Complexity of Solving Strategic Games

The basic problem:

NASH: Given a finite 2-player strategic game G, **find** a mixed strategy profile (α, β) that is a NE of G [if one exists, else return "no".]

⇐ Difference to SAT: existence of NE is guaranteed!

In this form NASH looks similar to other search problems, e.g.:

SAT: Given a Boolean formula φ in CNF, **find** a truth assignment that makes φ true if one exists, else return "no".

⇐ Search version of the usual decision problem

A **search problem** is given by a binary relation $R(x,y)$ over strings: Given x, find some y such that $R(x,y)$ holds **if** such a y exists, otherwise return "no".

Complexity classes for search problems:

FP: class of search problems that can be solved by a deterministic Turing machine in polynomial time.

FNP: ... (as above) ... by a non-deterministic Turing machine ...

TFNP: class of search problems in FNP where the relation R is known to be **total**, i.e. $\forall x \exists y\, R(x,y)$.

PPAD: class of search problems that can be polynomially reduced to END-OF-LINE.

Polynomial Parity Argument in Directed Graphs

END-OF-LINE: Consider a directed graph with node set $\{0,1\}^n$ such that each node has out- and indegree at most 1. The graph is specified by two poly-time functions f and g:

$f(v)$: successor candidate of v or empty

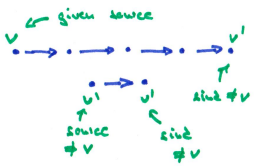
$g(v)$: predecessor candidate of v or empty

In the graph there is an arc $v \to v'$ if and only if $f(v) = v'$ and $g(v) = v'$.

Given a source node v in the graph, **find** some node $v' \neq v$ such that v' has outdegree 0 or indegree 0.

„source" „sink"

Example:



Notice:

* $FP \subseteq PPAD \subseteq TFNP \subseteq FNP$
* Lemke-Howson algorithm has exponential time complexity in the worst case.

Problem: Any „$\subseteq$"-relations proper?

Theorem (Daskalakis et al., 2006)

NASH is PPAD-complete.

2nd NASH: Given a finite 2-player game G and a NE of G, **find** a second NE of G if one exists, else return "no".

Theorem

2nd NASH is FNP-complete.

Proof idea: ← Reduction from 3SAT

Some further results: Given a finite 2-player game G, it is <u>NP-hard</u> to decide whether there exists a MSNE (α, β) in G that has one of the following properties:

(a) player 1 (or 2) receives a payoff $\geq k$. ⇒ Guaranteed payoff problem

(b) $U_1(\alpha, \beta) + U_2(\alpha, \beta) \geq k$. ⇒ Guaranteed social welfare problem

(c) (α, β) is <u>Pareto-optimal</u>, i.e., there is no strategy profile (α', β') such that

$$U_i(\alpha', \beta') \geq U_i(\alpha, \beta) \quad \text{for both } i \in \{1,2\}, \text{ and}$$
$$U_i(\alpha', \beta') > U_i(\alpha, \beta) \quad \text{for at least one } i \in \{1,2\}.$$

(d) player 1 (or 2) plays some given action a with probability > 0.

Extensive Games

So far: only simultaneous, one-shot games

<u>Question</u>: How to model the sequential structure of many games (e.g., chess ...)?

<u>Approach</u>: Use <u>extensive games</u> (≈ game trees)

Idea: Players have several choice points where they can decide how to play. Strategies, then, map choice points to applicable actions.

Definition: An <u>extensive game with perfect information</u> (EGWPI) is a tuple $\Gamma = \langle N, A, H, P, (u_i)_{i \in N} \rangle$ where:

- N is a finite, nonempty set of <u>players</u>.
- A is a nonempty set of <u>actions</u>.
- H is a set of (finite or infinite) sequences over A (called <u>histories</u>) such that:
  - * the empty sequence $\langle\rangle \in H$;
  - * if $\langle a^k \rangle_{k=1}^{K} \in H$ for some $K \in \mathbb{N} \cup \{\infty\}$ and $L < K$, then $\langle a^k \rangle_{k=1}^{L} \in H$; "closed under prefixes"
  - * if $\langle a^k \rangle_{k=1}^{\infty}$ is an action sequence such that $\langle a^k \rangle_{k=1}^{L} \in H$ for each $L \in \mathbb{N}$, then $\langle a^k \rangle_{k=1}^{\infty} \in H$. "closed under limits"

Assumption: All the ingredients of Γ are <u>common knowledge</u> amongst the players of the game.

A history is called <u>terminal</u> if it is infinite or if it is not the prefix of any longer history in H. The set of terminal histories is denoted by Z.

- $P: H \setminus Z \to N$ is the <u>player function</u> assigning to each non-terminal history $h \in H \setminus Z$ a player $P(h)$ whose turn it is to "move" after h.
- For each player $i \in N$, $u_i: Z \to \mathbb{R}$ is player i's <u>utility function</u>.

Some terminology:
- Γ is <u>finite</u> if H is finite.
- Γ <u>has finite horizon</u> if H contains no infinite history.

Γ finite $\overset{?}{\Longrightarrow}$ Γ has finite horizon

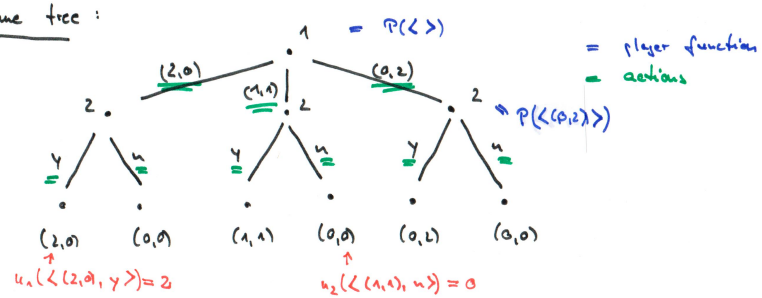
Example (Sharing Game): Two players have to share two indistinguishable objects.

- Player 1 proposes an allocation.
- Player 2 accepts or declines the proposal.

objects are allocated as proposed

no one gets anything

Game tree:



Formally, $\Gamma = \langle N, A, H, P, (u_i)_{i \in N} \rangle$ where

- $N = \{1, 2\}$
- $A = \{ (2,0), (1,1), (0,2), y, n \}$
- $H = \{ \langle \rangle, \langle (2,0) \rangle, \langle (1,1) \rangle, \langle (0,2) \rangle, \langle (2,0), y \rangle, \langle (2,0), n \rangle, \langle (1,1), y \rangle, \dots \}$
- $Z = \{ h \in H : |h| = 2 \}$
- $P(\langle \rangle) = 1, \quad P(h) = 2$ for $h \in H \setminus (Z \cup \{\langle \rangle\})$
- $u_1(\langle (2,0), y \rangle) = 2$, etc.