

LP encoding 2S6:

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

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

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

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

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

- $\sum_{i=1}^n \alpha(a_i) \cdot v_1(a_i, b_j) \geq v_2(b_j \in A_2)$

Subject to maximizing v_1 .

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

Because β leads just to linear comb.

the utility cannot get lower. This means, α 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 KIE. Because there is always a NE in MS, the maximizers form a KIE.

This implies that we can find KIEs 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 (α, β) is a MSNE with payoff profile (v, w) . Then the following holds:

$$\underline{\alpha} \cdot v - v_1(a_i, \beta) \geq 0 \quad (1 \leq i \leq m) \quad (\text{I})$$

$$w - v_2(\alpha, b_j) \geq 0 \quad (1 \leq j \leq n) \quad (\text{II})$$

$$\boxed{\alpha(a_i) \cdot (v - v_1(a_i, \beta)) = 0} \quad (1 \leq i \leq m) \quad (\text{III})$$

$$\beta(b_j) \cdot (w - v_2(\alpha, b_j)) = 0 \quad (1 \leq j \leq n) \quad (\text{IV})$$

$$\boxed{\alpha_i, \beta_j} \quad \begin{cases} \alpha(a_i) \geq 0 \\ \beta(b_j) \geq 0 \end{cases} \quad (1 \leq i \leq m) \quad (\text{V})$$

$$\boxed{\text{prob. dist.}} \quad \begin{cases} \sum \alpha(a_i) = 1 \\ \sum \beta(b_j) = 1 \end{cases} \quad (1 \leq i \leq m) \quad (\text{VI})$$

$$\boxed{\text{prob. dist.}} \quad \begin{cases} \sum \alpha(a_i) = 1 \\ \sum \beta(b_j) = 1 \end{cases} \quad (1 \leq i \leq m) \quad (\text{VII})$$

Proposition A mixed strategy profile (α, β) with payoff profile (u, v) is a NE if and only if there exist a solution to the above LCP with variables u, v, α, β .

Proof: \Rightarrow : Let (α, β) be a NE with payoff (u, 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 α and β are MS.

\Leftarrow : Assume we have a solution for the LCP. Because of V-VIII, α and β must be MS. For all $a_i \in A_1$, either $a_i \notin \text{supp}(\alpha)$ or $U_1(a_i, \beta) = u$. In addition, u is the best utility we can get playing a pure strategy a_i against β . This means

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

Naive approach at solving LCPs:

- ① Enumerate all pairs of possible supports:
~~exponentials~~ 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

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

You have to do that as well for $\beta(b_j) \cdot (v - U_2(a_i, b_j))$. 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:

NASL: Given a finite 2-player strategic game G , find a mixed strategy profile (α, β) that is a NE for G [if there exists one, otherwise return "no".]

SAT: Given a Boolean formula φ , decide whether the formula is satisfiable, i.e. whether there is a variable assignment that makes φ true).

FSAT: Given a Boolean formula φ , 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.

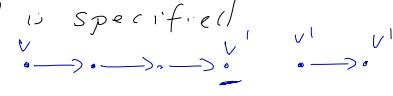
FNP:
... - - -
... non-det ...

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-OF-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 $f(v) =$ successor of v or empty and $g(v) =$ 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 (Daskalakis et al., 2006)

NASH is PPAD-complete.

$$FP \subseteq PPAD \subseteq TFNP \subseteq FNP$$