Algorithmic Game Theory - Computing Nash Equilibria

Branislav Bošanský

Czech Technical University in Prague

branislav.bosansky@agents.fel.cvut.cz

March 20, 2018

Computing a Nash Equilibrium

computing a Nash Equilibrium in Bimatrix Games

there are two matrices of utility values $A, B \in \mathbb{R}^{M \times N}$, where player 1 has m actions and player 2 has n actions

we are going to use the following indexes:

$$M = \{1, \ldots, m\} \qquad\qquad N = \{m+1, \ldots, m+n\}$$

Theorem (Best response condition)

Let x and y be mixed strategies of player 1 and 2, respectively. Then x is a best response to y if and only if for all $i \in M$

$$x_i > 0 \ \Rightarrow (Ay)_i = u = \max\{(Ay)_k : k \in M\}$$

Computing a Nash Equilibrium

Definition (Nondegenerate games)

A two-player game is called *nondegenerate* if no mixed strategy of support size k has more than k pure best responses.

Lemma (Nondegenerate games)

In any Nash equilibrium (x, y) *of a nondegenerate bimatrix game,* x *and* y *have supports of equal size.*

we can use this observation for the first algorithm:

Equilibria by support enumeration

Equilibria by support enumeration

Method: For each $k = 1, \ldots, \min\{m, n\}$ and each pair (I, J) of k-sized subsets of M and N, respectively, solve the equations:

$$\sum_{i \in I} x_i b_{ij} = v \qquad \text{for} \qquad \forall j \in J, \sum_{i \in I} x_i = 1,$$

$$\sum_{j \in J} a_{ij} y_j = u \qquad \text{for} \qquad \forall i \in I, \sum_{j \in J} y_j = 1,$$

and check that $x \geq \mathbf{0}$, $y \geq \mathbf{0}$, and that both x and y satisfy the best response condition.

Equilibria by Labeled Polytopes

we will use *best response polyhedra*: the set of mixed strategies together with the "upper envelope" of expected payoffs (and any larger payoffs) to the *other* player.

consider an example game

$$A = \begin{bmatrix} 3 & 3 \\ 2 & 5 \\ 0 & 6 \end{bmatrix}, \qquad B = \begin{bmatrix} 3 & 2 \\ 2 & 6 \\ 3 & 1 \end{bmatrix}$$

BR polyhedron $\overline{Q}$ is the set of triplets (y_4, y_5, u) that satisfy:

$$\begin{aligned} 3y_4 + 3y_5 &\leq u \\ 2y_4 + 5y_5 &\leq u \\ 0y_4 + 6y_5 &\leq u \\ y_4 \geq 0, y_5 &\geq 0, y_4 + y_5 = 1 \end{aligned}$$

Equilibria by Labeled Polytopes

Generally:

$$\overline{P} = \{(x, v) \in \mathbb{R}^M \times \mathbb{R} : x \geq \mathbf{0}, \mathbf{1}^\top x = 1, B^\top x \leq \mathbf{1}v\}$$
$$\overline{Q} = \{(y, u) \in \mathbb{R}^N \times \mathbb{R} : Ay \leq \mathbf{1}u, y \geq \mathbf{0}, \mathbf{1}^\top y = 1\}$$

each vertex of the polyhedron $\overline{Q}$ has *label* $k \in M \cup N$, for which k-th inequality in the definition of $\overline{Q}$ is binding:

$$\begin{cases} \sum_{j \in N} a_{kj} y_j = u & \text{if } k \in M \\ y_k = 0 & \text{if } k \in N \end{cases}$$

An equilibrium is a pair (x, y) of mixed strategies so that with the corresponding expected payoffs v and u, the pair $((x, v), (y, u))$ in $\overline{P} \times \overline{Q}$ is *completely labeled*.

Equilibria by Labeled Polytopes

We can simplify polyhedra by removing the expected values

$$P = \{x \in \mathbb{R}^M : x \geq \mathbf{0}, B^\top x \leq \mathbf{1}\}$$
$$Q = \{y \in \mathbb{R}^N : Ay \leq \mathbf{1}, y \geq \mathbf{0}\}$$

New vectors $x \in P$ and $y \in Q$ are not mixed strategies – they need to be scaled by $v = \frac{1}{\mathbf{1}^\top x}$, or $u = \frac{1}{\mathbf{1}^\top y}$, respectively.

This transformation preserves the labels on vertexes, since a binding inequality in $\overline{P}$ corresponds to a binding inequality in P (and the same holds for Q).

Equilibria by Vertex Enumeration

we can use the polytopes P and Q to improve the algorithm for finding all Nash equilibria

For each vertex x of $P - \{\mathbf{0}\}$, and each vertex y of $Q - \{\mathbf{0}\}$, if (x, y) is completely labeled, then $\left(x \cdot \frac{1}{\mathbf{1}^\top x}, y \cdot \frac{1}{\mathbf{1}^\top y}\right)$ is a Nash equilibrium.

A more efficient approach compared to the support enumeration.

The Lemke-Howson Algorithm

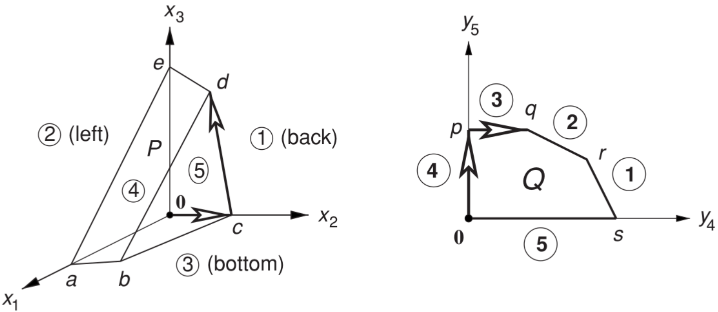
we assign labels to edges of the polytopes – since we are in nondegenerate polytopes, each vertex has m (or n, respectively) labels, and an edge has $m-1$ labels.

To *drop a label* l means to move from vertex x by an edge that has all labels but l.

LH starts from $(\mathbf{0}, \mathbf{0})$ by dropping some label.

At the end of the corresponding edge, a new label is picked-up that is a duplicate. Therefore, we must drop this label in the second polytope. If there is no duplicate, we can output a Nash equilibrium.

The Lemke-Howson Algorithm



Degenerate Games

What about degenerate games?

- there can be infinitely many Nash equilibria
- Lemke-Howson algorithm may fail since the continuation is not unique
- one needs to create a perturbed game

Theorem

Let (A, B) be a bimatrix game, and $(x, y) \in P \times Q$. Then (x, y) (rescaled) is a Nash equilibrium if and only if there is a set U of vertices of $P - \{\mathbf{0}\}$ and a set V of vertices of $Q - \{\mathbf{0}\}$ so that $x \in$ conv$U$ and $y \in$ conv$V$, and every $(u, v) \in U \times V$ is completely labeled.

Equilibria by LCP/MILP Mathematical Programs

LCP formulation:

$$
\begin{aligned}
\sum_{j \in N} a_{ij} y_j + q_i =& u \qquad \forall i \in M \\
\sum_{i \in M} b_{ij} x_i + p_j =& v \qquad \forall j \in N \\
\sum_{i \in M} x_i = 1 \quad \sum_{j \in N} y_j =& 1 \\
x_i \geq 0,\ p_i \geq 0,\ y_j \geq 0,\ q_j \geq& 0 \qquad \forall i \in M, \forall j \in N \\
x_i \cdot p_i = 0,\ y_j \cdot q_j =& 0 \qquad \forall i \in M, \forall j \in N
\end{aligned}
$$

Equilibria by LCP/MILP Mathematical Programs

MILP formulation:

$$
\begin{aligned}
\sum_{j \in N} a_{ij} y_j + q_i &= u && \forall i \in M \\
\sum_{i \in M} b_{ij} x_i + p_j &= v && \forall j \in N \\
\sum_{i \in M} x_i = 1 \quad \sum_{j \in N} y_j &= 1 \\
w_i, z_j \in \{0,1\},\ w_i \geq x_i \geq 0,\ z_j \geq y_j &\geq 0 && \forall i \in M, \forall j \in N \\
0 \leq p_i \leq (1 - w_i) Z, \quad 0 \leq q_j &\leq (1 - z_j) Z && \forall i \in M, \forall j \in N
\end{aligned}
$$

Nash and Correlated Equilibria in Bimatrix Games

Corollary

A nondegenerate bimatrix game has an odd number of Nash equilibria.

Can you construct a game that has a Nash equilibrium that cannot be found by the Lemke-Howson algorithm?

What about degenerate games? They may have infinite number of Nash equilibria (convex combinations of "extreme" equilibria).

What is the relation between CE and NE in bimatrix games?

Nash Equilibria in Bimatrix Games

There are 3 main algorithms:

- support enumeration search (PNS; R. Porter, E. Nudelman, and Y. Shoham, "Simple search methods for finding a Nash equilibrium," in AAAI, 2004, pp. 664669.)
- Lemke-Howson (LH; C. Lemke and J. Howson, "Equilibrium points of bimatrix games," SIAM J APPL MATH, vol. 12, no. 2, pp. 413423, 1964.)
- MILP variants (MILP; T. Sandholm, A. Gilpin, and V. Conitzer, "Mixed-integer programming methods for finding Nash equilibria," in AAAI, Pittsburgh, USA, 2005, pp. 495501.)

advantages/disadvantages:

- LH and PNS are typically faster than MILP
- MILP is much better when a specific equilibrium needs to be found
- MILP performance is getting better over time as the development of solver evolves

Homework assignment 1

Let $\Gamma = (N, S, u)$ and $\hat{\Gamma} = (N, S, \hat{u})$ be two normal-form games with the same sets of players and the same sets of pure strategies such that $u_i(s) \geq \hat{u}_i(s)$ for all players $i \in N$. Is it necessarily true that for each equilibrium σ of Γ there exists an equilibrium $\hat{\sigma}$ of $\hat{\Gamma}$ such that $u_i(\sigma) \geq \hat{u}_i(\hat{\sigma})$? Prove this claim or find a counterexample.