Settling the Complexity of Two-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 the problem of finding a Nash equilibrium in a two-player game is ***PPAD****-complete.*

1 Introduction

Almost sixty years ago, Morgenstern and von Neumann [17] advocated the study of game theory with its applications to economic behavior. A central mathematical result applied to their game theoretical study is von Neumann's existence theorem [22] of equilibrium in the two-player zero-sum game model, where one player's gain is the loss of the other. The proof is equivalent to the duality property of polytopes, which also forms the mathematical foundation of Dantzig's linear programming method for optimization problems [8] , as well as Yao's principle for finding algorithmic lower bounds [23]. Nash proposed in the middle of the last century to study the more general non-zero sum games, and proved that there exists a set of (mixed) strategies, now commonly referred to as a Nash equilibrium, one for each player, such that no player can benefit if it changes its own strategy unilaterally.

While the problem of computing a Nash equilibrium in a two-player zero-sum game is solvable in polynomial time since linear programming is, as by Khachiyan's ellipsoid algorithm [15], the existence proof of Nash equilibria relied on Kakutani's fixed point theorem (a generalization of Brouwer's fixed point theorem [1]), for which it is known that, in the functional oracle model, any algorithm solving the fixed point problem would unconditionally require an exponential number of function evaluations [12, 3]. The particular path following algorithm developed by Lemke and Howson [16] was recently proven to require, even in the best case for some instances, an exponential number of steps, by Rahul Savani and Bernhard Von Stengel [21]. For the original two-player Nash equilibrium problem, despite much effort in the last half century, no significant progress has been made on characterizing its algorithmic complexity, though both hardness results and algorithms have been developed for various modified versions.

An exciting breakthrough, which stated that computing Nash equilibria is indeed hard, was recently made by Daskalakis, Goldberg and Papadimitriou [9], for games among four players or more. An approximation version, within an exponentially small factor, 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 [19]. The work was improved to the three-player case by Chen and Deng [4], Daskalakis and Papadimitriou [10], independently, and with different proofs. Those results leave the two-player Nash equilibrium the last open problem, which has been referred to as one of the two "most concrete open problems" at the boundary of **P** [18], in the long sequel of search for an efficient solution.

Finding a Nash equilibrium in a two-player game could be easier for several reasons. Firstly, the zero-sum version can be solved in polynomial time by linear programming. Secondly, it admits a rational number solution of polynomial size [7], while it is not known if every game among three or more players has an exact solution of polynomial size. Finally, an important technique employed in the previous hardness proofs, that colors vertices (or players) of a graphical game, does not seem possible to work down to the two-player case.

The reduction of Daskalakis, Goldberg and Papadimitriou [9] started with a **PPAD**-complete problem named 3-DIMENSIONAL BROUWER, a discrete fixed point problem in 3D space; then reduced it to a zero point problem by using a small sampling cube, and approximated through a set of selected points in the cube. They continued to reduce the zero point problem to the problem of finding an approximate Nash equilibrium in a degree-3 graphical game (graphical games were first proposed in [14]), and further to the four-player Nash equilibrium problem by a recent result of Goldberg and Papadimitriou [11]. The construction shows deep ways of manipulations one can make use of Nash's equilibrium theorem. For example, it induces an alternative proof for Brouwer's fixed point theorem using Nash's equilibrium theorem.

In this work, we settle the computational complexity of the two-player Nash equilibrium problem with a **PPAD**-completeness proof, with innovative ideas to simplify the proof structure, and most importantly, to overcome difficulties in the previous proofs. In the overall structure, we bypass the graphical game model and derive a direct reduction from 3-DIMENSIONAL BROUWER to 2-NASH, the approximation version of the two-player Nash equilibrium problem. Because any exact equilibrium is also an approximate equilibrium by the definition of 2-NASH, the problem of finding an exact equilibrium of a two-player game (which is denoted by NASH in [20]) is **PPAD**-hard. On the other hand, it was known by Cottle and Dantzig [7] that the two-player case admits a rational solution of polynomial size and thus, NASH lies in **PPAD** by Papadimitriou using the Lemke-Howson algorithm [16, 20]. Our result immediately implies that NASH is **PPAD**-complete.

Several ideas are crucial to our reduction. First, we use the matching pennies game by splitting each choice into a pair of strategies [11], so that in every Nash equilibrium of the game, the sum of the probabilities for each pair of split-strategies is a constant. This property allows us to encode boolean variables and (approximations of) real numbers in an innovative way. Second, we perturb the matching pennies game by adding a set of logic and arithmetic gadgets, which require the encoding of paired strategies to follow a set of approximate functional relationships independently. By adding a gadget, we perturb some rows or columns of the matching pennies. An important idea is to make the coefficients of the matching pennies game significantly larger than the magnitude of the perturbations. In every approximate Nash equilibrium of the resulting game, the sum of the probabilities for each pair of split-strategies will remain very close to uniform. At the same time, the perturbations will force the two probability vectors to satisfy all the relationships (or constraints) dictated by the gadgets.

The structure of our reduction is clear (and simpler than previous work), and the proof of the correctness is carried out step by step. However, there are indeed quite a few innovations in this series of work [11, 9, 4, 10], and it may require extra effort to understand many of the intermediate results. For example, in the definition of 3-DIMENSIONAL BROUWER, a fixed point is a unit cube which has all four function values (or colors) on its eight vertices. But not every such unit cube would have an interior sampling cube that averages the values on the eight vertices to zero. This is the trickiest part in the concept of discrete fixed points proposed by Daskalakis, Goldberg, and Papadimitriou [9]. They define it in one version (a unit cube with all four function values) for its **PPAD**-hardness, and use it in a more restricted version (the values on the unit cube allow an interior sampling cube to average them to zero), in a mathematically sound way as guaranteed by Nash's equilibrium theorem, for the **PPAD**-hardness result of 4-NASH. Nevertheless, one can verify the existence of such a cube, independent of this proof, by using either Brouwer's degree theory, or its discrete version [3]. This is especially clear if we start our proof from the 2D discrete fixed point result of **PPAD**-completeness [2]. We can enumerate other puzzles that are not easily understandable at first glance, but we refrain ourselves to do so. The first time readers would benefit from going through all the proofs presented in the paper. Persistent thinkers can carefully figure out the subtleties, by getting familiar to the methodology of using Nash equilibria to solve the fixed point problem.

The novelty of our proof techniques not only settles this open problem in a direction regarded impossible by many but also opens up new possibilities for stronger hardness results for related problems (see, e.g., [6, 13]). We expect them to be useful not only to Algorithmic Game Theory but also to a wider range of problems arisen from Economics and Operations Research.

2 Preliminaries

A two-player game $\mathcal{G}$ is defined by a pair of $m \times n$ matrices $(\mathbf{A} = (a_{i,j}), \mathbf{B} = (b_{i,j}))$, where the m rows and the n columns, respectively, are the pure strategies of the first and the second players. If the first player chooses strategy i and the second player chooses strategy j, then their payoffs are $a_{i,j}$ and $b_{i,j}$, respectively.

A mixed strategy of a player is a probability distribution over its pure strategies. We use $\mathbb{P}^n$ to denote the set of all *probability vectors* in $\mathbb{R}^n$, i.e., non-negative vectors whose entries sum to 1. A profile of mixed strategies is a pair of mixed strategies $(\mathbf{x} \in \mathbb{P}^m, \mathbf{y} \in \mathbb{P}^n)$, one for each player.

We use $\mathbf{A}_i$ to denote the ith row vector of $\mathbf{A}$, and $\mathbf{B}_i$ to denote the ith column vector of $\mathbf{B}$.

Definition 1. *A Nash equilibrium of game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is a profile of mixed strategies $(\mathbf{x}, \mathbf{y})$ such that*

$$\mathbf{A}_i\mathbf{y} > \mathbf{A}_j\mathbf{y} \implies x_j = 0, \ \forall\, 1 \le i, j \le m;$$
$$\mathbf{x}^T\mathbf{B}_i > \mathbf{x}^T\mathbf{B}_j \implies y_j = 0, \ \forall\, 1 \le i, j \le n.$$

Informally speaking, a Nash equilibrium is a profile of mixed strategies such that no player can gain by unilaterally changing to a different mixed strategy, while the other player remains unchanged. As a tool to carry out the proof, we will also use the following notion of approximate Nash equilibria, an innovation introduced by Daskalakis, Goldberg, and Papadimitriou [9].

Definition 2. *An ϵ-Nash equilibrium of game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is a profile of mixed strategies $(\mathbf{x}, \mathbf{y})$ such that*

$$\mathbf{A}_i\mathbf{y} > \mathbf{A}_j\mathbf{y} + \epsilon \implies x_j = 0, \forall\, 1 \le i, j \le m;$$
$$\mathbf{x}^T\mathbf{B}_i > \mathbf{x}^T\mathbf{B}_j + \epsilon \implies y_j = 0, \forall\, 1 \le i, j \le n.$$

Obviously, an exact equilibrium is also an approximate equilibrium as defined above. Therefore, hardness in finding an approximate Nash equilibrium implies hardness in finding an exact Nash equilibrium.

2.1 PPAD and 2-Nash

PPAD is a class of search problems defined by its complete problem: ANOTHER END OF DIRECTED LINES (or END OF LINE) [20].

The input instance of the problem is a directed graph of an exponential number of vertices (which are numbered from 1 to 2^n), each with at most one incoming edge and at most one outgoing edge. In the graph, vertex 1 is a source vertex which has one outgoing edge but no incoming edge. The graph is generated by a pair of polynomial-time computable functions out and in such that, given a vertex k, $out(k)$ returns the vertex pointed to by k and $in(k)$ returns the vertex points to k, where *nil* is returned if no such vertex exists. The output is a sink vertex, or a source other than 1.

A search problem belongs to class **PPAD** if there is a polynomial-time reduction which reduces the problem to ANOTHER END OF DIRECTED LINES.

Definition 3 (2-NASH and NASH). *The input instance of problem 2-*NASH *is a pair $(\mathcal{G}, 0^k)$ where $\mathcal{G}$ is a two-player game, and the output is a 2^{-k}-Nash equilibrium of $\mathcal{G}$.*

The input of problem NASH *is a two-player game $\mathcal{G}$ and the output is an exact Nash equilibrium of $\mathcal{G}$.*

2.2 3-Dimensional Brouwer

Definition 4. *For each $n \ge 1$, we define a set*

$$A^n = \{\, \mathbf{r} \in \mathbb{Z}^3 \;\big|\; 0 \le r_i \le 2^n - 1, \forall\, 1 \le i \le 3 \,\}.$$

and its boundary $B^n = \{\, \mathbf{r} \in A^n \mid \exists\, i, r_i = 0$ or $2^n - 1 \,\}$.

In [9], the following search problem was proposed and proven to be complete in **PPAD**.

Definition 5 (3-DIMENSIONAL BROUWER). *The input is a pair $(C, 0^n)$ where C is a circuit with $3n$ input bits and 6 output bits Δ_1^+, Δ_1^-, Δ_2^+, Δ_2^-, Δ_3^+ and Δ_3^-. It specifies a function ϕ of a very special form. For each $\mathbf{r} \in A^n$, we define a cubelet in $[0, 1]^3$ as*

$$\{\, \mathbf{q} \in \mathbb{R}^3 \;\big|\; r_i 2^{-n} \le q_i \le (r_i + 1)2^{-n},\ \forall\, 1 \le i \le 3 \,\}$$

and use $\mathbf{c}_\mathbf{r}$ to denote its center point. Function ϕ is defined on the set of 2^{3n} centers. For every center $\mathbf{c}_\mathbf{r}$, where $\mathbf{r} \in A^n$, $\phi(\mathbf{c}_\mathbf{r}) \in \{\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3, \mathbf{e}^4\} \subset \mathbb{Z}^3$ and is specified by the output bits of $C(\mathbf{r})$ as follows:

- $\Delta_1^+ = 1$, *other five bits are* 0*: $\phi(\mathbf{c}_\mathbf{r}) = \mathbf{e}^1 = (1, 0, 0)$;*
- $\Delta_2^+ = 1$, *other five bits are* 0*: $\phi(\mathbf{c}_\mathbf{r}) = \mathbf{e}^2 = (0, 1, 0)$;*
- $\Delta_3^+ = 1$, *other five bits are* 0*: $\phi(\mathbf{c}_\mathbf{r}) = \mathbf{e}^3 = (0, 0, 1)$;*
- $\Delta_1^- = \Delta_2^- = \Delta_3^- = 1$, *and other three bits are* 0*: $\phi(\mathbf{c}_\mathbf{r}) = \mathbf{e}^4 = (-1, -1, -1)$.*

For all $\mathbf{r} \in A^n$, the six output bits of $C(\mathbf{r})$ are guaranteed to fall into one of the four cases above. C also satisfies the following boundary condition:

> *For each $\mathbf{r} \in B^n$, if there exists $1 \le i \le 3$ such that $r_i = 0$, letting l be the largest index such that $r_l = 0$, then $\phi(\mathbf{c}_\mathbf{r}) = \mathbf{e}^l$; otherwise, $\phi(\mathbf{c}_\mathbf{r}) = \mathbf{e}^4$.*

*A vertex of a cubelet is said to be panchromatic if, among the eight cubelets adjacent to it, there're four that have all four vectors $\mathbf{e}^1$, $\mathbf{e}^2$, $\mathbf{e}^3$ and $\mathbf{e}^4$. The output of problem 3-*DIMENSIONAL BROUWER *is a panchromatic vertex of ϕ.*

Notation 1. *We use Size$[C]$ to denote the number of logic gates plus the number of input and output variables in C, and $|\,C\,|$ to denote the number of bits used in its binary encoding. Clearly, we always have $3n <$ Size$[C] < |\,C\,|$.*

3 The Reduction

In this section, we will describe a reduction from problem 3-DIMENSIONAL BROUWER to 2-NASH. In section 4, we will prove the correctness of the reduction.

3.1 Sketch of the Reduction

Let pair $U = (C, 0^n)$ be an input instance of problem 3-DIMENSIONAL BROUWER, and m be the smallest integer such that $K = 2^m > |\,(C, 0^n)\,|^2$ where $|\,(C, 0^n)\,| = |C| + n$. We will construct a two-player game $\mathcal{G}^U$ in which both players have $2K$ strategies. It has the following property:

Property 1. *Given any ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game $\mathcal{G}^U$, where $\epsilon = 2^{-(m+4n)}$, a panchromatic vertex of function ϕ can be computed in polynomial time.*

To build $\mathcal{G}^U = (\mathbf{A}^U, \mathbf{B}^U)$, we insert a (valid) collection of gadgets $\mathcal{S}^U = \{T_1, T_2, ..., T_l\}$ into a prototype game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$, where $l \le K$. Formally, from each gadget $T_i \in \mathcal{S}^U$, one can generate (Figure 2) a pair of $2K \times 2K$ matrices $(\mathbf{L}[T_i], \mathbf{R}[T_i])$. Then

$$\mathbf{A}^U = \mathbf{A}^* + \sum_{T \in \mathcal{S}^U} \mathbf{L}[T] \quad \text{and} \quad \mathbf{B}^U = \mathbf{B}^* + \sum_{T \in \mathcal{S}^U} \mathbf{R}[T].$$

We will prove that, every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game $\mathcal{G}^U$ must satisfy a set of $l+1$ constraints. First, we prove that $0 \le \mathbf{A}^U - \mathbf{A}^*, \mathbf{B}^U - \mathbf{B}^* \le 1$, which implies a constraint $\mathcal{P}$ on vectors $\mathbf{x}$ and $\mathbf{y}$ (Lemma 1). Secondly, every gadget T_i in $\mathcal{S}^U$ requires $(\mathbf{x}, \mathbf{y})$ to satisfy a constraint $\mathcal{P}[T_i]$. These constraints $\{\mathcal{P}, \mathcal{P}[T_1], \mathcal{P}[T_2], ..., \mathcal{P}[T_l]\}$ on $(\mathbf{x}, \mathbf{y})$ allow us to prove Property 1, and thus, the correctness of the reduction.

3.2 Nodes, Values and Capacities

Arbitrarily choose two sets V_A and V_I such that $|V_A| = |V_I| = K$. Here we don't care what the elements actually are. Elements in V_A are called arithmetic nodes, and elements in V_I are called internal nodes.

To clarify our presentation, we always use v to denote a node in V_A and w to denote a node in V_I. Furthermore, we arbitrarily pick two $1-1$ correspondences $\mathcal{C}_A$ and $\mathcal{C}_I$. $\mathcal{C}_A$ maps V_A to $[K] = \{1, 2, ..., K\}$ and $\mathcal{C}_I$ maps V_I to $[K]$.

Clearly, one choice is that $V_A = V_I = [K]$ and $\mathcal{C}_A = \mathcal{C}_I = I$, where I is the identity map from $[K]$ to itself.

Let $(\mathbf{x}, \mathbf{y} \in \mathbb{P}^{2K})$ be a profile of mixed strategies. For every arithmetic node $v \in V_A$, we let $\mathbf{x}[v] = x_{2k-1}$ and $\mathbf{x}_C[v] = x_{2k-1} + x_{2k}$, where $k = \mathcal{C}_A(v)$. $\mathbf{x}[v]$ and $\mathbf{x}_C[v]$ are called the value and capacity of node v in $(\mathbf{x}, \mathbf{y})$, respectively. Similarly, for every $w \in V_I$, we let $\mathbf{y}[w] = y_{2t-1}$ and $\mathbf{y}_C[w] = y_{2t-1} + y_{2t}$, where $t = \mathcal{C}_I(w)$. $\mathbf{y}[w]$ and $\mathbf{y}_C[w]$ are called the value and capacity of w in $(\mathbf{x}, \mathbf{y})$ respectively.

3.3 The Prototype Game $\mathbf{G}^*$

$\mathcal{G}^U$ is obtained by perturbing the payoffs of a zero-sum game $\mathcal{G}^* = (\mathbf{A}^*, \mathbf{B}^*)$, which is a variation of the matching pennies game [9] with an exponentially large payoff parameter $M = 2^{4(m+n)+1}$. $\mathbf{A}^*$ is a $K \times K$ block diagonal matrix where each diagonal block is a 2×2 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 $\mathbf{B}^* = -\mathbf{A}^*$. Note that, at an exact Nash equilibrium of $\mathcal{G}^*$, the sum of probabilities at strategies $2k-1$ and $2k$ is a constant for all k, for both players. In addition, we define a class $\mathcal{L}$ of two-player games, and prove that all the ϵ-Nash equilibria of games in $\mathcal{L}$ satisfy a constraint $\mathcal{P}$. All the games we will construct belong to this class.

Definition 6. *A $2K \times 2K$ two-player game $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ belongs to class $\mathcal{L}$ if*

$$0 \le a_{i,j} - a^*_{i,j}, b_{i,j} - b^*_{i,j} \le 1, \ \forall\, 1 \le i, j \le 2K.$$

Notation 2. *By $x = y \pm \epsilon$ where $\epsilon > 0$, we mean $y - \epsilon \le x \le y + \epsilon$. Let $\mathbf{x}, \mathbf{y}$ be two points in $\mathbb{R}^3$, then $\mathbf{x} = \mathbf{y} \pm \epsilon$ means $x_i = y_i \pm \epsilon$, for all $i : 1 \le i \le 3$.*

Lemma 1 (Constraint $\mathcal{P}$). *Let $(\mathbf{x}, \mathbf{y})$ be an f-Nash equilibrium of $\mathcal{G} = (\mathbf{A}, \mathbf{B}) \in \mathcal{L}$ with $f \le 1$. Then for all $v \in V_A$ and $w \in V_I$, $\mathbf{x}_C[v] = 1/K \pm \epsilon$ and $\mathbf{y}_C[w] = 1/K \pm \epsilon$, where $\epsilon = 2^{-(m+4n)}$.*

Proof. We let $\mathbf{A}_i$ denote the i^{th} row vector of $\mathbf{A}$, and $\mathbf{B}_j$ denote the j^{th} column vector of $\mathbf{B}$.

By the definition of class $\mathcal{L}$, for each $1 \le t \le K$, the $2t-1^{st}$ and $2t^{th}$ entries of rows $\mathbf{A}_{2t-1}$ and $\mathbf{A}_{2t}$ are within $[M, M+1]$ and all other entries in them are within $[0, 1]$. Let $w = \mathcal{C}_I^{-1}(t) \in V_I$, then for any $\mathbf{y} \in \mathbb{P}^{2K}$,

$$M\mathbf{y}_C[w] \le \mathbf{A}_{2t-1}\mathbf{y}, \mathbf{A}_{2t}\mathbf{y} \le M\mathbf{y}_C[w] + 1. \qquad (1)$$

Similarly, for each $1 \le k \le K$, letting $v = \mathcal{C}_A^{-1}(k)$, then

$$-M\mathbf{x}_C[v] \le \mathbf{x}^T\mathbf{B}_{2k-1}, \mathbf{x}^T\mathbf{B}_{2k} \le -M\mathbf{x}_C[v] + 1. \qquad (2)$$

Now suppose $(\mathbf{x}, \mathbf{y})$ is an f-Nash equilibrium of $\mathcal{G}$ with $f \le 1$. We first prove that for each pair of v and w with $\mathcal{C}_A(v) = \mathcal{C}_I(w)$, say they equal to l, if $\mathbf{y}_C[w] = 0$ then $\mathbf{x}_C[v] = 0$. First note that $\mathbf{y}_C[w] = 0$ implies there exists a node $w' \in V_I$ with capacity $\mathbf{y}_C[w'] > 1/K$. Suppose $\mathcal{C}_I(w') = k \ne l$. By Inequality (1),

$$\mathbf{A}_{2k}\mathbf{y} - \max(\mathbf{A}_{2l}\mathbf{y}, \mathbf{A}_{2l-1}\mathbf{y}) > M/K - 1 > 1 \ge f.$$

Since $(\mathbf{x}, \mathbf{y})$ is an f-Nash equilibrium of game $\mathcal{G}$, we have $x_{2l} = x_{2l-1} = 0$, and thus, $\mathbf{x}_C[v] = 0$.

Next we show that $\mathbf{x}_C[v] = 1/K \pm \epsilon$ for all $v \in V_A$. To derive a contradiction, we assume that this statement is not true. Then there exist $v, v' \in V_A$ such that $\mathbf{x}_C[v] - \mathbf{x}_C[v'] > \epsilon$. Let $k = \mathcal{C}_A(v)$ and $k' = \mathcal{C}_A(v')$. By Inequality (2),

$$\mathbf{x}^T\mathbf{B}_{2k'} - \max\left(\mathbf{x}^T\mathbf{B}_{2k}, \mathbf{x}^T\mathbf{B}_{2k-1}\right) > M\epsilon - 1 \ge f.$$

This implies $\mathbf{y}_C[w] = 0$ for node $w \in V_I$ with $\mathcal{C}_I(w) = k$, and in turn implies $\mathbf{x}_C[v] = 0$, which contradicts with our assumption that $\mathbf{x}_C[v] > \mathbf{x}_C[v'] + \epsilon > 0$.

We can similarly prove that $\mathbf{y}_C[w] = 1/K \pm \epsilon$, for all nodes $w \in V_I$. □

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

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

3.4 Gadgets

First, we define a function named BUILDGAME. Given a collection of gadgets $\mathcal{S}$, it builds a two-player game $\mathcal{G} = (\mathbf{A}, \mathbf{B}) = \text{BUILDGAME}(\mathcal{S})$ as follows:

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

The construction of $\mathbf{L}$ and $\mathbf{R}$ is presented in Figure 2.

Formally, a gadget $T = (G, v_1, v_2, v, c, w)$ is a 6-tuple. Here $G \in \{G_\zeta, G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\wedge, G_\vee, G_\neg\}$ is the type of the gadget. We totally need nine types of gadgets. Each of them implements an arithmetic or logic constraint $\mathcal{P}[T]$, which requires the values of nodes v, v_1 and v_2 to satisfy certain functional relationship. The requirements for logic gadgets will hold exactly and the requirements for arithmetic ones will hold approximately. Their functionalities are similar to those proposed in [9] but there are subtle differences in the construction. For example, we don't have a simple multiplication gadget but a gadget for multiplication with a predefined constant ($G_{\times\zeta}$).

In gadget T, $v_1 \in V_A \cup \{\textit{nil}\}$ and $v_2 \in V_A \cup \{\textit{nil}\}$ are the first and second input nodes, respectively. $v \in V_A$ is the output node, and $w \in V_I$ is the internal node. Parameter $c \in \mathbb{R} \cup \{\textit{nil}\}$ is only used in G_ζ and $G_{\times\zeta}$ gadgets.

Every gadget has an output node $v \in V_A$ and an internal node $w \in V_I$, but different types of gadgets may have different number $(0,1$ or $2)$ of input nodes: a G_ζ gadget has no input node, so when type $G = G_\zeta$, $v_1 = v_2 = \textit{nil}$; $G_=$, $G_{\times\zeta}$ and $G_\neg$ gadgets have one input node, so when $G \in \{G_=, G_{\times\zeta}, G_\neg\}$, we have $v_1 \in V_A$ and $v_2 = \textit{nil}$; when $G \in \{G_+, G_-, G_<, G_\wedge, G_\vee\}$, $v_1 \neq v_2 \in V_A$.

Parameter c is only used in G_ζ and $G_{\times\zeta}$ gadgets: when type $G = G_\zeta$, we have $c \in \mathbb{R}$ and $0 \leq c \leq 1/K - \epsilon$; when $G = G_{\times\zeta}$, $0 \leq c \leq 1$; otherwise, $c = \textit{nil}$.

Definition 7 (Valid Collection). *Let $\mathcal{S}$ be a collection of gadgets. $\mathcal{S}$ is said to be* valid *if for every two gadgets T and T' in $\mathcal{S}$, where $T = (G, v_1, v_2, v, c, w)$ and $T' = (G', v_1', v_2', v', c', w')$, $v \neq v'$ and $w \neq w'$.*

We will prove that if $\mathcal{S}$ is a *valid* collection, then every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G} = \text{BUILDGAME}(\mathcal{S})$ must satisfy a set of $|\mathcal{S}| + 1$ constraints $\{\mathcal{P}\} \cup \{\mathcal{P}[T], T \in \mathcal{S}\}$. More exactly, let $T = (G, v_1, v_2, v, c, w)$, then $\mathcal{P}[T]$ requires the values of nodes v_1, v_2 and v (that is, $\mathbf{x}[v], \mathbf{x}[v_1]$ and $\mathbf{x}[v_2]$) to approximately satisfy certain arithmetic or logic relationship, which depends on the type G of T. For example, if $G = G_+$, then

$$\mathcal{P}[T] = [\, \mathbf{x}[v] = \min(\mathbf{x}[v_1] + \mathbf{x}[v_2], \mathbf{x}_C[v]) \pm \epsilon \,].$$

The nine types of constraints are summarized in Figure 1.

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

Figure 1. Constraint P[T]

Remark 1. *Notice that, no matter what G is, the constraint $\mathcal{P}[T]$ has nothing to do with the value of its internal node.*

Notation 3. *For a mixed strategy profile $(\mathbf{x}, \mathbf{y})$, $\mathbf{x}[v] =_B 1$ means $\mathbf{x}[v] = \mathbf{x}_C[v]$, and $\mathbf{x}[v] =_B 0$ means $\mathbf{x}[v] = 0$.*

Among the nine types of gadgets, $G_\wedge$, $G_\vee$ and $G_\neg$ are logic gadgets. They will be used to simulate the logic gates in C. Associated with $(\mathbf{x}, \mathbf{y})$, the value of $v \in V_A$ represents Boolean 1 ($\mathbf{x}[v] =_B 1$) if $\mathbf{x}[v] = \mathbf{x}_C[v]$; it encodes Boolean 0 ($\mathbf{x}[v] =_B 0$) if $\mathbf{x}[v] = 0$.

From Figure 1, the logic constraints implemented by the three logic gadgets are effective only when the values of their input nodes are representations of binary bits.

Formally, we have the following theorem, with proof in Section 3.5.

Theorem 1. *Let $\mathcal{S}$ be a valid collection of gadgets, game $\mathcal{G} = \text{BUILDGAME}(\mathcal{S})$, and $(\mathbf{x}, \mathbf{y})$ be any ϵ-Nash equilibrium of $\mathcal{G}$. Then for every $T \in \mathcal{S}$, constraint $\mathcal{P}[T]$ is satisfied by $(\mathbf{x}, \mathbf{y})$.*

Next, we prove that pair $(\mathbf{x}, \mathbf{y})$ also satisfies constraint $\mathcal{P}$. From Lemma 1, it's only necessary to prove that $\mathcal{G} \in \mathcal{L}$. For any gadget T, $\mathbf{L}[T]$ and $\mathbf{R}[T]$ are generated by the algorithm in Figure 2. They have the following property:

Property 2. *Let $T = (G, v_1, v_2, v, c, w)$, $\mathbf{L}[T] = (L_{i,j})$, and $\mathbf{R}[T] = (R_{i,j})$. Let $k = \mathcal{C}_A(v)$ and $t = \mathcal{C}_I(v)$. Then*

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

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

Set $\mathbf{L}[T] = (L_{i,j}) = \mathbf{R}[T] = (R_{i,j}) = 0$
$k = \mathcal{C}_A(v)$, $k_1 = \mathcal{C}_A(v_1)$, $k_2 = \mathcal{C}_A(v_2)$, and $t = \mathcal{C}_I(w)$

G_+:	$L_{2k-1,2t-1} = L_{2k,2t} = 1$ $R_{2k_1-1,2t-1} = R_{2k_2-1,2t-1} = R_{2k-1,2t} = 1$
G_ζ:	$L_{2k-1,2t} = L_{2k,2t-1}1$ $R_{2k-1,2t-1} = 1,\ R_{i,2t} = c, \forall\, 1 \le i \le 2K$
$G_{\times\zeta}$:	$L_{2k-1,2t-1} = L_{2k,2t} = 1$ $R_{2k_1-1,2t-1} = c,\ R_{2k-1,2t} = 1$
$G_=$:	$L_{2k-1,2t-1} = L_{2k,2t} = 1$ $R_{2k_1-1,2t-1} = R_{2k-1,2t} = 1$
G_-:	$L_{2k-1,2t-1} = L_{2k,2t} = 1$ $R_{2k_1-1,2t-1} = R_{2k_2-1,2t} = R_{2k-1,2t} = 1$
$G_<$:	$L_{2k-1,2t} = L_{2k,2t-1} = 1$ $R_{2k_1-1,2t-1} = R_{2k_2-1,2t} = 1$
$G_\vee$:	$L_{2k-1,2t-1} = L_{2k,2t} = R_{2k_1-1,2t-1} = 1$ $R_{2k_2-1,2t-1} = 1,\ R_{i,2t} = 1/(2K), \forall\, 1 \le i \le 2K$
$G_\wedge$:	$L_{2k-1,2t-1} = L_{2k,2t} = R_{2k_1-1,2t-1} = 1$ $R_{2k_2-1,2t-1} = 1,\ R_{i,2t} = 3/(2K), \forall\, 1 \le i \le 2K$
$G_\neg$:	$L_{2k-1,2t} = L_{2k,2t-1} = 1$ $R_{2k_1-1,2t-1} = R_{2k_1,2t} = 1$

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

So $(\mathbf{L}[T], \mathbf{R}[T])$ can be viewed as a perturbation to the prototype game $\mathcal{G}^*$. Lemma 2 below follows directly from Definition 7 and Property 2.

Lemma 2. *If $\mathcal{S}$ is valid, then $\mathcal{G} = \textsc{BuildGame}(\mathcal{S}) \in \mathcal{L}$.*

As a corollary of Lemma 1, every ϵ-Nash equilibrium of game $\mathcal{G}$ satisfies constraint $\mathcal{P}$.

3.5 Proof of Theorem 1

Theorem 1 is a direct corollary of nine propositions, one for each type of gadgets. Below we only include the proof for the property of G_+ gadgets to illustrate how such properties are established.

Proposition 1 (Gadget G_+). *Let gadget $T = (G_+, v_1, v_2, v, nil, w)$. Let $k = \mathcal{C}_A(v)$ and $t = \mathcal{C}_I(w)$.*

We let $\mathbf{A}^*_i$ and $\mathbf{L}_i$ denote the i^{th} row vectors of $\mathbf{A}^*$ and $\mathbf{L}[T]$ respectively, $\mathbf{B}^*_j$ and $\mathbf{R}_j$ denote the j^{th} column vectors of $\mathbf{B}^*$ and $\mathbf{R}[T]$, respectively.

If $\mathcal{G} = (\mathbf{A}, \mathbf{B})$ is a two-player game in $\mathcal{L}$, and satisfies

$$\mathbf{A}_{2k-1} = \mathbf{A}^*_{2k-1} + \mathbf{L}_{2k-1},\ \mathbf{A}_{2k} = \mathbf{A}^*_{2k} + \mathbf{L}_{2k}; \quad (3)$$
$$\mathbf{B}_{2t-1} = \mathbf{B}^*_{2t-1} + \mathbf{R}_{2t-1},\ \mathbf{B}_{2t} = \mathbf{B}^*_{2t} + \mathbf{R}_{2t}, \quad (4)$$

then every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}$ satisfies $\mathcal{P}[T]$.

Proof. Since $\mathbf{A}^*_{2k-1} = \mathbf{A}^*_{2k}$ and $\mathbf{B}^*_{2t-1} = \mathbf{B}^*_{2t}$, from (3), (4) and Figure 2, we have

$$\mathbf{x}^T\mathbf{B}_{2t-1} - \mathbf{x}^T\mathbf{B}_{2t} = \mathbf{x}[v_1] + \mathbf{x}[v_2] - \mathbf{x}[v] \text{ and}$$
$$\mathbf{A}_{2k-1}\mathbf{y} - \mathbf{A}_{2k}\mathbf{y} = \mathbf{y}[w] - (\mathbf{y}_C[w] - \mathbf{y}[w]).$$

If $\mathbf{x}[v] - (\mathbf{x}[v_1] + \mathbf{x}[v_2]) > \epsilon$, then the first equation shows $\mathbf{y}[w] = y_{2t-1} = 0$, since $(\mathbf{x}, \mathbf{y})$ is an ϵ-Nash equilibrium. On the other hand, since $\mathcal{G} \in \mathcal{L}$, $(\mathbf{x}, \mathbf{y})$ satisfies $\mathcal{P}$ and thus, $\mathbf{y}_C[w] = 1/K \pm \epsilon \gg \epsilon$. By the second equation, we have $\mathbf{x}[v] = x_{2k-1} = 0$, which contradicts with our assumption that $\mathbf{x}[v] > \mathbf{x}[v_1] + \mathbf{x}[v_2] + \epsilon > 0$.

If $\mathbf{x}[v] - (\mathbf{x}[v_1] + \mathbf{x}[v_2]) < -\epsilon$, then the first equation implies that $\mathbf{y}[w] = \mathbf{y}_C[w]$. By the second equation, we have $\mathbf{x}[v] = \mathbf{x}_C[v]$. Since $\mathbf{x}_C[v] = \mathbf{x}[v] < \mathbf{x}[v_1] + \mathbf{x}[v_2]$, we have $\mathbf{x}[v] = \mathbf{x}_C[v] > \mathbf{x}_C[v] - \epsilon = \min(\mathbf{x}[v_1] + \mathbf{x}[v_2], \mathbf{x}_C[v]) - \epsilon$ and the proposition is proven. □

Proof of Theorem 1. Let $T = (G, v_1, v_2, v, c, w)$ be one of the gadgets in the valid collection $\mathcal{S}$. We only consider the case that $G = G_+$.

Since $\mathcal{S}$ is valid, we know from Lemma 2 that $\mathcal{G} \in \mathcal{L}$. From Property 2 and Definition 7, none of the rows $\mathbf{A}_{2k-1}$, $\mathbf{A}_{2k}$ and columns $\mathbf{B}_{2t-1}$, $\mathbf{B}_{2t}$ is modified by gadgets in $\mathcal{S}$ except the T above, and $\mathcal{G}$ satisfies both conditions (3) and (4) in Proposition 1. As a result, every ϵ-Nash equilibrium of $\mathcal{G}$ satisfies constraint $\mathcal{P}[T]$. □

3.6 Sampling Network

In this subsection, we build a network of gadgets which will be referred as a *sampling network*. It is a basic component in the construction of $\mathcal{S}^U$ and $\mathcal{G}^U$.

Notation 4. *Let $\mathcal{S}$ be a valid collection of gadgets. A node $v \in V_A$ (or $w \in V_I$) is* unused *in $\mathcal{S}$ if none of the gadgets in $\mathcal{S}$ uses v (or w) as its output node (or internal node).*

Given a valid collection $\mathcal{S}$, we use $\textsc{Unused}[\mathcal{S}]$ to denote the number of unused arithmetic nodes in $\mathcal{S}$.

Let $\mathcal{S}$ be a valid collection, and T be a gadget such that $\mathcal{S} \cup \{T\}$ is sill valid. We will use $\textsc{Insert}(\mathcal{S}, T)$ to denote the insertion of T into $\mathcal{S}$.

Notation 5. *For every $\mathbf{r} \in A^n$, we use $\Delta^+_i(\mathbf{r})$ and $\Delta^-_i(\mathbf{r})$ to denote the output bits Δ^+_i and Δ^-_i of C evaluated at $\mathbf{r}$.*

Definition 8. *A real number $a \in \mathbb{R}^+$ is* poorly-positioned *if there exists an integer $1 \le t \le 2^n - 1$, such that $|\,a - t2^{-n}\,| \le (3n+1)K\epsilon$. Otherwise, a is* well-positioned.

For a well-positioned $a \in \mathbb{R}^+$, we let $\pi(a)$ denote the integer $k : 0 \le k \le 2^n - 1$ such that

$$\left|\,a - (k + 1/2)2^{-n}\,\right| = \min_{0 \le k' \le 2^n - 1} \left|\,a - (k' + 1/2)2^{-n}\,\right|.$$

Since a is well-positioned, k is unique and well defined.

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

Figure 3. Function ExtractBits

First, we combine different types of gadgets to extract bits of integer $\pi(a)$ from a. The technique was first developed in [9].

Let $\mathcal{S}$ be a valid collection with Unused$[\mathcal{S}] \ge 4n+1$. Let v be an arithmetic node, and $v^1, v^2, ..., v^n \in V_A$ be n *unused* nodes in $\mathcal{S}$. We now insert $4n+1$ gadgets into $\mathcal{S}$ by invoking the function ExtractBits$(\mathcal{S}, v, v^1..., v^n)$ in Figure 3. We let $\mathcal{S}'$ denote the collection $\mathcal{S}$ after executing ExtractBits.

Let $\mathcal{G}' =$ BuildGame$(\mathcal{S}')$. We claim that, in every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}'$, if $K\mathbf{x}[v]$ is well-positioned, then the values of nodes $v^1, v^2, ..., v^n$ are all boolean, and $\mathbf{x}[v^i] =_B b_i$, where integer $\pi(K\mathbf{x}[v]) = b_1 b_2 ... b_{n-1} b_n$.

Lemma 3. *In every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of game $\mathcal{G}'$ if $K\mathbf{x}[v]$ is well-positioned, then $\mathbf{x}[v^i] =_B b_i$, $\forall\ 1 \le i \le n$.*

Proof. For each $1 \le k \le n$, let $c_k = \sum_{j=k}^{n} b_j 2^{-(m+j)}$.

We need to consider the following three cases:

1). $2^{-(m+n)} - (3n+1)\epsilon > \mathbf{x}[v] - c_1 > (3n+1)\epsilon$;
2). $\mathbf{x}[v] \le (3n+1)\epsilon$; 3). $\mathbf{x}[v] \ge 1/K - (3n+1)\epsilon$.

Here we only give a proof for the first case. The other two cases can be proved easily.

We prove by induction that, for all $1 \le k \le n+1$,

$$2^{-(m+n)} - 3l\epsilon > \mathbf{x}[v_k] - c_k > 3l\epsilon$$

and $\mathbf{x}[v^{k-1}] =_B b_{k-1}$, where $l = n+1-k$.

The basis is trivial, since $\mathbf{x}[v_1] = \mathbf{x}[v] \pm \epsilon$. Now assume the statement is true for $k < n+1$. Assume $b_k = 1$, then by the inductive hypothesis, $\mathbf{x}[v_k] > c_k + 3(n+1-k)\epsilon \ge 2^{-(m+k)} + 3(n+1-k)\epsilon$. On the other hand, the gadget in line 5 requires $\mathbf{x}[v_{k1}]$ to be $2^{-(m+k)} \pm \epsilon$. As a result, we have $\mathbf{x}[v^k] =_B 1$, and $\mathbf{x}[v^k] = \mathbf{x}_C[v^k] = 1/K \pm \epsilon$. By the gadget in line 7, we have

$$\mathbf{x}[v_{k2}] = 2^{-k}\mathbf{x}[v^k] \pm \epsilon = 2^{-(m+k)} \pm 2\epsilon.$$

Finally, the gadget in line 8 implies

$$\mathbf{x}[v_{k+1}] = (\mathbf{x}[v_k] - (2^{-(m+k)} \pm 2\epsilon)) \pm \epsilon,$$

and thus, $\mathbf{x}[v_{k+1}] - c_{k+1} = \mathbf{x}[v_k] - c_k \pm 3\epsilon$. The case for $b_k = 0$ can be proved similarly. □

Based on function ExtractBits, we now construct a larger network of gadgets to simulate the evaluation of C.

Definition 9. $\mathbf{q} \in \mathbb{R}^3_+$ *is a* well-positioned *point if none of its components is* poorly-positioned. *Otherwise, we refer to it as a* poorly-positioned *point.*

For a well-positioned point $\mathbf{q}$, we let $\pi(\mathbf{q})$ denote the point $\mathbf{r} \in A^n$ such that $r_i = \pi(q_i)$, for all $i : 1 \le i \le 3$.

Let $\mathcal{S}$ be a valid collection with Unused$[\mathcal{S}] \ge 3(4n+1) +$ Size$[C]$. Let $\{v_i\}_{1 \le i \le 3} \subset V_A$ and $\{v_i^+, v_i^-\}_{1 \le i \le 3}$ be six *unused* nodes in V_A.

We now add a network of $3(4n+1) +$ Size$[C]$ gadgets to connect $\{v_i\}_{1 \le i \le 3}$ with $\{v_i^+, v_i^-\}_{1 \le i \le 3}$. Let $\mathcal{S}'$ be the collection of gadgets $\mathcal{S}$ after inserting the network into $\mathcal{S}$, and game $\mathcal{G}' =$ BuildGame$(\mathcal{S}')$.

Given an ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}'$, we view the values of $\{v_i\}_{1 \le i \le 3}$ as a point $\mathbf{q} \in \mathbb{R}^3$, with $q_i = K\mathbf{x}[v_i]$, for all $1 \le i \le 3$. We claim that, in every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}'$, if $\mathbf{q}$ is well-positioned, then

$$\mathbf{x}[v_i^+] =_B \Delta_i^+(\mathbf{r}) \text{ and } \mathbf{x}[v_i^-] =_B \Delta_i^-(\mathbf{r}),\ \ \forall\ 1 \le i \le 3,$$

where $\mathbf{r} = \pi(\mathbf{q})$. The network is divided into two parts.

Part 1. Pick $3n$ unused nodes $\{v_{ij} \in V_A\}_{1 \le i \le 3, 1 \le j \le n}$ of collection $\mathcal{S}$. Call ExtractBits$(\mathcal{S}, v_i, v_{i1}, v_{i2}...v_{in})$ for each $i : 1 \le i \le 3$.

Lemma 3 shows that, if $(\mathbf{x}, \mathbf{y})$ is an ϵ-Nash equilibrium of $\mathcal{G}'$ and $\mathbf{q}$ is well-positioned, letting $\mathbf{r} = \pi(\mathbf{q}) \in A^n$, then $\mathbf{x}[v_{ij}] =_B b_{ij}$ for all $1 \le i \le 3$ and $1 \le j \le n$, where $b_{i1} b_{i2} ... b_{in}$ is the binary representation of integer r_i.

Part 2. We view the $3n$ nodes $\{v_{ij}\}_{1 \le i \le 3, 1 \le j \le n}$ as the encoding of the $3n$ input bits of circuit C and insert Size$[C]$ logic gadgets $G_\wedge$, $G_\vee$, $G_\neg$ to simulate the evaluation of C on these bits, one for each logic gate in C. The six output bits are then placed in $\{v_i^+, v_i^-\}$. The simulation works perfectly, since we assumed $\mathbf{q}$ is well-positioned and thus, the values of $\{v_{ij}\}$ are all representations of boolean bits.

Lemma 4. *In every ϵ-Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of $\mathcal{G}'$, if $\mathbf{q}$ is well-positioned, letting $\mathbf{r} = \pi(\mathbf{q})$, then $\mathbf{x}[v_i^+] =_B \Delta_i^+(\mathbf{r})$ and $\mathbf{x}[v_i^-] =_B \Delta_i^-(\mathbf{r})$, for all $i : 1 \le i \le 3$.*

We will refer to this network as a *sampling network*. It works correctly only if $\mathbf{q}$ is well-positioned, and when $\mathbf{q}$ is not, the values of $\{\, v_i^+, v_i^- \,\}_{1 \le i \le 3}$ could be arbitrary. But even in the latter case, we know that $0 \le \mathbf{x}[v_i^+], \mathbf{x}[v_i^-] \le 1/K + \epsilon$, because $\mathcal{S}'$ is valid and $(\mathbf{x}, \mathbf{y})$ should satisfy $\mathcal{P}$.

3.7 Construction of S^{U} and G^{U}

The idea behind the construction of $\mathcal{S}^U$ and $\mathcal{G}^U$ is similar to the one in [9]. There are three distinguished nodes $\{v_i\}_{1\le i\le 3}$ in V_A. Their values are viewed as an encoding of a point $\mathbf{q}\in\mathbb{R}^3$ where $q_i = K\mathbf{x}[v_i]$ for all $1\le i\le 3$. Starting from $\mathcal{S}^U=\emptyset$, we first add sampling networks to evaluate C on 41^3 points around $\mathbf{q}$, and calculate the average of all the vectors on these sampling points. More gadgets will be inserted into $\mathcal{S}^U$ to make sure in every ϵ-Nash equilibrium of $\mathcal{G}^U$, this average vector is close to zero. Such a property implies the existence of a panchromatic vertex near $\mathbf{q}$, which can be located very efficiently.

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

Part 1. Let $\{v_i^k\}_{1\le i\le 3,-20\le k\le 20}$ be $3\cdot 41$ unused nodes in V_A. For all i and k, by inserting G_ζ, G_- and G_+ gadgets, we make sure in any ϵ-Nash equilibrium $(\mathbf{x},\mathbf{y})$ of $\mathcal{G}^U$,

$$\mathbf{x}[v_i^k] = \min(\mathbf{x}[v_i]+k\alpha', \mathbf{x}_C[v_i^k]) \pm 4\epsilon, \text{ if } k\ge 0,$$
$$\mathbf{x}[v_i^k] = \max(\mathbf{x}[v_i]+k\alpha', 0) \pm 4\epsilon, \text{ if } k< 0,$$

where $\alpha'=\alpha/K$ and $\alpha=2^{-2n}$.

Part 2. Let $I=\{-20,-19,...,20\}^3$. Pick $6\cdot 41^3$ unused nodes $\{v_i^{\mathbf{t}+}, v_i^{\mathbf{t}-}\}_{1\le i\le 3,\mathbf{t}\in I}$ in V_A. For every triple $\mathbf{t}\in I$, insert a sampling network into $\mathcal{S}^U$, to connect nodes $\{v_1^{t_1}, v_2^{t_2}, v_3^{t_3}\}$ with $\{v_i^{\mathbf{t}+}, v_i^{\mathbf{t}-}\}_{1\le i\le 3}$.

Part 3. Let $\{v_i^+, v_i^-\}_{1\le i\le 3}$ be 6 unused nodes in V_A. By inserting $G_{\times\zeta}$ and G_+ gadgets into $\mathcal{S}^U$, we make sure in every ϵ-Nash equilibrium $(\mathbf{x},\mathbf{y})$ of $\mathcal{G}^U$, for all $1\le i\le 3$,

$$\mathbf{x}[v_i^+] = \Big(\sum_{\mathbf{t}\in I}\frac{\alpha}{41^3}\,\mathbf{x}[v_i^{\mathbf{t}+}]\Big) \pm 3\cdot 41^3\epsilon,$$
$$\mathbf{x}[v_i^-] = \Big(\sum_{\mathbf{t}\in I}\frac{\alpha}{41^3}\,\mathbf{x}[v_i^{\mathbf{t}-}]\Big) \pm 3\cdot 41^3\epsilon.$$

Part 4. For each $1\le i\le 3$, we pick unused nodes: $v', v''\in V_A$, $w_1,w_2,w_3\in V_I$, and insert the following 3 gadgets into collection $\mathcal{S}^U$:

INSERT $(\mathcal{S}^U, (G_+, v_i, v_i^+, v', nil, w_1))$;
INSERT $(\mathcal{S}^U, (G_-, v', v_i^-, v'', nil, w_2))$;
INSERT $(\mathcal{S}^U, (G_=, v'', nil, v_i, nil, w_3))$.

4 Correctness of the Reduction

First, given any input instance $U=(C,0^n)$ of problem 3-DIMENSIONAL BROUWER, one can build $(\mathcal{G}^U, 0^{m+4n})$ in polynomial time, which is regarded as an input instance of 2-NASH. The only thing left is to prove Property 1 in Section 3.1, which is a corollary of the following theorem.

Theorem 2. *Let $(\mathbf{x},\mathbf{y})$ be an ϵ-Nash equilibrium of game $\mathcal{G}^U$ and $\mathbf{q}\in\mathbb{R}^3$ be the point encoded by nodes $\{v_i\}_{1\le i\le 3}$, where $q_i = K\mathbf{x}[v_i]$ for all $i: 1\le i\le 3$. Then there must exist three integers $1\le m_1,m_2,m_3\le 2^n-1$ such that*

$$(m_i-1)2^{-n}+30\alpha < q_i < (m_i+1)2^{-n}-30\alpha,$$

where $\alpha=2^{-2n}$, for all $i: 1\le i\le 3$.

Furthermore, $(m_1 2^{-n}, m_2 2^{-n}, m_3 2^{-n})$ is a panchromatic vertex of ϕ.

Corollary 1. 2-NASH *and* NASH *are* **PPAD**-*complete.*

One can check that the number of gadgets in set $\mathcal{S}^U$ is $O(|(C,0^n)|)\ll K$, and $\mathcal{S}^U$ is valid. As a result, every ϵ-Nash equilibrium $(\mathbf{x},\mathbf{y})$ of $\mathcal{G}^U$ satisfies a set of $|\mathcal{S}^U|+1$ constraints.

We use these constraints to prove Theorem 2. Proofs of Lemma 5, 6, 7, and 8 can be found in the full version [5]. We start with some notations.

Suppose $(\mathbf{x},\mathbf{y})$ is an ϵ-Nash equilibrium of game $\mathcal{G}^U$. For each $\mathbf{t}\in I$, we use $\mathbf{q}^{\mathbf{t}}$ to denote the point encoded by $\{v_1^{t_1}, v_2^{t_2}, v_3^{t_3}\}$, where $q_i^{\mathbf{t}} = K\mathbf{x}[v_i^{t_i}]$, for all $i: 1\le i\le 3$. Let I_G and I_B denote, respectively, the sets of indices of well-positioned and poorly-positioned points: $I_G=\{\,\mathbf{t}\in I \mid \mathbf{q}^{\mathbf{t}}$ is well-positioned$\}$ and $I_B=\{\mathbf{t}\in I\,|\,\mathbf{q}^{\mathbf{t}}$ is poorly-positioned$\}$. By the following lemma, there cannot be too many poorly-positioned points.

Lemma 5. *In every ϵ-Nash equilibrium $(\mathbf{x},\mathbf{y})$ of $\mathcal{G}^U$, we have $|\,I_B\,|\le 3\cdot 41^2$ and thus, $|\,I_G\,|\ge 38\cdot 41^2$.*

For each $\mathbf{t}\in I_G$, we let point $\mathbf{r}^{\mathbf{t}}=\pi(\mathbf{q}^{\mathbf{t}})\in A^n$ and use $1\le l_{\mathbf{t}}\le 4$ to denote the integer such that $\phi(\mathbf{c}_{\mathbf{r}^{\mathbf{t}}})=\mathbf{e}^{l_{\mathbf{t}}}$. Recall that $\mathbf{e}^1=(1,0,0)$, $\mathbf{e}^2=(0,1,0)$, $\mathbf{e}^3=(0,0,1)$ and $\mathbf{e}^4=(-1,-1,-1)$. For every $\mathbf{t}\in I$, let $\mathbf{z}^{\mathbf{t}}\in\mathbb{R}^3$ denote the vector such that $z_i^{\mathbf{t}}=\mathbf{x}[v_i^{\mathbf{t}+}]-\mathbf{x}[v_i^{\mathbf{t}-}]$, $\forall\, 1\le i\le 3$.

By Lemma 4, we have

Lemma 6. *Let $(\mathbf{x},\mathbf{y})$ be an ϵ-Nash equilibrium of game $\mathcal{G}^U$. Then for every $\mathbf{t}\in I_G$, $\mathbf{z}^{\mathbf{t}}=\mathbf{e}^{l_{\mathbf{t}}}/K\pm\epsilon$.*

Let $\mathbf{z}$ denote the 3-dimensional vector such that $z_i=\mathbf{x}[v_i^+]-\mathbf{x}[v_i^-]$, for every $1\le i\le 3$. In **Part 4** of the construction, we hope to establish $\|\mathbf{z}\|_\infty=O(\epsilon)$, but whether or not this condition holds depends on the values of v_1, v_2 and v_3. Formally, we have the following weaker property.

Lemma 7. *Let $(\mathbf{x},\mathbf{y})$ be an ϵ-Nash equilibrium of game $\mathcal{G}^U$, then for every $i: 1\le i\le 3$, 1). if $\mathbf{x}[v_i]>4\epsilon$, then $z_i>-4\epsilon$; 2). if $\mathbf{x}[v_i]<1/K-2\alpha/K$, then $z_i<4\epsilon$.*

For each $1\le i\le 4$, we use k_i to denote the number of indices $\mathbf{t}$ in I_G such that $l_{\mathbf{t}}=i$. Lemma 8 below was first used in [9].

Lemma 8. *Suppose that for nonnegative integers $k_1 ... k_4$, all three coordinates of $\sum_{i=1}^{4} k_i \mathbf{e}^i$ are smaller in absolute value than $k/5$ where $k = \sum_{i=1}^{4} k_i$. Then all four k_i are positive.*

Finally, we prove Theorem 2.

Proof of Theorem 2. First, we give an analysis on $\mathbf{z}^{\mathbf{t}}$ and $\mathbf{z}$:

$$\begin{aligned}
\mathbf{z} &= \frac{\alpha}{41^3} \sum_{\mathbf{t} \in I} \mathbf{z}^{\mathbf{t}} \pm O(\epsilon) \\
&= \frac{\alpha}{41^3} \sum_{\mathbf{t} \in I_G} \mathbf{z}^{\mathbf{t}} + \frac{\alpha}{41^3} \sum_{\mathbf{t} \in I_B} \mathbf{z}^{\mathbf{t}} \pm O(\epsilon) \\
&= \frac{\alpha}{41^3} \sum_{\mathbf{t} \in I_G} \left(\mathbf{e}^{l_{\mathbf{t}}} \pm \epsilon \right) + \frac{\alpha}{41^3} \sum_{\mathbf{t} \in I_B} \mathbf{z}^{\mathbf{t}} \pm O(\epsilon) \\
&= \frac{\alpha}{41^3 K} \sum_{1 \le i \le 4} k_i \mathbf{e}^i + \frac{\alpha}{41^3} \sum_{\mathbf{t} \in I_B} \mathbf{z}^{\mathbf{t}} \pm O(\epsilon) \\
&= \mathbf{z}^G + \mathbf{z}^B \pm O(\epsilon).
\end{aligned}$$

As $\mathcal{S}^U$ is valid, $(\mathbf{x}, \mathbf{y})$ must satisfy constraint $\mathcal{P}$, and thus, $\mathbf{x}[v] \le 1/K + \epsilon$ for every $v \in V_A$. Since $|I_B| \le 3 \cdot 41^2$,

$$\| \mathbf{z}^B \|_\infty \le \frac{\alpha}{41^3} \cdot 3 \cdot 41^2 \cdot (1/K + \epsilon) = \frac{3\alpha}{41K} + O(\epsilon).$$

Next, we prove that point $\mathbf{q}$ cannot be too close to the boundary of cube $[0,1]^3$. This immediately implies the existence of integers m_1, m_2 and m_3.

Suppose that there is an integer $i : 1 \le i \le 3$ such that $q_i \le 30\alpha$, then for all $\mathbf{t} \in I_G$, $q_i^{\mathbf{t}} \ll 2^{-n}$ and thus, $r_i^{\mathbf{t}} = 0$. The boundary condition on C implies $k_4 = 0$. Let l be the integer such that $k_l = \max_{1 \le j \le 3} k_j \ge |I_G|/3$, then

$$\begin{aligned}
z_l &= z_l^G + z_l^B \pm O(\epsilon) \\
&\ge \frac{\alpha}{41^3 K} \frac{38 \cdot 41^2}{3} - \frac{3\alpha}{41K} - O(\epsilon) \gg 4\epsilon.
\end{aligned}$$

We need to discuss the following two cases:

1. If $\mathbf{x}[v_l] < 1/K - 2\alpha/K$, then we get a contradiction to Lemma 7.
2. If $\mathbf{x}[v_l] \ge 1/K - 2\alpha/K$, then for all $\mathbf{t} \in I_G$, $r_l^{\mathbf{t}} = 2^n - 1 > 0$. From the boundary condition on C, we have $l_{\mathbf{t}} \ne l$ and thus, $k_l = 0$, which contradicts with our assumption on l.

We can similarly show that $q_i < 1 - 30\alpha, \forall\, i : 1 \le i \le 3$.

Finally, we prove that vertex $(m_1 2^{-n}, m_2 2^{-n}, m_3 2^{-n})$ is panchromatic. It's only necessary to show that all four k_i are positive. Since point $\mathbf{q}$ is not close to the boundary of $[0,1]^3$, we have $\| \mathbf{z} \|_\infty \le 4\epsilon$ from Lemma 7, and thus, $\| \mathbf{z}^G \|_\infty \le \| \mathbf{z}^B \|_\infty + O(\epsilon)$. So all the three coordinates of $\sum_{1 \le i \le 4} k_i \mathbf{e}^i$ is smaller that $\sum_{1 \le i \le 4} k_i/5 = |I_G|/5$. By Lemma 5, all four k_i are positive. □

5 Concluding Remarks

Even though many people thought the problem of finding Nash equilibria is hard in general, and it has been proven so for games among three or more players recently, it's not clear whether the two-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, since half a century ago, 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 [19].

The new proof techniques which made the inclusion of problem r-NASH into this class possible, started in Goldberg and Papadimitriou [11], have shown a variety of structures, as exhibited in the hardness proofs of 4-NASH, 3-NASH, and finally the two-player Nash equilibrium problem. They may find their use in other related problems and complexity classes. A recent work of ours with Teng [6] shows that the approach, with significantly deep extra work, can be applied to the issues of polynomial size approximation of Nash equilibria, as well as its smoothed complexity, regarded as a central open problem in smoothed analysis.

6 Acknowledgements

In order to help readers who might be interested in both this paper and [6], the authors of both papers decided to use some common notations and terminologies in the presentation.

Xiaotie Deng would like to acknowledge the support by a travel grant of Department of Computer Science, City University of Hong Kong.

References

[1] L. Brouwer. Über abbildung von mannigfaltigkeiten. *Mathematische Annalen*, 71:97–115, 1910.

[2] X. Chen and X. Deng. 2d-sperner is ppad-complete. In *ICALP 2006*.

[3] X. Chen and X. Deng. On algorithms for discrete and approximate brouwer fixed points. In *STOC 2005*.

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

[5] X. Chen and X. Deng. Settling the complexity of 2-player nash-equilibrium. In *ECCC TR05-140*, 2005.

[6] X. Chen, X. Deng, and S.-H. Teng. Computing nash equilibria: Approximation and smoothed complexity. In *FOCS 2006*.

[7] R. Cottle and G. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra Appl.*, 1:103–125, 1968.

[8] G. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

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

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

[11] P. Goldberg and C. Papadimitriou. Reducibility among equilibrium problems. In *STOC 2006*.

[12] M. Hirsch, C. Papadimitriou, and S. Vavasis. Exponential lower bounds for finding brouwer fixed points. *Journal of Complexity*, 5:379–416, 1989.

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

[14] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of UAI*, 2001.

[15] L. Khachian. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk*, SSSR 244: *English translation in Soviet Math. Dokl. 20*, 1979.

[16] C. Lemke and J. J.T. Howson. Equilibrium points in bimatrix games. *J. Soc. Indust. Appl. Math.*, 12, 1964.

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

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

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

[20] C. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 1994.

[21] R. Savani and B. von Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74:397–429, 2006.

[22] J. von Neumann. zur theorie der gesellshaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[23] A.-C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *FOCS 1977*.