^2) + O(\epsilon) < n^3/(8K^2) + O(\epsilon) \ll 1/(8K).$$

As a result, (14) is true.

Next, we prove $|Q| \le n+1$. Note that (14) implies that there exist integers $t_1 < t_2 < \cdots < t_{|Q|} \in I_{\mathrm{W}}$ such that $\mathbf{q}^{t_i}$ is strictly dominated by $\mathbf{q}^{t_{i+1}}$, that is,

$$\mathbf{q}^{t_i} \ne \mathbf{q}^{t_{i+1}} \quad \text{and} \quad q_j^{t_i} \le q_j^{t_{i+1}}, \quad \text{for all } j \in [n].$$

On the one hand, for every $1 \leq l \leq |Q| - 1$, there exists an integer $1 \leq k_l \leq n$ such that $q_{k_l}^{t_{l+1}} = q_{k_l}^{t_l} + 1$. On the other hand, for every $1 \leq k \leq n$, (14) implies that there is at most one $1 \leq l \leq |Q| - 1$ such that $q_k^{t_{l+1}} = q_k^{t_l} + 1$. Therefore, $|Q| \leq n + 1$. $\square$

For each $t \in I_{\mathrm{W}}$, let $c_t \in \{1, 2, \ldots, n + 1\}$ be the color of point $\mathbf{q}^t = \pi(\mathbf{p}^t)$ assigned by Color_C, and for each $i \in \{1, 2, \ldots, n+1\}$, let $W_i = |\{t \in I_{\mathrm{W}} \mid c_t = i\}|$.

The construction in Part 2 and Lemma 8.4 guarantees that:

LEMMA 8.7 (CORRECT ENCODING OF COLORS). *For each $1 \leq k \leq n^3$, let $\mathbf{r}^k$ denote the vector that satisfies $r_i^k = \mathbf{x}[v_i^{k+}] - \mathbf{x}[v_i^{k-}]$ for all $i \in [n]$. Then, for each $t \in I_{\mathrm{W}}$, $\mathbf{r}^t = K\mathbf{z}^{c_t} \pm 2\epsilon$, and for each $t \in I_{\mathrm{P}}$, $\|\mathbf{r}^t\|_\infty \leq 1/K + 2\epsilon$.*

Let $\mathbf{r}$ denote the vector in $\mathbb{R}^n$ such that $r_i = \mathbf{x}[v_i^+] - \mathbf{x}[v_i^-]$ for all $i \in [n]$. From the constraints of the gates inserted in Part 4, we aim to establish $\|\mathbf{r}\|_\infty < 4\epsilon$. However, whether or not this condition holds depends on the values of $\mathbf{x}[v_i^1]$. 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 the correctness proof of the reduction.

LEMMA 8.8 (WELL-CONDITIONED SOLUTION). *For all $i \in [n]$,*

—*if $\mathbf{x}[v_i^1] > 4\epsilon$, then $r_i = \mathbf{x}[v_i^+] - \mathbf{x}[v_i^-] > -4\epsilon$; and*

—*if $\mathbf{x}[v_i^1] < 1/K - 2n^3/K^2$, then $r_i = \mathbf{x}[v_i^+] - \mathbf{x}[v_i^-] < 4\epsilon$.*

PROOF. In order to set up a proof-by-contradiction of the first if-statement, we assume there exists some i such that $\mathbf{x}[v_i^1] > 4\epsilon$ and $\mathbf{x}[v_i^+] - \mathbf{x}[v_i^-] \leq -4\epsilon$.

From the condition imposed by the first gate $(G_+, v_i^1, v_i^+, v_i', nil)$ inserted in Part 4, we have

$$\begin{aligned}\mathbf{x}[v_i'] &= \min(\mathbf{x}[v_i^1] + \mathbf{x}[v_i^+], 1/K) \pm \epsilon \leq \mathbf{x}[v_i^1] + \mathbf{x}[v_i^+] \\ &\quad + \epsilon \leq \mathbf{x}[v_i^1] + \mathbf{x}[v_i^-] - 3\epsilon. \end{aligned} \tag{15}$$

From the condition imposed by the the second gate $(G_-, v_i', v_i^-, v_i'', nil)$, we have

$$\mathbf{x}[v_i''] \leq \max(\mathbf{x}[v_i'] - \mathbf{x}[v_i^-], 0) + \epsilon \leq \max(\mathbf{x}[v_i^1] - 3\epsilon, 0) + \epsilon = \mathbf{x}[v_i^1] - 2\epsilon, \tag{16}$$

where the last equality follows from the assumption that $\mathbf{x}[v_i^1] > 4\epsilon$. Since $\mathbf{x}[v_i^1] \leq 1/K + \epsilon$, we have $\mathbf{x}[v_i''] \leq \mathbf{x}[v_i^1] - 2\epsilon \leq 1/K - \epsilon < 1/K$. So, from the condition imposed by the last gate $(G_=, v_i'', nil, v_i^1, nil)$, we have

$$\mathbf{x}[v_i^1] = \min(\mathbf{x}[v_i''], 1/K) \pm \epsilon = \mathbf{x}[v_i''] \pm \epsilon,$$

which contradicts (16).

Similarly, to prove the second if-statement we assume there exists some $1 \leq i \leq n$ such that $\mathbf{x}[v_i^1] < 1/K - 2n^3/K^2$ and $\mathbf{x}[v_i^+] - \mathbf{x}[v_i^-] \geq 4\epsilon$ in order to derive a contradiction.

From Part 3, we can see that $\mathbf{x}[v_i^+] \leq n^3/K^2 + O(n^3\epsilon)$. Together with the assumption, we have

$$\mathbf{x}[v_i^1] + \mathbf{x}[v_i^+] \leq 1/K - n^3/K^2 + O(n^3\epsilon) < 1/K.$$

Thus, from the condition imposed by the first gate G_+, we have

$$\mathbf{x}[v_i'] = \min(\mathbf{x}[v_i^1] + \mathbf{x}[v_i^+], 1/K) \pm \epsilon = \mathbf{x}[v_i^1] + \mathbf{x}[v_i^+] \pm \epsilon \geq \mathbf{x}[v_i^1] + \mathbf{x}[v_i^-] + 3\epsilon$$

and

$$\mathbf{x}[v_i'] \leq \mathbf{x}[v_i^1] + \mathbf{x}[v_i^+] + \epsilon \leq 1/K - n^3/K^2 + O(n^3\epsilon) < 1/K.$$

From the condition imposed by the second gate G_-,

$$\mathbf{x}[v_i''] \geq \min(\mathbf{x}[v_i'] - \mathbf{x}[v_i^-], 1/K) - \epsilon = \mathbf{x}[v_i'] - \mathbf{x}[v_i^-] - \epsilon \geq \mathbf{x}[v_i^1] + 2\epsilon. \quad (17)$$

We also have

$$\mathbf{x}[v_i''] \leq \max(\mathbf{x}[v_i'] - \mathbf{x}[v_i^-], 0) + \epsilon \leq \mathbf{x}[v_i'] + \epsilon < 1/K.$$

Moreover, the last gate $G_=$ implies that $\mathbf{x}[v_i^1] = \min(\mathbf{x}[v_i''], 1/K) \pm \epsilon = \mathbf{x}[v_i''] \pm \epsilon$, which contradicts (17). □

Now, we show that Q is a panchromatic simplex of C. By Lemma 8.6, it suffices to prove that $W_i > 0$, for all $i \in [n+1]$.

By Part 3 of the construction and Lemma 8.7,

$$\begin{aligned}\mathbf{r} &= \frac{1}{K}\sum_{1\leq i\leq n^3} \mathbf{r}^i \pm O(n^3\epsilon) = \frac{1}{K}\sum_{i\in I_{\mathrm{W}}} \mathbf{r}^i + \frac{1}{K}\sum_{i\in I_{\mathrm{P}}} \mathbf{r}^i \pm O(n^3\epsilon)\\ &= \sum_{i\in I_{\mathrm{W}}} \mathbf{z}^{c_i} + \frac{1}{K}\sum_{i\in I_{\mathrm{P}}} \mathbf{r}^i \pm O(n^3\epsilon) = \sum_{1\leq i\leq n+1} W_i\, \mathbf{z}^i + \frac{1}{K}\sum_{i\in I_{\mathrm{P}}} \mathbf{r}^i \pm O(n^3\epsilon)\\ &= \mathbf{r}^G + \mathbf{r}^B \pm O(n^3\epsilon),\end{aligned}$$

where we let

$$\mathbf{r}^G = \sum_{1\leq i\leq n+1} W_i \mathbf{z}^i \quad \text{and} \quad \mathbf{r}^B = \sum_{i\in I_{\mathrm{P}}} \mathbf{r}^i/K.$$

As $|I_{\mathrm{P}}| \leq n$ and $\|\mathbf{r}^i\|_\infty \leq 1/K + \epsilon$ for each $i \in I_{\mathrm{P}}$, we have $\|\mathbf{r}^B\|_\infty = O(n/K^2)$.

Since $|I_{\mathrm{W}}| \geq n^3 - n$, we have $\sum_{1\leq i\leq n+1} W_i \geq n^3 - n$. The next lemma shows that, if one of W_i is equal to zero, then $\|\mathbf{r}^G\|_\infty$ is much greater than $\|\mathbf{r}^B\|_\infty$.

LEMMA 8.9. *If one of W_i is equal to zero, then $\|\mathbf{r}^G\|_\infty \geq n^2/(3K^2)$, and thus $\|\mathbf{r}\|_\infty > 4\epsilon$.*

PROOF. We divide the proof into two cases. First, assume $W_{n+1} = 0$. Let $l \in [n]$ be the integer such that $W_l = \max_{1\leq i\leq n} W_i$, then we have $W_l > n^2 - 1$. Thus, $r_l' = W_l/K \geq (n^2-1)/K > n^2/(3K^2)$.

Otherwise, assume $W_t = 0$ for some $1 \leq t \leq n$. We have the following two cases:

—$W_{n+1} \geq n^2/2$: $r_t' = -W_{n+1}/K \leq -n^2/(2K^2) < -n^2/(3K^2)$.

—$W_{n+1} < n^2/2$: Let l be the integer such that $W_l = \max_{1\leq i\leq n+1} W_i$. It is easy to see that $l \neq t, n+1$ and $W_l > n^2 - 1$. Then,

$$r_l' = (W_l - W_{n+1})/K > (n^2/2 - 1)/K^2 > n^2/(3K^2).$$

The lemma then follows. □

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

LEMMA 8.10 (WELL CONDITIONED). *For all $i \in [n]$,*

$$4\epsilon < \mathbf{x}[v_i^1] < 1/K - 2n^3/K^2.$$

By Lemma 8.10 and Lemma 8.8, we have $\|\mathbf{r}\|_\infty < 4\epsilon$. It then follows from Lemma 8.9 that all the W_i's are nonzero, and thus, Q is a panchromatic simplex.

PROOF OF LEMMA 8.10. In the proof, we will use the following boundary properties of Color_C: For each $\mathbf{q} \in B^n$ (recall $B^n = \{0, 1, \dots, 7\}^n$) and $1 \leq k \neq l \leq n$,

B.1: If $q_k = 0$, then $\text{Color}_C\,[\mathbf{q}] \neq n+1$;
B.2: If $q_k = 0$ and $q_l > 0$, then $\text{Color}_C\,[\mathbf{q}] \neq l$;
B.3: If $q_k = 7$, then $\text{Color}_C\,[\mathbf{q}] \neq k$; and
B.4: If $q_k = 7$ and $\text{Color}_C\,[\mathbf{q}] = l \neq k$, then $q_l = 0$.

All these properties follow directly from the definition of valid Brouwer circuits.

First, if there exists an integer $k \in [n]$ such that $\mathbf{x}[v_k^1] \leq 4\epsilon$, then $q_k^t = 0$ for all $t \in I_{\mathrm{W}}$. By B.1, $W_{n+1} = 0$. Let l be the integer such that $W_l = \max_{1 \leq i \leq n} W_i$. As $\sum_{1 \leq i \leq n+1} W_i = |I_{\mathrm{W}}| \geq n^3 - n$, we have $W_l \geq n^2 - 1$. As a result,

$$r_l \geq W_l/K^2 - O(n/K^2) - O(n^3\epsilon) > 4\epsilon.$$

Now consider the following two cases:

—If $\mathbf{x}[v_l^1] < 1/K - 2n^3/K^2$, then we get a contradiction from Lemma 8.8.
—If $\mathbf{x}[v_l^1] \geq 1/K - 2n^3/K^2$, then for all $t \in I_{\mathrm{W}}$,

$$p_l^t = 8K\Big(\min\big(\mathbf{x}[v_l^1] + (t-1)/(8K^2), 1/K\big) \pm O(\epsilon)\Big) > 1$$

and hence $q_l^t > 0$. By B.2, $W_l = 0$, contradicting the inequality $W_l \geq n^2 - 1$.

Otherwise, if there exists an integer $k \in [n]$ such that $\mathbf{x}[v_k^1] \geq 1/K - 2n^3/K^2$, then for all $t \in I_{\mathrm{W}}$, we have $q_k^t = 7$. By B.3, $W_k = 0$. If $W_{n+1} \geq n^2/2$, then

$$r_k \leq -W_{n+1}/K^2 + O(n/K^2) + O(n^3\epsilon) < -4\epsilon,$$

which, by the first part of Lemma 8.8, contradicts the assumption that $\mathbf{x}[v_k^1] \geq 1/K - 2n^3/K^2 > 4\epsilon$. Consider the remaining case where $W_{n+1} < n^2/2$.

Let l be the integer such that $W_l = \max_{1 \leq i \leq n+1} W_i$. Since $W_k = 0$, we have $W_l \geq n^2 - 1$ and $l \neq k$. As $W_{n+1} < n^2/2$, $W_l - W_{n+1} > n^2/2 - 1$ and thus,

$$r_l \geq (W_l - W_{n+1})/K^2 - O(n/K^2) - O(n^3\epsilon) > 4\epsilon.$$

We now consider the following two cases:

—If $\mathbf{x}[v_l^1] < 1/K - 2n^3/K^2$, then we get a contradiction by the second part of Lemma 8.8;
—If $\mathbf{x}[v_l^1] \geq 1/K - 2n^3/K^2$, then $p_l^t > 1$ and thus $q_l^t > 0$ for all $t \in I_{\mathrm{W}}$. By B.4, we have $W_l = 0$, which contradicts the assumption.

The lemma then follows. □

9. **PPAD**-*Completeness of* BROUWER

To prove Theorem 6.6, we reduce BROUWER^{f_2} to BROUWER^{f}. Recall that $f_2(n) = \lceil n/2 \rceil$ and an instance of BROUWER^{f_2} is a valid 3-coloring of a 2-dimensional grid. Like its 3-dimensional analog introduced in Daskalakis et al. [2006a], BROUWER^{f_2} is **PPAD**-complete [Chen and Deng 2006]. Thus, our reduction will show that BROUWER^{f} is also **PPAD**-complete.

The basic idea of the reduction is to iteratively embed an instance of BROUWER^{f_2} into a hypergrid one dimension higher. This iterative process eventually folds or embeds the 2-dimensional input instance into the desired hypergrid. We use the following notion to describe the embedding process. A triple $T = (C, d, \mathbf{r})$ is a *coloring triple* if $\mathbf{r} \in \mathbb{Z}^d$ with $r_i \geq 3$ for all $1 \leq i \leq d$ and C is a valid Brouwer-mapping circuit with parameters d and $\mathbf{r}$. Let Size $[C]$ denote the number of gates plus the number of input and output variables in a circuit C.

The embedding is carried out 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' (i.e., the volume of the search space of T' is greater than the one of T) *such that from every panchromatic simplex of T', one can find a panchromatic simplex of T efficiently.*

For the sake of clarity, in the context of this section, we rephrase the definition of BROUWER^{f} as follows: In the original definition, each valid Brouwer-mapping circuit C defines a color assignment from the search space to $\{1, \ldots, d, d+1\}$. In this section, we replace the color $d+1$ by a special color "red". In other words, if the output bits of C evaluated at $\mathbf{p}$ satisfy Case i with $1 \leq i \leq d$, then $\text{Color}_C[\mathbf{p}] = i$; otherwise, the output bits satisfy Case $d+1$ and $\text{Color}_C[\mathbf{p}] = \text{red}$.

We first prove a useful property of valid Brouwer-mapping circuits.

PROPERTY 2 (BOUNDARY CONTINUITY). *Let C be a valid Brouwer-mapping circuit with parameters d and $\mathbf{r}$. If $\mathbf{p} \in \partial(A^d_{\mathbf{r}})$ satisfies $1 \leq p_t \leq r_t - 2$ for some $t \in [d]$, then $\text{Color}_C[\mathbf{p}] = \text{Color}_C[\mathbf{p}']$, where $\mathbf{p}' = \mathbf{p} + \mathbf{e}_t$.*

PROOF. First, it is easy to check that $\mathbf{p}' \in \partial(A^d_{\mathbf{r}})$. Second, by the definition, if C is a valid Brouwer-mapping circuit, then for every $\mathbf{p} \in \partial(A^d_{\mathbf{r}})$, $\text{Color}_C[\mathbf{p}]$ only depends on the set $\{i \mid p_i = 0\}$. When $1 \leq p_t \leq r_t - 2$, we have $\{i \mid p_i = 0\} = \{i \mid p'_i = 0\}$, and thus, $\text{Color}_C[\mathbf{p}] = \text{Color}_C[\mathbf{p}']$. $\square$

9.1. REDUCTIONS AMONG COLORING TRIPLES.

Both $\mathbf{L}^1(T, t, u)$ and $\mathbf{L}^2(T, u)$ are very simple operations:

—Given a coloring triple $T = (C, d, \mathbf{r})$ and two integers $1 \leq t \leq d$ and $u > r_t$, $\mathbf{L}^1(T, t, u)$ pads dimension t to size u, that is, it builds a new coloring triple $T' = (C', d, \mathbf{r}')$ with $r'_t = u$ and $r'_i = r_i$, for all other $i \in [d]$.

—For integer $u \geq 3$, $\mathbf{L}^2(T, u)$ adds a dimension to T by constructing $T' = (C', d+1, \mathbf{r}')$ such that $\mathbf{r}' \in \mathbb{Z}^{d+1}$, $r'_{d+1} = u$ and $r'_i = r_i$, for all $i \in [d]$.

These two transformations are described in Figure 5 and Figure 6, respectively. We prove their properties in the following two lemmas.

LEMMA 9.1 ($\mathbf{L}^1(T, t, u)$: PADDING A DIMENSION). *Given a coloring triple $T = (C, d, \mathbf{r})$ and two integers $1 \leq t \leq d$ and $u > r_t$, $\mathbf{L}_1$ constructs a new*

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

```
if p ∈ ∂(A^d_{r'}) then
    if there exists i such that p_i = 0 then
        Color_{C'}[p] = max{ i | p_i = 0 }
    else
        Color_{C'}[p] = red
else if p_t ≤ r_t − 1 then
    Color_{C'}[p] = Color_C[p]
else
    Color_{C'}[p] = red
```

FIG. 5. How $\mathbf{L}^1(T, t, u)$ extends the coloring triple $T = (C, d, \mathbf{r})$.

coloring triple $T' = (C', d, \mathbf{r}')$ that satisfies the following two conditions:

A. *$r'_t = u$, and $r'_i = r_i$ for all other $i \in [d]$. In addition, there exists a polynomial $g_1(n)$ such that* $\text{Size}\left[C'\right] = \text{Size}\left[C\right] + O(g_1(\text{Size}\left[\mathbf{r}'\right]))$, *and T' can be computed in time polynomial in* $\text{Size}\left[C'\right]$. *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 build circuit C' according to its color assignment described in Figure 5. It has $\text{Size}\left[\mathbf{r}'\right]$ input bits, which encode a point $\mathbf{p} \in A^d_{\mathbf{r}'}$. It first checks whether $\mathbf{p}$ is on the boundary of $A^d_{\mathbf{r}'}$ or not. If $\mathbf{p} \in \partial(A^d_{\mathbf{r}'})$, then C' outputs its color according to the boundary condition. Otherwise, C' checks whether $\mathbf{p} \in A^d_{\mathbf{r}}$ or not. If $\mathbf{p} \in A^d_{\mathbf{r}}$, then C' runs C on $\mathbf{p}$ and outputs $\text{Color}_C[\mathbf{p}]$; Otherwise, C' simply outputs red. Property **A** immediately follows from this construction.

To show Property **B**, let P' be a panchromatic simplex of T', and $K_{\mathbf{p}}$ be the hypercube containing P'. We first note that $p_t < r_t - 1$, because if $p_t \geq r_t - 1$, $K_{\mathbf{p}}$ would not contain color t according to the color assignment. We note that $\text{Color}_{C'}[\mathbf{q}] = \text{Color}_C[\mathbf{q}]$ for all $\mathbf{q} \in A^d_{\mathbf{r}}$. Thus, P' is also a panchromatic simplex of the coloring triple T. $\square$

LEMMA 9.2 ($\mathbf{L}^2(T, u)$: ADDING A DIMENSION). *Given a coloring triple $T = (C, d, \mathbf{r})$ and an integer $u \geq 3$, $\mathbf{L}^2$ constructs a new coloring triple $T' = (C', d + 1, \mathbf{r}')$ satisfying the following conditions:*

A. *$r'_{d+1} = u$, and $r'_i = r_i$ for all $i \in [d]$. Furthermore, there exists a polynomial $g_2(n)$ such that* $\text{Size}\left[C'\right] = \text{Size}\left[C\right] + O(g_2(\text{Size}\left[\mathbf{r}'\right]))$. *$T'$ can be computed in time polynomial in* $\text{Size}\left[C'\right]$. *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.*

PROOF. For each point $\mathbf{q} \in A^{d+1}_{\mathbf{r}'}$, we use $\hat{\mathbf{q}}$ to denote the point in $A^d_{\mathbf{r}}$ with $\hat{q}_i = q_i$, for all $i \in [d]$. The color assignment of circuit C' is given in Figure 6, from which Property **A** follows.

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

```
if p ∈ ∂(A^d_r') then
    if there exists i such that p_i = 0 then
        Color_C' [p] = max{ i | p_i = 0 }
    else
        Color_C' [p] = red
else if p_{d+1} = 1 then
    Color_C' [p] = Color_C [p̂], where p̂ ∈ Z^d satisfies p̂_i = p_i for all i ∈ [d]
else
    Color_C' [p] = red
```

FIG. 6. How $\mathbf{L}^2(T, u)$ extends the coloring triple $T = (C, d, \mathbf{r})$.

To prove Property **B**, we let $P' \subset K_{\mathbf{p}}$ be a panchromatic simplex of T'. Note that $p_{d+1} = 0$, for otherwise, $K_{\mathbf{p}}$ does not contain color $d + 1$. Also note that $\text{Color}_{C'}[\mathbf{q}] = d+1$ for every $\mathbf{q} \in A^{d+1}_{\mathbf{r}'}$ with $q_{d+1} = 0$. Thus, for every $\mathbf{q} \in P'$ with $\text{Color}_{C'}[\mathbf{q}] \neq d + 1$, we have $q_{d+1} = 1$. Finally, because $\text{Color}_{C'}[\mathbf{q}] = \text{Color}_C[\hat{\mathbf{q}}]$ for every $\mathbf{q} \in A^{d+1}_{\mathbf{r}'}$ with $q_{d+1} = 1$, $P = \{\hat{\mathbf{q}} \,|\, \mathbf{q} \in P' \text{ and } \text{Color}_{C'}[\mathbf{q}] \neq d+1\}$ must be a panchromatic simplex of T. □

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

LEMMA 9.3 ($\mathbf{L}^3(T, t, a, b)$: SNAKE EMBEDDING). *Given a coloring triple $T = (C, d, \mathbf{r})$ and an integer $1 \leq t \leq d$, if $r_t = a(2b+1)+5$ for two integers $a, b \geq 1$, then $\mathbf{L}^3$ constructs a new coloring triple $T' = (C', d+1, \mathbf{r}')$ that satisfies:*

A. *$r'_t = a+5$, $r'_{d+1} = 4b+3$, and $r'_i = r_i$ for all other $i \in [d]$. Moreover, there exists a polynomial $g_3(n)$ such that* $\text{Size}[C'] = \text{Size}[C] + O(g_3(\text{Size}[\mathbf{r}']))$ *and T' can be computed in time polynomial in* $\text{Size}[C']$. *We write $T' = \mathbf{L}^3(T, t, a, b)$;*

B. *From each panchromatic simplex P' of coloring triple T', we can compute a panchromatic simplex P of T in polynomial time.*

PROOF. Consider the domains $A^d_{\mathbf{r}} \subset \mathbb{Z}^d$ and $A^{d+1}_{\mathbf{r}'} \subset \mathbb{Z}^{d+1}$ of our coloring triples. We form the reduction $\mathbf{L}^3(T, t, a, b)$ in three steps. First, we define a d-dimensional set $W \subset A^{d+1}_{\mathbf{r}'}$ that is large enough to contain $A^d_{\mathbf{r}}$. Second, we define a map ψ from W to $A^d_{\mathbf{r}}$ that (implicitly) specifies an embedding of $A^d_{\mathbf{r}}$ into W. Finally, we build a circuit C' for $A^{d+1}_{\mathbf{r}'}$ and show that from each panchromatic simplex of C', we can, in polynomial time, compute a panchromatic simplex of C.

A two dimensional view of $W \subset A^{d+1}_{\mathbf{r}'}$ is illustrated in Figure 7. We use a snake-pattern to realize the longer tth dimension of $A^d_{\mathbf{r}}$ in the two-dimensional space defined by the shorter tth and $(d+1)$th dimensions of $A^{d+1}_{\mathbf{r}'}$. Formally, W consists of points $\mathbf{p} \in A^{d+1}_{\mathbf{r}'}$ satisfying $1 \leq p_{d+1} \leq 4b + 1$ and

—if $p_{d+1} = 1$, then $2 \leq p_t \leq a + 4$;
—if $p_{d+1} = 4b + 1$, then $0 \leq p_t \leq a + 2$;

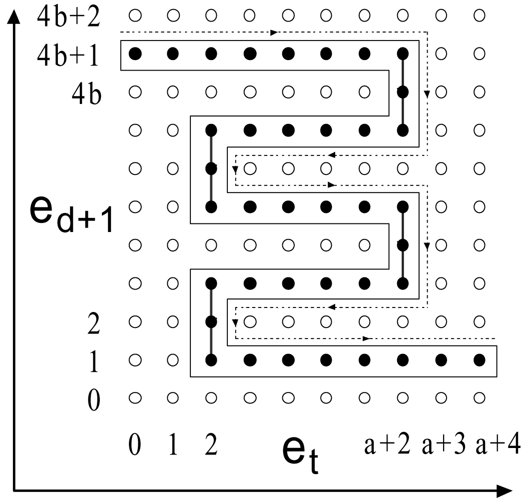

FIG. 7. The two dimensional view of set $W \subset A^{d+1}_{\mathbf{r}'}$.

—if $p_{d+1} = 4(b - i) - 1$ where $0 \le i \le b - 1$, then $2 \le p_t \le a + 2$;
—if $p_{d+1} = 4(b - i) - 3$ where $0 \le i \le b - 2$, then $2 \le p_t \le a + 2$;
—if $p_{d+1} = 4(b - i) - 2$ where $0 \le i \le b - 1$, then $p_t = 2$;
—if $p_{d+1} = 4(b - i)$ where $0 \le i \le b - 1$, then $p_t = a + 2$.

To build T', we embed the coloring triple T into W. The embedding is implicitly given by a surjective map ψ from W to $A^d_{\mathbf{r}}$, a map that will play a vital role in the construction and analysis. For each $\mathbf{p} \in W$, we use $\mathbf{p}[m]$ to denote the point $\mathbf{q}$ in $\mathbb{Z}^d$ with $q_t = m$ and $q_i = p_i$ for all other $i \in [d]$. We define $\psi(\mathbf{p})$ according to the following cases:

—if $p_{d+1} = 1$, then $\psi(\mathbf{p}) = \mathbf{p}[2ab + p_t]$
—if $p_{d+1} = 4b + 1$, then $\psi(\mathbf{p}) = \mathbf{p}[p_t]$;
—if $p_{d+1} = 4(b - i) - 1$ where $0 \le i \le b - 1$, then $\psi(\mathbf{p}) = \mathbf{p}[(2i + 2)a + 4 - p_t]$;
—if $p_{d+1} = 4(b - i) - 3$ where $0 \le i \le b - 2$, then $\psi(\mathbf{p}) = \mathbf{p}[(2i + 2)a + p_t]$;
—if $p_{d+1} = 4(b - i) - 2$ where $0 \le i \le b - 1$, then $\psi(\mathbf{p}) = \mathbf{p}[(2i + 2)a + 2]$;
—if $p_{d+1} = 4(b - i)$ where $0 \le i \le b - 1$, then $\psi(\mathbf{p}) = \mathbf{p}[(2i + 1)a + 2]$.

We let $\psi_i(\mathbf{p})$ denote the ith component of $\psi(\mathbf{p})$.

Essentially, we map W bijectively to $A^d_{\mathbf{r}}$ along its tth dimension with exception that when the snake pattern of W is making a turn, we stop the advance in $A^d_{\mathbf{r}}$, and continue the advance after it completes the turn.

The circuit C' specifies a color assignment of $A^{d+1}_{\mathbf{r}'}$ in accordance with Figure 8. We first prove that $\text{Color}_{C'}$ is valid:

PROPERTY 3 (BOUNDARY PRESERVING). *The coloring described in* Figure 8 *is valid.*

PROOF. It suffices to show that $\text{Color}_{C'}$ satisfies the boundary condition for all $\mathbf{p} \in W \cap \partial(A^{d+1}_{\mathbf{r}'})$.

Color$_{C'}$ $[\mathbf{p}]$ **of a point** $\mathbf{p} \in A^{d+1}_{\mathbf{r}'}$ **assigned by** $(C', d+1, \mathbf{r}') = \mathbf{L}^3(T, t, a, b)$

```
if p ∈ W then
    Color_C' [p] = Color_C [ψ(p)]
else if p ∈ ∂(A^{d+1}_r') then
    if there exists i such that p_i = 0 then
        Color_C' [p] = max{ i | p_i = 0 }
    else
        Color_C' [p] = red
else if p_{d+1} = 4i where 1 ≤ i ≤ b and 1 ≤ p_t ≤ a + 1 then
    Color_C' [p] = d + 1
else if p_{d+1} = 4i + 1, 4i + 2 or 4i + 3 where 0 ≤ i ≤ b − 1 and p_t = 1 then
    Color_C' [p] = d + 1
else
    Color_C' [p] = red
```

FIG. 8. How $\mathbf{L}^3(T, t, a, b)$ extends the coloring triple $T = (C, d, \mathbf{r})$.

Let $\mathbf{p}$ be a point in $W \cap \partial(A^{d+1}_{\mathbf{r}'})$. One can show that $\{i \mid p_i = 0\} = \{i \mid \psi_i(\mathbf{p}) = 0\}$. If there exists i such that $p_i = 0$, then

$$\text{Color}_{C'}[\mathbf{p}] = \text{Color}_C[\psi(\mathbf{p})] = \max\{i \mid \psi_i(\mathbf{p}) = 0\} = \max\{i \mid p_i = 0\},$$

since Color_C is valid.

Otherwise $\{i \mid p_i = 0\} = \emptyset$ and there exists l such that $p_l = r'_l - 1$. In this case, we have $\{i \mid \psi_i(\mathbf{p}) = 0\} = \emptyset$, and $\psi_l(\mathbf{p}) = r_l - 1$. As a result,

$$\text{Color}_{C'}[\mathbf{p}] = \text{Color}_C[\psi(\mathbf{p})] = \text{red},$$

since Color_C is valid. $\square$

By Property 3, we know that C' is a valid Brouwer-mapping circuit with parameters $d+1$ and $\mathbf{r}'$. Property **A** follows from the construction in Figure 8, since whether $\mathbf{p} \in W$ or not can be decided efficiently.

We now establish Property **B** of the lemma. The intuition behind the proof is as follows. In $\text{Color}_{C'}$, points to the right of W are colored in red, and points to the left are colored in $d+1$. Every (unit-size) hypercube $K_{\mathbf{p}} \subset A^{d+1}_{\mathbf{r}'}$ consists of $K_{\mathbf{p}} \cap W$, whose image $\psi(K_{\mathbf{p}} \cap W)$ is a (unit-size) hypercube in $A^d_{\mathbf{r}}$, and either points to the right or left of W. Let P' be a panchromatic simplex of T' in $A^{d+1}_{\mathbf{r}'}$, and $K_{\mathbf{p}^*}$ be the hypercube containing P'. Since hypercubes to the right of W do not contain color $d+1$, $K_{\mathbf{p}^*}$ must lie to the left of W. We will show that, except the point with color $d+1$, every point $\mathbf{p} \in P'$

—either belongs to $W \cap K_{\mathbf{p}^*}$; or

—can be mapped to a point $\mathbf{q} \in W \cap K_{\mathbf{p}^*}$ such that $\text{Color}_{C'}[\mathbf{q}] = \text{Color}_{C'}[\mathbf{p}]$.

Thus, from P', we can recover $d+1$ points in $W \cap K_{\mathbf{p}^*}$ with $d+1$ distinct colors $\{1, \ldots, d, \text{red}\}$. Since $\text{Color}_{C'}[\mathbf{p}] = \text{Color}_C[\psi(\mathbf{p})]$ for all $\mathbf{p} \in W$, we can apply ψ to get a panchromatic simplex P of Color_C.

We prove a collection of statements to cover all the possible cases of the given panchromatic simplex P' of T'. We use the following notation: For each $\mathbf{p} \in A^{d+1}_{\mathbf{r}'}$, let $\mathbf{p}[m_1, m_2]$ denote the point $\mathbf{q} \subset \mathbb{Z}^{d+1}$ such that $q_t = m_1, q_{d+1} = m_2$ and $q_i = p_i$ for all other $i \in [d]$.

STATEMENT 1. *If $p^*_t = 0$, then $p^*_{d+1} = 4b$ and moreover, for every point $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \neq d+1$, $Color_C[\psi(\mathbf{p}[p_t, 4b+1])] = Color_{C'}[\mathbf{p}]$.*

PROOF. First, note that $p^*_{d+1} \neq 4b+1$, for otherwise, $K_{\mathbf{p}^*}$ does not contain color $d+1$. Second, if $p^*_{d+1} < 4b$, then since $p^*_t = 0$, each point $\mathbf{q} \in K_{\mathbf{p}^*}$ is colored according one of the conditions in line 3, 8 or 10 of Figure 8. Let $\mathbf{q}^* \in K_{\mathbf{p}^*}$ be the red point in P'. Then, $\mathbf{q}^*$ must satisfy the condition on line 6 and hence there exists l such that $q^*_l = r'_l - 1$. By our assumption, $p^*_t = 0$. Thus, if $p^*_{d+1} < 4b$, then $l \notin \{t, d+1\}$, implying for each $\mathbf{q} \in K_{\mathbf{p}^*}$, $q_l > 0$ (as $r'_l = r_l \geq 3$ and thus, $q_l \geq q^*_l - 1 > 0$) and $\mathrm{Color}_{C'}[\mathbf{q}] \neq l$. Then, $K_{\mathbf{p}^*}$ does not contain color l, contradicting the assumption of the statement. Putting these two cases together, we have $p^*_{d+1} = 4b$.

We now prove the second part of the statement. If $p_{d+1} = 4b+1$, then we are done, because $\mathrm{Color}_C[\psi(\mathbf{p})] = \mathrm{Color}_{C'}[\mathbf{p}]$ according to lines 1 and 2 of Figure 8. Let us assume $p_{d+1} = 4b$. Since the statement assumes $\mathrm{Color}_{C'}[\mathbf{p}] \neq d+1$, $\mathbf{p}$ satisfies the condition in line 3 and hence $\mathbf{p} \in \partial(A^{d+1}_{\mathbf{r}'})$. By Property 2, we have $\mathrm{Color}_{C'}[\mathbf{p}[p_t, 4b+1]] = \mathrm{Color}_{C'}[\mathbf{p}]$. Since $\mathbf{p}[p_t, 4b+1] \in W$ when $p_t \in \{0, 1\}$, we have $\mathrm{Color}_C[\psi(\mathbf{p}[p_t, 4b+1])] = \mathrm{Color}_{C'}[\mathbf{p}[p_t, 4b+1]] = \mathrm{Color}_{C'}[\mathbf{p}]$, completing the proof of the statement. □

STATEMENT 2. *If $p^*_t = a+2$ or $a+3$, then $p^*_{d+1} = 0$. Moreover for each $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \neq d+1$, $\mathbf{p} \in W$ (and thus, $Color_C[\psi(\mathbf{p})] = Color_{C'}[\mathbf{p}]$).*

PROOF. If $p^*_{d+1} > 0$, then $K_{\mathbf{p}^*}$ does not contain color $d+1$, and $p^*_{d+1} = 0$. For this case, p_{d+1} must be 1, since $\mathrm{Color}_{C'}[\mathbf{q}] = d+1$ for all $\mathbf{q} \in A^{d+1}_{\mathbf{r}'}$ with $q_{d+1} = 0$. Because $p_t \in \{a+2, a+3, a+4\}$, we have $\mathbf{p} \in W$. □

STATEMENT 3. *If $p^*_{d+1} = 4b$, then $0 \leq p^*_t \leq a+1$. Moreover, for every point $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \neq d+1$, $Color_C[\psi(\mathbf{p}[p_t, 4b+1])] = Color_{C'}[\mathbf{p}]$.*

PROOF. If $p^*_t > a+1$, then $K_{\mathbf{p}^*}$ does not contain color $d+1$. So $0 \leq p^*_t \leq a+1$. Similar to the proof of Statement 1, we can prove the second part for the case when $0 \leq p_t \leq a+1$.

When $p_t = a+2$, both $\mathbf{p}$ and $\mathbf{p}[p_t, 4b+1]$ are in W, and we have $\psi(\mathbf{p}) = \psi(\mathbf{p}[p_t, 4b+1])$. Thus, $\mathrm{Color}_C[\psi(\mathbf{p}[p_t, 4b+1])] = \mathrm{Color}_C[\psi(\mathbf{p})] = \mathrm{Color}_{C'}[\mathbf{p}]$. □

We can similarly prove the following statements.

STATEMENT 4. *If $p^*_{d+1} = 4i+1$ or $4i+2$ for some $0 \leq i \leq b-1$, then $p^*_t = 1$. Moreover, for each $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \neq d+1$, $Color_C[\psi(\mathbf{p}[2, p_{d+1}])] = Color_{C'}[\mathbf{p}]$.*

STATEMENT 5. *If $p^*_{d+1} = 4i$ for some $1 \leq i \leq b-1$, then $1 \leq p^*_t \leq a+1$. In addition, for every $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \neq d+1$, if $2 \leq p_t \leq a+1$, then $Color_C[\psi(\mathbf{p}[p_t, 4i+1])] = Color_{C'}[\mathbf{p}]$; and if $p_t = 1$, then $Color_C[\psi(\mathbf{p}[2, 4i+1])] = Color_{C'}[\mathbf{p}]$.*

STATEMENT 6. *If $p^*_{d+1} = 4i - 1$ for some $1 \le i \le b$, then $1 \le p^*_t \le a+1$. In addition, for every $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \ne d+1$, if $2 \le p_t \le a+1$, then $Color_C[\psi(\mathbf{p}[p_t, 4i-1])] = Color_{C'}[\mathbf{p}]$; and if $p_t = 1$, then $Color_C[\psi(\mathbf{p}[2, 4i-1])] = Color_{C'}[\mathbf{p}]$.*

STATEMENT 7. *If $p^*_{d+1} = 0$, then $1 \le p^*_t \le a+3$. In addition, for every point $\mathbf{p} \in P'$ such that $Color_{C'}[\mathbf{p}] \ne d+1$, if $2 \le p^*_t \le a+3$, then $\mathbf{p} \in W$ (and thus, $Color_C[\psi(\mathbf{p})] = Color_{C'}[\mathbf{p}]$); if $p^*_t = 1$, then $Color_C[\psi(\mathbf{p}[2,1])] = Color_{C'}[\mathbf{p}]$.*

In addition,

STATEMENT 8. $p^*_{d+1} \ne 4b+1$.

PROOF. If $p^*_{d+1} = 4b+1$, then $K_{\mathbf{p}^*}$ does not contain color $d+1$. □

Suppose that P' is a panchromatic simplex of T', and $K_{\mathbf{p}^*}$ is the hypercube containing P'. Then, P' and $\mathbf{p}^*$ must satisfy the conditions of one of the statements above. By that statement, we can transform every point $\mathbf{p} \in P'$, (aside from the one that has color $d+1$) back to a point $\mathbf{q}$ in $A^d_{\mathbf{r}}$ to obtain a set P from P'. Since P is accommodated, it is a panchromatic simplex of C. Thus, with all the statements above, we specify an efficient algorithm to compute a panchromatic simplex P of T given a panchromatic simplex P' of T'.

9.2. **PPAD**-COMPLETENESS OF PROBLEM BROUWER^f. We are now ready to prove the main result of this section.

PROOF OF THEOREM 6.6. We reduce BROUWER^{f_2} to BROUWER^f. Because the former is known to be complete in **PPAD** [Chen and Deng 2006], the latter is also complete in **PPAD**. Suppose $(C, 0^{2n})$ is an input instance of BROUWER^{f_2} (here, we assume n is large enough so that $3 \le f(8n) \le 4n$). Let

$$3 \le l = f(8n) \le 4n\,, \quad m = \left\lceil \frac{8n}{l} \right\rceil \ge 2, \quad \text{and} \quad m' = \left\lceil \frac{n}{l-2} \right\rceil.$$

We need to construct a coloring triple $(C', m, \mathbf{r}')$ (which can also be viewed as an input instance $(C', 0^{8n})$ of BROUWER^f) where $\mathbf{r}' \in \mathbb{Z}^m$ and $r'_i = 2^l$ for all $i \in [m]$. Every panchromatic simplex of C' can be used to find a panchromatic simplex of C efficiently.

We first consider the case when $l \ge n$. As $m \ge 2$, the hypergrid $\{0, 1, \ldots, 2^n - 1\}^2$ of C is contained in $A^m_{\mathbf{r}'}$. Therefore, we can build $(C', m, \mathbf{r}')$ by iteratively applying $\mathbf{L}^1$ and $\mathbf{L}^2$ to $(C, 2, (2^n, 2^n))$ with appropriate parameters. It follows from Properties **A** of Lemma 9.1 and 9.2 that $(C', m, \mathbf{r}')$ can be built in polynomial time (more exactly, poly$(n, \text{Size}[C])$). On the other hand, given a panchromatic simplex of C', one can use Properties **B** of Lemma 9.1 and 9.2 to recover a panchromatic simplex of C efficiently.

In the rest of the proof, we assume $l < n$. For this case, we iteratively build a sequence of coloring triples

$$\mathcal{T} = \left\{ T^0, T^1, \ldots, T^{w-1}, T^w \right\}, \quad \text{for some } w = O(m),$$

starting with $T^0 = (C, 2, (2^n, 2^n))$ and ending with $T^w = (C^w, m, \mathbf{r}^w)$, where $\mathbf{r}^w \in \mathbb{Z}^m$ and $r^w_i = 2^l$, for all $i \in [m]$. At the tth iteration, we apply either $\mathbf{L}^1$, $\mathbf{L}^2$ or $\mathbf{L}^3$ with properly chosen parameters to build T^{t+1} from T^t.

The Construction of $T^{3m'-14}$ from T^1

for t from 0 to $m'-6$ **do**

By the inductive hypothesis (18), $T^{3t+1} = (C^{3t+1}, d^{3t+1}, \mathbf{r}^{3t+1})$ satisfies $d^{3t+1} = t+2$, $r_1^{3t+1} = 2^{(m'-t)(l-2)}$, $r_2^{3t+1} = 2^n$ and $r_i^{3t+1} = 2^l$ for all $3 \le i \le t+2$

let $u = (2^{(m'-t-1)(l-2)} - 5)(2^{l-1} - 1) + 5$

$T^{3t+2} = \mathbf{L}^1(T^{3t+1}, 1, u)$

$T^{3t+3} = \mathbf{L}^3(T^{3t+2}, 1, 2^{(m'-t-1)(l-2)} - 5, 2^{l-2} - 1)$

$T^{3t+4} = \mathbf{L}^1(T^{3t+3}, t+3, 2^l)$

FIG. 9. The Construction of $T^{3m'-14}$ from T^1.

We further assume that $m' \ge 5$. The special case when $m' = 2, 3$ or 4 can be proved easily using the procedure in Figure 10. Below we give details of the construction.

In the first step, we call $\mathbf{L}^1(T^0, 1, 2^{m'(l-2)})$ to get $T^1 = (C^1, 2, (2^{m'(l-2)}, 2^n))$. This step is possible because $m'(l-2) \ge n$. We then invoke the procedure in Figure 9. We inductively show that for all $0 \le t \le m'-5$, $T^{3t+1} = (C^{3t+1}, d^{3t+1}, \mathbf{r}^{3t+1})$ satisfies

$$d^{3t+1} = t+2,\ r_1^{3t+1} = 2^{(m'-t)(l-2)},\ r_2^{3t+1} = 2^n,\ r_i^{3t+1} = 2^l,\ \ \forall i : 3 \le i \le t+2. \tag{18}$$

So in each for-loop, the first component of $\mathbf{r}$ decreases by a factor of 2^{l-2}, while the dimension of the space increases by 1.

The basis when $t = 0$ is trivial. Assume (18) is true for $0 \le t < m'-5$. We prove it for $t+1$. T^{3t+4} is constructed from T^{3t+1} in Figure 9. We only need to verify the following inequality:

$$r_1^{3t+1} \le u = \left(2^{(m'-t-1)(l-2)} - 5\right)(2^{l-1} - 1) + 5, \tag{19}$$

since otherwise, the first call to $\mathbf{L}^1$ is illegal. (19) follows from the inductive hypothesis (18) that $r_1^{3t+1} = 2^{(m'-t)(l-2)}$, and the assumption that $t < m'-5$ and $l \ge 3$. By induction, we know (18) is true for all $0 \le t \le m'-5$.

So, after running the for-loop in Figure 9 for $(m'-5)$ times, we get a coloring triple $T^{3m'-14} = (C^{3m'-14}, d^{3m'-14}, \mathbf{r}^{3m'-14})$ that satisfies[10]

$$\begin{aligned} &d^{3m'-14} = m'-3\,,\ r_1^{3m'-14} = 2^{5(l-2)}\,,\\ &\quad r_2^{3m'-14} = 2^n \ \text{ and } \ r_i^{3m'-14} = 2^l\,, \qquad \forall i : 3 \le i \le m'-3. \end{aligned}$$

Next, we call the procedure described in Figure 10. It is easy to check that the while-loop must terminate in at most four iterations (because we start with

$$r_1^{3m'-14} = 2^{5(l-2)},$$

[10] Here we use the superscripts of C, d, r_i to denote the index of the iterative step. It is not an exponent!

The Construction of $T^{w'}$ from $T^{3m'-14}$

let $t = 0$
while $T^{3(m'+t)-14} = (C^{3(m'+t)-14}, m'+t-3, \mathbf{r}^{3(m'+t)-14})$ satisfies $r_1^{3(m'+t)-14} > 2^l$ **do**
  let $k = \lceil (r_1^{3(m'+t)-14} - 5)/(2^{l-1}-1) \rceil + 5$
  $T^{3(m'+t)-13} = \mathbf{L}^1 (T^{3(m'+t)-14}, 1, (k-5)(2^{l-1}-1)+5)$
  $T^{3(m'+t)-12} = \mathbf{L}^3 (T^{3(m'+t)-13}, 1, k-5, 2^{l-2}-1)$
  $T^{3(m'+t)-11} = \mathbf{L}^1 (T^{3(m'+t)-12}, m'+t-2, 2^l)$, set $t = t+1$
let $w' = 3(m'+t) - 13$ and $T^{w'} = \mathbf{L}^1(T^{3(m'+t)-14}, 1, 2^l)$

FIG. 10. The Construction of $T^{w'}$ from $T^{3m'-14}$.

and in each while-loop, it decreases by a factor of almost 2^{l-1}). At the end, the procedure returns a coloring triple $T^{w'} = (C^{w'}, d^{w'}, \mathbf{r}^{w'})$ that satisfies

$$w' \leq 3m' - 1\,,\; d^{w'} \leq m' + 1\,,\; r_1^{w'} = 2^l\,,\; r_2^{w'} = 2^n,\; r_i^{w'} = 2^l\,,\; \forall i : 3 \leq i \leq d^{w'}.$$

We note that the second coordinate is ignored in the above procedure; thus, by symmetry, we may repeat the whole process above on the second coordinate, and obtain a coloring triple $T^{w''} = (C^{w''}, d^{w''}, \mathbf{r}^{w''})$ that satisfies

$$w'' \leq 6m' - 2\,,\; d^{w''} \leq 2m' \text{ and } r_i^{w''} = 2^l\,,\; \forall i : 1 \leq i \leq d^{w''}.$$

By the definition of m', m and the assumption that $l < n$,

$$d^{w''} \leq 2m' \leq 2\left(\frac{n}{l-2} + 1\right) \leq 2\left(\frac{n}{l/3}\right) + 2 = \frac{6n}{l} + 2 < \frac{8n}{l} \leq m.$$

Finally, by applying $\mathbf{L}^2$ on coloring triple $T^{w''}$ for $m - d^{w''}$ times with parameter $u = 2^l$, we obtain $T^w = (C^w, m, \mathbf{r}^w)$ with $\mathbf{r}^w \in \mathbb{Z}^m$ and $r_i^w = 2^l$ for all $i \in [m]$. It then follows from the construction that $w = O(m)$.

To see why this sequence $\mathcal{T}$ gives a reduction from BROUWER^{f_2} to BROUWER^f, let $T^i = (C^i, d^i, \mathbf{r}^i)$ (again the superscripts of C, d and $\mathbf{r}$ denote the index of the iteration). As sequence $\{\text{Size}\left[\mathbf{r}^i\right]\}_{0 \leq i \leq w}$ is nondecreasing and $w = O(m) = O(n)$, by Property **A** of Lemma 9.1, 9.2, 9.3, there exists a polynomial $g(n)$ such that

$$\begin{aligned} \text{Size}\left[C^w\right] &\leq \text{Size}\left[C\right] + w \cdot O\big(g\left(\text{Size}\left[\mathbf{r}^w\right]\right)\big) \\ &= \text{Size}\left[C\right] + w \cdot O\big(g(lm)\big) = \text{poly}(n, \text{Size}\left[C\right]). \end{aligned}$$

By these Properties **A** again, we can construct the whole sequence $\mathcal{T}$ and in particular, coloring triple $T^w = (C^w, m, \mathbf{r}^w)$, in time poly$(n, \text{Size}\left[C\right])$.

Pair $(C^w, 0^{8n})$ is an input instance of problem BROUWER^f. Given any panchromatic simplex P of $(C^w, 0^{8n})$, we can use the algorithms in Property **B** of Lemma 9.1, 9.2, and 9.3 to compute a sequence of panchromatic simplices $P^w = P, P^{w-1}, \ldots, P^0$ iteratively in polynomial time, where P^t is a panchromatic simplex of T^t and is computed from the panchromatic simplex P^{t+1} of T^{t+1}. In the end, we obtain P^0, which is a panchromatic simplex of $(C, 0^{2n})$.

As a result, BROUWER^f is **PPAD**-complete, and Theorem 6.6 is proved. $\square$

10. *Discussions*

10.1. EXTENSIONS. As 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. So far, our complexity results on the approximation of Nash equilibria have been extended to Arrow-Debreu equilibria [Huang and Teng 2007]. They can also be naturally extended to both r-player games [Nash 1951] and r-graphical games [Kearns et al. 2001], for every fixed $r \geq 3$. Since the announcement of our work, it has been shown that Nash equilibria are **PPAD**-hard to approximate in fully polynomial time even for bimatrix games with some special payoff structures, such as bimatrix games in which all payoff entries are either 0 or 1 [Chen et al. 2007], or in which most of the payoff entries are 0. In the latter case, we can strengthen our gadgets to prove the following theorem:

THEOREM 10.1 (SPARSE BIMATRIX). *Nash equilibria remain* **PPAD**-*hard to approximate in fully polynomial time for sparse bimatrix games in which each row and column of the two payoff matrices contains at most* 10 *nonzero entries.*

The reduction needed in proving this theorem is similar to the one used in proving Theorem 6.1. The key difference is that we first reduce BROUWER^{f_1} to a sparse generalized circuit, where a generalized circuit is *sparse* if each node is used by at most two gates as their input nodes. We then refine our gadget games for G_ζ, $G_\wedge$ and $G_\vee$, to guarantee that the resulting bimatrix game is sparse. Details of the proof can be found in Chen et al. [2006].

Our approach for studying the approximation complexity of Nash equilibra has recently been applied to several network games. For example, Poplawski et al. [2008] proved that it is **PPAD**-hard to compute an approximate equilibrium in a fractional BGP game [Haxell and Wilfong 2008] in fully polynomial time. They also proved a similar statement about a network connection game introduced in Laoutaris et al. [2008].

10.2. OPEN QUESTIONS AND CONJECTURES. There remains a complexity gap in the approximation of two-player Nash equilibria: Lipton et al. [2004] show that an ϵ-approximate Nash equilibrium can be computed in $n^{O(\log n/\epsilon^2)}$-time, while this article shows that, for ϵ of order $1/\text{poly}(n)$, no algorithm can find an ϵ-approximate Nash equilibrium in $\text{poly}(n, 1/\epsilon)$ time, unless **PPAD** is contained in **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 search problems in **PPAD** would be solvable in $n^{O(\log n)}$-time.

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

CONJECTURE 1 (PTAS FOR BIMATRIX). *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 BIMATRIX). *There is an algorithm for* BIMATRIX *with smoothed complexity $O(n^{k+\sigma^{-c}})$, under perturbations with magnitude σ, for some constants c and k.*

Recently, for sufficiently large constant ϵ, polynomial-time algorithms are developed for the computation of an ϵ-approximate Nash equilibrium [Daskalakis et al. 2006, 2007; Kontogiannis et al. 2006; Bosse et al. 2007; Tsaknakis and Spirakis 2007]. Currently, the constant ϵ achieved by the best algorithm is 0.3393 (due to Tsaknakis and Spirakis [2007]). However, new techniques are needed to prove Conjecture 1 [Feder et al. 2007]. Lemma 3.2 implies that Conjecture 1 is true for ϵ-well-supported Nash equilibrium if and only if it is true for ϵ-approximate Nash equilibrium. For bimatrix games $(\mathbf{A}, \mathbf{B})$ such that $\text{rank}(\mathbf{A}+\mathbf{B})$ is a constant, Kannan and Theobald [2007] found a fully-polynomial-time algorithm to approximate Nash equilibria. For two-player *planar* win-lose games, Addario-Berry et al. [2007] gave a polynomial-time algorithm for computing a Nash equilibrium.

Etessami and Yannakakis [2007] studied the complexity of approximating Nash equilibria using a different approximation concept: a *strong* ϵ-approximate Nash equilibrium is a mixed strategy profile that is *geometrically close* (e.g., in $\|\cdot\|_\infty$) to an exact equilibrium. They introduced a new complexity class **FIXP**, and proved that the strong approximation of Nash equilibria in three-player games is **FIXP**-complete. It was also shown that the linear version of **FIXP** is exactly **PPAD**.

One might be able to prove a weaker version of Conjecture 2 by extending the analysis of Bárány et al. [2005] to show that there is an algorithm for BIMATRIX with smoothed complexity $n^{O(\log n/\sigma^2)}$. We also conjecture that Corollary 6.4 remains true without any complexity assumption on **PPAD**, namely, that it could be proved without assuming **PPAD** $\not\subset$ **RP**. A positive answer would extend the result of Savani and von Stengel [2004] to smoothed bimatrix games. Another interesting question is whether the average-case complexity of the Lemke–Howson algorithm is polynomial.

Of course, the fact that two-player Nash equilibria and Arrow–Debreu equilibria are **PPAD**-hard to compute in the smoothed model does not necessarily imply that game and market problems are hard to solve in practice. In addition to possible noise and imprecision in inputs, practical problems might have other special structure that makes equilibrium computation or approximation more tractable. The game and market problems and their hardness results might provide an opportunity and a family of concrete problems for discovering new input models that can help us rigorously evaluate the performance of practical equilibrium algorithms and heuristics.

Theorem 6.2 implies that for any $r > 2$, the computation of an r-player Nash equilibrium can be reduced in polynomial time to the computation of a two-player Nash equilibrium. However, the implied reduction is not very natural: The r-player Nash equilibrium problem is first reduced to END-OF-LINE, then to BROUWER, and then to BIMATRIX. It remains an interesting question to find a more direct reduction from r-player Nash equilibria to two-player Nash equilibria.

The following complexity question about Nash equilibria is due to Vijay Vazirani: Are the counting versions of all **PPAD**-complete problems as hard as the counting version of BIMATRIX? Gilboa and Zemel [1989] showed that deciding whether a bimatrix game has a unique Nash equilibrium is **NP**-hard. Their technique was extended in Conitzer and Sandholm [2003] to prove that counting the number of Nash equilibria is **#P**-hard. Because the reduction between search problems only requires a many-to-one map between solutions, the number of solutions is not necessarily preserved. More restricted reductions are needed to answer Vazirani's question.

Finally, even though the results in this article and the results of Daskalakis et al. [2006a], Chen and Deng [2005a], and Daskalakis and Papadimitriou [2005] provide strong evidence that equilibrium computation might not be solvable in polynomial time, very little is known about the hardness of **PPAD** [Johnson 2007]. On one hand, Megiddo [1988] proved that if BIMATRIX is **NP**-hard, then **NP** = **coNP**. On the other hand, there are oracles that separate **PPAD** from **P**, and various discrete fixed point problems such as the computational version of Sperner's Lemma, require an exponential number of functional evaluations in the query model, deterministic [Hirsch et al. 1989; Chen and Deng 2005b] or randomized [Chen and Teng 2007], and also in the quantum query model [Friedl et al. 2005; Chen et al. 2008]. It is desirable to find stronger evidences that **PPAD** is not contained in **P**. Does the existence of one-way functions imply that **PPAD** is not contained in **P**? Does "FACTORING is not in **P**" imply that **PPAD** is not contained in **P**? Characterizing the hardness of the **PPAD** class is a fascinating and challenging problem.

Appendix

A. *Perturbation and Probabilistic Approximation*

In this section, we prove Lemma 4.2. To help explain the probabilistic reduction from the approximation of bimatrix games to the solution of perturbed bimatrix games, we first define the notion of many-way polynomial reductions among **TFNP** problems.

Definition A.1 (*Many-Way Reduction*). Let $\mathcal{F}$ be a set of polynomial-time functions, and g be a polynomial-time function.

A search problem SEARCH^{R_1} $\in$ **TFNP** is *($\mathcal{F}$, g)-reducible* to SEARCH^{R_2} $\in$ **TFNP** if, for all $y \in \{0,1\}^*$, $(f(x), y) \in R_2$ implies $(x, g(y)) \in R_1$ for every input x of R_1 and for every function $f \in \mathcal{F}$.

PROOF OF LEMMA 4.2. We will only give a proof of the lemma under uniform perturbations. With a slightly more complex argument to handle the low probability case when the absolute value of the perturbation is large, we can similarly prove the lemma under Gaussian perturbations.

Suppose J is an algorithm with polynomial smoothed complexity for BIMATRIX. Let $T_J(\mathbf{A}, \mathbf{B})$ be the time complexity of J for solving the bimatrix game defined by $(\mathbf{A}, \mathbf{B})$. Let $N_\sigma()$ denote the uniform perturbation with magnitude σ. Then, there exists constants c, k_1 and k_2 such that for all $0 < \sigma < 1$,

$$\max_{\bar{\mathbf{A}}, \bar{\mathbf{B}} \in \mathbb{R}^{n\times n}_{[0,1]},} \mathrm{E}_{\mathbf{A}\leftarrow N_\sigma(\bar{\mathbf{A}}), \mathbf{B}\leftarrow N_\sigma(\bar{\mathbf{B}})} \left[T_J(\mathbf{A}, \mathbf{B})\right] \leq c \cdot n^{k_1} \sigma^{-k_2}.$$

For each pair of perturbation matrices $\mathbf{S}, \mathbf{T} \in \mathbb{R}^{n\times n}_{[-\sigma,\sigma]}$, we can define a function $f_{(\mathbf{S},\mathbf{T})}$ from $\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ to $\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ as

$$f_{(\mathbf{S},\mathbf{T})}((\bar{\mathbf{A}}, \bar{\mathbf{B}})) = (\bar{\mathbf{A}} + \mathbf{S}, \bar{\mathbf{B}} + \mathbf{T}).$$

Let $\mathcal{F}_\sigma$ be the set of all such functions, that is,

$$\mathcal{F}_\sigma = \left\{ f_{(\mathbf{S},\mathbf{T})} \,\middle|\, \mathbf{S}, \mathbf{T} \in \mathbb{R}^{n\times n}_{[-\sigma,\sigma]} \right\}.$$

Let g be the identity function from $\mathbb{R}^n \times \mathbb{R}^n$ to $\mathbb{R}^n \times \mathbb{R}^n$.

We now show that the problem of computing an ϵ-approximate Nash equilibrium is $(\mathcal{F}_{\epsilon/2}, g)$-reducible to the problem of finding a Nash equilibrium of perturbed instances. More specifically, we prove that for every bimatrix game $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$ and for every $f_{(\mathbf{S},\mathbf{T})} \in \mathcal{F}_{\epsilon/2}$, a Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $f_{(\mathbf{S},\mathbf{T})}((\bar{\mathbf{A}}, \bar{\mathbf{B}}))$ is an ϵ-approximate Nash equilibrium of $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$.

Let $\mathbf{A} = \bar{\mathbf{A}} + \mathbf{S}$ and $\mathbf{B} = \bar{\mathbf{B}} + \mathbf{T}$. Then,

$$|\mathbf{x}^T \mathbf{A}\mathbf{y} - \mathbf{x}^T \bar{\mathbf{A}}\mathbf{y}| = |\mathbf{x}^T \mathbf{S}\mathbf{y}| \leq \epsilon/2 \tag{20}$$

$$|\mathbf{x}^T \mathbf{B}\mathbf{y} - \mathbf{x}^T \bar{\mathbf{B}}\mathbf{y}| = |\mathbf{x}^T \mathbf{T}\mathbf{y}| \leq \epsilon/2. \tag{21}$$

Thus, for each Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $(\mathbf{A}, \mathbf{B})$, for any $(\mathbf{x}', \mathbf{y}')$,

$$(\mathbf{x}')^T \bar{\mathbf{A}}\mathbf{y} - \mathbf{x}^T \bar{\mathbf{A}}\mathbf{y} \leq \big((\mathbf{x}')^T \mathbf{A}\mathbf{y} - \mathbf{x}^T \mathbf{A}\mathbf{y}\big) + \epsilon \leq \epsilon.$$

Similarly, $\mathbf{x}^T \bar{\mathbf{B}}\mathbf{y}' - \mathbf{x}^T \bar{\mathbf{B}}\mathbf{y} \leq \epsilon$. Therefore, $(\mathbf{x}, \mathbf{y})$ is an ϵ-approximate Nash equilibrium of $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$.

Now given the algorithm J with polynomial smoothed time-complexity for BIMATRIX, we can apply the following randomized algorithm to find an ϵ-approximate Nash equilibrium of $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$:

Algorithm `NashApproximationByPerturbations` $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$

1. Randomly choose a pair of perturbation matrices $\mathbf{S}, \mathbf{T}$ of magnitude $\epsilon/2$ and set $\mathbf{A} = \bar{\mathbf{A}} + \mathbf{S}$ and $\mathbf{B} = \bar{\mathbf{B}} + \mathbf{T}$.
2. Apply algorithm J to find a Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $(\mathbf{A}, \mathbf{B})$.
3. Return $(\mathbf{x}, \mathbf{y})$.

The expected time complexity of the algorithm is bounded from above by the smoothed complexity of J, and hence is at most $2^{k_2} c \cdot n^{k_1} \epsilon^{-k_2}$, which is polynomial in n and $1/\epsilon$. $\square$

B. *Padding Generalized Circuits: Proof of Theorem 5.7*

Suppose $\mathcal{S} = (V, \mathcal{T})$ is a generalized circuit. Let $K = |V|$. First, $\mathcal{S}$ has a $1/K^3$-approximate solution because POLY^3-GCIRCUIT is reducible to POLY^{12}-BIMATRIX (see Lemma 6.8 and Section 7), and every two-player game has a Nash equilibrium. Thus, the theorem is true for $c = 3$.

For the case when $c > 3$, we reduce POLY^c-GCIRCUIT to POLY^3-GCIRCUIT as follows. Suppose $c = 2b + 1$, where $b > 1$. We build a new circuit $\mathcal{S}' = (V', \mathcal{T}')$ by inserting some dummy nodes into $\mathcal{S}$ as follows:

—$V \subset V'$, $|V'| = K^b > K$ and $|\mathcal{T}'| = |\mathcal{T}|$;

—For each gate $T = (G, v_1, v_2, v, \alpha) \in \mathcal{T}$, if $G \notin \{G_\zeta, G_{\times\zeta}\}$ (and thus, $\alpha = nil$), then $T \in \mathcal{T}'$; otherwise, gate $(G, v_1, v_2, v, K^{1-b}\alpha) \in \mathcal{T}'$.

Let $\mathbf{x}'$ be any $1/|V'|^3$-approximate solution of $\mathcal{S}'$ (note that $|V'|^3 = 1/K^{3b}$). We now construct a new assignment $\mathbf{x} : V \to \mathbb{R}$ by setting $\mathbf{x}[v] = K^{b-1}\mathbf{x}'[v]$ for every $v \in V$. It is easy to check that $\mathbf{x}$ is a $1/K^{2b+1}$-approximate solution to the original circuit $\mathcal{S}$. We then apply $1/K^{2b+1} = 1/K^c$.

C. *Padding Bimatrix Games: Proof of Lemma 6.9*

Let c be the constant such that POLY^c-BIMATRIX is known to be **PPAD**-complete. If $c < 2$, then finding an n^{-2}-approximate Nash equilibrium is harder, and thus is also complete in **PPAD**. With this, without loss of generality, we assume that $c \geq 2$. To prove the lemma, it suffices to show that for every constant c' such that $0 < c' < c$, POLY^c-BIMATRIX is polynomial-time reducible to POLY$^{c'}$-BIMATRIX.

Suppose $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is an $n \times n$ positively normalized bimatrix game. We transform it into a new $n \times n$ game $\mathcal{G}' = (\mathbf{A}', \mathbf{B}')$ as follows:

$$a'_{i,j} = a_{i,j} + \Big(1 - \max_{1\leq k\leq n} a_{k,j}\Big) \quad \text{and} \quad b'_{i,j} = b_{i,j} + \Big(1 - \max_{1\leq k\leq n} b_{i,k}\Big),$$
$$\forall i, j : 1 \leq i, j \leq n.$$

One can verify that any ϵ-approximate Nash equilibrium of $\mathcal{G}'$ is also an ϵ-approximate Nash equilibrium of $\mathcal{G}$. Besides, every column of $\mathbf{A}'$ and every row of $\mathbf{B}'$ has at least one entry with value 1.

Next, we construct an $n'' \times n''$ game $\mathcal{G}'' = (\mathbf{A}'', \mathbf{B}'')$ where $n'' = n^{\frac{2c}{c'}} > n$ as follows: $\mathbf{A}''$ and $\mathbf{B}''$ are both 2×2 block matrices with

$$\mathbf{A}''_{1,1} = \mathbf{A}',\ \mathbf{B}''_{1,1} = \mathbf{B}',\ \mathbf{A}''_{1,2} = \mathbf{B}''_{2,1} = \mathbf{1},\ \mathbf{A}''_{2,1} = \mathbf{A}''_{2,2} = \mathbf{B}''_{1,2} = \mathbf{B}''_{2,2} = \mathbf{0}.$$

Let $(\mathbf{x}'', \mathbf{y}'')$ be any $1/n''^{c'} = 1/n^{2c}$-approximate Nash equilibrium of $\mathcal{G}'' = (\mathbf{A}'', \mathbf{B}'')$. By the definition of ϵ-approximate Nash equilibria, one can show that

$$0 \leq \sum_{n<i\leq n''} x''_i,\ \sum_{n<i\leq n''} y''_i \leq n^{1-2c} \ll 1/2,$$

since we assumed that $c \geq 2$. Let $a = \sum_{1\leq i\leq n} x''_i$ and $b = \sum_{1\leq i\leq n} y''_i$. We construct a pair of mixed strategies $(\mathbf{x}', \mathbf{y}')$ of $\mathcal{G}'$ as follows: $x'_i = x''_i/a$ and $y'_i = y''_i/b$ for all $i \in [n]$. Since $a, b > 1/2$, one can show that $(\mathbf{x}', \mathbf{y}')$ is a $4/n^{2c}$-approximate Nash equilibrium of $\mathcal{G}'$, which is also a $1/n^c$-approximate Nash equilibrium of the original game $\mathcal{G}$.

D. *Gadget Games: Completing the Proof of Lemma 7.4*

PROOF FOR G_ζ GATES. From Figure 3, we have

$$\left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k-1}\right\rangle - \left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k}\right\rangle = \overline{\mathbf{x}}[v] - \alpha, \qquad \text{and}$$
$$\left\langle \mathbf{a}^{\mathcal{S}}_{2k-1}|\mathbf{y}\right\rangle - \left\langle \mathbf{a}^{\mathcal{S}}_{2k}|\mathbf{y}\right\rangle = \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big) - \overline{\mathbf{y}}[v].$$

If $\overline{\mathbf{x}}[v] > \alpha + \epsilon$, then from the first equation, we have $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$. But the second equation implies $\overline{\mathbf{x}}[v] = 0$, which contradicts our assumption that $\overline{\mathbf{x}}[v] > 0$.

If $\overline{\mathbf{x}}[v] < \alpha - \epsilon$, then from the first equation, we have $\overline{\mathbf{y}}[v] = 0$. But the second equation implies that $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v] \geq 1/K - \epsilon$, which contradicts the assumption that $\overline{\mathbf{x}}[v] < \alpha - \epsilon$ and $\alpha \leq 1/K$. $\square$

PROOF FOR $G_{\times\zeta}$ GATES. From (3), (4) and Figure 3, we have

$$\left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k-1}\right\rangle - \left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k}\right\rangle = \alpha\,\overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v], \qquad \text{and}$$
$$\left\langle \mathbf{a}^{\mathcal{S}}_{2k-1}|\mathbf{y}\right\rangle - \left\langle \mathbf{a}^{\mathcal{S}}_{2k}|\mathbf{y}\right\rangle = \overline{\mathbf{y}}[v] - \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big).$$

If $\overline{\mathbf{x}}[v] > \min(\alpha\,\overline{\mathbf{x}}[v_1], 1/K) + \epsilon$, then $\overline{\mathbf{x}}[v] > \alpha\,\overline{\mathbf{x}}[v_1] + \epsilon$ as $\overline{\mathbf{x}}[v] \le \overline{\mathbf{x}}_C[v] \le 1/K + \epsilon$. By the first equation, we have $\overline{\mathbf{y}}[v] = 0$ and the second one implies that $\overline{\mathbf{x}}[v] = 0$, which contradicts the assumption that $\overline{\mathbf{x}}[v] > \min(\alpha\,\overline{\mathbf{x}}[v_1], 1/K) + \epsilon > 0$.

If $\overline{\mathbf{x}}[v] < \min(\alpha\overline{\mathbf{x}}[v_1], 1/K) - \epsilon \le \alpha\overline{\mathbf{x}}[v_1] - \epsilon$, then the first equation shows that $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$ and thus, by the second equation, we have $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v] \ge 1/K - \epsilon$, which contradicts the assumption that $\overline{\mathbf{x}}[v] < \min(\alpha\overline{\mathbf{x}}[v_1], 1/K) - \epsilon \le 1/K - \epsilon$. □

PROOF FOR $G_=$ GATES. $G_=$ is a special case of $G_{\times\zeta}$, with parameter $\alpha = 1$. □

PROOF FOR G_- GATES. From (3), (4) and Figure 3, we have

$$\left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k-1}\right\rangle - \left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k}\right\rangle = \overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2] - \overline{\mathbf{x}}[v], \quad \text{and}$$
$$\left\langle \mathbf{a}^{\mathcal{S}}_{2k-1}|\mathbf{y}\right\rangle - \left\langle \mathbf{a}^{\mathcal{S}}_{2k}|\mathbf{y}\right\rangle = \overline{\mathbf{y}}[v] - \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big).$$

If $\overline{\mathbf{x}}[v] > \max(\overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2], 0) + \epsilon \ge \overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2] + \epsilon$, then the first equation shows that $\overline{\mathbf{y}}[v] = 0$. By the second equation, we have $\overline{\mathbf{x}}[v] = 0$ which contradicts the assumption that $\overline{\mathbf{x}}[v] > \max(\overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2], 0) + \epsilon > 0$.

If $\overline{\mathbf{x}}[v] < \min(\overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2], 1/K) - \epsilon \le \overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2] - \epsilon$, then by the first equation, we have $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$. By the second equation, we have $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v] \ge 1/K - \epsilon$, contradicting the assumption that $\overline{\mathbf{x}}[v] < \min(\overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2], 1/K) - \epsilon \le 1/K - \epsilon$. □

PROOF FOR $G_<$ GATES. From (3), (4) and Figure 3, we have

$$\left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k-1}\right\rangle - \left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k}\right\rangle = \overline{\mathbf{x}}[v_1] - \overline{\mathbf{x}}[v_2], \qquad \text{and}$$
$$\left\langle \mathbf{a}^{\mathcal{S}}_{2k-1}|\mathbf{y}\right\rangle - \left\langle \mathbf{a}^{\mathcal{S}}_{2k}|\mathbf{y}\right\rangle = \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big) - \overline{\mathbf{y}}[v].$$

If $\overline{\mathbf{x}}[v_1] < \overline{\mathbf{x}}[v_2] - \epsilon$, then $\overline{\mathbf{y}}[v] = 0$ according to the first equation. By the second equation, we have $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v] = 1/K \pm \epsilon$ and thus, $\overline{\mathbf{x}}[v] =^{\epsilon}_{B} 1$.

If $\overline{\mathbf{x}}[v_1] > \overline{\mathbf{x}}[v_2] + \epsilon$, then $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$ according to the first equation. By the second one, we have $\overline{\mathbf{x}}[v] = 0$ and thus, $\overline{\mathbf{x}}[v] =^{\epsilon}_{B} 0$. □

PROOF FOR $G_\vee$ GATES. From (3), (4) and Figure 3, we have

$$\left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k-1}\right\rangle - \left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k}\right\rangle = \overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] - 1/(2K), \text{ and}$$
$$\left\langle \mathbf{a}^{\mathcal{S}}_{2k-1}|\mathbf{y}\right\rangle - \left\langle \mathbf{a}^{\mathcal{S}}_{2k}|\mathbf{y}\right\rangle = \overline{\mathbf{y}}[v] - \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big).$$

If $\overline{\mathbf{x}}[v_1] =^{\epsilon}_{B} 1$ or $\overline{\mathbf{x}}[v_2] =^{\epsilon}_{B} 1$, then $\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] \ge 1/K - \epsilon$. By the first equation, we have $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$. By the second equation, we have $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v] = 1/K \pm \epsilon$ and thus, $\overline{\mathbf{x}}[v] =^{\epsilon}_{B} 1$.

If $\overline{\mathbf{x}}[v_1] =^{\epsilon}_{B} 0$ and $\overline{\mathbf{x}}[v_2] =^{\epsilon}_{B} 0$, then $\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] \le 2\epsilon$. From the first equation, $\overline{\mathbf{y}}[v] = 0$. Then, the second equation implies $\overline{\mathbf{x}}[v] =^{\epsilon}_{B} 0$. □

PROOF FOR $G_\wedge$ GATES. From (3), (4) and Figure 3, we have

$$\left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k-1}\right\rangle - \left\langle \mathbf{x}|\mathbf{b}^{\mathcal{S}}_{2k}\right\rangle = \overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] - 3/(2K), \text{ and}$$
$$\left\langle \mathbf{a}^{\mathcal{S}}_{2k-1}|\mathbf{y}\right\rangle - \left\langle \mathbf{a}^{\mathcal{S}}_{2k}|\mathbf{y}\right\rangle = \overline{\mathbf{y}}[v] - \big(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v]\big).$$

If $\overline{\mathbf{x}}[v_1] =^{\epsilon}_{B} 0$ or $\overline{\mathbf{x}}[v_2] =^{\epsilon}_{B} 0$, then $\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] \le 1/K + 2\epsilon$. From the first equation, we have $\overline{\mathbf{y}}[v] = 0$. By the second equation, we have $\overline{\mathbf{x}}[v] = 0$ and thus, $\overline{\mathbf{x}}[v] =^{\epsilon}_{B} 0$.

If $\overline{\mathbf{x}}[v_1] =_B^\epsilon 1$ and $\overline{\mathbf{x}}[v_2] =_B^\epsilon 1$, then $\overline{\mathbf{x}}[v_1] + \overline{\mathbf{x}}[v_2] \geq 2/K - 2\epsilon$. The first equation shows $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$. By the second equation, $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v] = 1/K \pm \epsilon$ and thus, $\overline{\mathbf{x}}[v] =_B^\epsilon 1$. $\square$

PROOF FOR $G_\neg$ GATES. From (3), (4) and Figure 3, we have

$$\left\langle \mathbf{x} | \mathbf{b}_{2k-1}^{\mathcal{S}} \right\rangle - \left\langle \mathbf{x} | \mathbf{b}_{2k}^{\mathcal{S}} \right\rangle = \overline{\mathbf{x}}[v_1] - \left(\overline{\mathbf{x}}_C[v_1] - \overline{\mathbf{x}}[v_1] \right), \text{ and}$$
$$\left\langle \mathbf{a}_{2k-1}^{\mathcal{S}} | \mathbf{y} \right\rangle - \left\langle \mathbf{a}_{2k}^{\mathcal{S}} | \mathbf{y} \right\rangle = \left(\overline{\mathbf{y}}_C[v] - \overline{\mathbf{y}}[v] \right) - \overline{\mathbf{y}}[v].$$

If $\overline{\mathbf{x}}[v_1] =_B^\epsilon 1$, then by the first equation, $\overline{\mathbf{y}}[v] = \overline{\mathbf{y}}_C[v]$. Then, by the second equation, we have $\overline{\mathbf{x}}[v] = 0$.

If $\overline{\mathbf{x}}[v_1] =_B^\epsilon 0$, then the first equation shows that $\overline{\mathbf{y}}[v] = 0$. By the second equation, we have $\overline{\mathbf{x}}[v] = \overline{\mathbf{x}}_C[v]$ and thus, $\overline{\mathbf{x}}[v] =_B^\epsilon 1$. $\square$

ACKNOWLEDGMENTS. We would like to thank the three referees for their great suggestions. We would like to thank Kyle Burke, Costis Daskalakis, Li-Sha Huang, Jon Kelner, Christos Papadimitriou, Laura Poplawski, Rajmohan Rajaraman, Dan Spielman, Ravi Sundaram, Paul Valiant, and Vijay Vazirani for helpful comments and suggestions. We would like to thank John Reif, Nicole Immorlica, Steve Vavasis, Christos Papadimitriou, Mohammad Mahdian, Ding-Zhu Du, Santosh Vempala, Aram Harrow, Adam Kalai, Imre Bárány, Adrian Vetta, and Jonathan Kelner for asking about the smoothed complexity of the Lemke-Howson algorithm.

REFERENCES

ABBOTT, T., KANE, D., AND VALIANT, P. 2005. On the complexity of two-player win-lose games. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 113–122.

ADDARIO-BERRY, L., OLVER, N., AND VETTA, A. 2007. A polynomial time algorithm for finding Nash equilibria in planar win-lose games. *J. Graph Algor. Applica. 11,* 1, 309–319.

ARROW, K., AND DEBREU, G. 1954. Existence of an equilibrium for a competitive economy. *Econometrica 22*, 265–290.

BÁRÁNY, I., VEMPALA, S., AND VETTA, A. 2005. Nash equilibria in random games. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 123–131.

BLUM, L., SHUB, M., AND SMALE, S. 1989. On a theory of computation over the real numbers; NP completeness, recursive functions and universal machines. *Bull. AMS 21*, 1 (July), 1–46.

BORGWARDT, K.-H. 1982. The average number of steps required by the simplex method is polynomial. *Z. Oper. Res. 26*, 157–177.

BOSSE, H., BYRKA, J., AND MARKAKIS, E. 2007. New algorithms for approximate Nash equilibria in bimatrix games. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*. 17–29.

BROUWER, L. 1910. Über Abbildung von Mannigfaltigkeiten. *Math. Ann. 71*, 97–115.

CHEN, X., AND DENG, X. 2005a. 3-Nash is PPAD-complete. In *Electronic Colloquium in Computational Complexity*. TR05–134.

CHEN, X., AND DENG, X. 2005b. On algorithms for discrete and approximate Brouwer fixed points. In *STOC '05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, New York, 323–330.

CHEN, X., AND DENG, X. 2006. On the complexity of 2D discrete fixed point problem. In *ICALP '06: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*. 489–500.

CHEN, X., DENG, X., AND TENG, S.-H. 2006a. Sparse games are hard. In *Proceedings of the 2nd Workshop on Internet and Network Economics*. 262–273.

CHEN, X., HUANG, L.-S., AND TENG, S.-H. 2006b. Market equilibria with hybrid linear-Leontief utilities. In *Proceedings of the 2nd Workshop on Internet and Network Economics*. 274–285.

CHEN, X., SUN, X., AND TENG, S.-H. 2008. Quantum separation of local search and fixed point computation. In *Proceedings of the 14th Annual International Computing and Combinatorics Conference*. 169–178.

CHEN, X., AND TENG, S.-H. 2007. Paths beyond local search: A tight bound for randomized fixed-point computation. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 124–134.

CHEN, X., TENG, S.-H., AND VALIANT, P. 2007. The approximation complexity of win-lose games. In *SODA '07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 159–168.

CODENOTTI, B., SABERI, A., VARADARAJAN, K., AND YE, Y. 2006. Leontief economies encode nonzero sum two-player games. In *SODA '06: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 659–667.

CONDON, A., EDELSBRUNNER, H., EMERSON, E., FORTNOW, L., HABER, S., KARP, R., LEIVANT, D., LIPTON, R., LYNCH, N., PARBERRY, I., PAPADIMITRIOU, C., RABIN, M., ROSENBERG, A., ROYER, J., SAVAGE, J., SELMAN, A., SMITH, C., TARDOS, E., AND VITTER, J. 1999. Challenges for theory of computing: Report of an NSF-sponsored workshop on research in theoretical computer science. *SIGACT News 30*, 2, 62–76.

CONITZER, V., AND SANDHOLM, T. 2003. Complexity results about Nash equilibria. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. ACM, New York, 765–771.

DASKALAKIS, C., GOLDBERG, P., AND PAPADIMITRIOU, C. 2006a. The complexity of computing a Nash equilibrium. In *STOC '06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*. ACM, New York, 71–78.

DASKALAKIS, C., GOLDBERG, P., AND PAPADIMITRIOU, C. H. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* To appear.

DASKALAKIS, C., MEHTA, A., AND PAPADIMITRIOU, C. H. 2006b. A note on approximate Nash equilibria. In *Proceedings of the 2nd Workshop on Internet and Network Economics*. 297–306.

DASKALAKIS, C., MEHTA, A., AND PAPADIMITRIOU, C. 2007. Progress in approximate Nash equilibria. In *Proceedings of the 8th ACM Conference on Electronic Commerce*. ACM, New York, 355–358.

DASKALAKIS, C., AND PAPADIMITRIOU, C. 2005. Three-player games are hard. In *Electronic Colloquium in Computational Complexity*. TR05–139.

DENG, X., PAPADIMITRIOU, C., AND SAFRA, S. 2003. On the complexity of price equilibria. *J. Comput. Syst. Sci. 67*, 2, 311–324.

EDELSBRUNNER, H. 2006. *Geometry and Topology for Mesh Generation (Cambridge Monographs on Applied and Computational Mathematics)*. Cambridge University Press, New York.

ETESSAMI, K., AND YANNAKAKIS, M. 2007. On the complexity of Nash equilibria and other fixed points. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 113–123.

FEDER, T., NAZERZADEH, H., AND SABERI, A. 2007. Approximating Nash equilibria using small-support strategies. In *Proceedings of the 8th ACM Conference on Electronic Commerce*. ACM, New York, 352–354.

FRIEDL, K., IVANYOS, G., SANTHA, M., AND VERHOEVEN, F. 2005. On the black-box complexity of Sperner's lemma. In *Proceedings of the 15th International Symposium on Fundamentals of Computation Theory*. 245–257.

FRIEDL, K., IVANYOS, G., SANTHA, M., AND VERHOEVEN, F. 2006. Locally 2-dimensional Sperner problems complete for the polynomial parity argument classes. In *Proceedings of the 6th Conference on Algorithms and Complexity*. 380–391.

GILBOA, I., AND ZEMEL, E. 1989. Nash and correlated equilibria: Some complexity considerations. *Games Econ. Behav. 1*, 1, 80–93.

GOLDBERG, P., AND PAPADIMITRIOU, C. 2006. Reducibility among equilibrium problems. In *STOC '06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*. ACM, New York, 61–70.

HAXELL, P. E., AND WILFONG, G. T. 2008. A fractional model of the border gateway protocol (BPG). In *SODA '08: Proceedings of the 19th annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 193–199.

HIRSCH, M., PAPADIMITRIOU, C., AND VAVASIS, S. 1989. Exponential lower bounds for finding Brouwer fixed points. *J. Complex. 5*, 379–416.

HOLT, C., AND ROTH, A. 2004. The Nash equilibrium: A perspective. *Proc. Nat. Acad. Sci. USA 101*, 12, 3999–4002.

HUANG, L.-S., AND TENG, S.-H. 2007. On the approximation and smoothed complexity of Leontief market equilibria. In *Proceedings of the 1st International Frontiers of Algorithmics Workshop*. 96–107.

JOHNSON, D. 2007. The NP-completeness column: Finding needles in haystacks. *ACM Trans. Algor. 3*, 2, 24.

KAKUTANI, S. 1941. A generalization of Brouwer's fixed point theorem. *Duke Math. J. 8*, 457–459.

KANNAN, R., AND THEOBALD, T. 2007. Games of fixed rank: A hierarchy of bimatrix games. In *SODA '07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 1124–1132.

KARMARKAR, N. 1984. A new polynomial-time algorithm for linear programming. *Combinatorica 4*, 373–395.

KEARNS, M., LITTMAN, M., AND SINGH, S. 2001. Graphical models for game theory. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. 253–260.

KHACHIAN, L. 1979. A polynomial algorithm in linear programming. *Doklady Akademia Nauk SSSR 244*, 1093–1096, (English translation in *Soviet Math. Dokl. 20, 191–194*).

KLEE, V., AND MINTY, G. 1972. How good is the simplex algorithm? In *Inequalities – III*, O. Shisha, Ed. Academic Press, Orlando, FL, 159–175.

KO, K. 1991. *Complexity Theory of Real Functions*. Birkhauser Boston Inc., Cambridge, MA.

KONTOGIANNIS, S., PANAGOPOULOU, P., AND SPIRAKIS, P. 2006. Polynomial algorithms for approximating Nash equilibria of bimatrix games. In *Proceedings of the 2nd Workshop on Internet and Network Economics*. 286–296.

LAOUTARIS, N., POPLAWSKI, L. J., RAJARAMAN, R., SUNDARAM, R., AND TENG, S.-H. 2008. Bounded budget connection (BBC) games or how to make friends and influence people, on a budget. In *PODC '08: Proceedings of the 27th ACM Symposium on Principles of Distributed Computing*. ACM, New York, 165–174.

LEMKE, C. 1965. Bimatrix equilibrium points and mathematical programming. *Manag. Sci. 11*, 681–689.

LEMKE, C., AND HOWSON, JR., J. 1964. Equilibrium points of bimatrix games. *J. Soc. Indust. Appl. Math. 12*, 413–423.

LEONARD, R. 1994. Reading Cournot, reading Nash: The creation and stabilisation of the Nash equilibrium. *Economic Journal 104*, 424, 492–511.

LIPTON, R., AND MARKAKIS, E. 2004. Nash equilibria via polynomial equations. In *Proceedings of the 6th Latin American Symposium on Theoretical Informatics*. 413–422.

LIPTON, R., MARKAKIS, E., AND MEHTA, A. 2004. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*. ACM, New York, 36–41.

MEGIDDO, N. 1988. A note on the complexity of P-matrix LCP and computing an equilibrium. Res. Rep. RJ6439 IBM Almaden Research Center, San Jose.

MEGIDDO, N., AND PAPADIMITRIOU, C. 1991. On total functions, existence theorems and computational complexity. *Theoretical Computer Science 81*, 317–324.

MORGENSTERN, O., AND VON NEUMANN, J. 1947. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ.

NASH, J. 1950. Equilibrium points in n-person games. *Proc. Nat. Acad. USA 36*, 1, 48–49.

NASH, J. 1951. Noncooperative games. *Ann. Math. 54*, 289–295.

PAPADIMITRIOU, C. 1991. On inefficient proofs of existence and complexity classes. In *Proceedings of the 4th Czechoslovakian Symposium on Combinatorics*.

PAPADIMITRIOU, C. 1994. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 498–532.

PAPADIMITRIOU, C. 2001. Algorithms, games, and the internet. In *STOC '01: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*. ACM, New York, 749–753.

POPLAWSKI, L. J., RAJARAMAN, R., SUNDARAM, R., AND TENG, S.-H. 2008. Preference games and personalized equilibria, with applications to fractional BGP. In *arXiv:0812.0598*.

SANDHOLM, T. 2000. Issues in computational Vickrey auctions. *Int. J. Electron. Comm. 4*, 3, 107–129.

SAVANI, R., AND VON STENGEL, B. 2004. Exponentially many steps for finding a Nash equilibrium in a bimatrix game. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 258–267.

SCARF, H. 1967a. The approximation of fixed points of a continuous mapping. *SIAM J. Appl. Math. 15*, 997–1007.

SCARF, H. 1967b. On the computation of equilibrium prices. In *Ten Economic Studies in the Tradition of Irving Fisher*, W. Fellner, Ed. Wiley, New York.

SPERNER, E. 1928. Neuer Beweis fur die Invarianz der Dimensionszahl und des Gebietes. *Abhandlungen aus dem Mathematischen Seminar Universitat Hamburg 6*, 265–272.

SPIELMAN, D., AND TENG, S.-H. 2004. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM 51*, 3, 385–463.

SPIELMAN, D., AND TENG, S.-H. 2006. Smoothed analysis of algorithms and heuristics: Progress and open questions. In *Foundations of Computational Mathematics*, L. Pardo, A. Pinkus, E. Süli, and M. Todd, Eds. Cambridge University Press, Cambridge, MA, 274–342.

TSAKNAKIS, H., AND SPIRAKIS, P. 2007. An optimization approach for approximate Nash equilibria. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*. 42–56.

VON NEUMANN, J. 1928. Zur theorie der gesellschaftsspiele. *Math. Ann. 100*, 295–320.

WILSON, R. 1971. Computing equilibria of n-person games. *SIAM J. Appl. Math. 21*, 80–87.

YE, Y. 2005. Exchange market equilibria with Leontief's utility: Freedom of pricing leads to rationality. In *Proceedings of the 1st Workshop on Internet and Network Economics*. 14–23.

RECEIVED APRIL 2007; REVISED SEPTEMBER 2008 AND JANUARY 2009; ACCEPTED JANUARY 2009