

L P encoding 256:

$$G = \langle \{1, 2\}, (A_1, A_2), (v_1, v_2) \rangle \text{ with}$$

$$A_1 = \{a_1, a_2, \dots, a_m\}$$

$$A_2 = \{b_1, b_2, \dots, b_n\}$$

$$-\alpha(a_i) \geq 0 \quad \forall a_i \in A_1$$

$$-\sum_{i=1}^n \alpha(a_i) = 1$$

$$-\sum_{i=1}^n \alpha(a_i) \cdot v_1(a_i, b_j) \geq 0 \quad \forall b_j \in A_2$$

Subject to maximizing  $v$ .

$\alpha$  will then be a maximizer for pure actions.

Because  $\beta$  leads just to linear comb.

the utility cannot get lower. This means,  
d is a maximizer for the whole game.

In the same way, we get a maximizer for  
player 2.

Maximizer Theorem: Pair of maximization  
is NE - provided the game has a NE. Because  
there is always a NE in MS, the maximizers  
form a NE

---

This implies that we can find  $x^*_E$  in 2S6  
in poly. time. Usually people use the simplex method,  
where we don't have a guarantee of poly.

How do we find NE in 2-player general  
games?

Instead of LP, we use Linear Complementarity  
problems (LCP):

- They do not have an optimization condition
- Instead, they have complementarity conditions;

For two vectors of variable  $(x_1, \dots, x_n)$  and  
 $(y_1, \dots, y_n)$ :

$$x_i \cdot y_i = 0$$

.

Let  $G = \langle \{1, 2\}, (A_1, A_2), (v_1, v_2) \rangle$  be a general game with  $A_1 = \{a_1, \dots, a_m\}$ ,  $A_2 = \{b_1, \dots, b_n\}$ . Suppose that  $(\alpha, \beta)$  is a MNE with payoff profile  $(v, v)$ . Then the following holds:

$$\rightarrow v - v_i(a_i, \beta) \geq 0 \quad (1 \leq i \leq m) \quad (\text{I})$$

$$v - v_j(\alpha, b_j) \geq 0 \quad (1 \leq j \leq n) \quad (\text{II})$$

$$\boxed{\alpha(a_i) \cdot (v - v_i(a_i, \beta)) = 0} \quad (1 \leq i \leq m) \quad (\text{III})$$

$\cancel{= 0 \rightarrow a_i \notin \text{Supp}(\alpha)}$

$\cancel{\alpha \text{ best response}}$

$$\beta(b_j) \cdot (v - v_j(\alpha, b_j)) = 0 \quad (1 \leq j \leq n) \quad (\text{IV})$$

$$\alpha, \beta \quad \boxed{\alpha(a_i) \geq 0} \quad (1 \leq i \leq m) \quad (\text{V})$$

$$a_{1+} \quad \boxed{\beta(b_j) \geq 0} \quad (1 \leq j \leq n) \quad (\text{VI})$$

*prob. dist.*

$$\sum \alpha(a_i) = 1 \quad (1 \leq i \leq m) \quad (\text{VII})$$

$$\sum \beta(b_j) = 1 \quad (1 \leq j \leq n) \quad (\text{VIII})$$

Proposition A mixed strategy profile  $(\alpha, \beta)$  with payoff profile  $(v, v)$  is a NE if and only if there exist a solution to the above LCP with variables  $v, v, \alpha, \beta$ .

Proof:  $\Rightarrow$ : Let  $(\alpha, \beta)$  be a NE with payoff  $(v, v)$ . By the Support Lemma, for every player for every pure strategy in the support, it is a best response to the remaining profile. Therefore I-IV are satisfied. V-VIII is true, because  $\alpha, \beta$  are MS.

$\Leftarrow$ : Assume we have a solution for the LCP. Because of V-VIII,  $\alpha$  and  $\beta$  must be MS. For all  $a_i \in A_i$ , either  $a_i \notin \text{supp}(\alpha)$  or  $v_i(a_i; \beta) = v$ . In addition,  $v$  is the best utility we can get playing a pure strategy  $a_i$  against  $\beta$ . This means

$U$  is the utility of the best response against  $\beta$ . The same arguments also show that  $b_j$  with  $\beta(b_j) > 0$  are best responses against  $\alpha$ . With the support Lemma, it follows that  $(\alpha, \beta)$  is a NE.  $\square$

Naive approach at solving LCPs:

- ① Enumerate all pairs of possible supports:  
~~exponentially many~~  $(2^m - 1) \cdot (2^n - 1)$  such pairs.
- ② For each pair of supports  $(\text{supp}(\alpha), \text{supp}(\beta))$  you do the following:  
Convert the LCP to a LP as follows

Replace conditions of the form  $\alpha(a_i) \cdot (v - U_1(a_i; \beta))$   
by

$$\begin{cases} v - U_1(a_i; \beta) = 0 & , \text{ if } a_i \in \text{supp}(\alpha) \\ \alpha(a_i) = 0 & , \text{ if } a_i \notin \text{supp}(\alpha) \end{cases}$$

You have do that as well for  $\beta(b_j) \cdot (v - U_2(a_j; b_j))$ .

Then we have a linear program!

Can be solved by any LP-Solver. If there  
is a solution, then this is a solution to the  
original LCP.

Lemke-Howson algorithm is a direct way  
of solving games,

# Complexity of Solving Strategic Games

Usually, we consider decision problem.  
This is not helpful here.

Consider the search problem:

NASH: Given a finite 2-player strategic game  $G$ ,  
find a mixed strategy profile  $(\alpha, \beta)$  that is  
a NE for  $G$  [if there exist one, otherwise  
return "no".]

SAT: Given a Boolean formula  $\varphi$ , decide  
whether the formula is satisfiable (i.e. whether  
there is a variable assignment that makes  $\varphi$  true)?

FSAT: Given a Boolean formula  $\varphi$ , find a satisfying  
assignment if one exists, otherwise return "no".

A search problem is given by a binary relation over strings

$R(x, y)$ : Given an  $x$ , find a  $y$  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 on a deterministic Turing machine in polynomial time.

FNP: ... - - -

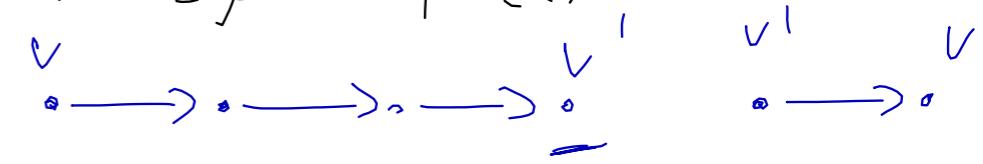
... non-det ...

TFNP: class of search problems in FNP where the function is known to be total.

$\forall x \exists y : R(x, y)$

Clearly finding an NE belongs to TFNP.

PPAD: Complexity class that is specified by the following problem



END-OFF-LINE-PROBLEM: Consider a directed graph,

such that each node has out- and indegree at most

1. This graph is specified by a poly-time functions  $f, g$  such that  $f(v) = \text{successor of } v$  or empty and  $g(v) = \text{predecessor of } v$  or empty.

Given a source node  $v$  in  $G$ , find another node  $v' \neq v$  such that  $v'$  either has outdegree 0 or indegree 0.

Theorem (Paskalakis et al., 2006)

NASH is PPAD-complete.

$$\text{FP} \subseteq \text{PPAD} \subseteq \text{TFNP} \subseteq \text{FNP}$$

