Algorithmic game theory

Martin Balko

4th lecture

October 25th 2024

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 iff

$$\forall\, i \in A_1 : x_i > 0 \Longrightarrow M_{i,*}y = \max\{M_{k,*}y \colon k \in A_1\}.$$

Analogously, y is a best response to x iff

$$\forall\, j \in A_2 : y_j > 0 \Longrightarrow N^\top_{j,*}x = \max\{N^\top_{k,*}x \colon k \in A_2\}.$$

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 iff

$$\forall\, i \in A_1 : x_i > 0 \implies M_{i,*}y = \max\{M_{k,*}y \colon k \in A_1\}.$$

  Analogously, y is a best response to x iff

$$\forall\, j \in A_2 : y_j > 0 \implies N^{\top}_{j,*}x = \max\{N^{\top}_{k,*}x \colon k \in A_2\}.$$

- 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

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$,

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1}v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1}u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

$$u_1(s)$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron
$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1}v\}.$$
Similarly, we define the best response polyhedron for player 2 in G as
$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1}u\}.$$
- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always
$$u_1(s) = x^\top My$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

$$u_1(s) = x^\top My \leq x^\top \mathbf{1} u$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

$$u_1(s) = x^\top My \leq x^\top \mathbf{1} u = u \sum_{i \in A_1} x_i$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

$$u_1(s) = x^\top My \leq x^\top \mathbf{1} u = u \sum_{i \in A_1} x_i = u.$$

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

$$u_1(s) = x^\top My \leq x^\top \mathbf{1} u = u \sum_{i \in A_1} x_i = u.$$

- Analogously, the condition $N^\top x \leq \mathbf{1} v$ says that $u_2(s) \leq v$.

Best response polyhedra

- To reveal the geometric structure, we define labeled polyhedra for a bimatrix game G.
- The best response polyhedron for player 1 in G is a polyhedron

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

  Similarly, we define the best response polyhedron for player 2 in G as

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- Let $(x, v) \in \overline{P}$, $(y, u) \in \overline{Q}$ and let $s = (s_1, s_2)$ be a strategy profile with mixed strategy vectors x and y.
- Then, $My \leq \mathbf{1}u$ says that $u_1(s) \leq u$, as the expected payoff is always

$$u_1(s) = x^\top My \leq x^\top \mathbf{1} u = u \sum_{i \in A_1} x_i = u.$$

- Analogously, the condition $N^\top x \leq \mathbf{1} v$ says that $u_2(s) \leq v$.
- Thus, points of $\overline{P}$ and $\overline{Q}$ are the mixed strategies with the "upper envelope" of expected payoffs of the other player.

Labelings of the Best response polyhedra

Labelings of the Best response polyhedra

- Now, we define the labels of points from the best response polyhedra

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

and

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

Labelings of the Best response polyhedra

- Now, we define the labels of points from the best response polyhedra

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

and

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- We say that a point $(x, v) \in \overline{P}$ has label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $x_i = 0$ or if $i \in A_2$ and $N_{i,*}^\top x = v$.

Labelings of the Best response polyhedra

- Now, we define the labels of points from the best response polyhedra

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

and

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- We say that a point $(x, v) \in \overline{P}$ has label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $x_i = 0$ or if $i \in A_2$ and $N_{i,*}^\top x = v$.
  - That is, if the inequality in the definition of $\overline{P}$ is binding

Labelings of the Best response polyhedra

- Now, we define the labels of points from the best response polyhedra

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1}v\}.$$

  and

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1}u\}.$$

- We say that a point $(x, v) \in \overline{P}$ has label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $x_i = 0$ or if $i \in A_2$ and $N_{i,*}^\top x = v$.
  - That is, if the inequality in the definition of $\overline{P}$ is binding ($i \in A_1$ is not in the support or $i \in A_2$ is a best response to x).

Labelings of the Best response polyhedra

- Now, we define the labels of points from the best response polyhedra

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1} v\}.$$

and

$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1} u\}.$$

- We say that a point $(x, v) \in \overline{P}$ has label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $x_i = 0$ or if $i \in A_2$ and $N^\top_{i,*} x = v$.
  - That is, if the inequality in the definition of $\overline{P}$ is binding ($i \in A_1$ is not in the support or $i \in A_2$ is a best response to x).
- Similarly, a point $(y, u) \in \overline{Q}$ has a label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $M_{i,*} y = u$, or if $i \in A_2$ and $y_i = 0$.

Labelings of the Best response polyhedra

- Now, we define the labels of points from the best response polyhedra

$$\overline{P} = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} \colon x \geq \mathbf{0}, \mathbf{1}^\top x = 1, N^\top x \leq \mathbf{1}v\}.$$

  and

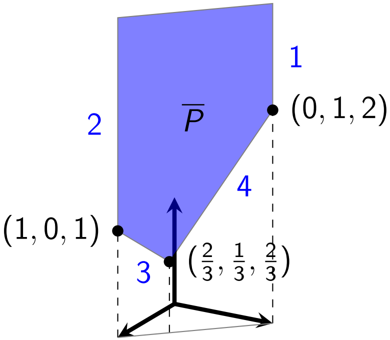
$$\overline{Q} = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} \colon y \geq \mathbf{0}, \mathbf{1}^\top y = 1, My \leq \mathbf{1}u\}.$$

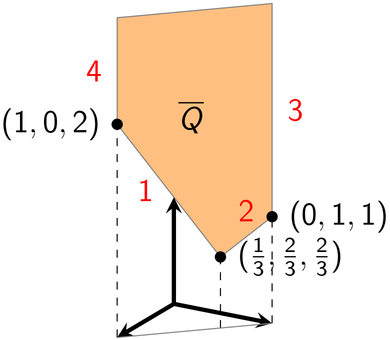
- We say that a point $(x, v) \in \overline{P}$ has label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $x_i = 0$ or if $i \in A_2$ and $N^\top_{i,*}x = v$.
  - That is, if the inequality in the definition of $\overline{P}$ is binding ($i \in A_1$ is not in the support or $i \in A_2$ is a best response to x).
- Similarly, a point $(y, u) \in \overline{Q}$ has a label $i \in A_1 \cup A_2$ if either $i \in A_1$ and $M_{i,*}y = u$, or if $i \in A_2$ and $y_i = 0$.

- Each point from $\overline{P}$ or $\overline{Q}$ may have more labels.

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





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

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

NE and Best response polyhedra

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof:

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.
- A missing label $i \in A_1$ means that a pure strategy $i \in Supp(s_1)$ $(x_i > 0)$ was not a best response $(M_{i,*}y < u)$.

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.
- A missing label $i \in A_1$ means that a pure strategy $i \in Supp(s_1)$ $(x_i > 0)$ was not a best response $(M_{i,*}y < u)$. Analogously for $i \in A_2$.

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.
- A missing label $i \in A_1$ means that a pure strategy $i \in Supp(s_1)$ $(x_i > 0)$ was not a best response $(M_{i,*}y < u)$. Analogously for $i \in A_2$. Then, s is not NE.

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.
- A missing label $i \in A_1$ means that a pure strategy $i \in Supp(s_1)$ $(x_i > 0)$ was not a best response $(M_{i,*}y < u)$. Analogously for $i \in A_2$. Then, s is not NE.
- If all labels appear, then s_1 and s_2 are mutually best responses, as each pure strategy is a best response or is not in the support.

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.
- A missing label $i \in A_1$ means that a pure strategy $i \in Supp(s_1)$ $(x_i > 0)$ was not a best response $(M_{i,*}y < u)$. Analogously for $i \in A_2$. Then, s is not NE.
- If all labels appear, then s_1 and s_2 are mutually best responses, as each pure strategy is a best response or is not in the support. Then, s is NE.

NE and Best response polyhedra

- What are the labels (and Best response polyhedra) for?
- They help us identify NE in G!

Proposition 2.27

A strategy profile s is NE of G iff the pair $((x, v), (y, u)) \in \overline{P} \times \overline{Q}$ is completely labeled, that is, every label $i \in A_1 \cup A_2$ appears as a label of either (x, v) or (y, u).

- Proof: By the Best response condition, for every player $i \in P$, a mixed strategy s_i is a best response to s_{-i} if and only if all pure strategies in the support of s_i are best responses to s_{-i}.
- A missing label $i \in A_1$ means that a pure strategy $i \in Supp(s_1)$ $(x_i > 0)$ was not a best response $(M_{i,*}y < u)$. Analogously for $i \in A_2$. Then, s is not NE.
- If all labels appear, then s_1 and s_2 are mutually best responses, as each pure strategy is a best response or is not in the support. Then, s is NE. □

Best response polytopes

Best response polytopes

- That is nice.

Best response polytopes

- That is nice. But we will make it even nicer!

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates).

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.
- We assume that M and $N^{\top}$ are non-negative and have no zero column.

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.
- We assume that M and $N^\top$ are non-negative and have no zero column. (simply add a large constant to the payoffs)

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.
- We assume that M and $N^\top$ are non-negative and have no zero column. (simply add a large constant to the payoffs)
- Then, we can divide each inequality $N_{i,*}^\top x \leq v$ with v, treating x_i/v as a new variable, and do the same for $\overline{Q}$.

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.
- We assume that M and $N^\top$ are non-negative and have no zero column. (simply add a large constant to the payoffs)
- Then, we can divide each inequality $N_{i,*}^\top x \leq v$ with v, treating x_i/v as a new variable, and do the same for $\overline{Q}$. This normalizes the payoffs to 1 and we get the following polytopes.

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.
- We assume that M and $N^\top$ are non-negative and have no zero column. (simply add a large constant to the payoffs)
- Then, we can divide each inequality $N^\top_{i,*}x \leq v$ with v, treating x_i/v as a new variable, and do the same for $\overline{Q}$. This normalizes the payoffs to 1 and we get the following polytopes.
- The (normalized) best response polytope for player 1 in G is a polytope

$$P = \{x \in \mathbb{R}^m \colon x \geq \mathbf{0}, N^\top x \leq \mathbf{1}\}.$$

Best response polytopes

- That is nice. But we will make it even nicer!
- The best response polyhedra have some unnecessary complications (they are unbounded and have u and v in their coordinates). We get rid of these under certain mild assumptions.
- We assume that M and $N^\top$ are non-negative and have no zero column. (simply add a large constant to the payoffs)
- Then, we can divide each inequality $N_{i,*}^\top x \leq v$ with v, treating x_i/v as a new variable, and do the same for $\overline{Q}$. This normalizes the payoffs to 1 and we get the following polytopes.
- The (normalized) best response polytope for player 1 in G is a polytope

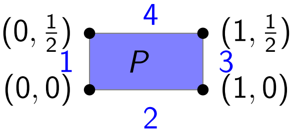
$$P = \{x \in \mathbb{R}^m \colon x \geq \mathbf{0}, N^\top x \leq \mathbf{1}\}.$$

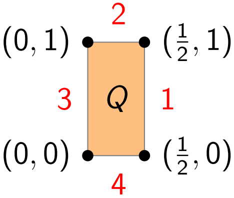
Similarly, the best response polytope for player 2 in G is a polytope

$$Q = \{y \in \mathbb{R}^n \colon y \geq \mathbf{0}, My \leq \mathbf{1}\}.$$

Best response polytopes P and Q for the Battle of sexes

Best response polytopes P and Q for the Battle of sexes





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

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

What did we get?

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^{\top} x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^{\top} x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^{\top} x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^\top x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.
- Disadvantage: coordinates of x and y do not sum up to 1.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^\top x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.
- Disadvantage: coordinates of x and y do not sum up to 1. We can rescale to $x/(\mathbf{1}^\top x)$ and $y/(\mathbf{1}^\top y)$.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^\top x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.
- Disadvantage: coordinates of x and y do not sum up to 1. We can rescale to $x/(\mathbf{1}^\top x)$ and $y/(\mathbf{1}^\top y)$.
- The polytope $\overline{P}$ is in a one-to-one correspondence with $P \setminus \{\mathbf{0}\}$ under the projective transformation $(x, v) \mapsto x/v$.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^\top x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.
- Disadvantage: coordinates of x and y do not sum up to 1. We can rescale to $x/(\mathbf{1}^\top x)$ and $y/(\mathbf{1}^\top y)$.
- The polytope $\overline{P}$ is in a one-to-one correspondence with $P \setminus \{\mathbf{0}\}$ under the projective transformation $(x, v) \mapsto x/v$. Similarly $\overline{Q}$ and $Q \setminus \{\mathbf{0}\}$.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^\top x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.
- Disadvantage: coordinates of x and y do not sum up to 1. We can rescale to $x/(\mathbf{1}^\top x)$ and $y/(\mathbf{1}^\top y)$.
- The polytope $\overline{P}$ is in a one-to-one correspondence with $P \setminus \{\mathbf{0}\}$ under the projective transformation $(x, v) \mapsto x/v$. Similarly $\overline{Q}$ and $Q \setminus \{\mathbf{0}\}$.
    - Projective transformations preserve incidences, so the labels stay the same.

What did we get?

- The inequalities have the same meaning: if $x_i \geq 0$ is binding, then $i \in A_1$ is not in the support and if $N_{j,*}^\top x \leq 1$ is binding, then $j \in A_2$ is a best response to s_1. Analogously for Q.
- From the assumption on M and N, the polyhedra P and Q are bounded (are polytopes) and have dimensions m and n.
- Disadvantage: coordinates of x and y do not sum up to 1. We can rescale to $x/(\mathbf{1}^\top x)$ and $y/(\mathbf{1}^\top y)$.
- The polytope $\overline{P}$ is in a one-to-one correspondence with $P \setminus \{\mathbf{0}\}$ under the projective transformation $(x, v) \mapsto x/v$. Similarly $\overline{Q}$ and $Q \setminus \{\mathbf{0}\}$.
  - Projective transformations preserve incidences, so the labels stay the same.

Corollary 2.30

A strategy profile (s_1, s_2) with mixed strategy vectors x and y is NE of G if and only if the point $(x/u_2(s), y/u_1(s)) \in P \times Q \setminus \{(\mathbf{0}, \mathbf{0})\}$ is completely labeled. □

NE in nondegenerate games

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.
- In these games NE correspond to pairs of completely labeled vertices.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.
- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m - k$ labels in A_1

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m - k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m - k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2. Analogously, each point of Q has at most n labels.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m - k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2. Analogously, each point of Q has at most n labels.
  - Thus, P and Q are both simple polytopes

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m - k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2. Analogously, each point of Q has at most n labels.
  - Thus, P and Q are both simple polytopes (each point of P or Q contained in more than m or n facets has more than m or n labels).

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m-k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2. Analogously, each point of Q has at most n labels.
  - Thus, P and Q are both simple polytopes (each point of P or Q contained in more than m or n facets has more than m or n labels).
  - Since $dim(P) = m$ and $dim(Q) = n$, only vertices of P and Q can have m and n labels.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.
- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m-k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2. Analogously, each point of Q has at most n labels.
  - Thus, P and Q are both simple polytopes (each point of P or Q contained in more than m or n facets has more than m or n labels).
  - Since $dim(P) = m$ and $dim(Q) = n$, only vertices of P and Q can have m and n labels.
  - By Corollary 2.30, only vertices of P and Q can be NE.

NE in nondegenerate games

- Recall that a bimatrix game is nondegenerate if there are at most k pure best responses to every mixed strategy with support of size k.

- In these games NE correspond to pairs of completely labeled vertices.
  - Since G is nondegenerate, each point of P has at most m labels. This is because if x has support of size k, then x has at most $m - k$ labels in A_1 and so if x had more than m labels, then x would have more than k best responses in A_2. Analogously, each point of Q has at most n labels.
  - Thus, P and Q are both simple polytopes (each point of P or Q contained in more than m or n facets has more than m or n labels).
  - Since $dim(P) = m$ and $dim(Q) = n$, only vertices of P and Q can have m and n labels.
  - By Corollary 2.30, only vertices of P and Q can be NE.

- $\Rightarrow$ Algorithm for finding NE: check all pairs of vertices and their labels!

Algorithm for finding NE with vertex enumeration

Algorithm for finding NE with vertex enumeration

Algorithm 0.2: VERTEX ENUMERATION(G)

Input : A nondegenerate 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 nondegenerate 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 nondegenerate 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 $1 < c < 2.9$.

Polytopes can be weird and complex!

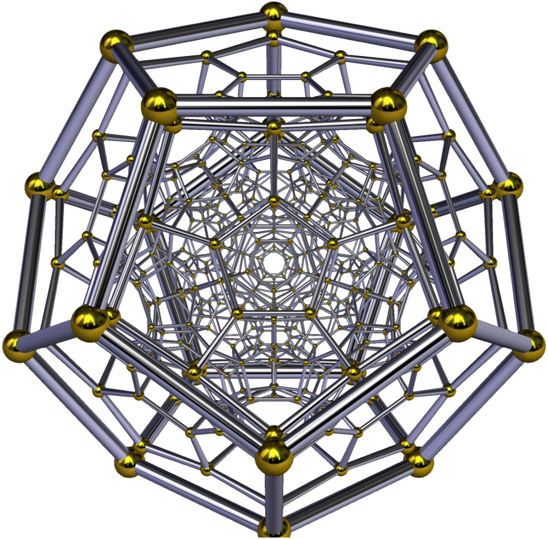


Figure: Schlegel diagram for the 120-cell.

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

Algorithm for finding NE with vertex enumeration

Algorithm 0.5: VERTEX ENUMERATION(G)

Input : A nondegenerate 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 nondegenerate 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 the labels.

The Lemke–Howson algorithm

The Lemke–Howson algorithm

- One of the best algorithms for finding NE in bimatrix games.

The Lemke–Howson algorithm

- One of the best algorithms for finding NE in bimatrix games.
- Discovered by Lemke and Howson in 1964.

The Lemke–Howson algorithm

- One of the best algorithms for finding 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 explained

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it. Then, a new label l is picked up.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it. Then, a new label l is picked up. This label l has a duplicate in the other polytope.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it. Then, a new label l is picked up. This label l has a duplicate in the other polytope. We drop the duplicate of l in the other polytope in the next step,

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it. Then, a new label l is picked up. This label l has a duplicate in the other polytope. We drop the duplicate of l in the other polytope in the next step, which leads to picking up a new label l'.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it. Then, a new label l is picked up. This label l has a duplicate in the other polytope. We drop the duplicate of l in the other polytope in the next step, which leads to picking up a new label l'. We iterate and stop when $l' = k$.

The Lemke–Howson algorithm explained

- Since P is simple and $dim(P) = m$, its vertices are incident to exactly m facets and m edges. So each vertex of P has m labels and each edge of P has $m - 1$ labels. Similarly for Q and n.
- Dropping a label $l \in A_1 \cup A_2$ in a vertex x of P means traversing the unique edge of P that is incident to x and has all m labels that x has besides l. The other endpoint of this edge has the same labels as x, only l is replaced with a new label, which is said to be picked up. Dropping and picking up a label in vertices of Q is defined analogously.
- The algorithm starts at $(\mathbf{0}, \mathbf{0})$ and alternately follows edges of P and Q.
- At the first step, it chooses a label $k \in A_1 \cup A_2$ and drops it. Then, a new label l is picked up. This label l has a duplicate in the other polytope. We drop the duplicate of l in the other polytope in the next step, which leads to picking up a new label l'. We iterate and stop when $l' = k$.
- Duplicate label is either a new best response, which gets a positive probability, or a pure strategy whose probability became 0 and we move away from its best response facet.

The Lemke–Howson algorithm: pseudocode

The Lemke–Howson algorithm: pseudocode

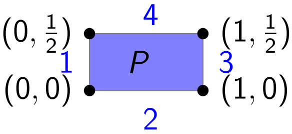
Algorithm 0.8: LEMKE–HOWSON(G)

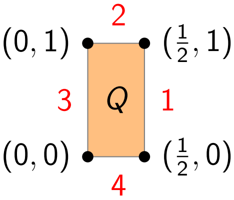
Input : A nondegenerate bimatrix game G.
Output : One Nash equilibrium of G.
$(x, y) \leftarrow (\mathbf{0}, \mathbf{0}) \in \mathbb{R}^m \times \mathbb{R}^n$,
$k \leftarrow$ arbitrary label from $A_1 \cup A_2$, $l \leftarrow k$,
while (*true*)
do {
In P, drop l from x and redefine x as the new vertex, 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 $(x/(\mathbf{1}^\top x), y/(\mathbf{1}^\top y))$.

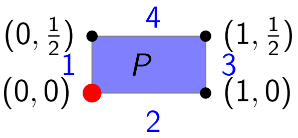
Lemke–Howson on the Battle of sexes ($k = 3$)

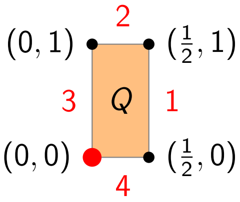
Lemke–Howson on the Battle of sexes ($k = 3$)



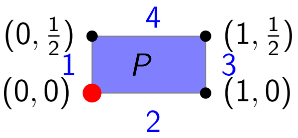


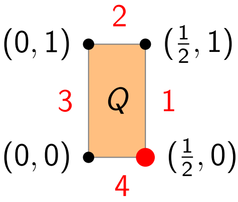
Lemke–Howson on the Battle of sexes ($k = 3$)



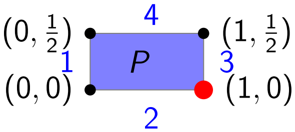


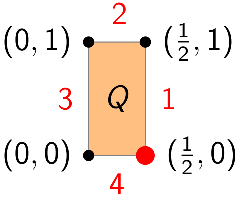
Lemke–Howson on the Battle of sexes ($k = 3$)





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





Correctness of the Lemke–Howson algorithm I

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof:

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y).

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \,\&\, yy' \in E(Q))$ or $(xx' \in E(P) \,\&\, y = y')$.

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \ \& \ yy' \in E(Q))$ or $(xx' \in E(P) \ \& \ y = y')$. Clearly, $\mathcal{G}$ is finite.

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \ \& \ yy' \in E(Q))$ or $(xx' \in E(P) \ \& \ y = y')$. Clearly, $\mathcal{G}$ is finite.
- $\mathcal{G}$ has degrees only 1 or 2

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \;\&\; yy' \in E(Q))$ or $(xx' \in E(P) \;\&\; y = y')$. Clearly, $\mathcal{G}$ is finite.
- $\mathcal{G}$ has degrees only 1 or 2 ($\mathcal{G}$ is a disjoint union of paths and cycles).

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \,\&\, yy' \in E(Q))$ or $(xx' \in E(P) \,\&\, y = y')$. Clearly, $\mathcal{G}$ is finite.
- $\mathcal{G}$ has degrees only 1 or 2 ($\mathcal{G}$ is a disjoint union of paths and cycles).
    - If (x, y) has all labels from $A_1 \cup A_2$, then (x, y) is connected to exactly one other vertex,

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \;\&\; yy' \in E(Q))$ or $(xx' \in E(P) \;\&\; y = y')$. Clearly, $\mathcal{G}$ is finite.
- $\mathcal{G}$ has degrees only 1 or 2 ($\mathcal{G}$ is a disjoint union of paths and cycles).
  - If (x, y) has all labels from $A_1 \cup A_2$, then (x, y) is connected to exactly one other vertex, as exactly one of x and y has the label k and we can drop k only from this one vertex.

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \;\&\; yy' \in E(Q))$ or $(xx' \in E(P) \;\&\; y = y')$. Clearly, $\mathcal{G}$ is finite.
- $\mathcal{G}$ has degrees only 1 or 2 ($\mathcal{G}$ is a disjoint union of paths and cycles).
    - If (x, y) has all labels from $A_1 \cup A_2$, then (x, y) is connected to exactly one other vertex, as exactly one of x and y has the label k and we can drop k only from this one vertex.
    - Otherwise, (x, y) has all labels from $A_1 \cup A_2 \setminus \{k\}$ and there is a unique label shared by x and y.

Correctness of the Lemke–Howson algorithm I

Proposition 2.31

The Lemke–Howson algorithm stops after a finite number of steps and outputs mixed strategy vectors of NE in G.

- Proof: Let k be the label chosen in the first step.
- We define a configuration graph $\mathcal{G}$ with the vertices formed by pairs (x, y) of vertices from $P \times Q$ that are k-almost completely labeled (every label from $A_1 \cup A_2 \setminus \{k\}$ is a label of x or y). A pair $\{(x, y), (x', y')\}$ is an edge of $\mathcal{G}$ if $(x = x' \;\&\; yy' \in E(Q))$ or $(xx' \in E(P) \;\&\; y = y')$. Clearly, $\mathcal{G}$ is finite.
- $\mathcal{G}$ has degrees only 1 or 2 ($\mathcal{G}$ is a disjoint union of paths and cycles).
    - If (x, y) has all labels from $A_1 \cup A_2$, then (x, y) is connected to exactly one other vertex, as exactly one of x and y has the label k and we can drop k only from this one vertex.
    - Otherwise, (x, y) has all labels from $A_1 \cup A_2 \setminus \{k\}$ and there is a unique label shared by x and y. Then (x, y) is adjacent to two vertices, as we can drop the duplicate label from x in P or y in Q.

Correctness of the Lemke–Howson algorithm II

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
    - The next vertex pair on the path is always unique

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).
  - Visiting the same vertex twice would give a vertex of degree larger than 2 in $\mathcal{G}$ since we started at a leaf.

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).
  - Visiting the same vertex twice would give a vertex of degree larger than 2 in $\mathcal{G}$ since we started at a leaf.
- Thus, the algorithm terminates after a finite number of steps in the other leaf (x^*, y^*) of the path.

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).
  - Visiting the same vertex twice would give a vertex of degree larger than 2 in $\mathcal{G}$ since we started at a leaf.
- Thus, the algorithm terminates after a finite number of steps in the other leaf (x^*, y^*) of the path. Since (x^*, y^*) is a leaf in the configuration graph, it is completely labeled.

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).
  - Visiting the same vertex twice would give a vertex of degree larger than 2 in $\mathcal{G}$ since we started at a leaf.
- Thus, the algorithm terminates after a finite number of steps in the other leaf (x^*, y^*) of the path. Since (x^*, y^*) is a leaf in the configuration graph, it is completely labeled.
- This endpoint is not of the form $(x, \mathbf{0})$ or $(\mathbf{0}, y)$ (Exercise).

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).
  - Visiting the same vertex twice would give a vertex of degree larger than 2 in $\mathcal{G}$ since we started at a leaf.
- Thus, the algorithm terminates after a finite number of steps in the other leaf (x^*, y^*) of the path. Since (x^*, y^*) is a leaf in the configuration graph, it is completely labeled.
- This endpoint is not of the form $(x, \mathbf{0})$ or $(\mathbf{0}, y)$ (Exercise).
- By Corollary 2.30, (x^*, y^*) corresponds to NE after rescaling.

Correctness of the Lemke–Howson algorithm II

- The Lemke–Howson algorithm starts at the leaf $(\mathbf{0}, \mathbf{0})$ of a path in $\mathcal{G}$.
- Then it walks along this path and does not visit any vertex of the configuration graph twice.
  - The next vertex pair on the path is always unique (we move to the other endpoint of the unique edge that contains the current vertex and corresponds to the dropped label).
  - Visiting the same vertex twice would give a vertex of degree larger than 2 in $\mathcal{G}$ since we started at a leaf.
- Thus, the algorithm terminates after a finite number of steps in the other leaf (x^*, y^*) of the path. Since (x^*, y^*) is a leaf in the configuration graph, it is completely labeled.
- This endpoint is not of the form $(x, \mathbf{0})$ or $(\mathbf{0}, y)$ (Exercise).
- By Corollary 2.30, (x^*, y^*) corresponds to NE after rescaling. □

The Lemke–Howson algorithm: remarks

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

- Degenerate games can have infinite number of NE.

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

- Degenerate games can have infinite number of NE.
- The Lemke–Howson algorithm finds only a single NE.

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

- Degenerate games can have infinite number of NE.
- The Lemke–Howson algorithm finds only a single NE.
- The algorithm can be implemented using so-called complementary pivoting (we will see it at the tutorials).

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

- Degenerate games can have infinite number of NE.
- The Lemke–Howson algorithm finds only a single NE.
- The algorithm can be implemented using so-called complementary pivoting (we will see it at the tutorials).
- The running time can still be exponential ($O(2^n)$ steps for $n = m$)!

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

- Degenerate games can have infinite number of NE.
- The Lemke–Howson algorithm finds only a single NE.
- The algorithm can be implemented using so-called complementary pivoting (we will see it at the tutorials).
- The running time can still be exponential ($O(2^n)$ steps for $n = m$)! It performs well in practise (polynomial on uniformly random games).

The Lemke–Howson algorithm: remarks

- By the Degree sum formula, the number of vertices of degree 1 is even.

Corollary 2.32

A nondegenerate bimatrix game has an odd number of NE.

- Degenerate games can have infinite number of NE.
- The Lemke–Howson algorithm finds only a single NE.
- The algorithm can be implemented using so-called complementary pivoting (we will see it at the tutorials).
- The running time can still be exponential ($O(2^n)$ steps for $n = m$)! It performs well in practise (polynomial on uniformly random games).

- Is there an efficient algorithm to find NE?

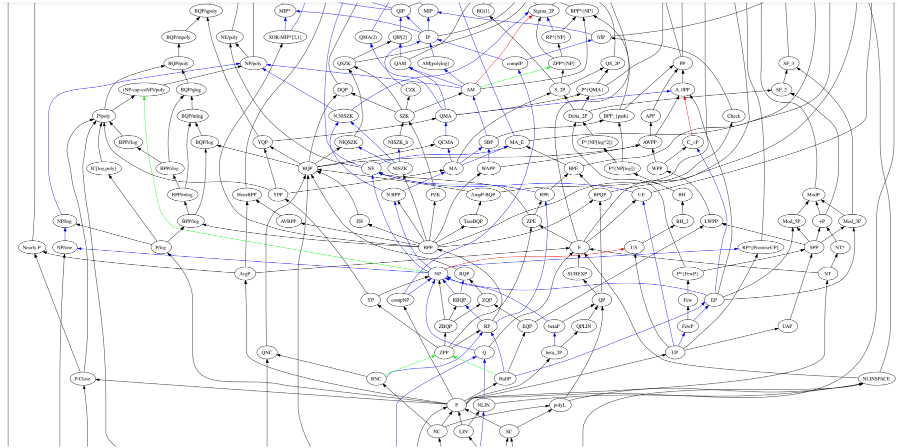


Figure: A view on the complexity classes classification.

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

- "P=NP" is one of the most important problems in computer science. The website https://www.win.tue.nl/~gwoegi/P-versus-NP.htm contains a collection of over 100 attempts to solve it.

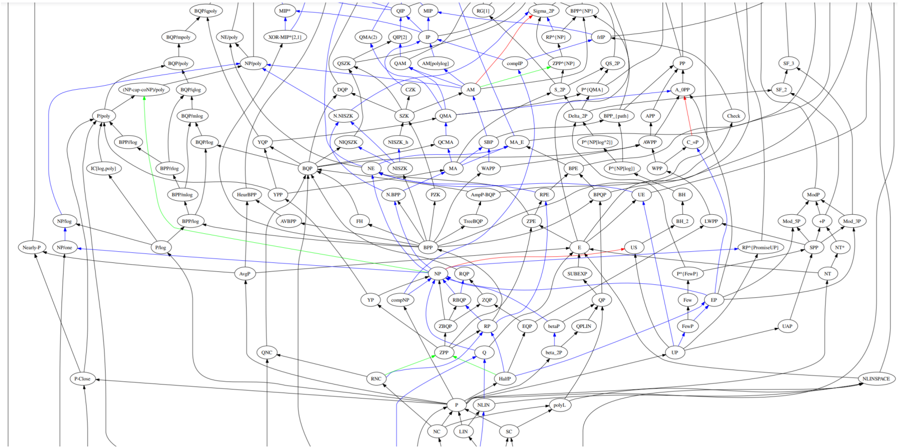


Figure: A view on the complexity classes classification.

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

- "P=NP" is one of the most important problems in computer science. The website https://www.win.tue.nl/~gwoegi/P-versus-NP.htm contains a collection of over 100 attempts to solve it.

Thank you for your attention.