Harvard CS 1360, Spring 2026
Instructor: Ariel Procaccia
Scribe: Taig Singh, Jakob de Raaij

Equilibrium Computation

—

Lecture 2

In the last lecture we saw that in every finite game there exists at least one, possibly mixed, Nash equilibrium. A natural next question for computer scientists to ask is whether we can efficiently compute it. In particular, given a normal-form game, *what is the computational complexity of finding a Nash equilibrium?*

1 The complexity class PPAD

To introduce the complexity class that Nash equilibrium computation falls under, we will begin by defining the End of the Line problem.

Definition 1 (End of the Line Problem)**.** Given is a directed graph $G = (V, E)$ with $V = \{0,1\}^n$, where every vertex has at most one predecessor and at most one successor. The edges E are implicitly given by polynomial-time computable functions $f_p : \{0,1\}^n \to \{0,1\}^n$ and $f_s : \{0,1\}^n \to \{0,1\}^n$ that respectively return the predecessor and successor of a given vertex, or that none exists. Given a vertex with no predecessor (the *source*), the problem is to find a vertex with no successor (a *sink*).

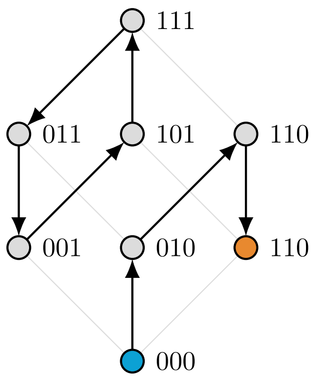


Figure 1: An instance of End of the Line with $n = 3$. The source is blue, the sink is orange.

For any input to End of the Line, the existence of a sink vertex is guaranteed: Imagine starting at at the source vertex and repeatedly taking the outgoing edge. On this path, we will never visit a vertex twice, since each vertex has only one predecessor and the source vertex has none. Since the number of vertices is finite, we must eventually end up in a vertex with no outgoing edge, a source. However, how do we find it?

In its nature, the End of the Line problem is very similar to the problem of finding a Nash equilibrium: We are guaranteed that a solution does exist, but it is not obvious how to find it efficiently. However, if presented with a solution, we can verify it efficiently.

Definition 2 (**TFNP**)**.** The complexity class of ***t**otal **f**unction problems that can be solved in **n**on-deterministic **p**olynomial time (**TFNP**)* includes all problems that are guaranteed to have a solution, and this solution can be verified in polynomial time.

Both End of the Line and finding a Nash equilibrium are in **TFNP**. Another well-known example of a problem in **TFNP** is the factoring problem of finding a prime factor of a given composite integer n: A solution always exists and the validity of any factorization can be quickly checked.

Definition 3 (**PPAD**)**.** The complexity class ***p****olynomial* ***p****arity* ***a****rguments on* ***d****irected graphs (****PPAD****)* includes all problems in **TFNP** that have polynomial-time reductions to the End of the Line problem.

This complexity class was introduced by Christos Papadimitriou (1949–present) in 1994, an influential theoretical computer scientist and a founder of algorithmic game theory. A depiction of how **PPAD** relates to the other complexity classes you may have learned about is shown in Figure 2. **FP** and **FNP** are the classes corresponding to **P** and **NP** for problems with not just a 0/1 decision (f.e., 'Is this graph 3-colorable?') but a string (f.e., the 3-coloring, if it exists) as their output. [1]

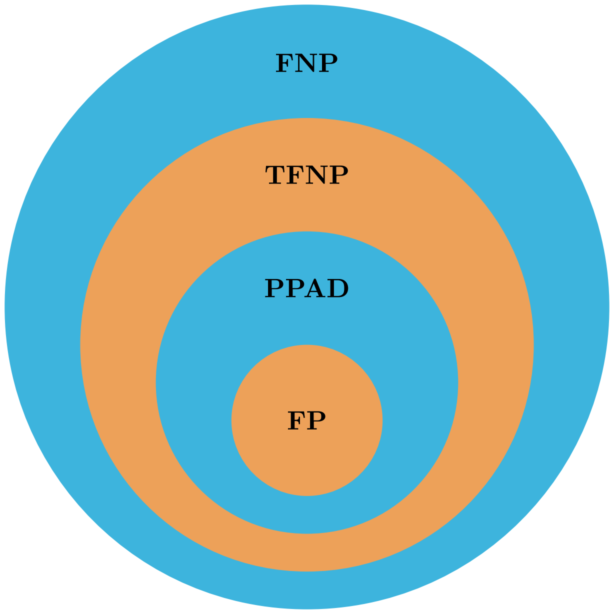


Figure 2: How PPAD relates to other complexity classes

Theorem 1. *For all $n \geq 2$, computing an (approximate) Nash equilibrium in an n-player normal-form game is* ***PPAD****-complete.*

PPAD-completeness behaves very similarly to **NP**-completeness: If we can solve one **PPAD**-complete problem in polynomial time, we can solve all problems in **PPAD** in polynomial time. Since the End of the Line problem is *believed* not to be solvable in polynomial time, it is also believed that we cannot compute a Nash equilibrium in polynomial time.

This may be bad news for the Nash equilibrium: Why should we expect games in the real world to be in an equilibrium state that is hard to find? In the remainder of the lecture, we explore two ways to get around the computational hardness of computing a Nash equilibrium. Both are based on Linear Programming.

2 An Introduction to Linear Programming

We will take a brief interlude to discuss linear programming, which will later help us show that some equilibrium concepts can be computed in polynomial time.

Definition 4 (Linear Programming)**.** *Linear programming (LP)* is an optimization technique used to find the best outcome within a linear mathematical model. The goal is to optimize a linear objective function

[1] In a similar vein, note that **PPAD** problems cannot be **NP**-complete, as problems that are **NP**-complete are decision problems, but there is always a solution to a problem in **PPAD**, just finding it is difficult. Thus, even though finding a Nash equilibrium in an n-player normal-form game is **PPAD**-complete, it is not **NP**-complete. Nonetheless, similar problems, like deciding whether a second Nash equilibrium exists, are **NP**-complete.

subject to linear constraints. An LP problem takes on the following form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{a} \\ & B\mathbf{x} \le \mathbf{b} \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable, and $\mathbf{c} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $\mathbf{a} \in \mathbb{R}^m$, $B \in \mathbb{R}^{k\times n}$, $\mathbf{b} \in \mathbb{R}^k$ are the problem data, specifying the objective and the constraints.

LPs are more general than the above form might suggest. In particular, if we want to maximize an objective function instead of minimizing it, we can multiply it by -1 to match the form. Similarly, inequality constraints that specify a lower bound on $\mathbf{x}$ can be turned into constraints that fit the form above by multiplying them by -1.

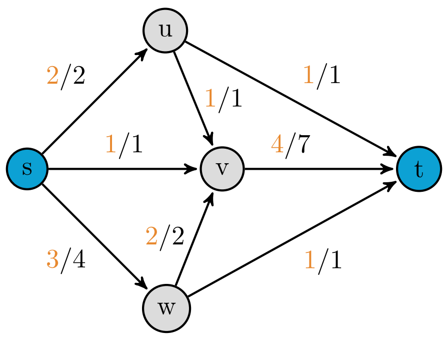
The key fact about linear programs for this lecture is that **LPs can be solved in polynomial time** using interior-point methods. If we can model a problem as an LP (with a polynomial number of variables and constraints), we can solve the problem in polynomial time.

Example 1 (The Max Flow Problem)**.** In the *max flow problem*, we are given a directed graph $G = (V, E)$ with a *source* vertex s, a *sink* vertex t, and a *capacity* α_{xy} for each $(x, y) \in E$. A *flow* is a function $f : E \to \mathbb{R}^+$ that satisfies

- $0 \le f_{xy} \le \alpha_{xy}$ for all $(x, y) \in E$ (the flow from x to y is non-negative and at most the capacity from x to y), and
- for all vertices $x \ne s, t$, $\sum_{(y,x)\in E} f_{yx} = \sum_{(x,y)\in E} f_{xy}$ (the total flow into vertex x is equal to the flow out of vertex x).

The *value* of a flow is $\sum_{(s,x)\in E} f_{sx}$, the total flow leaving the source s. We wish to find a flow with maximum value.

In the example below, the value of the max flow is 6. Along each edge, the value of the flow along this edge is displayed in orange, followed by the capacity of the edge in black.



We can formulate the max flow problem for this example using an LP:

$$\begin{aligned} \max \quad & f_{su} + f_{sv} + f_{sw} \\ \text{s.t.} \quad & 0 \le f_{su} \le 2, \quad 0 \le f_{sv} \le 1, \quad 0 \le f_{sw} \le 4, \\ & 0 \le f_{uv} \le 1, \quad 0 \le f_{ut} \le 1, \\ & 0 \le f_{wv} \le 2, \quad 0 \le f_{wt} \le 1, \\ & 0 \le f_{vt} \le 7, \\ & f_{su} = f_{uv} + f_{ut}, \\ & f_{sv} + f_{uv} + f_{wv} = f_{vt}, \\ & f_{sw} = f_{wv} + f_{wt}. \end{aligned}$$

3 Equilibrium Concepts in Polynomial Time

Going back to our discussion of the complexity of equilibrium concepts, we will consider two simplifications of the problem: Making the class of possible games smaller (normal-form games to zero-sum games) and making the class of solutions larger (Nash equilibria to correlated equilibria). We leverage our new tool of linear programming to find that in both cases, solutions to the games can be computed in polynomial time.

3.1 Zero-Sum Games

Definition 5 (Two-Player Zero-Sum Games)**.** A two-player *zero-sum* game is a game in normal form with $n = 2$ player where it holds that for every strategy profile $\mathbf{s}$,

$$u_1(\mathbf{s}) = -u_2(\mathbf{s}).$$

That is, the first player wins exactly as much as the second player looses. Consistently, every cell in the payoff matrix contains a number and their negation.[2]

Definition 6 (Maximin and Minimax)**.** A *maximin* (randomized) strategy of player 1 is

$$x_1^\star \in \arg \max_{x_1 \in \Delta(S_1)} \min_{s_2 \in S_2} u_1(x_1, s_2).$$

A *minimax* (randomized) strategy of player 2 is

$$x_2^\star \in \arg \min_{x_2 \in \Delta(S_2)} \max_{s_1 \in S_1} u_1(s_1, x_2).$$

The maximin strategy of player 1 is their optimal mixed strategy if player 2 is allowed to observe player 1's strategy and then react optimally to it (in the eyes of player 2). Given that player 1 plays strategy $x_1 \in \Delta(S_1)$, player 2 will pick a strategy $x_2 \in \Delta(S_2)$ to maximize $u_2(x_1, x_2)$, or equivalently to minimize $u_1(x_1, x_2)$. Knowing that player 2 is going to play a strategy giving player 1 utility $\min_{s_2 \in S_2} u_1(x_1, s_2)$, player 1 will pick their strategy x_1^* to maximize this expression. This is the maximin strategy of player 1 as defined above. Similarly, player 2's minimax strategy x_2^* minimizes player 1's utility (thus maximizing player 2's own utility) when player 1 observes player 2's strategy x_2 and then chooses a strategy s_1 that will maximize player 1's utility (thus minimizing player 2's utility).

An important observation is that once one player fixes a mixed strategy, there is always a best response for the other player that is pure. For example, there is a strategy $s_2 \in S_2$ that is a best response for player 2 given that player 1 plays $x_1^\star$: Since player 1's strategy is fixed, any mixed strategy x_2 of player 2 leads to a distribution over possible utilities $u_1(x_1, s_2)$, for s_2 sampled from x_2. In this distribution, there has to exist some pure strategies $s_2 \in S$ that minimize $u_1(x_1, s_2)$ — playing such a strategy s_2 deterministically will make player 2 no worse of than playing x_2. Thus, knowing player 1's strategy x_1, player 2 always has an optimal pure strategy.[3]

Example 2. Consider a two-player zero-sum game with the following payoff matrix for strategies A and B:

	A	B
A	$-1, 1$	$2, -2$
B	$2, -2$	$-2, 2$

We know that player 1's strategy is of the form $x_1 = (p, 1 - p)$ for some p, with p being the probability of them playing A and $1 - p$ being the probability of them playing B. When player 1 plays x_1, player two has

[2] For this reason you will find that in many sources, each cell of the payoff matrix only contains one number, the payoff of the row player.

[3] We can apply the same logic when verifying that a strategy profile (x_1, x_2) is a Nash equilibrium in a normal-form game: To verify that no player can deviate to a mixed strategy that will increase their payoff, it suffices to verify that no player can deviate to a pure strategy that will increase their payoff.

2 options (recall from Lecture 1 that player 2's optimal response will certainly be a pure strategy). If player 2 plays $s_2 = A$, the payoffs are

$$u_1(x_1, s_2) = -u_2(x_1, s_2) = -p + 2 \cdot (1 - p) = -3p + 2,$$

and if player 2 plays $s_2 = B$, the payoffs are

$$u_1(x_1, s_2) = -u_2(x_1, s_2) = 2p - 2 \cdot (1 - p) = 4p - 2.$$

Note that the first expression is smaller when $p \geq \frac{4}{7}$ (so player 2 will chose $s_2 = A$) and the second expression is smaller when $p \leq \frac{4}{7}$ (so player 2 will chose $s_2 = A$). Thus, it is in player 1's best interest to choose p to maximize

$$\min_{s_2 \in S_2} u_1(x_1, s_2) = \begin{cases} -3p + 2 & p \geq \frac{4}{7} \\ 4p - 2 & p < \frac{4}{7} \end{cases}.$$

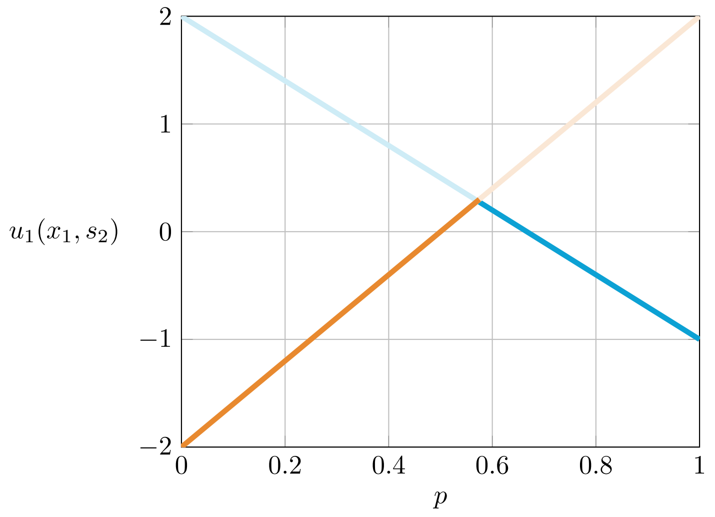


Figure 3: The utility of player 1 if they choose A with probability p, else B, and player 2 reacts optimally

Inspecting this function, for example by looking at its plot in Figure 3, we can see that it is optimal for player 1 to choose $p = \frac{4}{7}$ to maximize this value. Thus their maximin strategy is $x_1^\star = (\frac{4}{7}, \frac{3}{7})$.

A maximin strategy can also be computed with an LP:

$$\begin{aligned} \max \quad & w \\ \text{s.t.} \quad & \forall s_2 \in S, \quad \sum_{s_1 \in S} p(s_1) u_1(s_1, s_2) \geq w, \\ & \sum_{s_1 \in S} p(s_1) = 1, \\ & \forall s_1 \in S, \quad p(s_1) \geq 0. \end{aligned}$$

Here, w corresponds to the maximum utility player 1 can achieve when player 2 responds to their strategy to minimize player 1's utility. The first set of constraints says that every (pure) strategy player 2 plays must result in a utility for player 1 of at least w. The second and third constraints enforce that the $p(s_1)$ are a probability distribution, corresponding to player 1's mixed strategy.

A minimax strategy can be computed analogously with slight changes to the LP.

Theorem 2 (von Neumann, 1928)**.** *Every 2-player zero-sum game has a unique value v such that player 1 can guarantee utility at least v and player 2 can guarantee utility at least $-v$.*

Proof. By Nash's Theorem[4], every zero-sum game has a Nash equilibrium. Let (x_1, x_2) be such a Nash equilibrium and denote $v = u_1(x_1, x_2)$. For every $s_2 \in S_2$, $u_1(x_1, s_2) \geq v$ as (x_1, x_2) is a Nash equilibrium so player 1 can guarantee at least v by playing x_1. Similarly, for every $s_1 \in S_1$, $u_2(s_1, x_2) \geq -v$ so player 1 can guarantee at least $-v$ by playing x_2. □

John von Neumann (1903 - 1957) was a founder of game theory. He was also known for revolutionary contributions to mathematics, physics, computer science, and the Manhattan Project.

His above theorem, intuitively speaking, says that in zero-sum games it does not matters which player commits to a (mixed) strategy first. The payoff that player 1 can ensure by playing an maximin strategy x_1^* (with player 2 then optimally responding) is equal to the negative of the payoff that player 2 can ensure by playing an minimax strategy x_2^* (with player 1 then optimally responding). In other words,

$$\max_{x_1 \in \Delta(S_1)} \min_{s_2 \in S_2} u_1(x_1, s_2) = v = \min_{x_2 \in \Delta(S_2)} \max_{s_1 \in S_1} u_1(s_1, x_2).$$

Thus, we can use the maximin (or minimax) LP to find the value v of a zero-sum game in polynomial time!

3.2 Correlated Equilibrium

We now consider a generalization of the Nash equilibrium where correlation between the players' mixed strategies is allows. For simplicity, we only consider the wo player case $N = \{1, 2\}$, but the definition can be extended to more players.

Definition 7 (Correlated Equilibrium)**.** Let p be a distribution over all pairs of strategies $S \times S$. Now, assume a mediator chooses (s_1, s_2) according to p and reveals s_1 to player 1 and s_2 to player 2. Player 1 is *best responding* in (s_1, s_2) if they are playing their most preferred strategy knowing the conditional distribution of player 2's strategies, given that player 1 got s_1. In other words, player 1 is best responding if for all $s_1' \in S$,

$$\sum_{s_2 \in S} P(s_2 | s_1) u_1(s_1, s_2) \geq \sum_{s_2 \in S} P(s_2 | s_1) u_1(s_1', s_2),$$

where $P(s_2|s_1)$ is the distribution over strategies for player 2 conditional on player 1 getting s_1 from the mediator,

$$P(s_2|s_1) = \frac{P(s_1 \wedge s_2)}{P(s_1)} = \frac{p(s_1, s_2)}{\sum_{s_2' \in S} p(s_1, s_2')}.$$

Substituting in this expression, we get that player 1 is best responding if for all $s_1' \in S$,

$$\sum_{s_2 \in S} p(s_1, s_2) u_1(s_1, s_2) \geq \sum_{s_2 \in S} p(s_1, s_2) u_1(s_1', s_2).$$

A distribution p is a *correlated equilibrium (CE)* if both players are best responding.

Note that Nash equilibria are special cases of correlated equilibria, where each player's actions are drawn from an independent distribution. In Nash equilibria, conditioning on s_1 provides no additional information about s_2 and vice versa, but both players are still best responding to each other.

Example 3 (Game of Chicken)**.** Consider the classic game of chicken with the following payoff matrix:

	Dare	Chicken
Dare	$0, 0$	$4, 1$
Chicken	$1, 4$	$3, 3$

[4] We will reprove this theorem without referencing Nash's Theorem later in the course.

Each player is best off if they Dare while the other player Chickens. However, if both players Dare, they are worse off than if they both Chicken.

The *social welfare* is defined as the sum of utilities. The two pure NE in this game are (C, D) and (D, C) in which case the social welfare is 5. There is one additional mixed NE when both players play $(1/2, 1/2)$, choosing a strategy uniformly at random, in which case the social welfare is 4. However, the optimal social welfare is 6.

Consider the following correlated equilibrium: (D, D) with probability 0, and (D, C), (C, D) and (C, C) with probability $\frac{1}{3}$ each. In this case, the social welfare is $\frac{16}{3}$, higher than for both the pure and mixed NEs! We leave it as an exercise to verify that this is indeed a correlated equilibrium, by going through each possible strategy for each player and verifying that neither player would be better off switching their strategy, knowing the conditional distribution of the other player's strategies.

To implement a CE, one would need to implement the mediator. In the case of the correlated equilibrium above for the game of chicken, this could be implemented by putting two "Chicken" balls and one "Dare" ball in a hat. Each (blindfolded) player picks a ball and plays accordingly. This implementation achieves the desired joint distribution: (D, D) can never be selected and (C, C), (C, D), and (D, C) are all equally likely to be selected.

We can compute CEs via linear programming in polynomial time:

$$\begin{aligned}
\text{find} \quad & p(s_1, s_2) \quad \forall s_1, s_2 \in S \\
\text{s.t.} \quad & \forall s_1, s_1' \in S, \quad \sum_{s_2 \in S} p(s_1, s_2) u_1(s_1, s_2) \geq \sum_{s_2 \in S} p(s_1, s_2) u_1(s_1', s_2), \\
& \forall s_2, s_2' \in S, \quad \sum_{s_1 \in S} p(s_1, s_2) u_2(s_1, s_2) \geq \sum_{s_1 \in S} p(s_1, s_2) u_2(s_1, s_2'), \\
& \sum_{s_1, s_2 \in S} p(s_1, s_2) = 1, \\
& \forall s_1, s_2 \in S, \quad p(s_1, s_2) \in [0, 1].
\end{aligned}$$

Here, we are finding a distribution p over S^2 that is a CE, as enforced by the four sets of constraints: The first set of constraints ensures that player 1's strategies are the best responses given the conditional distributions of player 2's strategies under p. Similarly, the second set of constraints ensures that player 2's strategies are the best responses given the conditional distributions of player 1's strategies under p. The third and fourth sets of constraints ensure p is indeed a probability distribution over S^2.

Note that this LP does not have an objective. However, also checking whether a set of linear constraints can all simultaneously be satisfied can be done in polynomial time, with the same methods used to solve LPs in polynomial time.

Finally, one may wonder why a similar approach does not work for finding Nash equilibria. The issue is that if the two players chose mixed strategies $x_1(s_1)$ and $x_2(s_2)$ that are independent of each other, the variables $p(s_1, s_2)$ in the constraints become $x_1(s_1)x_2(s_2)$. The constraint is no longer linear.