Parametrized Two-Player Nash Equilibrium [5]

Danny Hemerlin, Chien-Chung Huang, Stefan Kratsch, and Magnus Wahlström

Merkouris Papamichail

IGP ALMA, AL1.20.0018

Spring 2021

Abstract

In this report we present a synopsis of Danny Hemerlin's et al. paper on *Parametrized Two-Player Nash Equilibrium*. The authors provide FPT algorithms for variants of the Nash problem, namely finding a Nash equilibrium in two player games. The first variant of the problem considered is in ℓ-sparse games, where they present an $\ell^{O(k\ell)} \cdot n^{O(1)}$ algorithm. The second case the authors consider regards games of *locally bounded treewidth*, where they obtain an algorithm of $f(k,\ell) \cdot n^{O(1)}$ time.

1 Introduction

Danny Hemerlin's et al. paper is rich in results regarding the computation of a Nash equilibrium in two-player games. In this report we only be considering the two main results, of ℓ-sparse games and games of locally bounded tree width.

We first introduce some elementary definitions form game theory. In order to define a two-player normal form game, of bi-matrix game, we first need to discuss the *mixed* strategy space of each player. We allow each player to use a probability mechanism, in order to mixed her strategies. Namely, if a player has d *pure* strategies, a mixed strategy is a vector in $\mathbb{R}^d$ space, which belongs to the $d-1$-probability simplex.

Definition 1 (d-Simplex)**.** Let $\langle e^1, e^2, \ldots, e^{d+1} \rangle$ be the normal base of $\mathbb{R}^{d+1}$, We call the set, $\Delta_d = \text{convex}(\langle e^1, e^2, \ldots, e^{d+1} \rangle)$, a d-Simplex.

Note that s d-Simplex is a d-dimentional object in $d+1$-dimentional space. Moreover, for all $x \in \Delta_d$, we have $x \geq 0$, while also $\sum_{i=1}^{d+1} x_i = 1$. Observe that each $x \in \Delta_d$ defines a *probability distribution* on d possibilities. Hence, by allowing a player to chose some $x \in \Delta_d$, we allow her to chose a probability distribution on her d strategies. We also will be needing the notion of *support*, for some $x \in \Delta_d$.

Definition 2. Let Δ_d be a d-Simplex, also let some $x \in \Delta_d$ an element of Δ_d. We call *support* of x the set $S(x) = \{i \in [d+1] \mid x_i > 0\}$.

We are now ready to present a definition of a bimatrix game.

Definition 3. A bimatrix game is a two-player game. Let player 1 have n *pure* strategies, while player 2 has m. Also, let $A, B \in \mathbb{R}^{m \times n}$ two real matrices, which we will be calling *payoff matrices*. In the single turn of the game, the two players choose a vector of their respective simplices *simultaneously*. The players obtain u_1, u_2 units of utility respectively, where

$$u_1 = x^t A y,$$
$$u_2 = x^t B y.$$

We make now some remarks regarding Definition 3. Firstly, note that *every* (non-cooperative) two player game can be expressed as a bimatrix game. Secondly, we can always assume $A = B$ [6], and that the payoof matrices have non-negative entries [6]. Lastly, we will denote as (A, B)-bimatrix game, a bimatrix game, with A, B as the payoff matrices.

In Definition 4 we present the most popular solution concept of bimatrix games, the *Nash equilibrium*. Note that for a (A, B)-bimatrix game, we call a strategy profile a pair (x, y), where $x \in \Delta_{m-1}$ and $y \in \Delta_{n-1}$.

Definition 4 (Nash equilibrium)**.** Let $(x^\star, y^\star) \in \Delta_{m-1} \times \Delta_{n-1}$ be a strategy profile. We say that $(x^\star, y^\star)$ is a *Nash equilibrium*, if and only if,

1. for all $x \in \Delta_{m-1}$, we have $(x^\star)^t Ay^\star \geq x^t Ay^\star$, *end*
2. for all $y \in \Delta_{n-1}$, we have $(x^\star)^t By^\star \geq (x^\star)^t By$.

Observe that, Definition 4 describes a situation where a player's strategy is optimal for a fixed strategy of the other player; and this is true for both players. The most important fact regarding the Nash equilibrium, and perhaps the reason for its wide acceptance as solution concept, is that *every normal form game has one*. This is a result due to John Nash proved around 1950. In the following lemma we give a characterization of Definition 4.

Lemma 5 (Support Lemma)**.** A strategy profile (x, y) is a Nash equilibrium for the (A, B)-bimatrix game, if and only if,

1. if $x_s > 0$, then $(Ay)_s \geq (Ay)_j$ for all $j \neq s$, *and*,
2. if $y_t > 0$, then $(x^t B)_t \geq (x^t B)$ for all $i \neq t$.

Perhaps the most important application of Support Lemma is that, given some support sets S, T we can compute a strategy profile (x, y), such that $S = S(x)$ and $T = S(y)$, if (x, y) is a Nash equilibrium. This way we can derive a simple algorithm for computing Nash equilibria, in literature this algorithm is often called *Support Enumeration*. Support Enumeration follows a "generate and test" philosophy. We "guess" the two sets S, T and then check if Support Lemma holds. In the following sections we will use the same philosophy, we will try to make educated guesses regarding S, T, in order to avoid exploring the whole support space.

Observe that, if support is bounded by k, then using support enumeration, we can compute a Nash equilibrium in $n^{O(k)}$ time. On the other hand, unless FPT = W[1], there is no $n^{o(k)}$ time algorithm for computing Nash equilibria of support size at most k [4]. Lastly, note that computing a Nash equilibrium is PPAD-complete, even for two player games [3, 1, 2]. Hence, we do not expect to find a polynomial-time algorithm for computing a Nash equilibrium.

2 Results

In this report we will present two results due to Hemerlin et al. in two classes of bimatrix games. For ℓ-sparse games, we give an algorithm of $\ell^{O(k\ell)}$ time, where k is an upper bound to the size of the support, and ℓ is the maximum number of non-zero entries in each row or column of the payoff martices. On the other hand, we consider the class of locally bounded treewidth games, as a sub-class of games with *at most* ℓ different values in their payoff matrices. In both cases the parameter will be $k + \ell$.

As with most parametrized algorithms, we will be needing to define a discrete structure to utilize. In this direction, we define a *game graph*, by considering the non-zero entries of the payoff matrices, and using the resulting $0, 1$-matrix as adjacency matrix for a graph. In ℓ-sparse games, we will have that the maximum degree is bounded by ℓ, i.e. $\Delta(G) \leq \ell$. In games with ℓ different values we will uutilize the *locally bounded treewidth.*

3 ℓ-Sparse Games

We start this section by defining the notion of the game graph for a bimatrix game.

Definition 6. [Game Graph] Let (A, B) be a bimatrix game. Let $\mathcal{G} = A \vee B$ be a matrix, where,

$$\mathcal{G}[i][j] = \begin{cases} 1, & A[i][j] \neq 0 \text{ or } B[i][j] \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

We call the bipartite graph $G = (S^1 \cup S^2, E)$, induced by $\mathcal{G}$ the *game graph* of the (A, B)-bimatrix game, where S^1, S^2 the set of pure strategies of the players.

We next define the notion of minimal Nash equilibrium. Intuitively, a *minimal* Nash equilibrium is a Nash equilibrium that does not contain an other Nash equilibrium induced by a subset of its support. Formally we give the following definition.

Definition 7. A Nash equilibrium (x, y) is *minimal*, if for any Nash equilibrium (x', y'), with $S(x') \subseteq S(x)$ and $S(y') \subseteq S(y)$, we have $S(x') = S(x)$ and $S(y') = S(y)$.

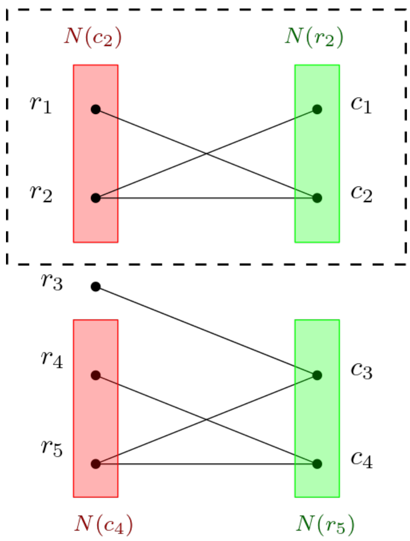
The main observation for our algorithm is stated in the following lemma. Note that for some sets $I \subseteq [m]$ and $J \subseteq [n]$ the matrix $A_{I,J}$ is the submatrix of A, induced on the rows of I and the columns of J.

Lemma 8. If (x, y) is a minimal Nash equilibrium for a game (A, B), with either $A_{S(x),S(y)} \neq 0$ or $B_{S(x),S(y)} \neq 0$, then the subgraph induced by $N[S(x) \cup S(y)]$ is *connected.*

We give the following example.

Example 9. *Note that the contrapositive of Lemma 8 states that if the induced subgraph is* not *connected, then the Nash equilibrium is* not *minimal.*

$$[0 \; x_1 \; 0 \; 0 \; x_2] \begin{bmatrix} 0 & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \\ 0 & 0 & a_{53} & a_{54} \end{bmatrix} \begin{bmatrix} 0 \\ y_1 \\ 0 \\ y_2 \end{bmatrix}$$



Since the induced graph is not *connected, then the Nash equilibrium (x, y) is* not *minimal. In dead, the strategy profile $x' = (0, 1, 0, 0, 0)$, $y' = (0,\ 1,\ 0,\ 0)$ is a Nash equilibrium.*

It can be proved that we can enumerate all the induced subgraphs on t vertices, with c connected components in $(\Delta + 1)^{2t} \cdot n^{c+O(1)}$ time. Hence, we have just derived an algorithm for computing an Nash equilibrium in ℓ-sparse bimatrix games. We enumerate all induced connected subgraphs on k vertices. Let V^1, V^2 be the corresponding support from the previous step. Check the Support Lemma for the given support. The resulting algorithm runs in time $\ell^{O(k\ell) \cdot n^{O(1)}}$, where k is an upper bound to the size of the support.

4 Locally Bounded Treewidth

We start this section by defining an new class of games.

Definition 10 (Bounded Valued Games)**.** Let $P \subset \mathbb{Q}$, and $|P| \leq \ell$. We consider a subcalss of bimatrix games (A, B), where $A, B \in P^{m \times n}$. We call this type of bimatrix game, an ℓ-bounded valued game.

Next, we present the notion of *locally bounded treewidth.*

Definition 11. A graph class has *locally bounded treewidth* if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph $G = (V, E)$ of the class, any vertex v and any distance $d \in \mathbb{N}$, the subgraph of G induced by all vertices within distance *at most* d from v has treewidth at most $f(d)$.

The crucial property of locally bounded treewidth graphs is that first-order queries can be answered in FPT time, when the parameter is the size of the first order formula. We next define some notion that we are going to need in our algorithm.

Definition 12 (Equilibrium Patterns)**.** a) Let I, J be two subsets of k elements from $[n]$. We say that two matrices $A^\star, B^\star \in \mathbb{Q}^{k \times k}$ *occur* in the bimatrix game (A, B) if $A^\star = A_{I,J}$ and $B^\star = B_{I,J}$.

b) On the other hand, we say that the pair $(A^\star, B^\star)$ *forms an equilibrium pattern* if there exist an equilibrium (x, y), where $(A^\star, B^\star)$ occurs in the game (A, B) at $(S(x), S(y))$.

Our algorithm will try all possible ℓ^{2k^2} pairs of matrices $(A^\star, B^\star)$. For each pair we determine whether it is an equilibrium pattern. Now, when does a pair of matrices $(A^\star, B^\star)$ form an equilibrium pattern? Firstly, it must occur in the game (A, B) for some indices sets I, J. Then, there must be an equilibrium (x, y) with $S(x) = I$ and $S(y) = J$, such that neither player has better alternative. The resulting algorithm will be of time $f(k, \ell) \cdot n^{O(1)}$, where $f(\cdot)$ is the function of Definition 11. Again, k is the upper bound to the size of the support.

Now, consider the following example.

Example 13. *Consider a win-lose game where the payoff matrices have values on* $\{0, 1\}$*. This* (A, B)*-game can be encoded into the relations of two arguments* $A/2$*,* $B/2$*, such that* $A(r, c)$ *is true if and only if we have* $A[i][j] = 1$*. Likewise for* B*.*

$$A^\star = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad B^\star = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{aligned} &\exists\, r_1, r_2, c_1, c_2 \\ &A(r_1, c_1) \land \neg A(r_1, c_2) \land \neg A(r_2, c_1) \land A(r_2, c_2) \land \\ &B(r_1, c_1) \land \neg B(r_1, c_2) \land \neg B(r_2, c_1) \land B(r_2, c_2) \land \\ &\forall r' (\neg A(r', c_1) \lor \neg A(r', c_2)) \land \forall c' (\neg B(r_1, c') \lor \neg B(r_2, c')) \end{aligned}$$

The first two lines of this formula encode the matrix patterns left. The last line enforces the there is no line with more than one "1", hence the player have no better choice. In general, with ℓ *different values, there would be* $\ell - 1$ *relations* A_i, B_i *encoding the game, where* $A_i(r, c)$ *is true if* $A_{r,c} = z_i$ *for every* $z_i \in P$ *except the zero value.*

5 Conclusions & Future Work

In this report we presented two FPT algorithms, from Hermelin's et al. paper, for computing the Nash equilibrium in two player games. We defined an underlying graph strtucture in bimartix games. We utilized this graphic-theoretical notion to study ℓ-sparse games, and games with locally bounded treewidth.

Hermelin's et al. paper is one of the first in the area of parametrized algorithms for computing a Nash equilibrium. Hence, there are many question still be unanswered. For example, is there a polynomial time algorithm for computing a Nash equilibrium in games of bounded treewidth? Can we remove the assumption regarding the number of different values in the algorithm for locally bounded treewidth?

References

[1] X. Chen and X. Deng, *3-nash is ppad-complete"*, Electronic Colloquium on Computational Complexity (ECCC), (2005).

[2] ——, *Settling the complexity of two-player nash equilibrium*, in 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), 2006, pp. 261–272.

[3] K. Daskalakis and C. Papadimitriou, *Three-player games are hard*, Electronic Colloquium on Computational Complexity (ECCC), (2005).

[4] V. Estivill-Castro and M. Parsa, *Computing nash equilibria gets harder - new results show hardness even for parameterized complexity*, Conferences in Research and Practice in Information Technology Series, 94 (2009).

[5] D. Hermelin, C.-C. Huang, S. Kratsch, and M. Wahlström, *Parameterized two-player nash equilibrium*, in Graph-Theoretic Concepts in Computer Science, P. Kolman and J. Kratochvíl, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 215–226.

[6] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, New York, NY, USA, 2007.