Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 140 (2005)

Settling the Complexity of 2-Player Nash-Equilibrium

Xi Chen
Department of Computer Science
Tsinghua University
Beijing, P.R.China
xichen00@mails.tsinghua.edu.cn

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

Abstract

We prove that finding a solution of two player Nash Equilibrium is **PPAD**-complete.

1 Introduction

Almost sixty years ago Morgenstern and von Neumann [14] initiated the study of game theory with their applications to Economic behavior. A particularly interesting mathematical result is their proof of the existence of equilibrium in the 2-player zero-sum game model where one player's gain is the loss of the other. They exploit duality properties of polytopes, which also lead to Dantzig's linear programming method [6] for optimization problems, as well as Yao's principle [17] for finding algorithmic lower bounds. Nash proposed in the middle of the last century to study the more general multiple person game model, and proved that there exists a set of (mixed) strategies, now called Nash-equilibrium point, one for each player, such that no player can benefit if it changes its own strategy unilaterally. While 2-player zero-sum game has a polynomial-time algorithm since linear programming has one, as by Khachian's ellipsoid algorithm [12], the existence proof of Nash equilibrium relied on the Kakutani's fixed point theorem (a generalization of the Brouwer's fixed point theorem [1]) which does not admit any polynomial-time algorithm [3, 10]. Despite much effort on the important problem, no significant progress has been made on algorithms for the original Nash-equilibrium problem in the last half century, though both hardness results and polynomial-time algorithms have been derived for various modified versions.

An exciting breakthrough was announced a few weeks ago which stated that finding Nash equilibriums is indeed hard, by Daskalakis, Goldberg and Papadimitriou [7], for games with four players or more. An ϵ approximation version was proven to be complete in the **PPAD** (polynomial parity argument, directed version) class, introduced by Papadimitriou in his seminal work about fifteen years ago [15]. The work was improved to the 3-player case by Chen and

ISSN 1433-8092

Deng [4], Daskalakis and Papadimitriou [8], independently, and with different proofs. Those results leave the two player Nash-equilibrium the last opening problem in the long sequel of search for an efficient solution.

Finding a Nash-equilibrium in a game between two players could be easier for several reasons. First, the zero-sum version can be solved in polynomial time by linear programming. Secondly, it admits a polynomial size rational number solution [5] while games between three or more players may only have solutions all in irrational numbers. Finally, an important technique employed in the hardness proofs, that colors vertices of a graphical game, does not seem possible to work down to the case of two players.

In this work, we settle the problem with a **PPAD**-complete proof for the 2-player Nash-equilibrium problem (2-Nash). Our proof gets rid of the graphical game model, and derives a direct reduction from 3-Dimensional Brouwer to 2-Nash. We need to design new gadgets for various arithmetic and logic operations [7], but they all work well in the new setting. Noticeably, an exact equilibrium is an approximate equilibrium in the definition of r-Nash. Therefore, the problem of finding an exact Nash equilibrium in a game between two players (which is denoted by Nash in [16]) is **PPAD**-hard. On the other hand, it was known by Cottle and Danzig [5] that the two player case admits a polynomial size rational solution and problem Nash is in **PPAD** [16]. Our result immediately implies that Nash is complete in **PPAD**, which settles a long standing open problem.

The paper is arranged as follows: We review the necessary definitions in Section 2. In section 3, we summarize the reduction from 3-Dimensional Brouwer to 3-Graphical Nash in [7], in particular the types of gadgets required by the reduction. In Section 4, we present our new gadgets and prove the correctness of the reduction. We conclude in Section 5 with remarks and discussion.

2 Preliminaries

2.1 Games, Graphical Games and Nash Equilibriums

A game $\mathcal{G}$ between $r \geq 2$ players is composed of two parts. First, every player $p \in [r]$ where $[r] = \{0, 1, ... r\}$ has a set S_p of pure strategies. Second, for each $p \in [r]$ and $s \in S$ where

$$S = S_1 \times S_2 \times ... \times S_r,$$

we have u_s^p as the payoff or utility of player p. Here s is called a pure strategy profile of the game. For any p, we use S_{-p} to denote the set of all strategy profiles of players other than p. For any $j \in S_p$ and $s \in S_{-p}$, we use js to denote the pure strategy profile in S, which is combined by j and s. A mixed strategy x^p of player p is a probability distribution over S_p, that is, real numbers $x_j^p \geq 0$ for any $j \in S_p$ and $\sum_{j \in S_p} x_j^p = 1$. A profile of mixed strategies $\mathbf{p}$ of game $\mathcal{G}$ consists of r mixed strategies x^p, $p = 1, 2, ... r$. For any $p \in [r]$, x^p is a mixed strategy of player p. For any $p \in [r]$ and $s \in S_{-p}$, we define x_s as

$$x_s = \prod_{p' \in [r],\ p' \neq p} x^{p'}_{s_{p'}}$$

Now we give the definition of both accurate and approximate Nash equilibriums of a game. Intuitively, a Nash equilibrium is a profile of mixed strategies **p** such that no player can gain by unilaterally choosing a different mixed strategy, while other mixed strategies in the profile are kept fixed. The concept of approximate Nash-equilibrium here was first proposed by [16].

Definition 1. *A Nash equilibrium of $\mathcal{G}$ is a profile of mixed strategies $\boldsymbol{p} = \{x^p\}$ such that*

$$\sum_{s \in S_{-p}} u^p_{is} x_s > \sum_{s \in S_{-p}} u^p_{js} x_s \implies x^p_j = 0$$

for any $p \in [r]$ and $i, j \in S_p$.

Definition 2. *An ϵ-Nash equilibrium of $\mathcal{G}$ is a profile of mixed strategies $\boldsymbol{p} = \{x^p\}$ such that*

$$\sum_{s \in S_{-p}} u^p_{is} x_s > \sum_{s \in S_{-p}} u^p_{js} x_s + \epsilon \implies x^p_j = 0$$

for any $p \in [r]$ and $i, j \in S_p$.

A useful class of games are graphical games, which was first defined in [11] and then generalized by [9]. Players in a graphical game are vertices of an underlying directed graph G. A player u can affect the payoffs of player v only if $uv \in G$. While general games require exponential data for their descriptions, graphical games have succinct representations. More exactly, when the in-degree of the underlying graph G is bounded, the representation of a graphical game is polynomial in the number of players and strategies.

2.2 TFNP, PPAD and r-Nash

Let $R \subset \Sigma^* \times \Sigma^*$ be a polynomial-time computable, polynomially balanced relation (that is, there exists a polynomial p such that for any x and y satisfy $(x, y) \in R$, $|y| \leq p(|x|)$). The NP search problem Q_R specified by R is this: given input $x \in \Sigma^*$, return a $y \in \Sigma^*$ such that $(x, y) \in R$ if such a y exists, and return the string "no" otherwise. An NP search problem is said to be total if for every x, there exists a y such that $(x, y) \in R$. We use **TFNP** [13] to denote the class of total NP search problems.

Definition 3. *Given two problems $Q_{R_1}, Q_{R_2} \in$ **TFNP**, we say that Q_{R_1} is reducible to Q_{R_2} if there exists a pair of polynomial-time computable functions (f, g) such that, for every input x of R_1, if y satisfies $(f(x), y) \in R_2$, then $(x, g(y)) \in R_1$.*

One of the most interesting sub-classes of **TFNP** is **PPAD** which is the directed version of class **PPA**. The totality of problems in **PPAD** is guaranteed by the following trivial fact: in a directed graph, where the in-degree and out-degree of every vertex are no more than one, if there exists a source, there must be another source or sink. Many important problems were identified to be in **PPAD** [16], e.g. the search versions of Brouwer's fixed point theorem, Kakutani's fixed point theorem, Smith's theorem and Borsuk-Ulam theorem. r-Nash, that is, the problem of finding an approximate Nash equilibrium in a game between r players, also belongs to **PPAD** [16].

Definition 4. *The input of problem r-*Nash *is a pair $(\mathcal{G}, 0^k)$ where $\mathcal{G}$ is an r-player game in normal form, and the output is a $(1/2^k)$-Nash equilibrium of game $\mathcal{G}$. The input of* Nash *is a 2-player game $\mathcal{G}$ in normal form, and the output is an exact Nash equilibrium of $\mathcal{G}$.*

3 Review of the Reduction in [7]

In this section, we briefly review the reduction from problem 3-Dimensional Brouwer to 3-Graphical Nash in [7]. First, we define the search problem 3-Dimensional Brouwer.

Definition 5 (3-Dimensional Brouwer**).** *The input of the problem is a pair $(C, 0^n)$ where C is a circuit with $3n$ input bits and 6 output bits Δx^+, Δx^-, Δy^+, Δy^-, Δz^+ and Δz^-. It specifies a Brouwer function ϕ of a very special form. For any $0 \le i,j,k \le 2^n - 1$, we define a cubelet K_{ijk} in the unit cube $[0,1]^3$ as*

$$K_{ijk} = \left\{ (x,y,z) \;\middle|\; i2^{-n} \le x \le (i+1)2^{-n}, j2^{-n} \le y \le (j+1)2^{-n}, k2^{-n} \le z \le (k+1)2^{-n} \right\}$$

and use c_{ijk} to denote its center. Brouwer function ϕ is a function on the set of centers. For any c_{ijk}, $\phi(c_{ijk}) = c_{ijk} + \delta$ where δ is one of the four increment vectors δ_1, δ_2, δ_3, δ_4 below, and is specified by the 6 output bits of $C(i,j,k)$ as follows:

case 1: $\Delta x^+ = 1$ *and other five bits are* $0 \implies \delta = \delta_1 = (\alpha, 0, 0)$;

case 2: $\Delta y^+ = 1$ *and other five bits are* $0 \implies \delta = \delta_2 = (0, \alpha, 0)$;

case 3: $\Delta z^+ = 1$ *and other five bits are* $0 \implies \delta = \delta_3 = (0, 0, \alpha)$;

case 4: $\Delta x^- = \Delta y^- = \Delta z^- = 1$ *and other three bits are* $0 \implies \delta = \delta_4 = (-\alpha, -\alpha, -\alpha)$,

where $\alpha = 2^{-2n}$ is much smaller than the cubelet side. For any $0 \le i,j,k \le 2^n - 1$, the six output bits of $C(i,j,k)$ are guaranteed to be one of the four cases above, and C satisfies the following conditions on the boundary:

$$\phi(c_{0jk}) = c_{0jk} + \delta_1 \quad \phi(c_{i0k}) = c_{i0k} + \delta_2 \quad \phi(c_{ij0}) = c_{ij0} + \delta_3$$
$$\phi(c_{(2^n-1)jk}) = c_{(2^n-1)jk} + \delta_4 \quad \phi(c_{i(2^n-1)k}) = c_{i(2^n-1)k} + \delta_4 \quad \phi(c_{ij(2^n-1)}) = c_{ij(2^n-1)} + \delta_4$$

with conflicts resolved arbitrarily. A vertex of a cubelet is said to be panchromatic if, among the eight cubelets adjacent to it, there are four that have all four increments δ_1, δ_2, δ_3 and δ_4. The output of the problem is a panchromatic vertex of ϕ which is specified by $(C, 0^n)$.

Theorem 1 ([7]). *Search problem* 3-Dimensional Brouwer *is* **PPAD***-complete.*

In [7], a binary graphical game $\mathcal{GG}$ with degree 3 is constructed from $(C, 0^n)$. Given any 2^{-4n}-Nash equilibrium of $\mathcal{GG}$, a panchromatic vertex of $(C, 0^n)$ can be identified efficiently.

There are two kinds of vertices in $\mathcal{GG}$, arithmetic vertices and interior vertices. For any arithmetic vertex v, $\mathbf{p}[v]$ is a meaningful real number in any Nash equilibrium $\mathbf{p}$, where $\mathbf{p}[v]$ is the probability of v choosing strategy 1. Gadgets are designed to implement arithmetic and logic operations among arithmetic vertices, and interior vertices are used to mediate between arithmetic vertices, so that the latter ones obey the intended arithmetic relationship.

Totally 9 gadgets are necessary, i.e. G_ζ, $G_{\times\zeta}$, $G_=$, G_+, G_-, $G_<$, $G_\wedge$, $G_\vee$ and $G_\neg$. Every gadget contains both arithmetic vertices and interior vertices. Furthermore, arithmetic vertices in a gadget are classified as input vertices and output vertices. For example, a G_+ gadget contains 4 vertices v_1, v_2, v_3 and w where w is an interior vertex and others are arithmetic vertices. v_3 is the output vertex of G_+ and v_1, v_2 are both input vertices. A gadget only decide payoffs of its interior vertex and output vertex. For example, by saying adding a G_+ gadget, we actually setup the payoffs of v_3 and w, so that in any ϵ-Nash equilibrium of $\mathcal{GG}$, we have $\mathbf{p}[v_3] = \max(\mathbf{p}[v_1] + \mathbf{p}[v_2], 1) \pm \epsilon$. For any arithmetic vertex v, there exists exactly one gadget of which v is the output vertex, while it can be an input vertex of arbitrarily many gadgets.

The main idea in the construction of $\mathcal{GG}$ comes from the following observation:

> Let $p = (x, y, z)$ be a point in the unit cube. If the increment of function ϕ at p (interpolated from centers of the adjacent cubelets) is close enough to zero, then there must exist a panchromatic vertex of Brouwer function ϕ near point p.

There are three distinguished vertices v_x, v_y and v_z which encode a point p in the unit cube. After extracting the $3n$ bits of $\mathbf{p}[v_x]$, $\mathbf{p}[v_y]$ and $\mathbf{p}[v_z]$, we simulate circuit C with logic gadgets $G_\wedge$, $G_\vee$, $G_\neg$ and calculate the increment vector of ϕ. The above computation is repeated for 41^3 points around $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$, and all the vectors are averaged as the displacement of ϕ at p. Finally, we add it to $\mathbf{p}[v_x]$, $\mathbf{p}[v_y]$, $\mathbf{p}[v_z]$, and use $G_=$ to make sure that, in any ϵ-Nash equilibrium, the displacement of ϕ at p is very close to zero. The averaging maneuver used in the interpolation here also resolves the problem caused by the brittle comparator $G_<$.

Let k_0 be an integer such that, for any input pair $(C, 0^n)$ of 3-Dimensional Brouwer,

$$\text{the number of arithmetic vertices in the graphical game } \mathcal{GG} \leq \left|(C, 0^n)\right|^{k_0}.$$

The hardness proof of 4-Nash in [7] is based on a combined reduction from 3-Dimensional Brouwer to 3-Graphical Nash to 4-Nash. In this work, we developed new structures (which are called nodes here) to perform the task of vertices in the reduction above. Gadgets are designed in the new setting, which allow us to directly reduce 3-Dimensional Brouwer to 2-Nash, and prove that the latter is also **PPAD**-complete.

4 Reduction from 3-Dimensional Brouwer to 2-Nash

In this section, we give a reduction from problem 3-Dimensional Brouwer to 2-Nash and prove that the latter is also **PPAD**-complete. Let $(C, 0^n)$ be any input of 3-Dimensional Brouwer, then a 2-player game $\mathcal{G}$ will be constructed. Given any ϵ-Nash equilibrium of the game where $\epsilon = 2^{-(m+4n)}$ and m is the smallest integer such that $2^m \geq |(C, 0^n)|^{k_0}$ (constant k_0 is defined at the end of section 3), a panchromatic vertex of ϕ can be identified easily.

Let's call the two players P_1 and P_2. For any $i \in \{1, 2\}$, P_i has a set of nodes N_i where $|N_i| = K = 2^m$. Each node v contains two strategies $(v, 0)$ and $(v, 1)$. Thus the strategy set S_i of player P_i consists of totally $2K$ strategies where

$$S_i = \left\{ \ (v, j) \ \middle| \ v \in N_i, \ j \in \{0, 1\} \ \right\} \quad \text{for any } i \in \{1, 2\}.$$

To clarify the presentation, we always use v to denote nodes in N_1 and w to denote nodes in N_2. Given a mixed strategy profile $\mathbf{p}$ of $\mathcal{G}$, we use $\mathbf{p}[v]$ ($\mathbf{p}[w]$) to denote the probability of P_1 choosing strategy $(v, 1)$ (P_2 choosing strategy $(w, 1)$) and $\mathbf{p}_C[v]$ ($\mathbf{p}_C[w]$) to denote the probability of P_1 choosing $(v, 1)$ and $(v, 0)$ (P_2 choosing $(w, 1)$ and $(w, 0)$). It's also called the capacity of node v (w) in the profile $\mathbf{p}$.

The idea of the construction is described informally as follows: The function of nodes in $N_1 \cup N_2$ is similar to the vertices in section 3. Nodes in N_2 are called interior nodes, while nodes in N_1 are called arithmetic nodes, as for any $v \in N_1$, $\mathbf{p}[v]$ would be a meaningful real number in any ϵ-Nash equilibrium $\mathbf{p}$ of game $\mathcal{G}$. Gadgets are designed to implement all the nine arithmetic and logic operations in the new setting. Every gadget contains exactly one interior node in N_2, which is used to mediate between arithmetic nodes in the gadget, so that the latter ones obey the intended arithmetic relationship.

Game $\mathcal{G}$ is built upon $\mathcal{G}^*$ which is a variation of the 2-player Matching Pennies [7] with an exponentially large integer $M = 2^{4(m+n)+1}$. $\mathcal{G}^*$ has the same number of players and same strategy sets as $\mathcal{G}$, and we use u^* to denote its payoffs. To get $\mathcal{G}$, we add a number of gadgets into $\mathcal{G}^*$, which form a network and perform a task similar to the graphical game in section 3. Every gadget contains exactly one interior node in N_2 and ≤ 3 arithmetic nodes in N_1. One of the arithmetic nodes is called the output node of the gadget, and others are called input nodes. Let $w \in N_2$ be the interior node and $v \in N_1$ be the output node of a gadget G. By saying adding G into $\mathcal{G}^*$, we actually modifies the following payoffs of $\mathcal{G}^*$ related to v and w:

the payoff u_s^{*1} to player P_1 where the pure strategy profile s contains node v,
the payoff u_s^{*2} to player P_2 where the pure strategy profile s contains node w.

More exactly, constants in $[0, 1]$ are added to these payoffs, while all the other payoffs of the game stay the same. For any arithmetic node $v \in N_1$, there is exactly one gadget of which v is the output node, while it can be an input node of arbitrarily many gadgets.

Payoffs u^* of Game $\mathcal{G}^*$

```
1: pick an arbitrary one-to-one correspondence C from N_1 to N_2
2: for any pure strategy profile s = ((v,i_1),(w,i_2)), v ∈ N_1, w ∈ N_2, i_1,i_2 ∈ {0,1} do
3:     if C(v) = w then
4:         set u_s^{*1} = M and u_s^{*2} = −M
5:     else
6:         set u_s^{*1} = u_s^{*2} = 0
```

Figure 1: Payoffs u^* of Game $\mathcal{G}^*$

In the left part of this section, we first give the payoffs of game $\mathcal{G}^*$ and define a class $\mathcal{L}$ of games based on it. For any $\mathcal{G}' \in \mathcal{L}$, players can only choose nodes uniformly in a ≤ 1-Nash equilibrium. Then, we design all the necessary gadgets in the new setting. Finally, we build game $\mathcal{G}$ by inserting gadgets into $\mathcal{G}^*$, and prove the correctness of the reduction.

4.1 Payoffs of Game $\mathcal{G}^*$

Payoffs u^* of game $\mathcal{G}^*$ are described in figure 1 where integer $M = 2^{4(m+n)+1} = 2K^4 2^{4n}$.

Definition 6. *A 2-player game $\mathcal{G}'$ (with same strategy sets as $\mathcal{G}$) belongs to $\mathcal{L}$ if its payoffs u' satisfy that $u_s^{'i} \in [u_s^{*i}, u_s^{*i} + 1]$ for any profile $s \in S_1 \times S_2$ and $i \in \{1, 2\}$.*

The following property of games in $\mathcal{L}$ is easy to prove.

Lemma 1. *Let $\mathbf{p}$ be any ≤ 1-Nash equilibrium of game $\mathcal{G}' \in \mathcal{L}$, then for any node $v \in N_1$, $w \in N_2$, the capacities of v and w in profile $\mathbf{p}$ satisfy*

$$\frac{1}{K} - \epsilon < \boldsymbol{p}_C[v],\ \boldsymbol{p}_C[w] < \frac{1}{K} + \epsilon. \qquad (\textit{recall that } \epsilon = \frac{1}{2^{m+4n}} = \frac{1}{K2^{4n}})$$

4.2 Design of Arithmetic and Logic Gadgets

In this part, we design all the nine necessary gadgets, i.e. G_ζ, $G_{\times\zeta}$, $G_=$, G_+, G_-, $G_<$, $G_\wedge$, $G_\vee$ and $G_\neg$ in the new setting. Functions of them are similar to those in [7]. One difference should be noticed here is the representation of bits. Let v be any node in N_1, we say v stores 1 if $\mathbf{p}[v] = \mathbf{p}_C[v]$ and v stores 0 if $\mathbf{p}[v] = 0$. We only prove the property of gadget G_+ below, while others can be verified similarly. Here by $x = y \pm \epsilon$ where $\epsilon > 0$, we mean $y - \epsilon \leq x \leq y + \epsilon$.

Proposition 1 (Gadget G_+). *Let $\mathcal{G}'$ (with payoffs u') be a 2-player game in $\mathcal{L}$ and nodes $v_1, v_2, v_3 \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v_1, 1), (w, 1))$, $s_2 = ((v_2, 1), (w, 1))$, $s_3 = ((v_3, 1), (w, 0))$, $s_4 = ((v_3, 1), (w, 1))$ and $s_5 = ((v_3, 0), (w, 0))$. If game $\mathcal{G}'$ satisfies*

> **1).** $u_{s_1}^{'2} = u_{s_1}^{*2} + 1$, $u_{s_2}^{'2} = u_{s_2}^{*2} + 1$ *and for any other s which contains $(w, 1)$,* $u_s^{'2} = u_s^{*2}$*;*
>
> **2).** $u_{s_3}^{'2} = u_{s_3}^{*2} + 1$ *and for any other s which contains $(w, 0)$,* $u_s^{'2} = u_s^{*2}$*;*

3). $u'^1_{s_4} = u^{*1}_{s_4} + 1$ *and for any other s which contains* $(v_3, 1)$, $u'^1_s = u^{*1}_s$;

4). $u'^1_{s_5} = u^{*1}_{s_5} + 1$ *and for any other s which contains* $(v_3, 0)$, $u'^1_s = u^{*1}_s$,

then in any ϵ-Nash equilibrium $\boldsymbol{p}$ of $\mathcal{G}'$, we have $\boldsymbol{p}[v_3] = \min(\boldsymbol{p}[v_1] + \boldsymbol{p}[v_2], \boldsymbol{p}_C[v_3]) \pm \epsilon$.

Proof. Properties **1)** − **4)** show that, in any mixed strategy profile $\mathbf{p}$ of game $\mathcal{G}'$, we have

$$\text{payoff to } P_2 \text{ if it chooses } (w, 1) - \text{payoff to } P_2 \text{ if it chooses } (w, 0) = \mathbf{p}[v_1] + \mathbf{p}[v_2] - \mathbf{p}[v_3]$$
$$\text{payoff to } P_1 \text{ if it chooses } (v_3, 1) - \text{payoff to } P_1 \text{ if it chooses } (v_3, 0) = \mathbf{p}[w] - (\mathbf{p}_C[w] - \mathbf{p}[w])$$

If $\mathbf{p}[v_3] - (\mathbf{p}[v_1] + \mathbf{p}[v_2]) > \epsilon$, then the first equation shows that $\mathbf{p}[w] = 0$ and the second one shows $\mathbf{p}[v_3] = 0$ which contradicts with our assumption that $\mathbf{p}[v_3] > \mathbf{p}[v_1] + \mathbf{p}[v_2] + \epsilon > 0$. If $\mathbf{p}[v_3] - (\mathbf{p}[v_1] + \mathbf{p}[v_2]) < -\epsilon$, then the first equation shows $\mathbf{p}[w] = \mathbf{p}_C[w]$ and the second one shows that $\mathbf{p}[v_3] = \mathbf{p}_C[v_3]$. As $\mathbf{p}_C[v_3] = \mathbf{p}[v_3] < \mathbf{p}[v_1] + \mathbf{p}[v_2]$, we have $\mathbf{p}[v_3] = \mathbf{p}_C[v_3] > \mathbf{p}_C[v_3] - \epsilon = \min(\mathbf{p}[v_1] + \mathbf{p}[v_2], \mathbf{p}_C[v_3]) - \epsilon$, and the proposition is proven. □

Proposition 2 (Gadget G_ζ where $\zeta \leq 1/K - \epsilon$). *Let $\mathcal{G}'$ (with payoffs u') be a game in $\mathcal{L}$ and nodes $v \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v, 1), (w, 1))$, $s_2 = ((v, 1), (w, 0))$ and $s_3 = ((v, 0), (w, 1))$. If the following conditions are satisfied*

1). $u'^2_{s_1} = u^{*2}_{s_1} + 1$ *and for any other s which contains* $(w, 1)$, $u'^2_s = u^{*2}_s$;

2). *for any s which contains* $(w, 0)$, $u'^2_s = u^{*2}_s + \zeta$;

3). $u'^1_{s_2} = u^{*1}_{s_2} + 1$ *and for any other s which contains* $(v, 1)$, $u'^1_s = u^{*1}_s$;

4). $u'^1_{s_3} = u^{*1}_{s_3} + 1$ *and for any other s which contains* $(v, 0)$, $u'^1_s = u^{*1}_s$,

then in any ϵ-Nash equilibrium $\boldsymbol{p}$ of game $\mathcal{G}'$, we have $\boldsymbol{p}[v] = \zeta \pm \epsilon$.

Proposition 3 (Gadget $G_{\times\zeta}$ where $\zeta \leq 1$). *Let $\mathcal{G}'$ (with payoffs u') be a game in $\mathcal{L}$ and nodes $v_1, v_2 \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v_1, 1), (w, 1))$, $s_2 = ((v_2, 1), (w, 0))$, $s_3 = ((v_2, 1), (w, 1))$ and $s_4 = ((v_2, 0), (w, 0))$. If the payoffs of game $\mathcal{G}'$ satisfy*

1). $u'^2_{s_1} = u^{*2}_{s_1} + \zeta$ *and for any other s which contains* $(w, 1)$, $u'^2_s = u^{*2}_s$;

2). $u'^2_{s_2} = u^{*2}_{s_2} + 1$ *and for any other s which contains* $(w, 0)$, $u'^2_s = u^{*2}_s$;

3). $u'^1_{s_3} = u^{*1}_{s_3} + 1$ *and for any other s which contains* $(v_2, 1)$, $u'^1_s = u^{*1}_s$;

4). $u'^1_{s_4} = u^{*1}_{s_4} + 1$ *and for any other s which contains* $(v_2, 0)$, $u'^1_s = u^{*1}_s$,

then in any ϵ-Nash equilibrium $\boldsymbol{p}$ of game $\mathcal{G}'$, we have $\boldsymbol{p}[v_2] = \min(\zeta\boldsymbol{p}[v_1], \boldsymbol{p}_C[v_2]) \pm \epsilon$.

Proposition 4 (Gadget $G_=$). *Gadget $G_=$ is a special case of $G_{\times\zeta}$. We just set the constant $\zeta = 1$, then in any ϵ-Nash equilibrium $\boldsymbol{p}$ of game $\mathcal{G}'$,* $\boldsymbol{p}[v_2] = \min(\boldsymbol{p}[v_1], \boldsymbol{p}_C[v_2]) \pm \epsilon$.

Proposition 5 (Gadget G_-). *Let $\mathcal{G}'$ (with payoffs u') be a 2-player game in $\mathcal{L}$ and nodes $v_1, v_2, v_3 \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v_1,1),(w,1))$, $s_2 = ((v_2,1),(w,0))$, $s_3 = ((v_3,1),(w,0))$, $s_4 = ((v_3,1),(w,1))$ and $s_5 = ((v_3,0),(w,0))$. If game $\mathcal{G}'$ satisfies*

1). $u'^2_{s_1} = u^{*2}_{s_1} + 1$ *and for any other s which contains $(w,1)$, $u'^2_s = u^{*2}_s$;*

2). $u'^2_{s_2} = u^{*2}_{s_2} + 1$, $u'^2_{s_3} = u^{*2}_{s_3} + 1$ *and for any other s which contains $(w,0)$, $u'^2_s = u^{*2}_s$;*

3). $u'^1_{s_4} = u^{*1}_{s_4} + 1$ *and for any other s which contains $(v_3,1)$, $u'^1_s = u^{*1}_s$;*

4). $u'^1_{s_5} = u^{*1}_{s_5} + 1$ *and for any other s which contains $(v_3,0)$, $u'^1_s = u^{*1}_s$,*

then in any ϵ-Nash equilibrium $\boldsymbol{p}$ of game $\mathcal{G}'$, we have

$$\min(\boldsymbol{p}[v_1] - \boldsymbol{p}[v_2], \boldsymbol{p}_C[v_3]) - \epsilon \le \boldsymbol{p}[v_3] \le \max(\boldsymbol{p}[v_1] - \boldsymbol{p}[v_2], 0) + \epsilon\,.$$

Proposition 6 (Gadget $G_<$). *Let $\mathcal{G}'$ (with payoffs u') be a 2-player game in $\mathcal{L}$ and nodes $v_1, v_2, v_3 \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v_1,1),(w,1))$, $s_2 = ((v_2,1),(w,0))$, $s_3 = ((v_3,1),(w,0))$ and $s_4 = ((v_3,0),(w,1))$. If the payoffs of game $\mathcal{G}'$ satisfy*

1). $u'^2_{s_1} = u^{*2}_{s_1} + 1$ *and for any other s which contains $(w,1)$, $u'^2_s = u^{*2}_s$;*

2). $u'^2_{s_2} = u^{*2}_{s_2} + 1$ *and for any other s which contains $(w,0)$, $u'^2_s = u^{*2}_s$;*

3). $u'^1_{s_3} = u^{*1}_{s_3} + 1$ *and for any other s which contains $(v_3,1)$, $u'^1_s = u^{*1}_s$;*

4). $u'^1_{s_4} = u^{*1}_{s_4} + 1$ *and for any other s which contains $(v_3,0)$, $u'^1_s = u^{*1}_s$,*

then in any ϵ-Nash equilibrium $\boldsymbol{p}$ of game $\mathcal{G}'$, we have $\boldsymbol{p}[v_3] = \boldsymbol{p}_C[v_3]$ if $\boldsymbol{p}[v_1] < \boldsymbol{p}[v_2] - \epsilon$ and $\boldsymbol{p}[v_3] = 0$ if $\boldsymbol{p}[v_1] > \boldsymbol{p}[v_2] + \epsilon$.

Proposition 7 (Gadget $G_\vee$). *Let $\mathcal{G}'$ (with payoffs u') be a 2-player game in $\mathcal{L}$ and nodes $v_1, v_2, v_3 \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v_1,1),(w,1))$, $s_2 = ((v_2,1),(w,1))$, $s_3 = ((v_3,1),(w,1))$ and $s_4 = ((v_3,0),(w,0))$. If the payoffs of game $\mathcal{G}'$ satisfy*

1). $u'^2_{s_1} = u^{*2}_{s_1} + 1$, $u'^2_{s_2} = u^{*2}_{s_2} + 1$ *and for any other s which contains $(w,1)$, $u'^2_s = u^{*2}_s$;*

2). *for any s which contains $(w,0)$, $u'^2_s = u^{*2}_s + 1/(2K)$;*

3). $u'^1_{s_3} = u^{*1}_{s_3} + 1$ *and for any other s which contains $(v_3,1)$, $u'^1_s = u^{*1}_s$;*

4). $u'^1_{s_4} = u^{*1}_{s_4} + 1$ *and for any other s which contains $(v_3,0)$, $u'^1_s = u^{*1}_s$,*

then in any ϵ-Nash equilibrium $\boldsymbol{p}$, we have $\boldsymbol{p}[v_3] = \boldsymbol{p}_C[v_3]$ if $\boldsymbol{p}[v_1] = \boldsymbol{p}_C[v_1]$ or $\boldsymbol{p}[v_2] = \boldsymbol{p}_C[v_2]$ and $\boldsymbol{p}[v_3] = 0$ if $\boldsymbol{p}[v_1] = \boldsymbol{p}[v_2] = 0$.

Proposition 8 (Gadget $G_\wedge$). *Gadget $G_\wedge$ is similar as $G_\vee$. We only change the constant in* **2)** *of Proposition 7 from $1/(2K)$ to $3/(2K)$, then in any ϵ-Nash equilibrium $\boldsymbol{p}$, $\boldsymbol{p}[v_3] = 0$ if $\boldsymbol{p}[v_1] = 0$ or $\boldsymbol{p}[v_2] = 0$, and $\boldsymbol{p}[v_3] = \boldsymbol{p}_C[v_3]$ if $\boldsymbol{p}[v_1] = \boldsymbol{p}_C[v_1]$ and $\boldsymbol{p}[v_2] = \boldsymbol{p}_C[v_2]$.*

Proposition 9 (Gadget $G_\neg$). *Let $\mathcal{G}'$ (with payoffs u') be a 2-player game in $\mathcal{L}$ and nodes $v_1, v_2 \in N_1$, $w \in N_2$. Let pure strategy profile $s_1 = ((v_1,1),(w,1))$, $s_2 = ((v_1,0),(w,0))$, $s_3 = ((v_2,1),(w,0))$ and $s_4 = ((v_2,0),(w,1))$. If the payoffs of game $\mathcal{G}'$ satisfy*

> **1).** $u'^2_{s_1} = u^{*2}_{s_1} + 1$ *and for any other s which contains $(w,1)$, $u'^2_s = u^{*2}_s$;*
>
> **2).** $u'^2_{s_2} = u^{*2}_{s_2} + 1$ *and for any other s which contains $(w,0)$, $u'^2_s = u^{*2}_s$;*
>
> **3).** $u'^1_{s_3} = u^{*1}_{s_3} + 1$ *and for any other s which contains $(v_2,1)$, $u'^1_s = u^{*1}_s$;*
>
> **4).** $u'^1_{s_4} = u^{*1}_{s_4} + 1$ *and for any other s which contains $(v_2,0)$, $u'^1_s = u^{*1}_s$,*

then in any ϵ-Nash equilibrium $\boldsymbol{p}$, $\boldsymbol{p}[v_2] = 0$ if $\boldsymbol{p}[v_1] = \boldsymbol{p}_C[v_1]$ and $\boldsymbol{p}[v_2] = \boldsymbol{p}_C[v_2]$ if $\boldsymbol{p}[v_1] = 0$.

4.3 Construction of Game $\mathcal{G}$

Now we are ready to use the gadgets designed so far to build the game $\mathcal{G}$. We use $G_\zeta(v,w)$ to denote the insertion of a G_ζ gadget into game G^* with v as its output node and w as its interior node. For gadgets with one input node ($G_{\times\zeta}$, $G_\neg$ and $G_=$), we use $G(v_1,v_2,w)$ to denote the insertion of such a gadget into game G^* with v_1, v_2, w as its input node, output node and interior node respectively. For gadgets with two input nodes, we use $G(v_1,v_2,v_3,w)$ to denote the insertion of such a gadget into game G^* with v_1 and v_2 as its first and second input node respectively, v_3 as its output node and w as its interior node.

The structure of the gadget network in $\mathcal{G}$ is similar to the one in [7]. There are 3 distinguished nodes v_x, v_y, v_z in N_1 and real numbers $\mathbf{p}[v_x]$, $\mathbf{p}[v_y]$, $\mathbf{p}[v_z]$ encode a point $\mathbf{t} = (x,y,z)$ in the unit cube $[0,1]^3$ where $x = K\mathbf{p}[v_x]$, $y = K\mathbf{p}[v_y]$, $z = K\mathbf{p}[v_z]$. (Strictly speaking, it may happen that $K\mathbf{p}[v_x] > 1$ according to Lemma 1, but we will prove that this is impossible in any ϵ-Nash equilibrium later.). Let K_{ijk} be the cubelet that contains point $\mathbf{t}$. Starting from v_x ,v_y and v_z, we extract the $3n$ bits which represent integer i,j,k (from the $(m+1)$th bit to the $(m+n)$th bit of $\mathbf{p}[v_x]$, $\mathbf{p}[v_y]$ and $\mathbf{p}[v_z]$), and use logic gadgets to simulate C.

But only getting the increment of ϕ at c_{ijk} is not enough, we need to repeat the above computation for 41^3 points of the form $(x+p\cdot\alpha, y+q\cdot\alpha, z+r\cdot\alpha)$ for $-20 \le p,q,r \le 20$ and finally calculate the average of all these increments. After adding the displacement to $\mathbf{p}[v_x], \mathbf{p}[v_y]$ and $\mathbf{p}[v_z]$, we insert gadgets $G_=$ to make sure that in any ϵ-Nash equilibrium $\mathbf{p}$, the average increment at $\mathbf{t}$ is very close to zero. This property guarantees the existence of a panchromatic vertex near $\mathbf{t}$, which can be identified from $\mathbf{p}$ very efficiently. The averaging maneuver used in the interpolation also resolves the problem caused by the brittle comparator $G_<$ at the same time.

The construction of game $\mathcal{G}$ is divided into 5 parts:

Part 1. Starting from the three distinguished nodes v_x, v_y, $v_z \in N_1$, for any $-20 \le i \le 20$, there are three nodes v_{x_i}, v_{y_i} and v_{z_i} in N_1. By adding gadgets G_ζ, G_- and G_+, we make sure that in any ϵ-Nash equilibrium $\mathbf{p}$ of $\mathcal{G}$, $\mathbf{p}[v_{x_i}] = \min(\mathbf{p}[v_x] + i\alpha', \mathbf{p}_C[v_{x_i}]) \pm 4\epsilon$ if $i \ge 0$

and $\mathbf{p}[v_{x_i}] = \max(\mathbf{p}[v_x] + i\alpha', 0) \pm 4\epsilon$ if $i < 0$ where $\alpha' = \alpha 2^{-m}$. Similar results also stand for nodes v_{y_i} and v_{z_i}.

Part 2. For any $-20 \le p \le 20$, we extract $3n$ bits (from the $(m+1)$th to the $(m+n)$th) of $\mathbf{p}[v_{x_p}]$, $\mathbf{p}[v_{y_p}]$ and $\mathbf{p}[v_{z_p}]$ and store them in $v^i_{x_p}$, $v^i_{y_p}$ and $v^i_{z_p} \in N_1$ where $1 \le i \le n$. Figure 2 shows how to extract the n bits of $\mathbf{p}[v_{x_p}]$. Although we hope $\mathbf{p}[v^i_{x_p}] = 0$ if the $(m+i)$th bit of $\mathbf{p}[v_{x_p}]$ is 0 and $\mathbf{p}[v^i_{x_p}] = \mathbf{p}_C[v^i_{x_p}]$ if it is 1, this may not be true because of the brittle comparator $G_<$. Lemma 2 below is easy to check, and similar results also stand for $v^i_{y_p}$, $v^i_{z_p}$.

Lemma 2. *If $\boldsymbol{p}[v_{x_p}] \ge 1/K - 61\alpha'$, then $\boldsymbol{p}[v^i_{x_p}] = \boldsymbol{p}_C[v^i_{x_p}]$ for any $1 \le i \le n$. If $\boldsymbol{p}[v_{x_p}] \le 61\alpha'$, then $\boldsymbol{p}[v^i_{x_p}] = 0$ for any $1 \le i \le n$. Otherwise, if $\boldsymbol{p}[v_{x_p}]$ satisfies*

$$\left| \, \boldsymbol{p}[v_{x_p}] - \frac{\lfloor 2^{n+m} \boldsymbol{p}[v_{x_p}] \rfloor}{2^{n+m}} \, \right| > n^2 \epsilon \,,$$

then $\boldsymbol{p}[v^i_{x_p}] = 0$ if the $(m+i)$th bit of real number $\boldsymbol{p}[v_{x_p}]$ is 0 and $\boldsymbol{p}[v^i_{x_p}] = \boldsymbol{p}_C[v^i_{x_p}]$ if it is 1, for any integer $1 \le i \le n$.

Part 3. For any $-20 \le p, q, r \le 20$, we recognize the $3n$ nodes $v^i_{x_p} v^i_{y_q} v^i_{z_r}$ where $1 \le i \le n$ as the input bits of circuit C and use logic gadgets $G_\wedge$, $G_\vee$, $G_\neg$ to simulate it. The outputs (6 bits) are stored in 6 nodes, Δx^+_{pqr}, Δx^-_{pqr}, Δy^+_{pqr}, Δy^-_{pqr}, Δz^+_{pqr} and Δz^-_{pqr} in N_1.

Part 4. Pick 6 unused nodes Δx^+, Δx^-, Δy^+, Δy^-, Δz^+ and Δz^- in N_1. By using gadgets $G_{\times\zeta}$ and G_+, we make sure that in any ϵ-Nash equilibrium $\mathbf{p}$ of game $\mathcal{G}$,

$$\mathbf{p}[\Delta x^+] = \Big(\sum_{p,q,r} \frac{\alpha}{41^3} \, \mathbf{p}[\Delta x^+_{pqr}] \Big) \pm 3 \cdot 41^3 \epsilon \qquad \mathbf{p}[\Delta x^-] = \Big(\sum_{p,q,r} \frac{\alpha}{41^3} \, \mathbf{p}[\Delta x^-_{pqr}] \Big) \pm 3 \cdot 41^3 \epsilon \,,$$

and similar results also stand for nodes Δy^+, Δy^-, Δz^+ and Δz^-.

Part 5. Pick unused nodes v_1, v_2, v_3, v'_x, v'_y, $v'_z \in N_1$, w_1 ... $w_9 \in N_2$, and add the following nine gadgets into game $\mathcal{G}^*$.

$$\begin{array}{lll} G_+(v_x, \Delta x^+, v_1, w_1) & G_-(v_1, \Delta x^-, v'_x, w_2) & G_=(v'_x, v_x, w_3) \\ G_+(v_y, \Delta y^+, v_2, w_4) & G_-(v_2, \Delta y^-, v'_y, w_5) & G_=(v'_y, v_y, w_6) \\ G_+(v_z, \Delta z^+, v_3, w_7) & G_-(v_3, \Delta z^-, v'_z, w_8) & G_=(v'_z, v_z, w_9) \end{array}$$

4.4 Correctness of the Reduction

Obviously, game $\mathcal{G}$ belongs to $\mathcal{L}$ and all the gadgets inserted work well in it. The size of game $\mathcal{G}$ is polynomial of $|(C, 0^n)|$, as both the number of strategies and the number of bits required to represent a payoff u^i_s are polynomial of $|(C, 0^n)|$, and game $\mathcal{G}$ can be computed from $(C, 0^n)$ in polynomial time. Furthermore, the following theorem shows that, given any ϵ-Nash equilibrium $\mathbf{p}$ of game $\mathcal{G}$ where $\epsilon = 2^{-(m+4n)}$, a panchromatic vertex of $(C, 0^n)$ can be identified very efficiently.

Implementation of Part 2

pick unused nodes v_1, v_2 ... $v_{n+1} \in N_1$ and $w \in N_2$
$G_=(v_{x_p}, v_1, w)$,
for any $1 \le i \le n$ **do**
    pick unused nodes v^1, v^2, $v^3 \in N_1$ and w^1, w^2, w^3, $w^4 \in N_2$
    $G_{2^{-(m+i)}}(v^1, w^1)$, $G_<(v^1, v_i, v^2, w^2)$, $G_{\times 2^{-i}}(v^2, v^3, w^3)$, $G_-(v_i, v^3, v_{i+1}, w^4)$

Figure 2: Implementation of **Part 2**

Theorem 2. *Let $\boldsymbol{p}$ be any ϵ-Nash equilibrium of the game $\mathcal{G}$ constructed above. $x = K\boldsymbol{p}[v_x]$, $y = K\boldsymbol{p}[v_y]$ and $z = K\boldsymbol{p}[v_z]$ where $K = 2^m$. Let p, q, r be three integers satisfying*

$$(p-1)2^{-n} < x - 30\alpha < x + 30\alpha < (p+1)2^{-n}\,;$$
$$(q-1)2^{-n} < y - 30\alpha < y + 30\alpha < (q+1)2^{-n}\,;$$
$$(r-1)2^{-n} < z - 30\alpha < z + 30\alpha < (r+1)2^{-n}\,,$$

then vertex $(p2^{-n}, q2^{-n}, r2^{-n})$ must be a panchromatic vertex of $(C, 0^n)$.

Similarly as the proof in [7], we need the following property of the four increments.

Lemma 3. *Suppose that for nonnegative integers k_1 ... k_4, all three coordinates of $\sum_{i=1}^4 k_i\delta_i$ are smaller in absolute value than $\alpha k/5$ where $k = \sum_{i=1}^4 k_i$. Then all four k_i are positive.*

Proof of Theorem 2. First, we prove that $\mathbf{t} = (x, y, z)$ cannot be close to the boundary of the unit cube. If $\mathbf{p}[v_x] \le 40\alpha'$, then Lemma 2 shows for any $-20 \le i, j, k \le 20$, we have $\mathbf{p}[\Delta x^+_{ijk}] = \mathbf{p}_C[\Delta x^+_{ijk}]$. Thus $\mathbf{p}[\Delta x^+]$ is very close to $\alpha' = \alpha 2^{-m}$, while $\mathbf{p}[\Delta x^-]$ is close to 0. As α' is much larger than ϵ, we get a contradiction in $G_=(v'_x, v_x, w_3)$ of **Part 5**. Similarly, we can prove that $\mathbf{p}[v_x] < 1/K - 40\alpha'$, which can be easily generalized to $\mathbf{p}[v_y]$ and $\mathbf{p}[v_z]$.

Now we see the existence of integers p, q, r which satisfy the three conditions above. Let T be the set of eight centers around $(p2^{-n}, q2^{-n}, r2^{-n})$, $V = \{\ (i, j, k),\ -20 \le i, j, k \le 20\ \}$ and V_1 be the subset of V such that, triple $(i, j, k) \in V_1$ iff

$$\left|\,\mathbf{p}[v_{x_i}] - p2^{-(n+m)}\,\right| \le n^2\epsilon \ \text{ or } \ \left|\,\mathbf{p}[v_{y_j}] - q2^{-(n+m)}\,\right| \le n^2\epsilon \ \text{ or } \ \left|\,\mathbf{p}[v_{z_k}] - r2^{-(n+m)}\,\right| \le n^2\epsilon\,.$$

As α' is much larger than ϵ, we have $|\,V_1\,| \le 3 \cdot 41^2$. For any triple $(i, j, k) \in V - V_1$, Lemma 2 shows that all the $3n$ bits of $\mathbf{p}[v_{x_i}]$, $\mathbf{p}[v_{y_j}]$ and $\mathbf{p}[v_{z_k}]$ are extracted successfully, and Δx^+_{ijk} etc. values should imply an increment which is same as one of those at centers in T. Let k_t, where $1 \le t \le 4$, be the number of triples in $V - V_1$ whose Δx^+_{ijk} etc. values imply the vector δ_t, then all four k_i must be positive according to Lemma 3 (otherwise, we could find a contradiction in one of the $G_=$ gadgets of **Part 5**), which shows that $(p2^{-n}, q2^{-n}, r2^{-n})$ is a panchromatic vertex of $(C, 2^n)$, and the theorem is proven. □

Theorem 3. *Search problem* 2-Nash *is* **PPAD***-complete.*

Corollary 1. *Search problem* Nash *is* **PPAD***-complete.*

5 Concluding Remarks

Even though many thought the problem of finding a Nash-equilibrium is hard in general, and has been proven so for three or more players recently, it is not clear whether the 2-player case can be shown in the same class of **PPAD**-complete problems. Our work settles this issue and a long standing open problem that has attracted Mathematicians, Economists, Operations Researchers, and most recently Computer Scientists. The result shows the richness of the **PPAD**-complete class introduced by Papadimitriou fifteen years ago [15]. The new proof techniques which made inclusion of r-Nash into this class possible, started in Goldberg and Papadimitriou [9], have shown a variety of structures, as exhibited in the hardness proofs of problem 4-Nash, 2D-SPERNER [2], 3-Nash, and finally 2-Nash, may find their use in other related problems and complexity classes.

References

[1] L.E.J. Brouwer. Über Abbildung von Mannigfaltigkeiten. *Mathematische Annalen*, 71:97–115, 1910.

[2] X. Chen and X. Deng. 2D-SPERNER is PPAD-complete. *submitted to ICALP 2006.*

[3] X. Chen and X. Deng. On Algorithms for Discrete and Approximate Brouwer Fixed Points. In *STOC 2005*, pages 323–330.

[4] X. Chen and X. Deng. 3-Nash is PPAD-complete. *ECCC, TR05-134*, 2005.

[5] R.W. Cottle and G.B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra Appl.*, 1:103–125, 1986.

[6] G.B. Danzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[7] C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *ECCC, TR05-115*, 2005.

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

[9] P.W. Goldberg and C.H. Papadimitriou. Reducibility Among Equilibrium Problems. *ECCC, TR05-90*, 2005.

[10] M.D. Hirsch, C.H. Papadimitriou, and S. Vavasis. Exponential lower bounds for finding Brouwer fixed points. *J.Complexity*, 5:379–416, 1989.

[11] M. Kearns, M. Littman, and S. Singh. Graphical Models for Game Theory. In *Proceedings of UAI*, 2001.

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

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

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

[15] C.H. Papadimitriou. On inefficient proofs of existence and complexity classes. In *Proceedings of the 4th Czechoslovakian Symposium on Combinatorics*, 1991.

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

[17] A.C-C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proceedings of FOCS 1977*, pages 222–227.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'