Parametrized Two-Player Nash Equilibrium [5]

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

Merkouris Papamichail[†]

[†]IGP in Algorithms, Logic and Discrete Mathematics
NKUA, NTUA

Parametrized Algorithms & Complexity,
Spring 2021

Table of Contents

Table of Constents

Results

Problems	Computation Time
ℓ-sparse Games	$\ell^{O(k\ell)} \cdot n^{O(1)}$
Locally Bounded Treewidth Games	$f(k,\ell) \cdot n^{O(1)}$
k-unbalanced Games	$\ell^{O(k^2)} \cdot n^{O(1)}$

Table of Constents

Probability Simplex

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 $\Delta_d = convex(\langle e^1, e^2, \ldots, e^{d+1} \rangle) \subset \mathbb{R}^{d+1}$, a d-Simplex.

1. A d-Simplex is a d-dimensional object in $d+1$-dimensional space.
2. For all $x \in \Delta_d$, we have $x \geq 0$.
3. For all $x \in \Delta_d$, we have $\sum_{i=1}^{d+1} x_i = 1$.
4. Each $x \in \Delta_d$ defines a *probability distribution* on d possibilities.
5. For some $x \in \Delta_d$, we call *support* the set $S(x) = \{i \in [d+1] \mid x_i > 0\}$

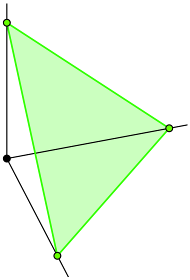


Figure 1: The 2-simplex in $\mathbb{R}^3$.

2-player Normal Form Games

Bi-matrix Game

1. 2-player game.
2. Player 1 has m strategies. while player 2 has n strategies.
3. Let $A, B \in \mathbb{R}^{m \times n}$ two real matrices.
4. The two players choose a vector of their respective simplices, *simultaneously*:
    1. Player 1 chooses some $x \in \Delta_{m-1}$.
    2. player 2 chooses some $y \in \Delta_{n-1}$
5. The players obtain u_1, u_2 units of utility, respectively, where:
    1. $u_1 = x^t A y$
    2. $u_2 = x^t B y$

Notes:

1. *"Every"* 2-player game can be represented as a Bi-matrix game.
2. We assume that $A = B$ [6].
3. We assume $A \in \mathbb{R}^{n \times n}_{\geq 0}$ [6].

Nash Equilibrium

Definition

Let $(x^\star, y^\star) \in \Delta_{m-1} \times \Delta_{n-1}$ be the two strategies of the two players. We say that the *strategy profile* $(x^\star, y^\star)$ is a *Nash Equilibrium* (NE), iff,

1. For all $x \in \Delta_m$, we have $(x^\star)^t A y^\star \geq x^t A y^\star$, *and*
2. for all $y \in \Delta_n$, we have $(x^\star)^t A y^\star \geq (x^\star)^t A y$.

I.e. my strategy is optimal, for a fixed strategy of the other player; and this is true for both players.

Notes:

1. The Nash Equilibrium is the most acceptable *solution concept* for a Normal Form Game.
2. John Nash proved (1950) that *every* Normal Form Game has a NE.

Computing a Nash Equilibrium

- Remember: For some $x \in \Delta_d$ we call *support* of x the set $S(x) = \{i \in [d+1] \mid x_i > 0\}$ containing the indices of the positive coordinates of x.

Support Lemma

The pair of strategy vectors (x, y) is a Nash equilibrium for the Bi-matrix game (A, B) iff:

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

Algorithm: Support Enumeration

1. "Guess" a support vector for each player.
2. Determine whether the inequalities of Support Lemma hold.

Complexity of Computing a NE

1. Computing a NE is PPAD-complete, *even* for a 2-player game [3, 1, 2].
   Hence, we do *not* expect to find a polynomial-time algorithm fro NE.
2. Computing a NE is *as difficult as* computing the support of a NE [6].
3. We can compute a NE of a Bi-matrix game, where the *support is bounded by k*, in $n^{O(k)}$ time (see Support Enum.).
4. Unless $\mathrm{FPT} = \mathrm{W}[1]$, there is no $n^{o(k)}$ time algorithm for computing a NE with support of size at most k in a Bi-matrix game [4].

Table of Constents

Towards a FPT Algorithm

1. Finding a proper parameter:
   - The obvious parameters, i.e. the number of players or the size of the support, *are not suitable*.
   - We consider ℓ-sparse Bi-matrix games, which have *at most* ℓ *non-zero values* in each row or column.
   - We, also, will consider (A, B) Bi-matrix games, where the matrices have at most ℓ different values.
   - Parameter (in both cases): $\ell + k$, where k is an upper bound to the size of the support.
2. Finding a structure:
   - It's not obvious which discrete structure to utilize.
   - We will extrapolate a *Game Graph*, by considering the non-zero entries of the payoff matrices.
   - In ℓ-sparse Bi-matrix games we will have $\Delta(G) \leq \ell$, where G is the Game Graph.
   - In games with ℓ different values, the Game Graph will exploit the *Locally Bounded Treewidth*.

Table of Constents

Game Graph

Game Graph

Let (A, B) be a Bi-matrix game. Let $\mathcal{G} = A \vee B$ be the 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* $G = (S^1 \cup S^2, E)$, induced by $\mathcal{G}$ the *game graph* of (A, B). Where S^1, S^2 the set of strategies of each player, respectively.

Note: For ease of the presentation, we (often) assume $A = B$, and $A, B \geq 0$.

Game Graph: Example

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \\ 0 & 0 & a_{53} & a_{54} \end{bmatrix}$$

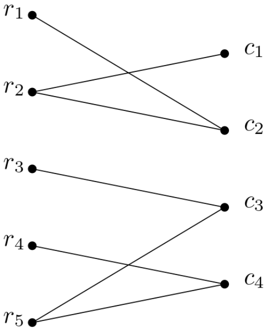


Figure 2: The game graph induced by the A.

Utilizing the Game Graph

Minimal Nash Equilibria

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 $S(y') = S(y)$.

Lemma

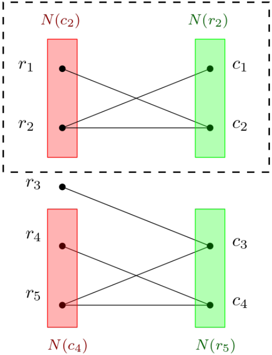
If *minimal* NE $\Rightarrow$ *connected*

Game Graph: Example 2 (a)

Lemma

If *not* connected $\Rightarrow$ *not* minimal NE

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



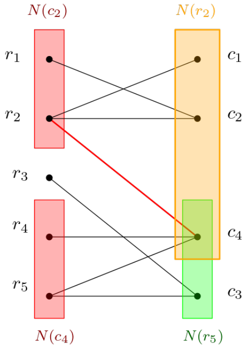
- **Remarks** Since the induced graph is *not connected*, then the NE (x, y) is not minimal. In deed the strategy profile $x' = (0, 1, 0, 0, 0)$, $y' = (0, 1, 0, 0)$ *is* a NE.

Game Graph: Example 2 (b)

Lemma

If minimal NE $\Rightarrow$ connected

$$[0\ x_1\ 0\ 0\ x_2] \begin{bmatrix} 0 & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & \color{red}{a_{24}} \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \\ 0 & 0 & a_{53} & a_{54} \end{bmatrix} \begin{bmatrix} 0 \\ y_1 \\ 0 \\ y_2 \end{bmatrix}$$



- Remarks If we assume that $a_{24} > 0$, then the same trick *won't work*, (x', y') will not be a NE.
- Note: That Lemma 2 is *not a characterization*. Hence, we could have a connected induced graph, that does not correspond to a NE.

An FPT Algorithm for ℓ-sparse Games (1)

Lemma

We can enumerate all the induced subgraphs on t vectices, with c connected components in,

$$(\Delta + 1)^{2t} \cdot n^{c+O(1)}$$

time.

Algorithm

1. Enumerate all induced *connected* subgraphs on k vertices.
2. Let V^1, V^2 be the corresponding support from the previous graph.
3. Check the Support Lemma for the given support.

An FPT Algorithm for ℓ-sparse Games (2)

Theorem

A Nash Equilibrium in a ℓ-sparse Bi-matrix game, where the support sizes is bounded by k, can be computed in $\ell^{O(k\ell)} \cdot n^{O(1)}$.

Corollary

The problem of determining whether a Nah Equilibrium with support sizes at most k exists in an ℓ-sparse Bi-matrix game, admits *no polynomial kernel*.

Table of Constents

Locally Bounded Treewidth

Definition

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 $d \in \mathbb{N}$, the subgraph of G induced by all vertices within distance *at most* d from v has treewidth at most $f(d)$.

- Note: *The crucial property of locally bounded treewidth graphs is that first-order queries can be answered in FPT time on such graphs, when the parameter is the size of the first-order formula.*

A New Class of Games: Bounded Valued Matrices

Bounded Valued Bi-matrix Games

Let $P \subset \mathbb{Q}$, and $|P| \leq \ell$, for some value ℓ. We consider a subclass of Bi-matrix games (A, B), where $A, B \in P^{m \times n}$.

- Note: *In this section we will present an algorithm that finds an equilibrium with support sizes at most k, when the game come from a graph with locally bounded treewidth.*
- Parameter: $\ell + k$

Theorem

A Nash Equilibrium in a *locally bounded treewidth* game, where the support sizes are bounded by k, and the payoff matrices have *at most ℓ different values*, can be computed in $f(k, \ell) \cdot n^{O(1)}$ time for some computable function $f(\cdot)$.

Equilibrium Patterns

Definition

(a) Let I, J be two subsets of k *elements* in $[n]$. We say that two matrices $A^\star, B^\star \in \mathbb{Q}^{k\times k}$ *occur* in the Bi-matrix game (A, B) if $A^\star = A_{I,J}$ and $B^\star = B_{I,J}$.
(b) The pair $(A^\star, B^\star)$ *forms an equilibrium pattern* if there exists an equilibrium (x, y) where $(A^\star, B^\star)$ occurs in the game (A, B), at $(S(x), S(y))$.

1. Our algorithm will try *all possible* ℓ^{2k^2} pairs of matrices $(A^\star, B^\star)$.
2. For each pair we determine whether it is an *equilibrium pattern*.
3. When does a pair of matrices $(A^\star, B^\star)$ form an equilibrium pattern?
   a) It must *occur* in the game (A, B) for some indices sets I, J.
   b) There must be an equilibrium (x, y) with $S(x) = I$ and $S(y) = J$, *such that neither player has better alternative*.

Quering an Equilibrium Pattern with a First-Order Formula

▶ Example: Consider a win-lose, $P = \{0,1\}$, game (A, B), encoded into relations $A/2, B/2$, such that $A(r,c)$ is *true* iff $A[r][c] = 1$. Likewise for B.

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

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

The pattern $(A^\star, B^\star)$.

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

The corresponding first-order formula.

▶ Note: In general, with ℓ different values, there would be $\ell - 1$ relations A_i and 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.

Table of Constents

Conclusions & Future Work

Conclusions

- We presented parametrised algorithms to compute a Nash Equilibrium in 2-player games.
- We defined an underlying graph structure in Bi-matrix Games.
- We utilized the graph-theoretical structure inherited in Bi-matrix Games.

Future Work

- Is there a polynomial-time algorithm for computing Nash equilibria in games of bounded treewidth?
- Can we remove the assumption regarding the number of different values in the algorithm for Locally Bounded Treewidth Games?

Thank you for your time!

Table of Constents

Bibliography

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

———, *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.

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

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).

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.

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