Algorithmic game theory

Martin Balko

4th lecture

October 26th 2023

Nash equilibria in bimatrix games

What we learned last time

What we learned last time

- Last lecture we introduced a brute-force algorithm to find all Nash equilibria in bimatrix games.

What we learned last time

- Last lecture we introduced a brute-force algorithm to find all Nash equilibria in bimatrix games.
- Recall that we have payoff matrices M and N with $(M)_{i,j} = u_1(i,j)$ and $(N)_{i,j} = u_2(i,j)$.

What we learned last time

- Last lecture we introduced a brute-force algorithm to find all Nash equilibria in bimatrix games.
- Recall that we have payoff matrices M and N with $(M)_{i,j} = u_1(i,j)$ and $(N)_{i,j} = u_2(i,j)$.
- The best response condition: If x and y are mixed strategy vectors of players 1 and 2, respectively, then x is a best response to y if and only if for all $i \in A_1$,

$$x_i > 0 \Longrightarrow (M)_i y = \max\{(M)_k y \colon k \in A_1\}.$$

  Analogously, y is the best response to x if and only if for all $j \in A_2$,

$$y_j > 0 \Longrightarrow (N^\top)_j x = \max\{(N^\top)_k x \colon k \in A_2\}.$$

What we learned last time

- Last lecture we introduced a brute-force algorithm to find all Nash equilibria in bimatrix games.
- Recall that we have payoff matrices M and N with $(M)_{i,j} = u_1(i,j)$ and $(N)_{i,j} = u_2(i,j)$.
- The best response condition: If x and y are mixed strategy vectors of players 1 and 2, respectively, then x is a best response to y if and only if for all $i \in A_1$,

  $$x_i > 0 \implies (M)_i y = \max\{(M)_k y \colon k \in A_1\}.$$

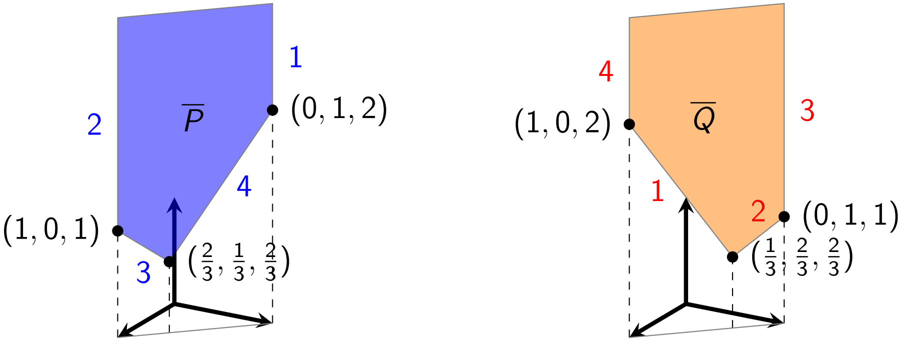
  Analogously, y is the best response to x if and only if for all $j \in A_2$,

  $$y_j > 0 \implies (N^\top)_j x = \max\{(N^\top)_k x \colon k \in A_2\}.$$

- Today, we reveal a geometric structure behind finding NE in bimatrix games and show one of the fastest known algorithms for this task.

Best response polyhedra $\overline{P}$ and $\overline{Q}$ for the Battle of sexes

Best response polyhedra $\overline{P}$ and $\overline{Q}$ for the Battle of sexes

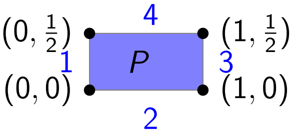


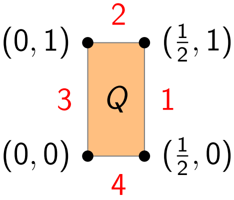
$$\overline{P} = \{(x_1, x_2, v) \in \mathbb{R}^2 \times \mathbb{R} \colon x_1, x_2 \geq 0, x_1 + x_2 = 1, x_1 \leq v, 2x_2 \leq v\}$$

$$\overline{Q} = \{(y_3, y_4, u) \in \mathbb{R}^2 \times \mathbb{R} \colon y_3, y_4 \geq 0, y_3 + y_4 = 1, 2y_3 \leq u, y_4 \leq u\}.$$

Best response polytopes P and Q for the Battle of sexes

Best response polytopes P and Q for the Battle of sexes





$$P = \{(x_1, x_2) \in \mathbb{R}^2 \colon x_1, x_2 \geq 0, x_1 \leq 1, 2x_2 \leq 1\}$$

$$Q = \{(y_3, y_4) \in \mathbb{R}^2 \colon y_3, y_4 \geq 0, 2y_3 \leq 1, y_4 \leq 1\}.$$

Algorithm for finding NE with vertex enumeration

Algorithm for finding NE with vertex enumeration

Algorithm 0.2: VERTEX ENUMERATION(G)

Input : A non-degenerate bimatrix game G.
Output : All Nash equilibria of G.
for each pair (x, y) of vertices from $(P \setminus \{\mathbf{0}\}) \times (Q \setminus \{\mathbf{0}\})$
$\begin{cases} \text{if } (x, y) \text{ is completely labeled,} \\ \text{then return } (x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y)) \text{ as a Nash equilibrium} \end{cases}$

Algorithm for finding NE with vertex enumeration

Algorithm 0.3: VERTEX ENUMERATION(G)

Input : A non-degenerate bimatrix game G.
Output : All Nash equilibria of G.
for each pair (x, y) of vertices from $(P \setminus \{\mathbf{0}\}) \times (Q \setminus \{\mathbf{0}\})$
$\begin{cases} \text{if } (x, y) \text{ is completely labeled,} \\ \text{then return } (x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y)) \text{ as a Nash equilibrium} \end{cases}$

- All vertices of a simple polytope in $\mathbb{R}^d$ with v vertices and N defining inequalities can be found in time $O(dNv)$ (Avis and Fukuda).

Algorithm for finding NE with vertex enumeration

Algorithm 0.4: VERTEX ENUMERATION(G)

Input : A non-degenerate bimatrix game G.
Output : All Nash equilibria of G.
for each pair (x, y) of vertices from $(P \setminus \{\mathbf{0}\}) \times (Q \setminus \{\mathbf{0}\})$
$\begin{cases} \text{if } (x, y) \text{ is completely labeled,} \\ \text{then return } (x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y)) \text{ as a Nash equilibrium} \end{cases}$

- All vertices of a simple polytope in $\mathbb{R}^d$ with v vertices and N defining inequalities can be found in time $O(dNv)$ (Avis and Fukuda).
- However, if $m = n$, the best response polytopes can have c^n vertices for some constant c with $1 < c < 2.9$.

Polytopes can be weird and complex!

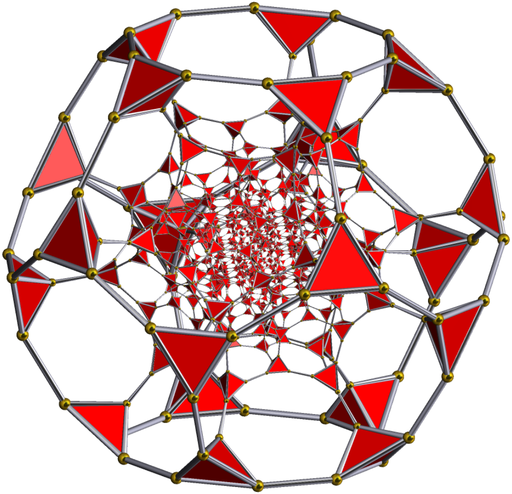


Figure: Schlegel diagram for the truncated 120-cell.

Source: https://en.wikipedia.org/

Algorithm for finding NE with vertex enumeration

Algorithm 0.5: VERTEX ENUMERATION(G)

Input : A non-degenerate bimatrix game G.
Output : All Nash equilibria of G.
for each pair (x, y) of vertices from $(P \setminus \{\mathbf{0}\}) \times (Q \setminus \{\mathbf{0}\})$
$\left\{ \begin{array}{l} \text{if } (x, y) \text{ is completely labeled,} \\ \text{then return } (x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y)) \text{ as a Nash equilibrium} \end{array} \right.$

- All vertices of a simple polytope in $\mathbb{R}^d$ with v vertices and N defining inequalities can be found in time $O(dNv)$ (Avis and Fukuda).
- However, if $m = n$, the best response polytopes can have c^n vertices for some constant c with $1 < c < 2.9$.

Algorithm for finding NE with vertex enumeration

Algorithm 0.6: VERTEX ENUMERATION(G)

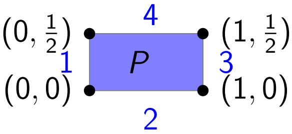
Input : A non-degenerate bimatrix game G.
Output : All Nash equilibria of G.
for each pair (x, y) of vertices from $(P \setminus \{\mathbf{0}\}) \times (Q \setminus \{\mathbf{0}\})$

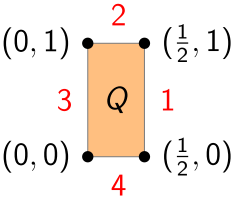
$$\begin{cases} \text{if } (x, y) \text{ is completely labeled,} \\ \text{then return } (x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y)) \text{ as a Nash equilibrium} \end{cases}$$

- All vertices of a simple polytope in $\mathbb{R}^d$ with v vertices and N defining inequalities can be found in time $O(dNv)$ (Avis and Fukuda).
- However, if $m = n$, the best response polytopes can have c^n vertices for some constant c with $1 < c < 2.9$.
- We can speed up the search by performing a walk on $(P \setminus \{\mathbf{0}\}) \times (Q \setminus \{\mathbf{0}\})$ guided by labels.

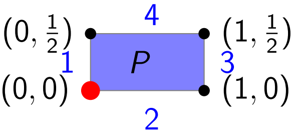
Lemke–Howson on the Battle of sexes ($k = 3$)

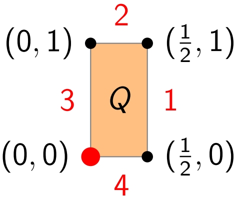
Lemke–Howson on the Battle of sexes ($k = 3$)





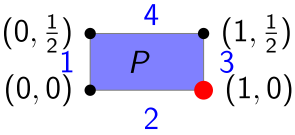
Lemke–Howson on the Battle of sexes ($k = 3$)

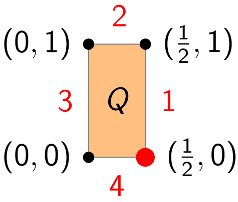




Lemke–Howson on the Battle of sexes ($k = 3$)

Lemke–Howson on the Battle of sexes ($k = 3$)





The Lemke–Howson algorithm

The Lemke–Howson algorithm

- Known as one of the best algorithms to find NE in bimatrix games.

The Lemke–Howson algorithm

- Known as one of the best algorithms to find NE in bimatrix games.
- Discovered by Lemke and Howson in 1964.

The Lemke–Howson algorithm

- Known as one of the best algorithms to find NE in bimatrix games.
- Discovered by Lemke and Howson in 1964.

Figure: Carlton E. Lemke (1920–2004) and J. T. Howson (1937–2022).

Source: https://oldurls.inf.ethz.ch

The Lemke–Howson algorithm: pseudocode

The Lemke–Howson algorithm: pseudocode

Algorithm 0.8: LEMKE–HOWSON(G)

Input : A nondegenerate bimatrix game G.
Output : One Nash equilibrium of G.
$(x, y) \leftarrow (\mathbf{0}, \mathbf{0}) \in \mathbb{R}^m \times \mathbb{R}^n$,
$k \leftarrow$ arbitrary label from $A_1 \cup A_2$, $l \leftarrow k$,
while (*true*)

do
- In P, drop l from x and redefine x as the new vertex, redefine l as the newly picked up label. Switch to Q. If $l = k$, stop looping.
- In Q, drop l from y and redefine y as the new vertex, redefine l as the newly picked up label. Switch to P. If $l = k$, stop looping.

Output $(x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y))$.

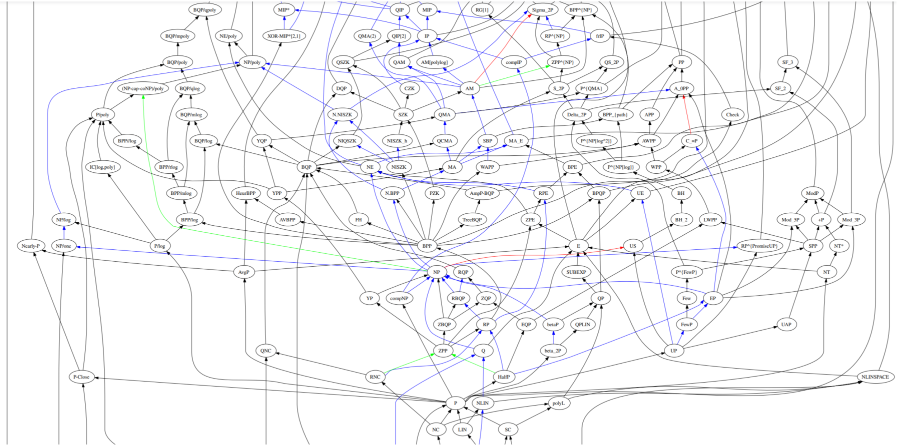


Figure: A view on the complexity classes classification.

Source: https://complexityzoo.uwaterloo.ca/Complexity_Zoo

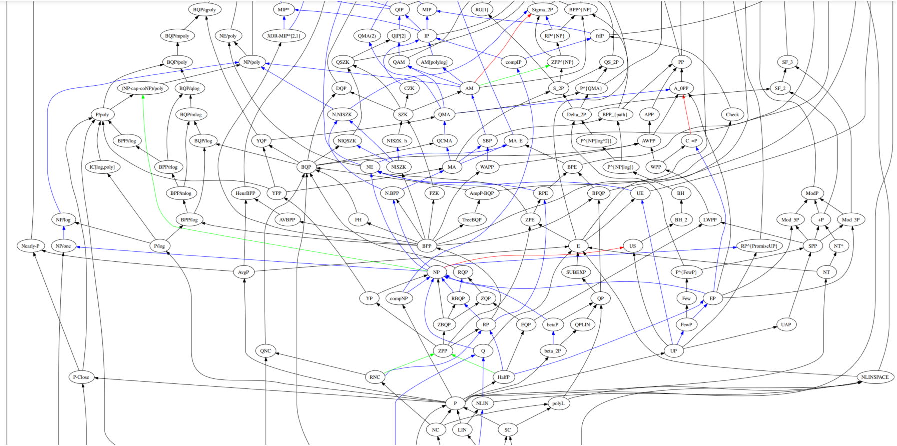


Figure: A view on the complexity classes classification.

Source: https://complexityzoo.uwaterloo.ca/Complexity_Zoo

Thank you for your attention.