Algorithmic Game Theory

Arno Pauly

Computer Laboratory
University of Cambridge

Outline

1. Bimatrix Games & Nash equilibria
2. The Lemke-Howson-Algorithm
3. Complexity class: PPAD
4. Network Games
5. Complexity class: PLS

Informal Introduction : Finite Games

- Several Actors (=players), each with his own goals
- Every player has some finite set of potential actions
- The final outcome depends on the actions chosen by all players
- Every player may evaluate outcomes differently

Informal Introduction: Equilibria

- An equilibrium is a choice of actions, so that no player can improve the final outcome (from her point of view) by unilaterally deviating.
- Such equilibria do not exist in general.
- Solution: Introduce randomization.
- Each player picks a probability distribution over her actions.
- and tries to maximize the *expected* value of the outcome
- Now (Nash) equilibria always exist.

Formal definitions

Definition

An $n \times m$ two-player game in normal form is given by two $n \times m$ matrices A, B.

Definition

The set of stochastic vectors of size n is defined via:

$$\mathcal{S}^n := \{x \in \mathbb{R}^n \mid \forall i \leq n \; x_i \geq 0 \; \wedge \; \sum_{i=1}^{n} s_i = 1\}$$

Definition

A Nash equilibrium of (A, B) is a pair $(x, y) \in \mathcal{S}^n \times \mathcal{S}^m$ satisfying:

1. $x^T Ay \geq z^T Ay$ for all $z \in \mathcal{S}^n$
2. $x^T By \geq x^T Bz$ for all $z \in \mathcal{S}^m$

Lemke-Howson-Algorithm: The Setting

- Assumption: A and B are non-degenerate integer (or rational) matrices.
- Consider the polytopes $P := \{x \in \mathbb{R}^n \mid x \geq 0 \ \wedge \ B^T x \leq 1\}$ and $Q := \{y \in \mathbb{R}^m \mid Ay \leq 1 \ \wedge \ y \geq 0\}$ (defined by $n + m$ inequalities each)
- $x \in P$ has label $k \leq n + m$, if the kth inequality is strict. Same for $y \in Q$.
- Vertices of the polytopes are rational.

Lemke-Howson-Algorithm: The Goal

Let x be a vertex of P and y be a vertex of Q, so that each label $k \leq n + m$ appears at x or y. Then either $(x, y) = (0, 0)$, or a Nash equilibrium can be obtained as $x' := (\sum_{i=1}^{n} x_i)^{-1} x$ and $y' := (\sum_{j=1}^{m} y_j)^{-1} y$.

Lemke-Howson-Algorithm: What we do

1. Start at $(0,0)$, pick some $k \leq n+m$ and move along the adjacent edge in P without the label k.
2. At the next vertex, some new label l appears. Move along the edge in Q without l.
3. Some new label appears. If it is k, we have found a completely labelled vertex $\neq (0,0)$.
4. Otherwise, move along the edge in P without the new label..
5. Repeat until termination.

Lemke-Howson-Algorithm: Abstract View

We search for sinks or (non-trivial) sources in an implicitly defined exponentially large directed graph consisting of paths and circles.

PPAD: The generic problem

Definition

The problem Source-or-Sink takes as its input 2 poly-sized circuits computing functions $P, S : \{0,1\}^n \rightarrow \{0,1\}^n$ with $\forall w \in \{0,1\}^n$ either $S(w) = w$ or $P(S(w)) = w$, and either $P(w) = w$ or $S(P(w)) = w$, as well as $P(0^n) = 0^n$. The solution is some $w \in \{0,1\}^n \setminus \{0^n\}$ with $P(w) = w$ or $S(w) = w$.

PPAD: The complexity class

Definition

Let PPAD denote the class of all search problems polynomial-time reducible to Source-or-Sink.

Proposition

$FP \subseteq PPAD \subseteq FNP$

Proposition

Relative to a generic oracle, all the inclusion above are proper.

PPAD-completeness

The following problems are PPAD-complete:

1. Find a Nash equilibrium (in a 2-player normal form game).
2. Find a (weak) approximation of a Nash equilibrium (in an *n*-player normal form game).
3. Find a (weak) approximation of a Nash equilibrium in a graphical game.
4. Find a Brouwer Fixed Point (of a suitably represented function).
5. Find a Sperner-colouring in 3 dimensions.

Network Congestion Games: Definition

- A network congestion game is played by N players on a directed graph.
- For each edge e, there is a monotone function $d_e : \{1, \ldots, N\} \to \mathbb{N}$.
- For each player p, there are vertices s_p and t_p (so that there is a path from s_p to t_p).
- Each player picks some path from his source vertex s_p to his target vertex t_p.
- If edge e is used by k players, then each player using k suffers a delay of $d_e(k)$.
- Each player tries to minimize the total delay on her path.

Network Congestion Games: Solutions

- We search for a path-assignment where no player has incentive to deviate.
- If all players have the same source and target vertex, we can use minimal cuts to find a solution in polynomial time.
- Otherwise, we can do local improvements by searching for an alternative path for a single player, so that the sum of delays incurred by all players decreases.
- Iteration converges to a solution, but might take exponentially many steps.

PLS: Abstract View

PLS: Search for a sink in an implicitly defined exponentially large directed acyclic graph.

PLS: Generic Problem

Definition

The problem Circuit-Flip takes as input a poly-sized circuit computing a function $F : \{0,1\}^n \rightarrow \{0,1\}^n$, and produces a $w \in \{0,1\}^n$, so that for all $v \in \{0,1\}^*$ with $|w, v| \leq 1$ we have $F(v) \leq F(w)$ lexicographically.

Definition

Let *PLS* denote the class of all search problems polynomial-time reducible to Circuit-Flip.

PLS: Complexity class

Proposition

$FP \subseteq PLS \subseteq FNP$.

Proposition

Relative to a generic oracle, all the inclusion above are proper.

Proposition

Solving network congestion games is PLS-complete.

If you want more…

N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani
Algorithmic Game Theory.
Cambridge University Press, 2007.