On the computational complexity of computing an approximate Nash equilibrium

Bassel El Mabsout[1], Nicolas AlHaddad[1]

November 17, 2020

1 Introduction

Game theory is a theoretical framework for modeling the strategic interactions among rational decision-makers, allowing us to study the science of logical decision making in humans, animals, and computers. The Nash equilibrium is a central concept in game theory used for predicting the behavior of such systems in terms of searching for an optimal strategy. It therefore touches a huge array of problems in, Social Science, economics, systems science and Computer Science. This paper discusses the computational complexity of finding Nash equilibria in games as it has direct implications on the plausibility of it being a good model for the behavior of real systems. Intuitively, if the fastest computers cannot compute it within many universe lifetimes, then how can we hope that a group of real agents would do so.

John Nash showed [Nas50] that every game has at least 1 mixed Nash equilibrium but proved it non-constructively leaving open the problem of whether there exists an efficient algorithm to compute it. A Nash equilibrium is a stable state of a game, where no participant can gain by changing their strategy if the strategies of the others remain the same. It is called mixed when those strategies are represented by probability distributions.

In the beginning John von Neumann showed [Neu] that the problem of finding the Nash equilibrium in a 2 player game where the sum of the payoffs is 0 for any strategy (think rock-paper-scissors where the loser pays the winner 1$, no matter what everyone chooses their total sum of money remains unchanged, meaning no new money is added or removed from the system) can be found in polynomial time on a deterministic Turing machine through the famous mini-max algorithm (computable polynomially in the size of the description of the game). In more complex scenarios such as with more players or non-zero-sum games, however, it remained an open problem whether one can always find such Nash equilibria efficiently.

This motivated the creation of the class called PPAD 1 (polynomial parity argument for directed graphs). A canonical language that is PPAD-complete is EOL (Definition 6). Interestingly, finding a Nash equilibrium (NASH) was not proven to be PPAD-complete immediately. The first paper shows [DGP09] that even ϵ-NASH (Definition 4) (which is NASH but approximated with $\epsilon = 1/2^{poly(n)}$ precision where n is the input size describing the game) with more than 2 players is PPAD-complete. Shortly after, another paper by Chen, Deng and Teng [CDT09] extended this result for 2 players. Meaning that finding the equilibria of games in general is most likely more difficult than the 2-player

[1]Authors contributed equally

zero-sum game case (and indeed even for > 2-player zero-sum games this problem is PPAD-complete [CD11]).

1.1 Outline

In Section 2 we give all the definitions and background required to prove the main result of the original paper[DGP09]. In Section 3 we go through the proofs of why ϵ-NASH is PPAD-complete. In Section 4 we provide a discussion about future work. Finally, in Section 5 we criticize the original paper [DGP09].

2 Preliminaries

PPAD is a class of search problems that is conjectured to be hard. A search problem is defined as a set of functions: $F : \{\{0,1\}^* \to \{0,1\}^*\}$. FP is defined as the subset of F such that every $f \in \mathsf{FP}$ is computable with a deterministic polytime Turing machine. Equivalently, FNP is defined as the subset of F such that every $f \in \mathsf{FNP}$ is computable with a non-deterministic polytime Turing machine. Consequently $\mathsf{FP} \subseteq \mathsf{FNP}$ analogous to the corresponding decision problems $\mathbf{P} \subseteq \mathbf{NP}$ ($\mathsf{FP} = \mathsf{FNP} \iff \mathbf{P} = \mathbf{NP}$). It turns out that $\mathsf{FP} \subseteq \mathsf{PPAD} \subseteq \mathsf{FNP}$, and it is likely that $\mathsf{FP} \subsetneq \mathsf{PPAD}$ because if $\mathsf{PPAD} \subseteq \mathsf{FP}$ then $\mathbf{NP} = \mathbf{coNP}$ [DGP09].

Definition 1. PPAD: is the class of all search problems that are poly-time reducible to EOL (Definition 6). This makes EOL trivially PPAD-complete. PPAD was introduced by Papadimitriou in 1994 [Pap94] so that he can study the computational complexity of NASH equilibrium. PPAD stands for the polynomial parity argument on directed graphs and it is based on the fact that every directed graph with an unbalanced vertex (it's unbalanced when the number of incoming edges $\neq$ outgoing edges) must also contain at least one other unbalanced vertex as well. The corresponding search problem is to find such a vertex (EOL). Notice that since such a vertex always exist, this makes PPAD a class of total functions (search problems). Alternatively, A binary relation P(x,y) is in PPAD if and only if there is a deterministic polynomial time algorithm that can determine whether P(x,y) holds given both x and y, and for every x, there exists a y such that P(x,y) holds based on a parity argument on directed graphs.

Definition 2. GAME: A game in normal form has has at least 2 players. Each player in the game must have a finite a set of strategies associated with that player denoted by S_p. The set of all strategies in the game is the Cartesian product of all personal strategies and is denoted by S. To refer to all strategies except for the strategies of player p they denote that by S_{-p}. To refer to a subset T of player strategies in S, they denote that by S_T. To denote the payoff or utility of a certain strategy of a player p (the value of an outcome when player p choose a certain strategy s from S, notice that the utility is calculated by also looking at other players strategies, because S is the Cartesian product of all players strategies). They refer to this by u_s^p and in normal form they assume that the payoff should always be ≥ 0.

Definition 3. Nash equilibrium: Each player chooses a probability distribution over their S_p (see definition 2), called a mixed strategy, so that no player can deviate from their mixed strategy and improve on their expected payoff, i.e. "no unilateral player has incentive to deviate". The authors do give a detailed algebraic definition of the problem but we omit it for the sake of brevity (and clarity). Also notice, that the solution to NASH (the probabilities chosen by the players) might be irrational, if that's the case then computers have to approximate that value (they have finite precision) as they can't compute the exact NASH equilibrium.

Definition 4. ϵ-NASH: If Nash equilibrium means "no incentive to deviate," then approximate Nash equilibrium stands for "low incentive to deviate". Specifically, if ϵ is a small positive quantity, we can define an ϵ-NASH equilibrium as a profile of mixed strategies where any player can improve their expected payoff by at most ϵ by switching to another strategy.

Definition 5. r-NASH is the problem of finding NASH with ϵ accuracy in a game that has r players. It's a search problem that is guaranteed to have a solution [Nas50].

Definition 6. EOL (End-of-Line): G is a (possibly exponentially large) directed graph specified by a poly-time computable function f (for example defined implicitly by a poly-sized circuit) with no isolated vertices:
Input: A description of a directed graph G and a specified unbalanced vertex of G.
Output: Some other unbalanced vertex.
As mentioned in the description of PPAD (Definition 1), any graph containing an unbalanced vertex must also contain another. Intuitively, every source must have a sink (otherwise it wouldn't be a source) and EOL is the problem of finding a sink.
Note: One may think that this class is easy to solve, one would assume that a simple graph traversal algorithm will be enough to find the other vertex. However, we can't afford to do the graph traversal if the graph itself is exponentially large and we are given a description of that graph. Remember G can be specified by a poly sized circuit but the graph itself can be exponential!

Definition 7. SEOL (Simplified-End-of-Line):
This is the same definition of EOL but restricted so that every vertex in G has at most 1 outgoing vertex and at most 1 incoming vertex (this definition is often used as the definition of EOL [CDT09]) and f is represented by a circuit C with n inputs such that when given the index of a vertex, it outputs the index of the vertex that it points to (when the index of a vertex is given to C and it maps to its self, this means it doesn't point at anything). Because the structure of G means that vertices with only one neighbour come in pairs, this keeps the problem PPAD-complete. In particular, given a source s, we can find a sink s' at the other end of the path starting at s. (Note that this may take exponential time if we just evaluate f repeatedly)

Theorem 8. *Brouwer's fixed-point theorem [UZA62] states that for any continuous function f mapping a compact convex set to itself there is a point x_0 such that $f(x_0) = x_0$, the corresponding problem of finding such a fixed point is denoted as* BROUWER *[Pap94] and as they proved, it is* PPAD*-complete (Definition 11). As an example lets take the disk in $\mathbb{R}^2$ where $x^2 + y^2 \leq 1$. Let our continuous transformation f be a rotation of the disk so that for every point (x, y) satisfying the constraint $x^2 + y^2 \leq 1$ will map to some point satisfying the same constraint (A.K.A. it maps the disk to itself). According to this theorem, there must exist at least a point that didn't change through the transformation. Note that since we're doing a rotation here and the disk stays 'in the same spot', then this forces our rotation to be about the point* $(0, 0)$ *(our fixed point).*
In the paper they solve their version of Brouwer which is a discrete and simplified version of the search problem associated with Brouwer's fixed point theorem and they define it formally as:
Brouwer Input: An efficient algorithm Π_F for the evaluation of a function $F : [0, 1]^m \longrightarrow [0, 1]^m$; a constant K such that F satisfies the Lipschitz condition (Definition 9); and the desired accuracy ϵ.

Output: A point x such that $d(F(x), x) \leq \epsilon$ This definition is motivated by the fact that representing irrational numbers is computationally intractable. Since F maps $[0,1]^m$ to its self, there is no guarantee that fixed points will be rational therefore a specific accuracy ϵ is required. The Lipschitz condition guarantees that any point close to the fixed point will remain in a region determined by ϵ, K and m. Meaning that the approximation guarantees completeness (when a fixed point exists in an ϵ sized region then any point within that region will map to some other point in this same region).

Definition 9. Lipschitz condition over an m-hypercube: F satisfies this condition if $\exists K, \forall x_1, x_2 \in [0,1]^m, d(F(x_1), F(x_2)) \leq K * d(x_1, x_2)$, where d is defined as the Euclidean distance. And x_1, x_2 are points in the $[0,1]$ subset of euclidean space of m dimensions (m-hypercube). For example, the linear function F(x) = x that is defined over the domain $[0,1]$ ($m = 1$), we can see that $d(F(x_1), F(x_2)) = d(x_1, x_2) \leq k * d(x_1, x_2)$ for $k \geq 1$. Hence, F satisfies the condition.

Lemma 10. *Sperner's lemma [Spe]: Combinatorial analog of the Brouwer fixed point theorem. The lemma states that every Sperner coloring of a triangulation of an n-dimensional simplex contains a cell colored with a complete set of colors. The grid in Figure 2 is colored according to a Sperner coloring. Notice, there exist triangles with vertices colored with all set of three colors (these are the shaded triangles).*

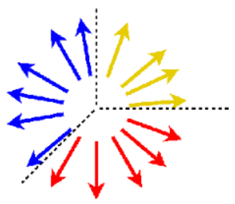

Figure 1: The colors assigned to the different directions of $F(x) - x$. There is a transition from red to yellow at 0 degrees, from yellow to blue at 90 degrees, and from blue to red at 225 degrees.

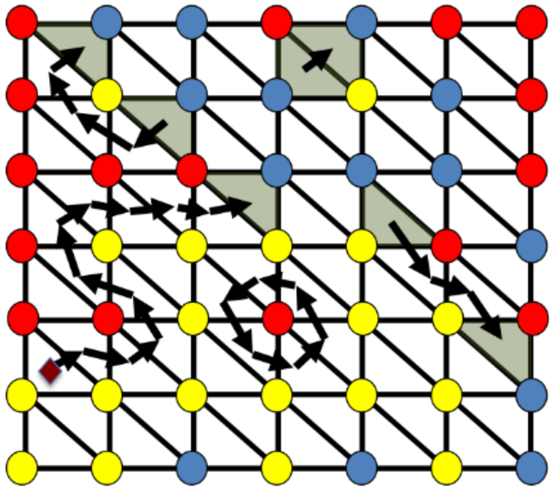

Figure 2: The subdivision of the square into smaller squares, and the coloring of the vertices of the subdivision according to the direction of $F(x) - x$. The arrows correspond to the EOL graph on the triangles of the subdivision; the source T is marked by a diamond

3 Main Results

The main result of the paper is showing that r-NASH is PPAD-complete for players $r \geq 3$. The paper we are summarizing is the 2009 paper where they updated their result, they also mention that it is possible for $r = 2$ as well. Warning: The proofs used here rely heavily on topology!

Lemma 11. BROUWER *is* PPAD*-complete*

Proof. To prove that BROUWER is PPAD-complete they prove that BROUWER $\in$ PPAD and that BROUWER is PPAD-hard.

1. BROUWER $\in$ PPAD: This is done through reduction from BROUWER to EOL. Since EOL $\in$ PPAD then BROUWER $\in$ PPAD:

   They take an instance of BROUWER (Definition 8) F and build an input to EOL (Definition 6), mainly a Graph and an unbalanced vertex to start the end of line search from. Focusing on the 2d case, they discretize the $[0, 1]^2$ square (domain of F) by forming a grid made of smaller squares (of side length determined by K and ϵ), and then divide each little square into two right triangles. Notice that we can find approximate fixed points by searching for points that remained within the same triangle after the transformation. Meaning the size of the triangle determines the accuracy of the approximation. Since they're working in a discrete setting, they use the discrete analog of Brouwer's theorem, otherwise known as Sperner's Lemma 10. Then, a Sperner coloring of this grid is performed: "We color each vertex of the triangles by one of three colors depending on the direction in which F maps x. In two dimensions, this can be taken to be the angle between vector $F(x) - x$ and the horizontal. Specifically, we color it red if the direction lies between 0 and -135 degrees, blue if it ranges between 90 and 225 degrees, and yellow otherwise,as shown in Figure 1. (If the direction is 0 degrees, we allow either yellow or red; similarly for the other two borderline cases.) Using the above coloring convention the vertices get colored in such a way that the Lipschitz (Definition 9) property is satisfied. None of the vertices on the lower side of the square uses red, no vertex on the left side uses blue, and no vertex on the other two sides uses yellow. Figure 2 shows a coloring of the vertices that could result from the function F". Since this is a Sperner coloring, by Sperner's lemma (Definition 10), this guarantees that there exists a triangle where the vertices are all colored differently (these are shaded in Figure 2). This indicates that within this triangle there is a fixed point. They reduce the problem of finding this triangle to EOL. There is always a path through the edges starting from the one going from yellow to red on the leftmost vertical line of the grid from which we can reach a tricolored triangle in the following way: From a red-yellow edge e we reach another one e' that is not visited before on the triangles connected to e. Once such an e' does not exist, the corresponding triangle will be tricolored. Now, we can do a poly-time transformation of this graph into an input accepted by EOL such that the source would be the starting edge on the left and the destination would be the edge of the tricolored triangle containing the fixed point (this is a slightly simplified but equivalent version of the rules in the paper). The transformation converts edges into vertices, triangles into edges such that the algorithm solving EOL in the new graph would solve BROUWER in the old one.

2. BROUWER is PPAD-hard: In the paper they reduce EOL to BROUWER. EOL is PPAD-complete which implies that BROUWER is PPAD-hard.
   The authors prove this by a poly-time transformation from any graph G and source s taken as inputs of EOL to a computable K-Lipschitz continuous function F mapping the 3-cube: $[0,1]^3$ to it's self where every fixed point is a sink and the source s is a specific point in the cube. This transformation however is quite complex, we therefore present a simpler proof by replacing EOL with SEOL and similarly polynomially reducing the problem of finding a vertex in our simpler G with no outgoing edges (the end of the line) to finding a fixed point for a computable K-Lipschitz continuous function F mapping $[0,1]$ to its self as follows:

   The number of vertices in G cannot be larger 2^n (n being the number of inputs to the circuit C describing G). There is therefore a bijection between every vertex v in G and a point p in the set $P : \{0, 1/2^n, 2/2^n, \dots 1 - 1/2^n\}$ lying in our [0,1] region. Then, we define our computable transformation F on every such p by mapping it to the point $p' = p + 1/2^n$ if $C(p) \neq p$ otherwise $p' = p$. Notice that since C maps vertex indices to themselves if they don't have an outgoing edge then every sink is mapped to a fixed point of F and every other point is not. Now F still needs to be a K-Lipschitz continuous total function over its domain, and to preserve this property every point in between the points in P is linearly interpolated: For every point x in $[0,1]$ lets define $\underline{x} = \frac{\lfloor x2^n \rfloor}{2^n}$. Then $F(x) = F(\underline{x}) + 2^n(x - \underline{x})(F(\underline{x} + 1/2^n) - F(\underline{x}))$, $F(1) = 1 - 2/2^n$ (to handle the special case of points between $1 - 2^n$ and 1). Notice that for any $x_1, x_2 \in [0,1]$, $d(F(x_1), F(x_2)) \leq d(x_1, x_2)$ meaning our Lipschitz constant is 1. Also due to our mapping, any time a point x has the property: $|x - F(x)| < 1/2^n$ will be a point where $\underline{x}$ is a fixed point making it obvious that finding an $\epsilon < 1/2^n$ approximated fixed point means finding the vertex in the graph that corresponds to a sink. This therefore concludes our proof that SEOL is poly-time reducible to BROUWER.

□

Lemma 12. ϵ-NASH $\leq_p$ BROUWER

Proof. The full proof from NASH to $\mathsf{BROUWER}$ was given by NASH himself in the 1950: [Nas50]. The paper argues that given that there is a reduction from NASH to $\mathsf{BROUWER}$ then there must a be reduction from ϵ-$\mathsf{NASH} \leq_p \mathsf{BROUWER}$. The idea behind Nash's proof is to construct a function f from the set all of all strategies to itself, that satisfies the conditions of Brouwer's fixed point theorem such that the fixed point x is a Nash equilibrium. Unless the players are happy with x (which means they have reached nash equilibrium). The players will keep changing x. This function will indicate the movement of the unsatisfied players. $\square$

Lemma 13. $\mathsf{BROUWER} \leq_p r$-$\mathsf{NASH}$ *where* $r = 3$

Proof. The key thing that the paper did is show how to make NASH games compute simple arithmetic functions and then use that to compute $\mathsf{BROUWER}$ functions. To do this they represented each arithmetic operation by a game of maximum of four players and then compose those games so that a Nash equilibrium in the final game represents an approximate fixed point in the original $\mathsf{BROUWER}$ function. We can represent any $\mathsf{BROUWER}$ function by a composition of these simple operators: addition, multiplication and comparison. Addition and multiplication is easy to simulate by a game (by inducing a player to pick an action with probability that is the sum or the product of 2 other player's probability of picking that action. On a high level, the system is giving incentive for the players to produce the right answer to the arithmetic operation at hand). However, comparison requires special handling, the details of which are in the paper. Since circuits are inherently composable, if we replace every operation by a game. Then the composition of those games (through using 2 players as inputs and a third player as an output) is going to give the same result as the circuit.
However, we now have a problem. The number of operations in the circuit will dictate how many players we have in the final composed game. Notice, that many of the players within the games that represent arithmetic operations are independent in the sense that they don't effect each other's strategies. In particular this structure forms what is known as a graphical game. Graphical games can always be simulated by a game of three players. Hence, $\mathsf{BROUWER}$ is as difficult as computing r-NASH with three players. $\square$

4 Conclusion

It seems that computing Nash equilibria may be difficult since ϵ-NASH is PPAD-complete. Actually, there has been many improvements over what we know about the complexity of finding Nash equilibria. For instance, In this paper [CDT09] they improve both the number of players required to play the game to the minimum possible of 2, and the approximation ϵ from $1/2^{poly(n)}$ in this paper to $1/poly(n)$ in theirs. But still, in order to discount NASH as a model, one has to prove that weakening the approximation even further would still be outside of FP. And even if it is, it could still be tractable by being computable using quasi-polynomials, as far as we know, this remains an open problem. Another avenue to explore is whether we can exploit the structure of realistic games further generating a subset of games where finding the Nash equilibrium is in FP. This is not unlikely as the algorithms with worst-case exponential time complexity often end up being efficient when applied on real problems.

5 Critiques

- Notational abuse: The authors override the conventional definition of **P** and **NP** where it seems they mean FP and FNP instead. "We think of **NP** as the class of search problems", this contradicts with the definitions provided in class where **P** and **NP** define the class of decision problems that can be solved with a polynomial time deterministic and non-deterministic Turing machine respectively.

- Zero-sum games are easy: The paper implies that finding the equilibrium of all zero-sum games can be done in poly-time. "if a game is zero-sum, like the rock-paper-scissors game, then it follows from the work of John von Neumann in the 1920s that Nash can be formulated in terms of linear programming (a subject identified by George Dantzig in the 1940s); linear programs can be solved efficiently". This is misleading as it has been shown that the problem of finding the Nash equilibria of 3-player zero-sum-games is PPAD-complete. This is true only when the zero-sum game involves 2 players.

- ϵ confusion: The paper defines NASH as the problem of finding an approximate Nash equilibrium in a game with $\epsilon = 1/2^{poly(n)}$ where n is the input length. However, they also define ϵ-NASH to mean the same problem but where $\epsilon = 1/poly(n)$. Both definitions have to deal with approximations, since in the case of mixed equilibrium, the values may be irrational and we cannot have infinite precision on a DTM. Calling one ϵ-NASH and the other NASH is misleading.

References

[Nas50] John F. Nash. "Equilibrium points in n-person games". In: *Proceedings of the National Academy of Sciences* 36.1 (1950), pp. 48–49. ISSN: 0027-8424. DOI: 10.1073/pnas.36.1.48. eprint: https://www.pnas.org/content/36/1/48.full.pdf. URL: https://www.pnas.org/content/36/1/48.

[UZA62] H. UZAWA. "Walras's existence theorem and Brouwer's fixpoint theorem". In: *Econ. Stud. Quart. 13* (1962), pp. 59–62.

[Pap94] Christos H. Papadimitriou. "On the complexity of the parity argument and other inefficient proofs of existence". In: *Journal of Computer and System Sciences* 48.3 (1994), pp. 498–532. ISSN: 0022-0000. DOI: https://doi.org/10.1016/S0022-0000(05)80063-7. URL: http://www.sciencedirect.com/science/article/pii/S0022000005800637.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. "Settling the Complexity of Computing Two-Player Nash Equilibria". In: *J. ACM* 56.3 (May 2009). ISSN: 0004-5411. DOI: 10.1145/1516512.1516516. URL: https://doi.org/10.1145/1516512.1516516.

[DGP09] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. "The Complexity of Computing a Nash Equilibrium". In: *CommunICatIons of the aCm* 52.2 (2009).

[CD11] Yang Cai and Constantinos Daskalakis. "On minmax theorems for multiplayer games". In: *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms.* SIAM. 2011, pp. 217–234.

[Neu] J. V. Neumann. “Zur Theorie der Gesellschaftsspiele”. In: *Mathematische Annalen* 100 (), pp. 295–320.

[Spe] E. Sperner. “Neuer beweis für die invarianz der dimensionszahl und des gebietes”. In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 6 (), pp. 265–272.