CSCE 631 — Algorithmic Game Theory

Lecture 6: Computing Equilibria II

Alan Kuhnle

Summer 2026

Texas A&M University

Today's Plan

LP for Zero-Sum Games: Detailed Formulation

Recap: Zero-Sum Game Setup

- Two-player zero-sum game with payoff matrix $A \in \mathbb{R}^{m \times n}$.
- Row player payoff: $x^\top Ay$; column player payoff: $-x^\top Ay$.
- Row player chooses $x \in \Delta_m$, column player chooses $y \in \Delta_n$.

Minimax Theorem (von Neumann, 1928)

$$\max_{x \in \Delta_m} \min_{y \in \Delta_n} x^\top Ay = \min_{y \in \Delta_n} \max_{x \in \Delta_m} x^\top Ay = v^*$$

Today: the full LP formulation and its dual interpretation.

Row Player's LP (Primal)

$$
\begin{aligned}
\text{maximize} \quad & v \\
\text{subject to} \quad & \sum_{i=1}^{m} A_{ij}\, x_i \geq v \quad \text{for } j = 1, \ldots, n \\
& \sum_{i=1}^{m} x_i = 1 \\
& x_i \geq 0 \quad \text{for } i = 1, \ldots, m
\end{aligned}
$$

- Variables: $x_1, \ldots, x_m$ and v.
- n inequality constraints (one per column / pure strategy of opponent).
- Optimal v^* is the **value of the game**.
- Optimal x^* is the **maxmin strategy**.

Column Player's LP (Dual)

$$
\begin{aligned}
\text{minimize} \quad & w \\
\text{subject to} \quad & \sum_{j=1}^{n} A_{ij}\, y_j \le w \quad \text{for } i = 1, \ldots, m \\
& \sum_{j=1}^{n} y_j = 1 \\
& y_j \ge 0 \quad \text{for } j = 1, \ldots, n
\end{aligned}
$$

- This is the LP dual of the row player's LP.
- **Strong duality:** $v^* = w^*$.
- Optimal y^* is the **minimax strategy**.
- (x^*, y^*) forms a Nash equilibrium of the zero-sum game.

Duality Interpretation

Complementary slackness gives equilibrium structure:

- If $x_i^* > 0$, then constraint $\sum_j A_{ij} y_j^* = w^*$ is tight (row i is a best response).
- If $y_j^* > 0$, then constraint $\sum_i A_{ij} x_i^* = v^*$ is tight (column j is a best response).

This recovers the **best-response characterization** of NE from LP duality:

support strategies are all equally good (tied for best).

Computational Cost

LP with $m + n + 2$ variables and $m + n + 2$ constraints.

Solvable in $O(\text{poly}(m, n))$ time via interior point methods.

Worked Example

Matching Pennies:

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Row player LP:

$$\max\ v \quad \text{s.t.} \quad x_1 - x_2 \geq v, \quad -x_1 + x_2 \geq v, \quad x_1 + x_2 = 1, \quad x \geq 0$$

Adding the two inequalities: $0 \geq 2v$, so $v^* \leq 0$.
Setting $x_1 = x_2 = \frac{1}{2}$: both constraints give $0 \geq v$, so $v^* = 0$.

NE: $x^* = y^* = (\frac{1}{2}, \frac{1}{2})$, value $v^* = 0$.

Correlated Equilibria via LP

Correlated Equilibrium (Recall)

Definition (Correlated Equilibrium)

A probability distribution p over action profiles $a = (a_1, \ldots, a_n)$ such that for every player i and every pair of actions a_i, a_i':

$$\sum_{a_{-i}} p(a_i, a_{-i}) \left[u_i(a_i, a_{-i}) - u_i(a_i', a_{-i}) \right] \geq 0$$

Interpretation: A mediator samples $a \sim p$ and recommends a_i to player i. No player wants to deviate from the recommendation.

Every NE is a CE (take p to be the product distribution).

CE can achieve **better social welfare** than any NE.

Computing CE: An LP!

Variables: $p(a)$ for each action profile $a \in A_1 \times \cdots \times A_n$.

LP for Correlated Equilibrium

$$
\begin{aligned}
\text{find} \quad & p \\
\text{s.t.} \quad & \sum_{a_{-i}} p(a_i, a_{-i}) \big[u_i(a_i, a_{-i}) - u_i(a_i', a_{-i})\big] \geq 0 \quad \forall\, i,\ \forall\, a_i,\ a_i' \\
& \sum_{a} p(a) = 1 \\
& p(a) \geq 0 \quad \forall\, a
\end{aligned}
$$

- All constraints are **linear** in p.
- Number of variables: $|A_1| \times \cdots \times |A_n|$.
- Number of incentive constraints: $\sum_i |A_i|(|A_i| - 1)$.

CE LP: Optimization Variants

Can optimize *any linear objective* over the CE polytope:

- **Max social welfare:** $\max \sum_a p(a) \sum_i u_i(a)$
- **Max fairness (egalitarian):** $\max \min_i \sum_a p(a)\, u_i(a)$ (requires auxiliary variable, still LP)
- **Min social cost:** $\min \sum_a p(a)\, c(a)$

All of these are **polynomial-time** solvable!

CE vs. NE: Computational Complexity

	Nash Equilibrium	**Correlated Equilibrium**
Existence	Guaranteed	Guaranteed
Computation	PPAD-complete	**Polynomial (LP)**
Set structure	Union of convex sets	**Convex polytope**
Optimization	NP-hard in general	**LP over polytope**
Welfare	Can be suboptimal	Can exceed NE welfare

Key insight: CE constraints are *linear* in p; NE requires the *product structure* $p = x \otimes y$, which is a nonlinear constraint.

CE Example: Traffic / Chicken Game

	D	C
D	$(0, 0)$	$(3, 1)$
C	$(1, 3)$	$(2, 2)$

Pure NE: (D, C) and (C, D) with payoffs $(3, 1)$ and $(1, 3)$.
Mixed NE: each plays D with prob $\frac{1}{2}$; expected payoff $\frac{3}{2}$ each.

Optimal CE: $p(D, C) = p(C, D) = p(C, C) = \frac{1}{3}$.
Expected payoff per player: $\frac{1}{3}(3 + 1 + 2) = 2 > \frac{3}{2}$.

CE achieves **higher social welfare** than any NE.

Approximate Nash Equilibria

ε-Nash Equilibrium

Since exact NE is PPAD-complete, can we *approximate*?

Definition (ε-Nash Equilibrium)

A strategy profile (x, y) is an ε-NE if for all players i and all alternative strategies s_i':

$$u_i(s_i, s_{-i}) \ \geq \ u_i(s_i', s_{-i}) - \varepsilon$$

- No player can gain more than ε by deviating.
- For $\varepsilon = 0$, this is an exact NE.
- Question: for what ε can we find an ε-NE in poly time?

Algorithms for Approximate NE

Known results for 2-player $n \times n$ games:

Algorithm	ε	**Time**
Tsaknakis–Spirakis (2007)	0.3393	$O(\text{poly}(n))$
Daskalakis–Mehta–Papadimitriou (2007)	0.5	$O(\text{poly}(n))$
Kontogiannis–Spirakis (2010)	0.3393	$O(\text{poly}(n))$
Lipton–Markakis–Mehta (2003)	any $\varepsilon > 0$	$n^{O(\log n/\varepsilon^2)}$

- Best known poly-time: $\varepsilon \approx 0.3393$.
- For any constant $\varepsilon > 0$: quasi-polynomial time (LMM).
- **Open:** poly-time ε-NE for any constant $\varepsilon < 0.3393$?

The Lipton–Markakis–Mehta Approach

Theorem (LMM 2003)
Every bimatrix game has an ε-NE where each player's strategy has support of size $k = O(\log n/\varepsilon^2)$.

Proof idea:

1. Start from any exact NE (x^*, y^*).
2. Sample k actions i.i.d. from x^* to form uniform mixture $\hat{x}$.
3. By Hoeffding's inequality, $\hat{x}$ is an ε-best response to y^* w.h.p.
4. Similarly for y^*.

Algorithm: Enumerate all $\binom{n}{k}^2 = n^{O(\log n/\varepsilon^2)}$ support pairs of size k.

Hardness of Approximation

Theorem (Rubinstein 2018)
Under the Exponential Time Hypothesis for PPAD, there is no polynomial-time algorithm for ε-NE with $\varepsilon = \text{poly}(1/n)$ in $n \times n$ bimatrix games.

- Finding NE to *inverse-polynomial* precision is hard.
- For *constant* ε: status is open!
- Gap: we can do $\varepsilon \approx 0.34$ in poly time, but cannot reach $\varepsilon = 1/\text{poly}(n)$.
- This gap between 0.34 and $1/\text{poly}(n)$ is one of the biggest open problems in AGT.

Practical Algorithms and Tools

Lemke–Howson in Practice

- Despite worst-case exponential complexity, often fast in practice.
- Implemented in most game theory software packages.
- Finds **one** NE (not all); different starting labels can yield different equilibria.
- Works only for **nondegenerate** bimatrix games (lexicographic perturbation for degenerate cases).

Practical considerations:

- Numerical stability matters: use rational arithmetic for exact results.
- For large games, heuristic methods often outperform exact algorithms.

Support Enumeration in Practice

Improvements over naive enumeration:

- Start with small supports (pure strategies first, then $k = 2, 3, \ldots$).
- Prune: if a partial support assignment is infeasible, skip all extensions.
- Conditional best-response: use symmetry and dominance to reduce the search space.

Use case: Finding **all** NE of small games.
Support enumeration is the standard approach when completeness is needed.

Gambit: A Software Suite

Gambit (McKelvey, McLennan, Turocy): open-source software for game theory computation.

- Implements: support enumeration, Lemke–Howson, homotopy methods, global Newton, simulated annealing.
- Supports: normal-form and extensive-form games, *n* players.
- Python API: `pygambit` for scripting.
- URL: `http://www.gambit-project.org/`

Gambit Algorithms

`gambit-enummixed` — support enumeration

`gambit-lcp` — Lemke–Howson (linear complementarity)

`gambit-lp` — LP for zero-sum games

Polynomial-Time Algorithms for Structured Games

Special game classes where NE is tractable:

Game Class	Key Property
Zero-sum (2-player)	LP duality
Potential games	NE $=$ local optima of potential
Congestion games	Isomorphic to potential games
Graphical games (bounded treewidth)	Dynamic programming
Symmetric games (2-player)	Single-variable search
Sparse / rank-k games	Low-rank structure

Theme: Exploit structure to avoid the PPAD barrier

Rank-Based Approaches

Definition (Game Rank)
The **rank** of a bimatrix game (A, B) is $\mathrm{rank}(A + B)$.

- Zero-sum: rank 0 (since $A + B = 0$).
- Rank-k games can be solved in $n^{O(k)}$ time.
- **Idea:** Decompose $A + B = \sum_{r=1}^{k} u_r v_r^\top$. Parametrize NE conditions using k free variables instead of n.
- Practical for small rank even when n is large.

Low rank captures games where players' interests are "almost aligned" or "almost opposed."

Summary

Summary

1. **Zero-sum LP:** Primal/dual pair; strong duality $\Leftrightarrow$ minimax theorem; complementary slackness $\Leftrightarrow$ NE best-response conditions.
2. **Correlated equilibria:** computable in poly time via LP; constraints are linear (no product structure); can optimize welfare.
3. **CE vs. NE:** CE is computationally easy (LP); NE is PPAD-complete. The product structure of NE is the source of hardness.
4. **ε-NE:** Best poly-time achieves $\varepsilon \approx 0.34$; quasi-poly for any constant ε; inverse-poly precision is hard.
5. **Structured games:** zero-sum, potential, low-rank, bounded treewidth all admit poly-time NE computation.
6. **Next time:** online decision making and regret minimization.