CS 599: Algorithmic Game Theory October 20, 2010

Lecture 9 : PPAD and the Complexity of Equilibrium Computation

Prof. Xi Chen *Scribes: Cheng Lu and Sasank Vijayan*

1 Complexity Class PPAD

1.1 What does PPAD mean?

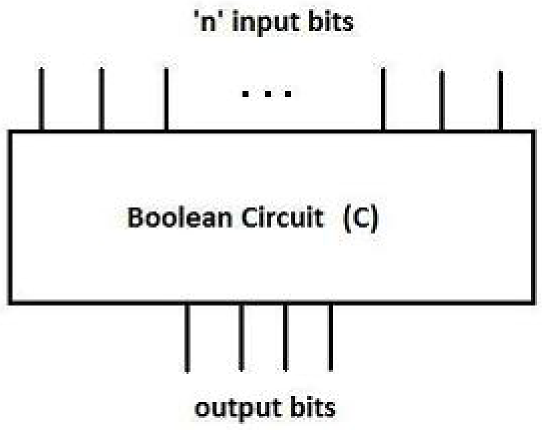
PPAD stands for Polynomial Parity Argument on Directed Graphs. It is a complexity class of search problems defined by Christos Papadimitriou in 1994. In general they are called Total Search Problems, i.e. problems which are guaranteed to have a solution. Sperner's Lemma is an example of one such problem which is guaranteed to have a trichromatic triangle.

To help define the PPAD class we start with a trival problem called **End-of-Line.**

Definition 1. *Given a PPAD graph G and a vertex $v \in G$, where v is a leaf node. Find a node $v^* \neq v$, such that v^* is also a leaf node.*

The above problem definition appears trivial with a linear time algorithm to obtain a solution. However there is a subtulty involved. Which is that the graph G is not actually given but is generated from a boolean circuit.

Boolean Circuit:



In the above Boolean circuit C,$C(u)$ is the computational result of the circuit and u denotes the binary steam. The circuit provides a predecessor and successor pair for all vertices in the graph, i.e. $C(u) = (v, w)$

If u denotes the vertices on the graph then, $|V| = 2^n$ $u \in [0, 2^n - 1]$ and $v, w \in [0, 2^n - 1] \bigcup \{\emptyset\}$

Since $|V| = 2^n$, checking if the vertex is a leaf node cannot take place in linear time and the appearace of a trivial linear solution to the algorithm is misleading.

We now define the PPAD complexity class.

1.2 PPAD definition:

Definition 2. *Any problem A is in PPAD if there is a polynomial time reduction from A to the End-of-Line problem,*

i.e. $A \leq_p B$, where $B \equiv$ End-of-Line Problem

In other words for any problem A in PPAD there exists two ploynomial time computable functions f and g satisfying the following mapping:

$r \longrightarrow f(r)$, where $r \equiv$ input for A & $f(r) \equiv$ input for B

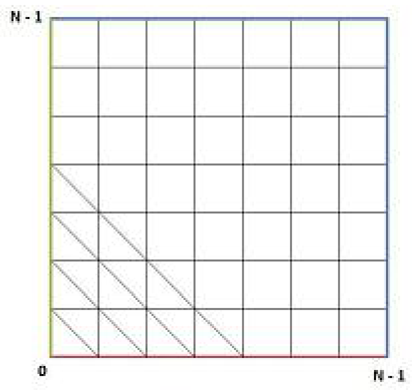
$g(S) \longleftarrow S$, where $S \equiv$ solution for B & $g(S) \equiv$ solution for A

1.3 Why is PPAD interesting?

Lots of game theoretic problems are in PPAD or even PPAD complete. Also PPAD has a close relationship with Fixed Point problems. To some extent we can say that PPAD characterizes a class of Fixed Point problems because of its close connection to Sperner's Lemma.

2 Two Dimensional Sperner's Lemma

2.1 The Problem



Given the following set of colors for the problem $\{Red, Greeen, Blue\}$, an $N \times N$ grid and a function f that maps the points on the grid to the colors,

i.e. $f : [0, N-1] \times [0, N-1] \longrightarrow \{Red, Green, Blue\}$

Lemma 3. *If f satisfies the above boundary condition, $\exists$ a trichromatic triangle in the grid.*

We also have a computation version of the 2D-Sperner's problem:

Definition 4. *Given a boolean circuit C with $2n$ input bits and $C(i,j)$; where $C(i,j) \equiv$ color of (i,j) in the grid, $i,j \in [0, N-1]$ and $|V| = 2^n$, the output of the problem is a tricromatic triangle.*

Note: The circuit only gives color to the interior points of the grid as the boundary colors are already fixed. Also since the domain of the problem is a 2-dimensional grid and hence is in a sense easier to generalize.

2.2 2D-Sperner's Lemma is PPAD-complete

To prove 2D-Sperner's Lemma is in PPAD we reduce the 2-dimensional version of the problem to the End-of-Line problem.

Reduction: Any triangle in the grid gives a vertex in the PPAD graph and an edge is a path going from one triangle to another. Following this representation we obtain a PPAD graph were any leaf of the graph is a trichromatic triangle. Using this reduction sketch we can prove that 2D-Sperner's Lemma is in PPAD.

Theorem 5. *Search problem 2D-Sperner's lemma is PPAD-complete.*

In fact 2-dimensional Sperner's Lemma is a PPAD-complete problem as proved by Xi Chen and Xiaotie Deng in 2006. For exact proof, please refer to [1]. The 3-dimensional version of the problem is also PPAD-complete which was proven earlier by C. Papadimitriou. It is also interesting to note that 1D-Sperner's lemma can be computed in polynomial time.

3 2-Player Nash Equilibrium

We revise the concept of a 2-Player Nash Equilibrium.

A 2-player game is a game where the strategies of the players are represented by a pair of $n \times n$ matrices A and B.

i.e. $G = (A, B)$, where $A, B \in \mathbb{R}^{n \times n}$

Definition 6. *Given that x and y are the strategies of players 1 and 2 respectively, then (x, y) is in Nash Equilibrium if,*

$xAy \geq x'Ay \ \forall x'$ and $xBy \geq xBy' \ \forall y'$

where xAy is the payoff of the 1st player and xBy is the payoff of the 2nd player.

There is also another equivalent version of the problem defined as follows:

Definition 7. *Given that x and y are the strategies of players 1 and 2 respectively, then (x, y) is in Nash Equilibrium if*

$A_i y > A_j y \Rrightarrow x_j = 0$ and $xB_i > xB_j \Rrightarrow y_j = 0$

where A_i, A_j refer to the ith row and jth row of utility matrix A and B_i, B_j refer to the ith row and jth row of utility matrix B.

The intuition behind Definition 7 is that $\forall i, j$, if row $i >$ row j then the 1st player never plays row j and if column $i >$ column j then the 2nd player never plays column j as their strategies.

4 The Complexity of 2-Player Nash Equilibrium

Now we come to the central theorem in the lecture.

Theorem 8. *Finding 2-player Nash equilibrium is PPAD-complete.*

Proof. For the full proof of Theorem 8, please refer to [2]. The following is the sketch of the main ideas.

The proof is divided into 2 steps:

1. Reduce from *2D-Sperner Problem* to *Generalized Circuit Problem.*
2. Reduce from *Generalized Circuit Problem* to *2-Player Nash Equilibrium.*

Step 1: Generalized circuits are generalizations of classical Boolean circuits. A typical generalized circuit is shown in Figure 1.

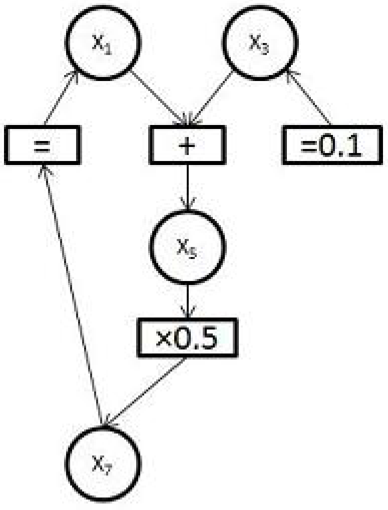


Figure 1: A typical generalized circuit

A generalized circuit contains the following components:

1. Variables: There are $K (K \in \mathbb{N}^+)$ real variables $X_1, X_3, \dots X_{2K-1}$ in the circuit. Indexing the numbers by odd numbers for ease of following reduction. $\forall 1 \le i \le K, X_{2i-1} \in [0, \frac{1}{K}]$.
2. Arithmetic Gates: There are some kinds of binary-to-one arithmetic gates in the circuit. As an example, an addition gate is shown in Figure 2:

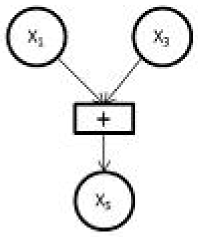


Figure 2: Addition gate

It performs the following operation: $X_5 = \min(X_1 + X_3, \frac{1}{K})$(Note that $X_5 \in [0, \frac{1}{K}]$). It is notable that a gate multiplying a variable with a constant is allowable in the circuit while a gate doing multiplication operation between two variables is not within the specifications of a generalized circuit.

3. Boolean Gates: Boolean gates *AND, OR* and *NOT* are inherited from Boolean circuits but they operate in a slightly different way. For any real variable X_{2i-1}, if $X_{2i-1} = 0$, it is equivalent to Boolean value *FALSE*. If $X_{2i-1} = \frac{1}{K}$, it is equivalent to Boolean value *TRUE*. If $X_{2i-1} \in (0, \frac{1}{K})$, it has value *N/A* or *UNKNOWN* to a Boolean gate. A Boolean gate is only applicable for two Boolean values. More specifically, if the inputs to a Boolean gate all have Boolean values *TRUE/FALSE*, the Boolean gate would run as classical Boolean gate. Otherwise the Boolean gate would output any value $x \in [0, \frac{1}{K}]$.

   It is important to note that every variable X_{2i-1} can be at most the output of one gate. This property is essential to the reduction of Step 2.

An assignment to variables $X_1, X_3, \ldots, X_{2K-1}$ is satisfiable if $\forall 1 \leq i \leq K$, $X_{2i-1} \in [0, \frac{1}{K}]$ and for every gate in the circuit, the value of the output variable matches exactly the value(s) of the input variable(s) to the gate according to the gate operation. Given a legal generalized circuit, since it may contain cycles, the existence of a satisfiable assignment is nontrivial. However, since we have a range to restrict every real variable, like a *cap*, Brouwer's fixed point theorem is applicable in this problem to prove that for every valid generalized circuit, there must be at least one satisfiable assignment. Therefore we can define *Generalized Circuit Problem*:

Definition 9. *Given a generalized circuit, find a satisfiable assignment.*

To complete the reduction in Step 1, let's consider an instance of *2D-Sperner Problem*. Let a simple triangle be any triangle with two right-angle edges of length 1. Consider any simple triangle T in the 2D-Sperner grid. For any point P in T, *color*(P) is the color of P which is defined as the color of the closest integer point (grid point) to P (See Figure 3).

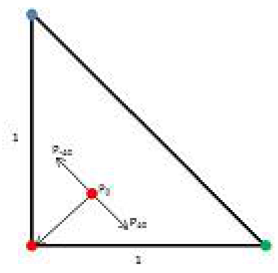


Figure 3: A simple triangle in a 2D-Sperner grid

Based on the color of P, we associate P with a function $vec(P) : \{red, green, blue\} \rightarrow \mathbb{R}^2$ which is defined as follows:

1. $color(P) = red$, $vec(P) = (\frac{1}{K}, 0)$,
2. $color(P) = green$, $vec(P) = (0, \frac{1}{K})$,
3. $color(P) = blue$, $vec(P) = (-\frac{1}{K}, -\frac{1}{K})$.

For a fixed point P, we define a line of 81 points $\{P_{-40}(x_{-40}, y_{-40}), P_{-39}(x_{-39}, y_{-39}), \ldots, P_{-1}(x_{-1}, y_{-1}), P = P_0(x_0, y_0), P_1(x_1, y_1), \ldots, P_{39}(x_{39}, y_{39}), P_{40}(x_{40}, y_{40})\}$ such that $\forall -40 \leq i \leq 40$, $P_i(x_i, y_i) = P_0(x_0, y_0) + (\frac{i}{n}, -\frac{i}{n})$. Here n is a sufficiently large number over 40 or just the input parameter n of 2D-Sperner problem. According to the above definitions, every point $P_i(-40 \leq i \leq 40)$ is associated with a color $color(P_i)$ and a vector $vec(P_i) = (x_i^v, y_i^v)$. We have the following geometry lemma:

Lemma 10. *If $|\bar{x} = \frac{1}{81}\Sigma_{i=-40}^{40} x_i^v| < \frac{1}{3K}$ and $|\bar{y} = \frac{1}{81}\Sigma_{i=-40}^{40} y_i^v| < \frac{1}{3K}$, the simple triangle containing $P = P_0(x_0, y_0)$ is trichromatic.*

The proof of Lemma 10 is based on contradiction. Suppose the simple triangle containing P is not trichromatic, then the triangle can have at most 2 colors. According to the setting of the 81 points, every point must have the color of one of the points of the simple triangle. Therefore there are only at most two types of different vectors in the vector set $\{vec(P_{-40}), vec(P_{-39}), \ldots, vec(P_{-1}), vec(P_0), vec(P_1), \ldots, vec(P_{39}), vec(P_{40})\}$. However, no matter what the two types of vectors are and how many vectors are for each type, averaging over all the vectors by the coordinates would definitely cause the absolute average value of at least one coordinate larger than $\frac{1}{3K}$.

If two coordinates of a vector are any numbers in $[0, \frac{1}{K}]$, we call the vector a *bad* vector. A stronger geometry lemma claims that if there are at most two bad vectors, and if the two average coordinates of the 81 points are both small as above, the simple triangle containing P is still trichromatic. This stronger conclusion ensures the correctness of the reduction in Step 1.

For any instance of 2D-Sperner problem, the corresponding polynomially reduced generalized circuit sketch looks like the following:

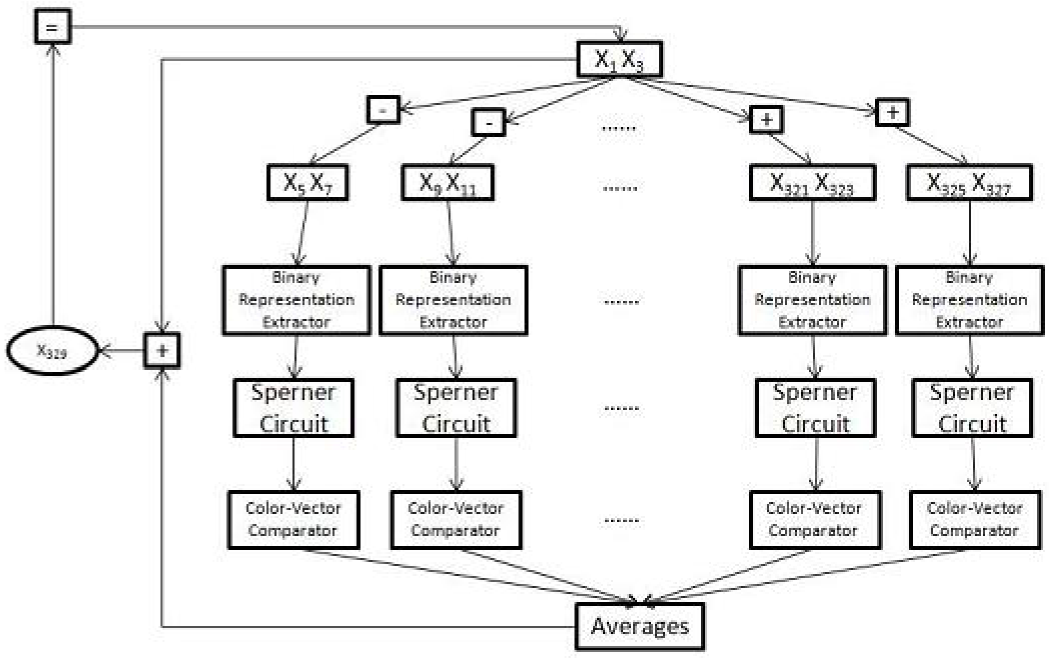


Figure 4: The reduced generalized circuit structure from 2D-Sperner problem

The basic construction idea is as follows. We first use variables X_1 and X_3 to encode point P's position. Then we use *addition* and *subtraction* gates to figure out the encodings of the positions of points $P_i(-40 \le i \le 40)$. Afterwards we use *binary representation extractors* to obtain the positions of the respective closest grid points of P_{-40} to P_{40}, followed by queries to the given Sperner circuits to get respective colors. By mapping the color information to the vectors with the help of *color-vector comparators*, we can finally calculate the averages. Last but not least, we need an *addition backward loop* to update the position of P to make it closer to the desired location.

The color-vector comparators would apply some Boolean gates. As mentioned before, some of the outputs of the Boolean gates may be *UNKNOWN*, which can be any value in $[0, \frac{1}{K}]$. They are certainly *bad* for us. That's why we need 40 pairs of additional points to overcome the fragility of the comparators and guarantee at most two vectors are bad among all the 81 vectors generated in the circuit. Then based on the stronger version of geometry lemma we guarantee that finally the satisfiable assignment to the circuit would be able to map back to a solution to the 2D-Sperner problem. 40 is a constant neither depending on n nor $\frac{1}{3K}$. It is guessed that changing 40 to 10 might work as well.

Step 2: In fact the reduction in Step 2 starts from a more generalized version of generalized circuit problem called *Approximate Generalized Circuit Problem*, which only differs at the specifications of a satisfiable assignment. In approximate generalized circuit problem, an assignment is approximately satisfiable if the values of all the real variables $X_{2i-1}(1 \le i \le K)$ are all within range $[0, \frac{1}{K} + \epsilon]$ and all the output values of gates are within an ϵ bias of the standard outputs($\epsilon > 0$). Take the circuit in Figure 2 for example, if $\min\{X_1 + X_3, \frac{1}{K}\} - \epsilon \le X_5 \le \min\{X_1 + X_3, \frac{1}{K}\} + \epsilon$, the assignment is approximately satisfiable. In our reduction approximate generalized circuit problem is to find an $\epsilon = \frac{1}{2K}$-approximately satisfiable assignment. Most importantly, it can be proved that 2D-Sperner problem can also be reduced to approximate generalized circuit problem.

We start with an instance of approximate generalized circuit C of K real variables $X_1, X_3, \ldots, X_{2K-1}$. We reduce it to an instance of a specific 2-player game $G = (A, B)$, where $A, B \in \mathbb{R}^{2K \times 2K}$ are respective utility matrices of the 2 players. Then for any Nash equilibrium $(x, y) \in \mathbb{R}^{2K+2K}$ of game G, we shall generate a $\frac{1}{2^K}$-approximate solution $(X_1, X_3, \ldots, X_{2K-1})$ to circuit C accordingly and thus complete the proof.

We choose *Matching Pennies Game* as out reduction target. For $\forall K \in \mathbb{N}^+$, we define a *standard* K matching pennies game as $G^*_K = (A^*, B^*), A^*, B^* \in \mathbb{R}^{2K \times 2K}$. For $\forall 1 \leq i \leq K$, $A^*_{2i-1,2i-1} = A^*_{2i-1,2i} = A^*_{2i,2i-1} = A^*_{2i,2i} = M = 2^{K+1}$, the rest parts of A^* are filled with 0s. $B^* = -(A^*)^T$.

For the Nash equilibria of game G^*_K, we have the following lemma:

Lemma 11. *(x, y) is a Nash equilibrium of G^*_K iff $x_{2i-1} + x_{2i} = y_{2i-1} + y_{2i} = \frac{1}{K}(1 \leq i \leq K)$.*

If $x_{2i-1} + x_{2i} = y_{2i-1} + y_{2i} = \frac{1}{K}(1 \leq i \leq K)$, it is easy to verify that (x, y) is a Nash equilibrium. If (x, y) is a Nash equilibrium, by contradiction, w.l.o.g we can assume that $x_1 + x_2 < x_3 + x_4$. The utility of player 2 on column 1 is $-M(x_1 + x_2)$ and the utility of player 2 on column 3 is $-M(x_3 + x_4)$. Hence $-M(x_1 + x_2) > -M(x_3 + x_4)$. According to Definition 7, we have $y_3 = y_4 = 0$. Then if we look at row 3 and row 4 of player 1, since $y_3 = y_4 = 0$, also based on Definition 7 we have $x_3 = x_4 = 0$. However, it contradicts the initial assumption that $x_3 + x_4 > x_1 + x_2 \geq 0$.

We can *perturb* a standard K matching pennies game to an *approximate* K matching pennies game.

Definition 12. *$G'_K = (A', B')$ is an approximate K matching pennies game if $\forall 1 \leq i \leq 2K, 1 \leq j \leq 2K, 0 \leq A'_{ij} - A^*_{ij}, B'_{ij} - B^*_{ij} \leq 1$.*

We have a similar lemma for an approximate K matching pennies game:

Lemma 13. *If $G'_K = (A', B')$ is an approximate K matching pennies game, for any Nash equilibrium (x, y) of G'_K, $\frac{1}{K} - \epsilon \leq x_{2i-1} + x_{2i}, y_{2i-1} + y_{2i} \leq \frac{1}{K} + \epsilon(1 \leq i \leq K)$, where $\epsilon = \frac{1}{2^K}$.*

By contradiction, w.l.o.g, assume $x_3 + x_4 > x_1 + x_2 + \epsilon$. The utility of player 2 on column 1 is $u_{c1} = \Sigma_{j \in [2K]} B'_{1j} x_j = -M(x_1 + x_2) + \Sigma_{j \in [2K]}(B'_{1j} - B^*_{1j})x_j \geq -M(x_1 + x_2)$. The utility of player 2 on column 3 is $u_{c3} = \Sigma_{j \in [2K]} B'_{3j} x_j = -M(x_3 + x_4) + \Sigma_{j \in [2K]}(B'_{3j} - B^*_{3j})x_j \leq -M(x_3 + x_4) + 1$. $u_{c1} - u_{c3} \geq M(x_3 + x_4 - x_1 - x_2) - 1 > M\epsilon - 1 = 1$. According to Definition 7, we thus have $y_3 = y_4 = 0$. Then if we look back at row 3 and row 4 of player 1, since $y_3 = y_4 = 0$, similarly based on Definition 7 we have $x_3 = x_4 = 0$. However, this contradicts the condition that $x_3 + x_4 > x_1 + x_2 + \epsilon > 0$.

The general reduction idea is as follow. For every type of gate g in C with K real variables, we reduce it to a feasible perturbation over the standard K matching pennies game G^*_K, thus generate an instance of approximate K matching pennies game G'_K. For each of the reduction, we prove that a Nash equilibrium of G'_K would correspond to a approximately satisfiable assignment over input and output variables of g. We take the addition gate in Figure 2 as an example and show that a Nash equilibrium (x, y) of G'_K must satisfy $x_5 \leq \min\{x_1 + x_3, \frac{1}{K}\} + \epsilon$, the right half of the approximately satisfiable condition of the addition gate.

Since the indices of input variables are 1 and 3 and the index of the output variable is 5, we perturb the standard K matching pennies game as

$$\begin{aligned}
A'_{5,5} &= A^*_{5,5} + 1 = M + 1,\\
A'_{6,6} &= A^*_{6,6} + 1 = M + 1,\\
B'_{1,5} &= B^*_{1,5} + 1 = 1,\\
B'_{3,5} &= B^*_{3,5} + 1 = 1,\\
B'_{5,6} &= B^*_{5,6} + 1 = -M + 1.
\end{aligned}$$

The rest parts of A' and B' are the same as A^* and B^* respectively. By contradiction, suppose in a Nash equilibrium (x, y), $x_5 > \min\{x_1 + x_3, \frac{1}{K}\} + \epsilon$. Since $x_5 + x_6 \leq \frac{1}{K} + \epsilon$, it can only be $x_5 > x_1 + x_3 + \epsilon$. Consider the utilities of column 5 u_{c5} and column 6 u_{c6} of player 2. $u_{c5} = x_1 + x_3 - M(x_5 + x_6), u_{c6} = (-M+1)x_5 - Mx_6$. $u_{c5} - u_{c6} = x_1 + x_3 - x_5 < -\epsilon$. Hence $y_5 = 0$. Consider the utilities of row 5 u_{r5} and row 6 u_{r6} of player 1. $u_{r5} - u_{r6} = y_5 - y_6 < 0$. Hence $x_5 = 0$. However, it should be $x_5 > x_1 + x_3 + \epsilon > 0$. Thus we have a contradiction and $x_5 \leq \min\{x_1 + x_3, \frac{1}{K}\} + \epsilon$.

For all the other arithmetic and Boolean gates in an approximate generalized circuit, we can always do the similar constructions and approximate satisfiability proofs. A common property share by all the constructions is that if the output variable of a gate is X_{2i-1}, only rows $(2i - 1)$, $(2i)$ and columns $(2i - 1)$, $(2i)$ of the standard K matching pennies game will be perturbed and all the other rows and columns won't change. Since every variable $X_{2i-1}(1 \leq i \leq K)$ can be the output variable of at most one gate, all the perturbations made by all the gates in the same circuit would not conflict with each other. Therefore, for a given generalized circuit C, we can incorporate all the perturbations made by all the gates in C in the same standard K matching pennies game G^*_K to generate a legal approximate K matching pennies game G'_K and show that for every Nash equilibrium $(x = (x_1, x_2, \ldots, x_{2K-1}, x_{2K}), y)$ in G'_K, $X_{2i-1} = x_{2i-1}(1 \leq i \leq K)$ is a $\frac{1}{2^K}$-approximately satisfiable assignment to circuit C. Thus we complete the reduction in Step 2.

To sum up, since 2D-Sperner Problem can be reduced to 2-Player Nash Equilibrium and 2D-Sperner Problem is PPAD-complete, 2-Player Nash Equilibrium is PPAD-complete. $\square$

References

[1] Xi Chen and Xiaotie Deng. On the Complexity of 2D Discrete Fixed Point Problem. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming.* Lecture Notes in Computer Science, 2006, pages 489-500, Springer-Verlag.

[2] Xi Chen and Xiaotie Deng. Settling the Complexity of 2-Player Nash-Equilibrium. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 261-272, 2006.