Complexity Theory, Game Theory, and Economics

Lecture Notes for the 29th McGill Invitational Workshop on Computational Complexity

Lectures by Tim Roughgarden, with a guest lecture by Omri Weinstein

Bellairs Institute
Holetown, Barbados

Foreword

This document collects the lecture notes from my mini-course "Complexity Theory, Game Theory, and Economics," taught at the Bellairs Research Institute of McGill University, Holetown, Barbados, February 19–23, 2017, as the 29th McGill Invitational Workshop on Computational Complexity.

The goal of this mini-course is twofold:

(i) to explain how complexity theory has helped illuminate several barriers in economics and game theory; and

(ii) to illustrate how game-theoretic questions have led to new and interesting complexity theory, including several breakthroughs in the past few months!

It consists of two five-lecture sequences: the *Solar Lectures,* focusing on the communication and computational complexity of computing equilibria; and the *Lunar Lectures,* focusing on applications of complexity theory in game theory and economics.[1] No background in game theory is assumed.

Thanks are due to many people: Denis Therien and Anil Ada for organizing the workshop and for inviting me to lecture; Omri Weinstein, for giving a guest lecture on simulation theorems in communication complexity; Alex Russell, for coordinating the scribe notes; the scribes[2], for putting together a terrific first draft of these notes; and all of the workshop attendees, for making the experience so unforgettable (if intense!).

The writing of these notes was supported in part by NSF award CCF-1524062, a Google Faculty Research Award, and a Guggenheim Fellowship. I would be very happy to receive any comments or corrections from readers.

Tim Roughgarden
Bracciano, Italy
December 2017

[1] Cris Moore: "So when are the *stellar* lectures?"

[2] Anil Ada, Amey Bhangale, Shant Boodaghians, Sumegha Garg, Valentine Kabanets, Antonina Kolokolova, Michal Koucký, Cristopher Moore, Pavel Pudlák, Dana Randall, Jacobo Torán, Salil Vadhan, Joshua R. Wang, and Omri Weinstein.

Contents

Part I

Solar Lectures

Solar Lecture 1

Introduction, Wish List, and Two-Player Zero-Sum Games

Lecturer: Tim Roughgarden *Scribe: Anil Ada and Shant Boodaghians*

1.1 The Plan

The topic of the week is Complexity Theory, Game Theory, and Economics. The theme is two-fold:

(i) how complexity theory has illuminated barriers in economics and game theory;

(ii) how studying fundamental complexity questions about game-theoretic concepts has led to the development of new and interesting complexity theory (including some major breakthroughs in the past few months!).

There will be 5 solar lectures and 5 lunar lectures. The solar lectures will focus on the communication and computational complexity of computing an (approximate) Nash equilibrium. The lunar lectures are meant to be understandable even after consuming a rum punch; they focus on applications of computational complexity theory to game theory and economics.

1.1.1 The Solar Lectures: Complexity of Equilibria

Lecture 1: Introduction and wish list. The goal of the first lecture is to get the lay of the land. We'll focus on the types of positive results about equilibria that we want, like fast algorithms and quickly converging distributed processes. Such positive results are possible in special cases (like zero-sum games), and the challenge for complexity theory is to prove that they cannot be extended to the general case. The topics in this lecture are mostly classical.

Lectures 2 and 3: The communication complexity of Nash equilibria. These two lectures cover the main ideas in the brand-new (STOC '17) paper of Babichenko and Rubinstein [9], which proves strong communication complexity lower bounds for computing an approximate Nash equilibrium. Discussing the proof will also give us an excuse to talk about "simulation theorems" in the spirit of Raz and McKenzie [120], which lift query complexity lower bounds to communication complexity lower bounds and have recently found a number of exciting applications.

Lecture 4: TFNP, PPAD, and all that. In this lecture we begin our study of the *computational* complexity of computing a Nash equilibrium, where we want conditional but super-polynomial lower bounds. To prove analogs of NP-completeness results, we will need to develop customized complexity classes appropriate for the study of equilibrium computation.[1] We will also discuss the existing evidence for the intractability of these complexity classes, including some very recent developments.

Lecture 5: The computational complexity of computing an approximate Nash equilibrium of a bimatrix game. The goal of this lecture is to give a high-level overview of Rubinstein's recent breakthrough result [135] that an ETH-type assumption for PPAD implies a quasi-polynomial-time lower bound for the problem of computing an approximate Nash equilibrium (which is tight, by Corollary 1.17).

1.1.2 The Lunar Lectures: Complexity-Theoretic Barriers in Economics

Most of the lunar lectures have the flavor of "applied complexity theory."[2] While the solar lectures build on each other to some extent, the lunar lectures will be episodic, and each can be read independently.

Lecture 1: The 2016 FCC Incentive Auction. This is a great case study of how computer science has influenced real-world auction design. It is also the only lecture that gives a broader glimpse of the vibrant field called *algorithmic game theory*, only about 10% of which concerns the complexity of computing equilibria.

Lecture 2: Barriers to near-optimal equilibria [126]. This lecture concerns the "price of anarchy," meaning the extent to which Nash equilibria approximate an optimal outcome. It turns out that nondeterministic communication complexity lower bounds can be translated, in black-box fashion, to lower bounds on the price of anarchy. We'll see how this translation enables a theory of "optimal simple auctions."

Lecture 3: Barriers in markets [131]. You've surely heard of the idea of "market-clearing prices," which are prices in a market such that supply equals demand. When the goods are divisible (milk, wheat, etc.), market-clearing prices exist under relatively mild technical assumptions. With indivisible goods (houses, spectrum licenses, etc.), market-clearing prices may or may not exist. It turns out complexity considerations play a huge role in when prices exist and when they do not. This is cool and surprising because the issue of equilibrium existence seems to have nothing to do with computation (in contrast to the Solar Lectures, where the questions studied are explicitly about computation).

Lecture 4: The borders of Border's theorem. [68]. Border's theorem is a famous result in auction theory from 1991, about single-item auctions. Despite its fame, no one has been able to generalize it to significantly more general settings. We'll see that complexity theory explains this mystery: significantly generalizing Border's theorem would imply that the polynomial hierarchy collapses!

Lecture 5: Tractable Relaxations of Nash Equilibria. Having spent the week largely on negative results for computing Nash equilibria, for an epilogue we'll switch to positive results for relaxations of Nash equilibria, such as for correlated equilibria.

[1] Why can't we just use the theory of NP-completeness? Because the guaranteed existence (Theorem 1.14) and efficient verifiability of a Nash equilibrium imply that computing one is an easier task than solving an NP-complete problems, under appropriate complexity assumption (see Theorem 4.1).

[2] Not an oxymoron!

1.2 Nash Equilibria in Two-Player Zero-Sum Games

1.2.1 Preamble

To an algorithms person (like your lecturer), complexity theory is the science of why you can't get what you want. So what is it we want? Let's start with some cool positive results for a very special class of games—two-player zero-sum games—and then we can study whether or not they extend to more general games. For the first positive result, we'll review the famous Minimax theorem, and see how it leads to a polynomial-time algorithm for computing a Nash equilibrium of a two-player zero-sum game. Then we'll show that there are natural "dynamics" (basically, a distributed algorithm) that converge rapidly to an approximate Nash equilibrium.

1.2.2 Rock-Paper-Scissors

Recall the game rock-paper-scissors (or roshambo, if you like)[3]: there are two players, each simultaneously picks a strategy from $\{\text{rock}, \text{paper}, \text{scissors}\}$. If both players choose the same strategy then the game is a draw; otherwise, rock beats scissors, scissors beats paper, and paper beats rock.[4]

Here's an idea: how about we play rock-paper-scissors, and you go first? This is clearly unfair—no matter what strategy you choose, I have a response that guarantees victory. But what if you only have to commit to a *probability distribution* over your three strategies (called a *mixed strategy*)? To be clear, the order of operations is: (i) you pick a distribution; (ii) I pick a response; (iii) nature flips coins to sample a strategy from your distribution. Now you can protect yourself—by picking a strategy uniformly at random, no matter what I do, you have an equal chance of a win, a loss, or a draw.

The *Minimax theorem* states that, in any game of "pure competition" like rock-paper-scissors, a player can always protect herself with a suitable randomized strategy—there is no disadvantage of having to move first. The proof of the Minimax theorem will also give as a byproduct a polynomial-time algorithm for computing a Nash equilibrium (by linear programming).

1.2.3 Formalism

We specify a two-player zero-sum game with an $m \times n$ payoff matrix A of numbers. The rows correspond to the possible choices of Alice (the "row player") and the columns correspond to possible choices for Bob (the "column player"). Entry A_{ij} contains Alice's payoff when Alice chooses row i and Bob chooses column j. In a zero-sum game, Bob's payoff is automatically defined to be $-A_{ij}$ (when Alice chooses row i and Bob chooses column j). Throughout the solar lectures, we normalize the payoff matrix so that $|A_{ij}| \leq 1$ for all i and j.[5]

For example, the payoff matrix corresponding to rock-paper-scissors is:

[3] https://en.wikipedia.org/wiki/Rock-paper-scissors

[4] Here are some fun facts about rock-paper-scissors. There's a World Series of RPS every year, with a top prize of at least $50K. If you watch some videos of them, you will see pure psychological welfare. Maybe this explains why some of the same players seem to end up in the later rounds of the tournament every year.

There's also a robot hand, built at the University of Tokyo, that plays rock-paper-scissors with a winning probability of 100% (check out the video). No surprise, a very high-speed camera is involved.

[5] This is without loss of generality, by scaling.

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

Mixed strategies for Alice and Bob correspond to probability distributions x and y over rows and columns, respectively. When speaking about Nash equilibria, one always assumes that players randomize independently. For a two-player zero-sum game A and mixed strategies x, y, we can write Alice's expected payoff as

$$x^\top A y = \sum_{i,j} A_{ij} x_i y_j \,.$$

Bob's expected payoff is the negative of this quantity.

1.2.4 The Minimax Theorem

The question that the Minimax theorem addresses is the following:

> If two players make choices *sequentially* in a zero-sum game, is it better to go first or second?

In a zero-sum game, there can only be a first-mover disadvantage. Going second gives a player the opportunity to adapt to what the other player does first. And the second player always has the option of choosing whatever mixed strategy she would have chosen had she gone first. But does going second ever strictly help? The Minimax theorem gives an amazing answer to the question above: *it doesn't matter!*

Theorem 1.1 (Minimax Theorem)**.** *Let A be the payoff matrix of a two-player zero-sum game. Then*

$$\max_x \left(\min_y \; x^\top A y \right) = \min_y \left(\max_x \; x^\top A y \right), \tag{1.1}$$

where x ranges over probability distributions over the rows of A and y ranges over probability distributions over the columns of A.

On the left-hand side of (1.1), the row player moves first and the column player second. The column player plays optimally given the strategy chosen by the row player, and the row player plays optimally anticipating the column player's response. On the right-hand side of (1.1), the roles of the two players are reversed. The Minimax theorem asserts that, under optimal play, the expected payoff of each player is the same in the two scenarios.

The first proof of the Minimax theorem was due to von Neumann [148] and used fixed-point-type arguments (which we'll have much more to say about later). von Neumann and Morgenstern [149], inspired by Ville [147], later realized that the Minimax theorem can be deduced from strong linear programming duality.[6]

[6]Dantzig [40, p.5] describes meeting John von Neumann on October 3, 1947: "In under a minute I slapped the geometric and the algebraic version of the [linear programming] problem on the blackboard. Von Neumann stood up and said 'Oh that!' Then for the next hour and a half, he proceeded to give me a lecture on the mathematical theory of linear programs.

"At one point seeing me sitting there with my eyes popping and my mouth open (after all I had searched the literature and found nothing), von Neumann said: 'I don't want you to think I am pulling all this out of my sleeve on the spur of the moment like a magician. I have just recently completed a book with Oskar Morgenstern on the Theory of Games. What I am doing is conjecturing that the two problems are equivalent."

This equivalence between strong linear programming duality and the Minimax theorem is made precise in Dantzig [39], Gale et al. [58], and Adler [2].

Proof. The idea is to formulate the problem faced by the first player as a linear program. The theorem will then follow from linear programming duality.

First, the player who moves second always has an optimal deterministic strategy—given the probability distribution chosen by the first player, the second player can just play the strategy with the highest expected payoff. This means the inner min and max in (1.1) may as well range over columns and rows, respectively, rather than over all probability distributions. The left expression in (1.1) then translates to the following linear program:

$$\begin{aligned} \max \quad & v \\ \text{s.t.} \quad & v \leq \sum_{i=1}^{m} A_{ij} x_i \quad \text{for all columns } j, \\ & x \text{ is a probability distribution over rows.} \end{aligned}$$

If the optimal point is (v^*, x^*), then v^* equals the left-hand-side of (1.1) and x^* belongs to the corresponding arg-max. In plain terms, x^* is what Alice should play if she has to move first, and v^* is the consequent expected payoff (assuming Bob responds optimally).

Similarly, we can write a second linear program that computes the optimal point (w^*, y^*) from Bob's perspective, where w^* equals the right-hand-side of (1.1) and y^* is in the corresponding arg-min:

$$\begin{aligned} \min \quad & w \\ \text{s.t.} \quad & w \geq \sum_{j=1}^{n} A_{ij} y_j \quad \text{for all rows } i, \\ & y \text{ is a probability distribution over columns.} \end{aligned}$$

It is straightforward to verify that these two linear programs are in fact duals of each other (left to the reader, or see Chvátal [38]). By strong linear programming duality, we know that the two linear programs have equal optimal objective function value and hence $v^* = w^*$. This means that the payoff that Alice can guarantee herself if she goes first is the same as the payoff that Bob can guarantee himself if he goes first, and this completes the proof. □

Definition 1.2 (Value of a two-player zero-sum game; min-max pair)**.** Let A be a payoff matrix of a two-player zero-sum game. Then the *value* of the game is defined to be the common value of

$$\max_x \left(\min_y \; x^\top A y \right) \quad \text{and} \quad \min_y \left(\max_x \; x^\top A y \right).$$

A *min-max strategy* is a strategy x^* in the arg-max of the left-hand side or a strategy y^* in the arg-min of the right-hind side. A *min-max pair* is a pair (x^*, y^*) where x^* and y^* are both min-max strategies.

For example, the value of the rock-paper-scissors game is 0 and (u, u) is its unique min-max pair, where u denotes the uniform probability distribution.

The min-max pairs are the optimal solutions of the two linear programs in the proof of Theorem 1.1. Since the optimal solution of a linear program can be computed in polynomial time, so can a min-max pair.

1.2.5 Nash Equilibrium

In zero-sum games, a min-max pair is closely related to the notion of a Nash equilibrium, defined next.[7]

Definition 1.3 (Nash equilibrium in a two-player zero-sum game)**.** Let A be a payoff matrix of a two-player zero-sum game. The pair $(\hat{x}, \hat{y})$ is a *Nash equilibrium* if

(i) $\hat{x}^\top A\hat{y} \geq x^\top A\hat{y}$ for all x (given that Bob plays $\hat{y}$, Alice cannot increase her expected payoff by deviating unilaterally to a strategy different from $\hat{x}$, i.e., given $\hat{y}$, $\hat{x}$ is optimal);

(ii) $\hat{x}^\top A\hat{y} \leq \hat{x}^\top Ay$ for all y (given $\hat{x}$, $\hat{y}$ is an optimal strategy for Bob).

The pairs in Definition 1.3 are sometimes called *mixed* Nash equilibria, to stress that players are allowed to randomize. (As opposed to a *pure* Nash equilibrium, where both players play deterministically.) Unless otherwise noted, we will always be concerned with mixed Nash equilibria.

Claim 1.4. *In a two-player zero-sum game, a pair (x^*, y^*) is a min-max pair if and only if it is a Nash equilibrium.*

Proof. Suppose (x^*, y^*) is a min-max pair, and so Alice's expected payoff is v^*, the value of the game. Since Alice plays her min-max strategy, Bob cannot make her payoff smaller than v^* via some other strategy. Since Bob plays his min-max strategy, Alice cannot make her payoff larger than v^*. Since neither player can do better with a unilateral deviation, (x^*, y^*) is a Nash equilibrium.

Conversely, suppose (x^*, y^*) is not a min-max pair with, say, Alice not playing a min-max strategy. If Alice's expected payoff is less than v^*, then (x^*, y^*) is not a Nash equilibrium (she could do better by deviating to a min-max strategy). Otherwise, since x^* is not a min-max strategy, Bob has a response y such that Alice's expected payoff would be strictly less than v^*. Thus Bob could do better by deviating unilaterally to y, and (x^*, y^*) is not a Nash equilibrium. □

There are several interesting consequences of Theorem 1.1 and Proposition 1.4:

1. The set of all Nash equilibria is a convex set, as the optimal solutions of a linear program form a convex set.

2. All Nash equilibria (x, y) lead to the same value of $x^\top Ay$. That is, each player receives the same expected payoff across all Nash equilibria.

3. Most importantly, since the proof of Theorem 1.1 uses linear programming duality and gives us a polynomial-time algorithm to compute a min-max pair (x^*, y^*), we have a polynomial-time algorithm to compute a Nash equilibrium of a two-player zero-sum game.

Corollary 1.5. *A Nash equilibrium of a two-player zero-sum game can be computed in polynomial time.*

1.2.6 Beyond Zero-Sum Games

There's no reason to be content with our positive results so far. Two-player zero-sum games are important—e.g., von Neumann was largely focused on them, with applications ranging from poker to war—but most game-theoretic situations are not purely oppositional.[8] Can we generalize any of their nice properties to

[7] If you think you learned this definition from the movie *A Beautiful Mind*, it's time to learn the correct definition!

[8] They can even be cooperative, for example if you and I want to meet at some intersection in Manhattan. Our strategies are intersections, and either we both get a high payoff (if we choose the same strategy) or we both get a low payoff (otherwise).

games with more players or without the zero-sum property? For example, what about *bimatrix games*, where there are still two players but the game is not necessarily zero-sum?[9] Solar Lectures 4 and 5 are devoted to this question, and provide evidence that there is no analog of Corollary 1.5 for bimatrix games.

1.2.7 Who Cares?

Before proceeding to our second cool fact about two-player zero-sum games, let's take a step back and be clear about what we're trying to accomplish. Why do we care about computing equilibria in games, anyway?

1. We might want fast algorithms to actually use in practice. The demand for equilibrium computation algorithms is significantly less than that for, say, linear programming solvers, but your lecturer regularly meets researchers who would make good use of better off-the-shelf solvers for computing an equilibrium of a game.

2. Perhaps most relevant for this week's audience, the study of equilibrium computation naturally leads to interesting and new complexity theory (e.g., definitions of new complexity classes, such as PPAD). As we'll see this week, the most celebrated results in the area are quite deep and draw on ideas from all across theoretical computer science.

3. Complexity considerations can be used to support or critique the practical relevance of an equilibrium concept such as the Nash equilibrium. It is tempting to interpret a polynomial-time algorithm for computing an equilibrium as a plausibility argument that players can figure one out quickly, and an intractability result as evidence that players will not generally reach an equilibrium in a reasonable amount of time.

   Of course, the real story is more complex. First, computational intractability is not necessarily first on the list of the Nash equilibrium's issues. For example, its non-uniqueness in non-zero-sum games already limits its predictive power.[10]

   Second, it's not particularly helpful to critique a definition without suggesting an alternative. Lunar Lecture 5 partially addresses this issue by discussing two tractable equilibrium concepts, correlated equilibria and coarse correlated equilibria.

   Third, does an arbitrary polynomial-time algorithm, such as one based on solving a non-trivial linear program, really suggest that independent play by strategic players will actually converge to an equilibrium? Algorithms for linear programming do not resemble how players typically make decisions in games. A stronger positive result would involve a behaviorally plausible distributed algorithm that players can use to efficiently converge to a Nash equilibrium through repeated play over time. We discuss such a result for two-player zero-sum games next.

1.3 Uncoupled Dynamics

In the first half of the lecture, we saw that a Nash equilibrium of a two-player zero-sum game can be computed in polynomial time. This was done by interpreting the Minimax Theorem in the framework of linear programming duality.

[9] Notice that three-player zero-sum games are already more general than bimatrix games—to turn one of the latter into one of the former, add a dummy third player with only one strategy whose payoff is the negative of the combined payoff of the original two players. Thus the most compelling negative results would be for the case of bimatrix games.

[10] Recall our "meeting in Manhattan" example—every intersection is a Nash equilibrium!

It would be more compelling, however, to come up with a definition of a plausible process by which players can learn a Nash equilibrium. Such a result requires a behavioral model for what players do when not at equilibrium. The goal is then to investigate whether or not the process converges to a Nash equilibrium (for an appropriate notion of convergence), and if so, how quickly.

1.3.1 The Setup

Uncoupled dynamics refers to a class of processes with the properties mentioned above. The idea is that each player initially knows only her own payoffs (and not those of the other players), a la the number-in-hand model in communication complexity. The game is then played repeatedly, with each player picking a strategy in each time step as a function only of her own payoffs and what transpired in the past.

Uncoupled Dynamics (Two-Player Version)

At each time step $t = 1, 2, 3, \ldots$:

1. Alice chooses a strategy x^t as a function only of her own payoffs and the previously chosen strategies $x^1, \ldots, x^{t-1}$ and $y^1, \ldots, y^{t-1}$.
2. Bob simultaneously chooses a strategy y^t as a function only of his own payoffs and the previously chosen strategies $x^1, \ldots, x^{t-1}$ and $y^1, \ldots, y^{t-1}$.
3. Alice learns y^t and Bob learns x^t.[11]

Uncoupled dynamics have been studied at length in both the game theory and computer science literatures (often under different names). Specifying such dynamics boils down to a definition of how Alice and Bob choose strategies as a function of their payoffs and the joint history of play. Let's look at some famous examples.

1.3.2 Fictitious Play

One natural idea is to best respond to the observed behavior of your opponent.

Example 1.6 (Fictitious Play)**.** In *fictitious play*, each player assumes that the other player will mix according to the relative frequencies of their past actions (i.e., the empirical distribution of their past play), and plays a best response.

Fictitious Play (Two-Player Version)

At each time step $t = 1, 2, 3, \ldots$:

1. Alice chooses a strategy x^t that is a best response against $\hat{y}^{t-1} = \frac{1}{t-1}\sum_{s=1}^{t-1} y^s$, the past actions of Bob (breaking ties arbitrarily).
2. Bob simultaneously chooses a strategy y^t that is a best response against $\hat{x}^{t-1} = \frac{1}{t-1}\sum_{s=1}^{t-1} x^s$, the past actions of Alice (breaking ties arbitrarily).

[11] When Alice and Bob use mixed strategies, there are actually two different natural feedback models, one where each player learns the actual mixed strategy chosen by the other player, and one where each learns only a sample (a pure strategy) from the other player's chosen distribution. It's generally easier to prove results in the first model, but such proofs usually can be extended with some additional work to hold (with high probability over the strategy realizations) in the second model as well.

3. Alice learns y^t and Bob learns x^t.

One way to interpret fictitious play is to imagine that each player assumes that the other is using the same mixed strategy every time step, and estimates this time-invariant mixed strategy with the empirical distribution of past actions.

Note that each player picks a pure strategy in each time step (modulo tie-breaking in the case of multiple best responses).

Fictitious play has an interesting history:

1. It was first proposed by G. W. Brown in 1949 (published in 1951 [20]) as a computer algorithm to compute a Nash equilibrium of a two-player zero-sum game. Note this is not so long after the birth of either game theory or computers!

2. In 1951, Julia Robinson (better known for her contributions to the resolution of Hilbert's tenth problem about Diophantine equations) proved that, in two-player zero-sum games, the time-averaged payoffs of the players converges to the value of the game [122]. Robinson's proof only gives an exponential (in the number of strategies) bound on the number of iterations required for convergence. In 1959, Samuel Karlin [84] conjectured that a polynomial bound should be possible (in two-player zero-sum games). Relatively recently (2014), Daskalakis and Pan [41] refuted the conjecture and proved an exponential lower bound for the case of adversarial (and not necessarily consistent) tie-breaking.

3. It is still an open question whether or not fictitious play converges quickly for natural (or even just consistent) tie-breaking rules! The goal here would be to show that $\text{poly}(n, 1/\epsilon)$ time steps suffice for the time-averaged payoffs to be within ϵ of the value of the game (where n is the number of strategies).

4. The situation for non-zero-sum games was murky until 1964, when Lloyd Shapley [138] discovered a 3×3 game (a non-zero-sum variation on rock-paper-scissors) where fictitious play never converges to a Nash equilibrium (which foreshadowed future developments on zero-sum vs. non-zero-sum games).

Next we'll look at a different choice of dynamics that has much better convergence properties.

1.3.3 Smooth Fictitious Play

Fictitious play is "all-or-nothing"—even if two strategies have almost the same expected payoff against the opponent's empirical distribution, the slightly worse one is completely ignored in favor of the slightly better one. A more stable approach, and perhaps a more behaviorally plausible one, is to assume that players randomize, biasing their decision toward the strategies with the highest expected payoffs (again, against the empirical distribution of the opponent). In other words, each player plays a "noisy best response" against the observed play of the other player. For example, already in 1957 Hannan [70] considered dynamics where each player chooses a strategy with probability proportional to its expected payoff (against the empirical distribution of the other player's past play), and proved polynomial convergence to the Nash equilibrium payoffs in two-player zero-sum games.

Even better convergence properties are possible if poorly performing strategies are abandoned more aggressively—this will correspond to a "softmax" version of fictitious play.

Example 1.7 (Smooth Fictitious Play)**.** In time t of *smooth fictitious play*, a player (Alice, say) computes the empirical distribution $\hat{y}^{t-1} = \sum_{s=1}^{t-1} x^s$ of the other player's past play, computes the expected payoff π_i^t of each pure strategy i under the assumption that Bob plays $\hat{y}^{t-1}$, and chooses x^t by playing each strategy with

probability proportional to $e^{\eta^t \pi_i^t}$. Here η^t is a tunable parameter that interpolates between always playing uniformly at random (when $\eta = 0$) and fictitious play (when $\eta = +\infty$). The choice $\eta^t \approx \sqrt{t}$ is often the best one for proving convergence results.

Smooth Fictitious Play (Two-Player Version)

Given: parameter family $\{\eta^t \in [0, \infty) : t = 1, 2, 3, \ldots\}$.

At each time step $t = 1, 2, 3, \ldots$:

1. Alice chooses a strategy x^t by playing each strategy i with probability proportional to $e^{\eta^t \pi_i^t}$, where π_i^t denotes the expected payoff of strategy i when Bob plays the mixed strategy $\hat{y}^{t-1} = \frac{1}{t-1} \sum_{s=1}^{t-1} y^s$.
2. Bob simultaneously chooses a strategy y^t by playing each strategy j with probability proportional to $e^{\eta^t \rho_j^t}$, where ρ_j^t is the expected payoff of strategy j when Alice plays the mixed strategy $\hat{x}^{t-1} = \frac{1}{t-1} \sum_{s=1}^{t-1} x^s$.
3. Alice learns y^t and Bob learns x^t.

Versions of smooth fictitious play have been studied independently in the game theory literature (beginning with Fudenberg and Levine [57]) and the computer science literature (beginning with Freund and Schapire [56]). It converges extremely quickly.

Theorem 1.8 ([57, 56]). *For a zero-sum two-player game with m rows and n columns and a parameter $\epsilon > 0$, after $T = O(\log(n+m)/\epsilon^2)$ time steps of smooth fictitious play, the empirical distributions $\hat{x} = \frac{1}{T} \sum_{t=1}^{T} x^t$ and $\hat{y} = \frac{1}{T} \sum_{t=1}^{T} y^t$ constitute an ϵ-approximate Nash equilibrium.*

The ϵ-approximate Nash equilibrium condition in Theorem 1.8 is exactly what it sounds like: neither player can improve their expected payoff by more than ϵ via a unilateral deviation (see also Definition 1.12, below). (Recall that payoffs are assumed scaled to lie in $[-1, 1]$.)

There are two steps in the proof of Theorem 1.8: (i) the noisy best response in smooth fictitious play is equivalent to the "Multiplicative Weights" algorithm, which has "vanishing regret;" and (ii) in a two-player zero-sum game, vanishing-regret guarantees translate to (approximate) Nash equilibrium convergence. The optional Sections 1.3.5–1.3.7 provide more details for the interested reader.

1.3.4 Implications for Communication Complexity

Theorem 1.8 implies that smooth fictitious play can be used to define a randomized $O(\log^2(n+m)/\epsilon^2)$-bit communication protocol for computing an ϵ-NE.[12] The goal of the next two lectures is to prove that there is no analogously efficient communication protocol for computing an approximate Nash equilibrium of a non-zero-sum game.[13] Ruling out low-communication protocols will in particular rule out any type of quickly converging uncoupled dynamics.

[12] Actually, this is for the variant of smooth fictitious play where Alice (respectively, Bob) only learns a random sample from y^t (respectively, x^t); see footnote 11. Each such sample can be communicated to the other player in $\log(n+m)$ bits. Theorem 1.8 continues to hold (with high probability over the samples) for this variant of smooth fictitious play [57, 56].

[13] The communication complexity of computing anything about a two-player zero-sum game is a silly problem—Alice knows the entire game at the beginning (since Bob's payoff is the negative of hers) and can unilaterally compute whatever she wants. But it still makes sense to ask if the communication bound implied by smooth fictitious play can be replicated in non-zero-games (where Alice and Bob initially know only their own payoff matrices).

1.3.5 Proof of Theorem 1.8, Part 1: Multiplicative Weights (Optional)

To elaborate on the first step of the proof of Theorem 1.8, we need to explain the standard setup for online decision-making.

Online Decision-Making

At each time step $t = 1, 2, \ldots, T$:

a decision-maker picks a probability distribution p^t over her actions Λ

an adversary picks a reward vector $r^t : \Lambda \to [-1, 1]$

an action a^t is chosen according to the distribution p^t, and the decision-maker receives reward $r^t(a^t)$

the decision-maker learns r^t, the entire reward vector

In smooth fictitious play, each of Alice and Bob are in effect solving the online decision-making problem (with actions corresponding to the game's strategies). For Alice, the reward vector r^t is induced by Bob's action at time step t (if Bob plays strategy j, then r^t corresponds to the jth column of the game matrix A), and similarly for Bob (with the ith row multiplied by -1). Next we interpret Alice and Bob's behavior in smooth fictitious play as algorithms for online decision-making.

An *online decision-making algorithm* specifies for each t the probability distribution p^t, as a function of the reward vectors $r^1, \ldots, r^{t-1}$ and realized actions $a^1, \ldots, a^{t-1}$ of the first $t-1$ time steps. An *adversary* for such an algorithm $\mathcal{A}$ specifies for each t the reward vector r^t, as a function of the probability distributions $p^1, \ldots, p^t$ used by $\mathcal{A}$ on the first t days and the realized actions $a^1, \ldots, a^{t-1}$ of the first $t-1$ days.

Here is a famous online decision-making algorithm, the "Multiplicative Weights (MW)" algorithm (see [100, 55, 27]).

Multiplicative Weights (MW) Algorithm

initialize $w^1(a) = 1$ for every $a \in \Lambda$

for each time step $t = 1, 2, 3, \ldots$ **do**

use the distribution $p^t := w^t/\Gamma^t$ over actions, where $\Gamma^t = \sum_{a \in \Lambda} w^t(a)$ is the sum of the weights

given the reward vector r^t, for every action $a \in \Lambda$ use the formula $w^{t+1}(a) = w^t(a) \cdot (e^{\eta^t r^t(a)})$ to update its weight (η^t is a parameter, canonically $\approx \sqrt{t}$)

The MW algorithm maintains a weight, intuitively a "credibility," for each action. At each time step the algorithm chooses an action with probability proportional to its current weight. The weight of each action evolves over time according to the action's past performance.

Inspecting the descriptions of smooth fictitious play and the MW algorithm, we see that we can rephrase the former as follows:

Smooth Fictitious Play (Rephrased)

Given: parameter family $\{\eta^t \in [0, \infty) : t = 1, 2, 3, \ldots\}$.

At each time step $t = 1, 2, 3, \ldots$:

1. Alice uses an instantiation of the MW algorithm to choose a mixed strategy x^t.
2. Bob uses a different instantiation of the MW algorithm to choose a mixed strategy y^t.
3. Alice learns y^t and Bob learns x^t.
4. Alice feeds her MW algorithm a reward vector r^t with $r^t(i)$ equal to the expected payoff of playing row i, given Bob's mixed strategy y^t over columns; and similarly for Bob.

How should we assess the performance of an online decision-making algorithm like the MW algorithm, and do guarantees for the algorithm have any implications for smooth fictitious play?

1.3.6 Proof of Theorem 1.8, Part 2: Vanishing Regret (Optional)

One of the big ideas in online learning is to compare the time-averaged reward earned by an online algorithm with that earned by the best *fixed action* in hindsight.[14]

Definition 1.9 ((Time-Averaged) Regret)**.** Fix reward vectors $r^1, \ldots, r^T$. The *regret* of the action sequence $a^1, \ldots, a^T$ is

$$\underbrace{\frac{1}{T}\max_{a \in A}\sum_{t=1}^{T} r^t(a)}_{\text{best fixed action}} - \underbrace{\frac{1}{T}\sum_{t=1}^{T} r^t(a^t)}_{\text{our algorithm}}. \tag{1.2}$$

Note that, by linearity, there is no difference between considering the best fixed action and the best fixed distribution over actions (there is always an optimal pure action in hindsight).

We'd like an online decision-making algorithm that achieves low regret, as close to 0 as possible. Since rewards lie in $[-1, 1]$, the regret can never be larger than 2. We think of regret $\Omega(1)$ (as $T \to \infty$) as an epic fail for an algorithm.

It turns out that the MW algorithm has the best-possible worst-case regret guarantee (up to constant factors).[15]

Theorem 1.10. *For every adversary, the MW algorithm has expected regret $O(\sqrt{(\log n)/T})$.*

See e.g. the book of Cesa-Bianchi and Lugosi [26] for a proof of Theorem 1.10, which is not overly difficult.

An immediate corollary is that the number of time steps needed to drive the expected regret down to a small constant is only logarithmic in the number of actions—this is surprisingly fast!

Corollary 1.11. *There is an online decision-making algorithm that, for every adversary and $\epsilon > 0$, has expected regret at most ϵ after $O((\log n)/\epsilon^2)$ time steps.*

[14] There is no hope of competing with the best action *sequence* in hindsight: consider two actions and an adversary that flips a coin each time step to choose between the reward vectors $(1, 0)$ and $(0, 1)$.

[15] For the matching lower bound, with n actions, consider an adversary that sets the reward of each action uniformly at random from $\{-1, 1\}$ at each time step. Every online algorithm earns expected cumulative reward 0, while the expected reward of the best action in hindsight is $\Theta(\sqrt{T} \cdot \sqrt{\log n})$.

1.3.7 Proof of Theorem 1.8, Part 3: Vanishing Regret Implies Convergence (Optional)

We now sketch the rest of the proof of Theorem 1.8. Consider a zero-sum game A with payoffs in $[-1, 1]$ and some $\epsilon > 0$. Let n denote the number of rows or the number of columns, whichever is larger, and set $T = \Theta((\log n)/\epsilon^2)$ so that the guarantee in Corollary 1.11 holds with error $\epsilon/2$. Let $x^1, \ldots, x^T$ and $y^1, \ldots, y^T$ be the mixed strategies used by Alice and Bob throughout T steps of smooth fictitious play. Let $\hat{x} = \frac{1}{T}\sum_{t=1}^{T} x^t$ and $\hat{y} = \frac{1}{T}\sum_{t=1}^{T} y^t$ denote the time-averaged strategies of Alice and Bob, respectively. We claim that $(\hat{x}, \hat{y})$ is an ϵ-NE.

In proof, let

$$v = \frac{1}{T}\sum_{t=1}^{T} (x^t)^\top A y^t$$

denote Alice's time-averaged payoff. Since both Alice and Bob effectively used the MW algorithm to choose their strategies, we can apply the vanishing regret guarantee in Corollary 1.11 once for each player and use linearity to obtain

$$v \geq \left(\max_x \frac{1}{T}\sum_{t=1}^{T} x^\top A y^t\right) - \frac{\epsilon}{2} = \left(\max_x x^\top A \hat{y}\right) - \frac{\epsilon}{2} \tag{1.3}$$

and

$$v \leq \left(\min_y \frac{1}{T}\sum_{t=1}^{T} (x^t)^\top A y\right) + \frac{\epsilon}{2} = \left(\min_y \hat{x}^\top A y\right) + \frac{\epsilon}{2}. \tag{1.4}$$

In particular, taking $x = \hat{x}$ in (1.3) and $y = \hat{y}$ in (1.4) shows that

$$\hat{x}^\top A \hat{y} \in \left[v - \frac{\epsilon}{2}, v + \frac{\epsilon}{2}\right]. \tag{1.5}$$

Now consider a (pure) deviation from $(\hat{x}, \hat{y})$, say by Alice to the row i. Denote this deviation by e_i. By inequality (1.3) (with $x = e_i$) we have

$$e_i^\top A \hat{y} \leq v + \frac{\epsilon}{2}. \tag{1.6}$$

Since Alice receives expected payoff at least $v - \frac{\epsilon}{2}$ in $(\hat{x}, \hat{y})$ (by (1.5)) and at most $v + \frac{\epsilon}{2}$ from any deviation (by (1.6)), her ϵ-NE conditions are satisfied. A symmetric argument applies to Bob, completing the proof.

1.4 General Bimatrix Games

Unlike a zero-sum game, a general bimatrix game has two independent payoff matrices, an $m \times n$ matrix A for Alice and an $m \times n$ matrix B for Bob. (In a zero-sum game, $B = -A$.) The definition of an (approximate) Nash equilibrium is what you'd think it would be:

Definition 1.12 (ϵ-Approximate Nash Equilibrium)**.** For a bimatrix game (A, B), row and column mixed strategies $\hat{x}$ and $\hat{y}$ constitute an ϵ-NE if

$$\hat{x}^\top A \hat{y} \geq x^\top A \hat{y} - \epsilon \qquad \forall x, \text{ and} \tag{1.7}$$

$$\hat{x}^\top B \hat{y} \geq \hat{x}^\top B y - \epsilon \qquad \forall y\,. \tag{1.8}$$

It has long been known that many of the nice properties of zero-sum games break down in general bimatrix games.[16]

[16] We already mentioned Shapley's 1964 example showing that fictitious play need not converge [138].

Example 1.13 (Strange Bimatrix Behavior)**.** Suppose two friends, Alice and Bob, want to go for dinner, and are trying to settle on a choice of restaurant. Alice prefers Italian over Thai, and Bob prefers Thai over Italian, but both would rather eat together than eat alone. Supposing the rows and columns are indexed by Italian and Thai, in that order, and Alice is the row player, we get the following payoff matrices:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \qquad \text{or, in shorthand,} \qquad (A, B) = \begin{bmatrix} (2,1) & (0,0) \\ (0,0) & (1,2) \end{bmatrix}.$$

There are two obvious Nash equilibria, both pure: either Alice and Bob go to the Italian restaurant, or they both go to the Thai restaurant. But there's a third Nash equilibrium, a mixed one[17]: Alice chooses Italian over Thai with probability $\frac{2}{3}$, and Bob chooses Thai over Italian with probability $\frac{2}{3}$. This is an undesirable Nash equilibrium, with Alice and Bob eating alone more than half the time.

Example 1.13 shows that, unlike in zero-sum games, different Nash equilibria can result in different expected player payoffs. Similarly, the Nash equilibria of a bimatrix game do not generally form a convex set (unlike in the zero-sum case).

Nash equilibria of bimatrix games are not completely devoid of nice properties, however. For starters, we have guaranteed existence.

Theorem 1.14 (Nash's Theorem [114, 113])**.** *Every bimatrix game has at least one (mixed) Nash equilibrium.*

The proof is a fixed-point argument that we will have more to say about in Solar Lecture 2.[18] Nash's theorem holds more generally for games with any finite number of players and strategies.

Nash equilibria of bimatrix games have nicer structure than those in games with three or more players. First, in bimatrix games with integer payoffs, there is a Nash equilibrium in which all probabilities are rational numbers with bit complexity polynomial in that of the game.[19] Second, there is a simplex-type pivoting algorithm, called the *Lemke-Howson algorithm* [96], which computes a Nash equilibrium of a bimatrix game in a finite number of steps (see von Stengel [150] for a survey). Like the simplex method, the Lemke-Howson algorithm takes an exponential number of steps in the worst case [109, 136]. Nevertheless, the similarities between Nash equilibria of bimatrix games and optimal solutions of linear programs originally led to some optimism that a polynomial-time algorithm might exist. Alas, as we'll see, this does not seem to be the case.

1.5 Approximate Nash Equilibria in Bimatrix Games

The last topic of this lecture is some semi-positive results about *approximate* Nash equilibria in general bimatrix games. While simple, these results are important and will show up repeatedly in the rest of the lectures.

1.5.1 Sparse Approximate Nash Equilibria

Here is a crucial result for us: there are always *sparse* approximate Nash equilibria.[20,21]

[17] Fun fact: outside of degenerate cases, every game has an *odd* number of Nash equilibria (see also Solar Lecture 4).

[18] Von Neumann's alleged reaction when Nash told him his theorem [112, P.94]: "That's trivial, you know. That's just a fixed point theorem."

[19] Exercise: prove this by showing that, after you've guessed the two support sets of a Nash equilibrium, you can recover the exact probabilities using two linear programs.

[20] Althöfer [4] and Lipton and Young [98] independently proved a precursor to this result in the special case of zero-sum games. The focus of the latter paper is applications in complexity theory (like "anticheckers").

[21] Exercise: there are arbitrarily large games where every exact Nash equilibrium has full support. Hint: generalize rock-paper-scissors. Alternatively, see Section 5.2.6 of Solar Lecture 5.

Theorem 1.15 (Lipton et al. [99]). *For every $\epsilon > 0$ and every $n \times n$ bimatrix game, there exists an ϵ-NE in which each player randomizes uniformly over a multi-set of $O((\log n)/\epsilon^2)$ pure strategies.*[22]

Proof idea. Fix an $n \times n$ bimatrix game (A, B).

1. Let (x^*, y^*) be an exact Nash equilibrium of (A, B). (One exists, by Theorem 1.14.)
2. As a thought experiment, sample $\Theta((\log n)/\epsilon^2)$ pure strategies for Alice i.i.d. (with replacement) from x^*, and similarly for Bob i.i.d. from y^*.
3. Let $\hat{x}, \hat{y}$ denote the empirical distributions of the samples (with probabilities equal to frequencies in the sample)—equivalently, the uniform distributions over the two multi-sets of pure strategies.
4. Use Chernoff bounds to argue that $(\hat{x}, \hat{y})$ is an ϵ-NE (with high probability). Specifically, because of our choice of the number of samples, the expected payoff of each row strategy w.r.t. $\hat{y}$ differs from that w.r.t. y^* by at most $\epsilon/2$ (w.h.p.). Since every strategy played with non-zero probability in x^* is an exact best response to y^*, every strategy played with non-zero probability in $\hat{x}$ is within ϵ of a best response to $\hat{y}$. Similarly with the roles of $\hat{x}$ and $\hat{y}$ reversed. This is a sufficient condition for being an ϵ-NE.[23]

□

1.5.2 Implications for Communication Complexity

Theorem 1.15 immediately implies the existence of an ϵ-NE of an $n \times n$ bimatrix game with description length $O((\log^2 n)/\epsilon^2)$, with $\approx \log n$ bits used to describe each of the $O((\log n)/\epsilon^2)$ pure strategies in the multi-sets promised by the theorem. Moreover, if an all-powerful prover writes down an alleged such description on a publicly observable blackboard, then Alice and Bob can privately verify that the described pair of mixed strategies is indeed an ϵ-NE. For example, Alice can use the (publicly viewable) description of Bob's mixed strategy to compute the expected payoff of her best response and check that it is at most ϵ more than her expected payoff when playing the mixed strategy suggested by the prover. Summarizing:

Corollary 1.16. *The nondeterministic communication complexity of computing an ϵ-NE of an $n \times n$ bimatrix game is $O((\log^2 n)/\epsilon^2)$.*

Thus, if there *is* a polynomial lower bound on the deterministic or randomized communication complexity of computing an approximate Nash equilibrium, the only way to prove it is via techniques that don't automatically apply also to the problem's nondeterministic communication complexity. This observation rules out many of the most common lower bound techniques. In Solar Lectures 2 and 3, we'll see how to thread the needle using a *simulation theorem*, which lifts a deterministic or random query (i.e., decision tree) lower bound to an analogous communication complexity lower bound.

1.5.3 Implications for Computational Complexity

The second important consequence of Theorem 1.15 is a limit on the worst-possible computational hardness we could hope to prove for the problem of computing an approximate Nash equilibrium of a bimatrix game: at worst, the problem is quasi-polynomial-hard.

[22] By a padding argument, there is no loss of generality in assuming that Alice and Bob have the same number of strategies.

[23] This sufficient condition has its own name: a *well-supported* ϵ-NE.

Corollary 1.17. *There is an algorithm that, given as input a description of an $n \times n$ bimatrix game and a parameter ϵ, outputs an ϵ-NE in $O(n^{(\log n)/\epsilon^2})$ time.*

Proof. The algorithm enumerates all $O(n^{(\log n)/\epsilon^2})$ possible choices for the multi-sets promised by Theorem 1.15. It is easy to check whether or not the mixed strategies induced by a choice constitute an ϵ-NE—just compute the expected payoffs of each strategy and of the players' best responses, as in the proof of Corollary 1.16. □

The quasi-polynomial-time approximation scheme (QPTAS) in Corollary 1.17 initially led to optimism that there should be a PTAS for the problem, in light of the paucity of natural problems for which the best-possible running time is quasi-polynomial. Also, if there *were* a reduction showing quasi-polynomial-time hardness for computing an approximate Nash equilibrium, what would it look like? Solar Lectures 4 and 5 will answer this question.

Solar Lecture 2

Communication Complexity Lower Bound for Computing an Approximate Nash Equilibrium of a Bimatrix Game (Part I)

Lecturer: Tim Roughgarden *Scribe: Valentine Kabanets and Antonina Kolokolova*

This lecture and the next consider the communication complexity of computing an approximate Nash equilibrium, culminating with a proof of the very recent breakthrough polynomial lower bound of Babichenko and Rubinstein [9]. This lower bound rules out the possibility of quickly converging uncoupled dynamics in general bimatrix games (see Section 1.3).

2.1 Preamble

Recall the setup: there are two players, Alice and Bob, each with their own payoff matrices A and B. Without loss of generality (by padding), the two players have the same number N of strategies. We consider a two-party model where, initially, Alice only knows A and Bob only knows B. The goal is then for Alice and Bob to compute an approximate Nash equilibrium (Definition 1.12) with as little communication as possible.

This lecture and the next explain all of the main ideas behind the following result:

Theorem 2.1 (Babichenko and Rubinstein [9])**.** *There is a constant $c > 0$ such that, for all sufficiently small constants $\epsilon > 0$ and sufficiently large N, the randomized communication complexity of computing an ϵ-NE is $\Omega(N^c)$.*

For our purposes, a randomized protocol with communication cost b always uses at most b bits of communication, and terminates with at least one player knowing an ϵ-NE of the game with probability at least $\frac{1}{2}$ (over the protocol's coin flips).

Thus, while there are lots of obstacles to players reaching an equilibrium of a game (see also Section 1.2.7), communication alone is already a significant bottleneck. A corollary of Theorem 2.1 is that there can be no uncoupled dynamics (Section 1.3) that converge to an approximate Nash equilibrium in a sub-polynomial number of rounds in general bimatrix games (cf., the guarantee in Theorem 1.8 for smooth fictitious play in zero-sum games). This is because uncoupled dynamics can be simulated by a randomized communication protocol with logarithmic overhead (to communicate which strategy gets played each round).[1] This corollary should be regarded as a fundamental contribution to pure game theory and economics.

The goal of this and the next lecture is to sketch a full proof of the lower bound in Theorem 2.1 for deterministic communication protocols. We do really care about randomized protocols, however, as these are the types of protocols induced by uncoupled dynamics (see Section 1.3.4). The good news is that

[1] See also footnote 12 in Solar Lecture 1.

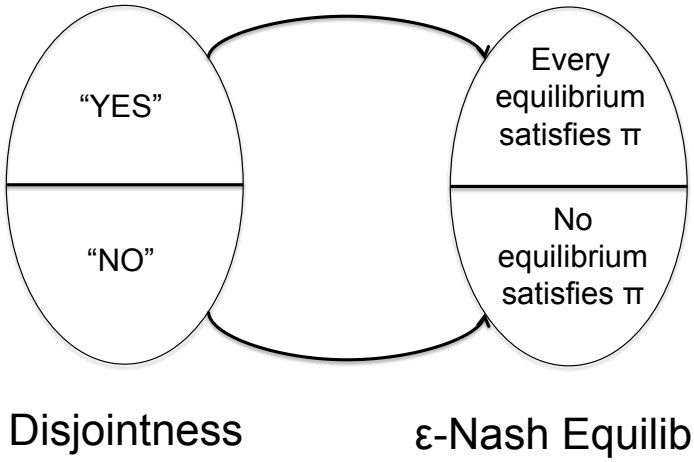

Figure 2.1: A naive attempt to reduce the Disjointness problem to the problem of computing an approximate Nash equilibrium.

the argument for the deterministic case will already showcase all of the conceptual ideas in the proof of Theorem 1.8. Extending the proof to randomized protocols requires substituting a simulation theorem for randomized protocols (we'll use only a simulation theorem for deterministic protocols, see Theorem 2.7) and a few other minor tweaks.[2]

2.2 Naive Approach: Reduction From Disjointness

To illustrate the difficulty of proving a result like Theorem 2.1, consider a naive attempt that tries to reduce, say, the Disjointness problem to the problem of computing an ϵ-NE, with YES-instances mapped to games in which all equilibria have some property Π, and NO-instances mapped to games in which no equilibrium has property Π (Figure 2.1).[3] For the reduction to be useful, Π needs to be some property that can be checked with little to no communication, such as "Alice plays her first strategy with positive probability" or "Bob's strategy has full support." The only problem is that *this is impossible!* The reason is that the problem of computing an approximate Nash equilibrium has polylogarithmic *nondeterministic* communication complexity (because of the existence of sparse approximate equilibria, see Theorem 1.15 and Corollary 1.16), while the Disjointness function does not (for 1-inputs). A reduction of the proposed form would translate a nondeterministic lower bound for the latter problem to one for the former, and hence cannot exist.

Our failed reduction highlights two different challenges. The first is to resolve the typechecking error that we encountered between a standard decision problem, where there might or might not be a witness

[2]When Babichenko and Rubinstein [9] first proved their result (in late 2016), the state-of-the-art in simultaneous theorems for randomized protocols was much more primitive than for deterministic protocols. This forced Babichenko and Rubinstein [9] to use a relatively weak simulation theorem for the randomized case (by Göös et al. [65]), which led to a number of additional technical details in the proof. Amazingly, a full-blown randomized simulation theorem was published shortly after this workshop [67, 5]! With this in hand, extending the argument here for deterministic protocols to randomized protocols is relatively straightforward.

[3]Recall the Disjointness function: Alice and Bob have input strings $a, b \in \{0,1\}^n$, and the output of the function is "0" if there is a coordinate $i \in \{1, 2, \ldots, n\}$ with $a_i = b_i = 1$ and "1" otherwise. One of the first things you learn in communication complexity is that the nondeterministic communication complexity of Disjointness (for certifying 1-inputs) is n (see e.g. [93, 130]). And of course one of the most famous and useful results in communication complexity is that the function's randomized communication complexity (with two-sided error) is $\Omega(n)$ [83, 121].

(like DISJOINTNESS, where a witness is an element in the intersection), and a total search problem where there is always a witness (like computing an approximate Nash equilibrium, which is guaranteed to exist by Nash's theorem). The second challenge is to figure out how to prove a strong lower bound on the deterministic or randomized communication complexity of computing an approximate Nash equilibrium without inadvertently proving the same (non-existent) lower bound for nondeterministic protocols. To resolve the second challenge, we'll make use of simulation theorems that lift query complexity lower bounds to communication complexity lower bounds (see Section 2.7); these are tailored to a specific computational model, like deterministic or randomized protocols. For the first challenge, we need to identify a total search problem with high communication complexity. That is, for total search problems, which should be the analog of 3SAT or DISJOINTNESS? The correct answer turns out to be *fixed-point computation*.

2.3 Finding Brouwer Fixed Points (The ϵ-BFP Problem)

This section and the next describe reductions from computing Nash equilibria to computing fixed points, and from computing fixed points to a path-following problem. These reductions are classical. The content of the proof in Theorem 2.1 are *reductions in the opposite direction*; these are discussed in Solar Lecture 3.

2.3.1 Brouwer's Fixed-Point Theorem

Brouwer's fixed-point theorem states that whenever you stir your coffee, there will be a point that ends up exactly where it began. Or if you prefer a more formal statement:

Theorem 2.2 (Brouwer's Fixed-Point Theorem (1910)). *If C is a compact convex subset of $\mathbb{R}^d$, and $f : C \to C$ is continuous, then there exists a* fixed point*: a point $x \in C$ with $f(x) = x$.*

All of the hypotheses are necessary.[4] We will be interested in a computational version of Brouwer's fixed-point theorem, the ϵ-BFP *problem*:

The ϵ-BFP Problem (Generic Version)

given a description of a compact convex set $C \subseteq \mathbb{R}^d$ and a continuous function $f : C \to C$, output an *ϵ-approximate fixed point*, meaning a point $x \in C$ such that $\|f(x) - x\| < \epsilon$.

The ϵ-BFP problem, in its many different forms, plays a starring role in the study of equilibrium computation. The set C is typically fixed in advance, for example to the d-dimensional hypercube. While much of the work on the ϵ-BFP problem has focused on the ℓ_∞ norm (e.g. [74]), one innovation in the proof of Theorem 2.1 is to instead use a normalized version of the ℓ_2 norm (following Rubinstein [135]).

Nailing down the problem precisely requires committing to a family of succinctly described continuous functions f. The description of the family used in the proof of Theorem 2.1 is technical and best left to Section 3.1. Often (and in these lectures), the family of functions considered contains only $O(1)$-Lipschitz functions.[5] In particular, this guarantees the existence of an ϵ-approximate fixed point with description length polynomial in the dimension and $\log \frac{1}{\epsilon}$ (by rounding an exact fixed point to its nearest neighbor on a suitably defined grid).

[4] If convexity is dropped, consider rotating an annulus centered at the origin. If boundedness is dropped, consider $x \mapsto x + 1$ on $\mathbb{R}$. If closedness is dropped, consider $x \mapsto \frac{x}{2}$ on $(0, 1]$. If continuity is dropped, consider $x \mapsto (x + \frac{1}{2}) \bmod 1$ on $[0, 1]$. Many more general fixed-point theorems are known, and find applications in economics and elsewhere; see e.g. [15, 103].

[5] Recall that a function f defined on a metric space (X, d) is *λ-Lipschitz* if $d(f(x), f(y)) \leq \lambda \cdot d(x, y)$ for all $x, y \in X$. That is, the function can only amplify distances between points by a λ factor.

2.3.2 From Brouwer to Nash

Fixed-point theorems have long been used to prove equilibrium existence results, including the original proofs of the Minimax theorem (Theorem 1.1) and Nash's theorem (Theorem 1.14).[6] Analogously, algorithms for computing (approximate) fixed points can be used to compute (approximate) Nash equilibria.

Fact 2.3. Existence/computation of ϵ-NE reduces to that of ϵ-BFP.

To provide further details, let's sketch why Nash's theorem (Theorem 1.14) reduces to Brouwer's fixed-point theorem (Theorem 2.2), following the version of the argument in Geanakoplos [60].[7] Consider a bimatrix game (A, B) and let S_1, S_2 denote the strategy sets of Alice and Bob (i.e., the rows and columns). The relevant convex compact set is $C = \Delta_1 \times \Delta_2$, where Δ_i is the simplex representing the mixed strategies over S_i. We want to define a continuous function $f : C \to C$, from mixed strategy profiles to mixed strategy profiles, such that the fixed points of f are the Nash equilibria of this game. We define f separately for each component $f_i : C \to \Delta_i$ for $i = 1, 2$. A natural idea is to set f_i to be a best response of player i to the mixed strategy of the other player. This does not lead to a continuous, or even well defined, function. We can instead use a "regularized" version of this idea, defining

$$f_1(x_1, x_2) = \operatorname*{argmax}_{x_1' \in \Delta_1} g_1(x_1', x_2), \tag{2.1}$$

where

$$g_1(x_1', x_2) = \underbrace{\mathbf{E}_{j \sim x_1', \ell \sim x_2}\left[A_{j\ell}\right]}_{\text{linear in } x_1'} - \underbrace{\|x_1' - x_1\|_2^2}_{\text{strictly convex}}, \tag{2.2}$$

and similarly for f_2 and g_2 (with Bob's payoff matrix B). The first term of the function g_i encourages a best response while the second "penalty term" discourages big changes to player i's mixed strategy. Because the function g_i is strictly concave in x_i', f_i is well defined. The function $f = (f_1, f_2)$ is continuous (as you should check). By definition, every Nash equilibrium of the given game is a fixed point of f. For the converse, suppose that (x_1, x_2) is not a Nash equilibrium, with Alice (say) able to increase her expected payoff by deviating unilaterally from x_1 to x_1'. A simple computation shows that, for sufficiently small $\epsilon > 0$, $g_1((1-\epsilon)x_1 + \epsilon x_1', x_2) > g_1(x_1, x_2)$, and hence (x_1, x_2) is not a fixed point of f (as you should check).

Summarizing, an oracle for computing Brouwer fixed points immediately gives an oracle for computing a Nash equilibrium of a bimatrix game. The same argument applies to games with any (finite) number of players. The same argument also shows that an oracle for computing an ϵ-approximate fixed point in the ℓ_∞ norm can be used to compute an $O(\epsilon)$-approximate Nash equilibrium of a game. The first high-level goal of the proof of Theorem 2.1 is to reverse the direction of the reduction—to show that the problem of computing an approximate Nash equilibrium is as general as computing an approximate fixed point, rather than merely being a special case.

Goal #1
ϵ-BFP $\leq$ ϵ-NE

[6] In fact, the story behind von Neumann's original proof of the Minimax theorem is a little more complicated and nuanced; see Kjeldsen [89] for a fascinating and detailed discussion.

[7] This discussion is borrowed from [129, Lecture 20].

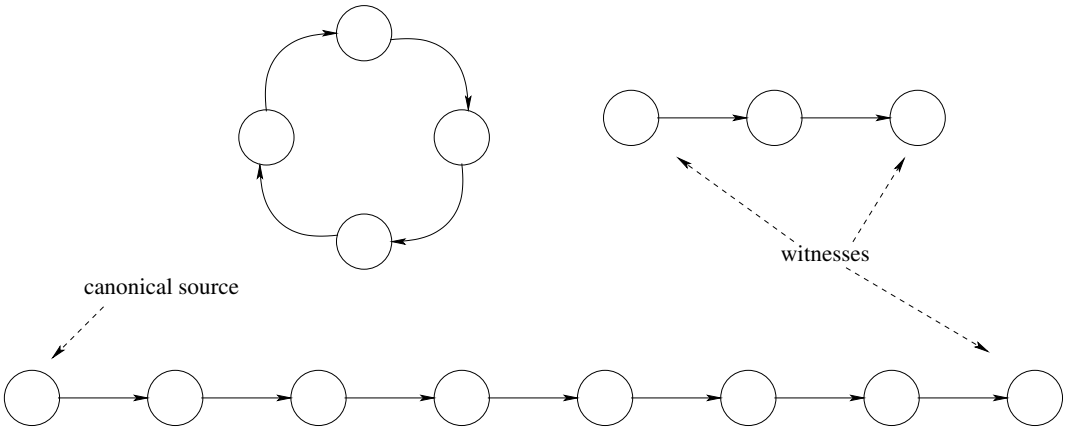

Figure 2.2: An instance of the EoL problem corresponds to a directed graph with all in- and out-degrees at most 1. Solutions correspond to sink vertices and source vertices other than the given one.

This goal follows in the tradition of a sequence of celebrated computational hardness results last decade for computing an exact Nash equilibrium (or an ϵ-approximate Nash equilibrium with ϵ polynomial in $\frac{1}{n}$) [44, 34].

There are a couple of immediate issues. First, it's not clear how to meaningfully define the ϵ-BFP problem in a two-party communication model—what are Alice's and Bob's inputs? We'll address this issue in Section 3.1. Second, even if we figure out how to define the ϵ-BFP problem and implement goal #1, so that the ϵ-NE problem is at least as hard as the ϵ-BFP problem, what makes us so sure that the latter is hard? This brings us to our next topic—a "generic" total search problem that is hard almost by definition and can be used to transfer hardness to other problems (like ϵ-BFP) via reductions.[8]

2.4 The End-of-the-Line (EoL) Problem

For equilibrium and fixed-point computation problems, it turns out that the appropriate "generic" problem involves following a path in a large graph; see also Figure 2.2.

The EoL Problem (Generic Version)

given a description of a directed graph G with maximum in- and out-degree 1, and a source vertex s of G, find either a sink vertex of G or a source vertex other than s.

The restriction on the in- and out-degrees forces the graph G to consist of vertex-disjoint paths and cycles, with at least one path (starting at the source s). The EoL problem is a total search problem—there is always a solution, if nothing else the other end of the path that starts at s. Thus an instance of EoL can always be solved by rotely following the path from s; the question is whether or not there is a more clever algorithm that always avoids searching the entire graph.

It should be plausible that the EoL problem is hard, in the sense that there is no algorithm that always improves over rote path-following; see also Section 2.6. But what does it have to do with the ϵ-BFP problem? A lot, it turns out.

[8] For an analogy, a "generic" hard problem for the complexity class NP is: given a description of a polynomial-time verifier, does there exist a witness (i.e., an input accepted by the verifier)?

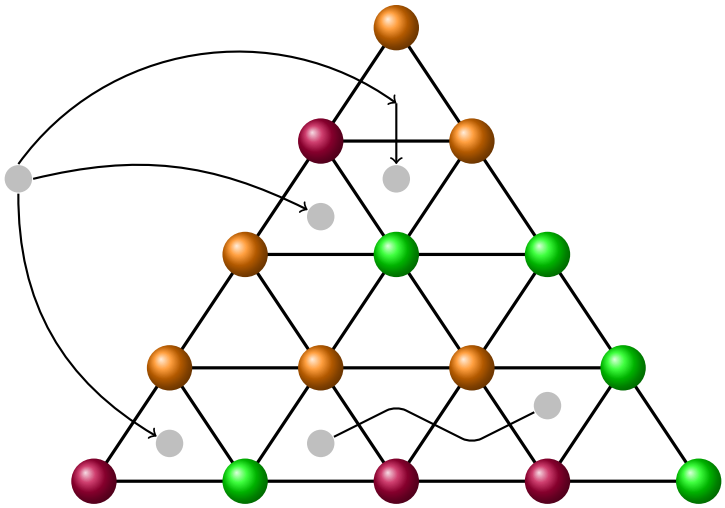

Figure 2.3: The proof of Sperner's lemma, in action.

Fact 2.4. The problem of computing an approximate Brouwer fixed point reduces to the EoL problem (i.e., ϵ-BFP $\leq$ EoL).

The basic reason that fixed-point computation reduces to path-following is *Sperner's Lemma*.

Lemma 2.5 (Sperner's Lemma)**.** *Consider a triangle with vertices A, B, C and a triangulation T of it. Suppose the vertices S of the triangulation T are 3-colored so that (1) A, B, and C receive distinct colors; and (2) each vertex on the boundary of ABC is colored with one of the two colors of the endpoints of its side of the triangle. Then, there exists an odd number of triangles of T for which all three corners have distinct colors.*

Figure 2.3 illustrates the proof of Sperner's lemma: starting at a source vertex outside the triangle, one can continue moving through the triangulation (entering via a bichromatic edge) until one finally hits a trichromatic triangle; in the picture, one enters each time via a red-yellow edge. In other words, finding a trichromatic triangle reduces to the EoL problem.[9]

Next we'll use Sperner's lemma to prove Brouwer's fixed-point theorem for a 2-dimensional simplex Δ. Let $f : \Delta \to \Delta$ be a λ-Lipschitz function (with respect to the ℓ_2 norm, say).

1. Subdivide Δ into sub-triangles with side length at most ϵ/λ. Think of the points of Δ as parameterized by three coordinates (x, y, z), with $x, y, z \geq 0$ and $x + y + z = 1$.

2. Associate each of the three coordinates with a distinct color. To color a point (x, y, z), consider its image (x', y', z') under f and choose the color of a coordinate that strictly decreased (if there are none, then (x, y, z) is a fixed point and we're done). Note that the conditions of Sperner's Lemma are satisfied.

[9] We're glossing over some details of the reduction. First, there is a canonical way to direct the edges of the graph induced by a coloring, resulting in a directed graph as required for the EoL problem. Second, the source vertex outside the triangle can have odd degree $2k - 1$ larger than 1; splitting it into k vertices (a source vertex with out-degree 1 and $k - 1$ vertices with in- and out-degree 1) yields a graph of the required form.

3. We claim that the center $(\bar{x}, \bar{y}, \bar{z})$ of a trichromatic triangle must be an $O(\epsilon)$-fixed point (in the ℓ_∞ norm). Because some corner of the triangle has its x-coordinate go down under f, $(\bar{x}, \bar{y}, \bar{z})$ is at distance at most ϵ/λ from this corner, and f is λ-Lipschitz, the x-coordinate of $f(\bar{x}, \bar{y}, \bar{z})$ is at most $\bar{x} + O(\epsilon)$. The same argument applies to $\bar{y}$ and $\bar{z}$, which implies that each of the coordinates of $f(\bar{x}, \bar{y}, \bar{z})$ is within $\pm O(\epsilon)$ of the corresponding coordinate of $(\bar{x}, \bar{y}, \bar{z})$.

By an inductive argument, Sperner's Lemma extends to simplices in any (finite) number of dimensions. The number of colors is one more than the number of dimensions, and a path-following argument implies the existence of (an odd number of) sub-simplices for which every corner has a different color. Analogs of Sperner's Lemma can also be proved by similar arguments for other natural sets, like hypercubes.[10]

This brings us to the second high-level goal of the proof of Theorem 2.1: to reverse the direction of the above reduction from ϵ-BFP to EoL. That is, we would like to show that the problem of computing an approximate Brouwer fixed point is as general as every path-following problem (of the form in EoL), and is not merely a special case.

Goal #2
EoL $\leq \epsilon$-BFP

If we succeed in implementing goals #1 and #2, and also prove directly that the EoL problem is hard, then we'll have proven hardness for the problem of computing an approximate Nash equilibrium.

2.5 Road Map for the Proof of Theorem 2.1

The high-level plan for the proof in the rest of this and the next lecture is to show that

a low-cost communication protocol for ϵ-NE

implies

a low-cost communication protocol for ϵ-2BFP,

where ϵ-2BFP is a two-party version of the problem of computing a fixed point (to be defined), which then implies

a low-cost communication protocol for 2EoL,

where 2EoL is a two-party version of the End-of-the-Line problem (to be defined), which then implies

a low-query algorithm for EoL.

Finally, we'll prove directly that the EoL problem does not admit a low-query algorithm. This gives us four things to prove (hardness of EoL and the three implications); we'll tackle them one-by-one in reverse order. The first step (Section 2.6)) is easy. The second step (Section 2.7) follows directly from one of the simulation theorems alluded to in Section 2.1. The last two steps, which correspond to goals #2 and #1, respectively, are harder and deferred to Solar Lecture 3.

[10] Brouwer's fixed-point theorem in its full generality follows easily from Sperner's Lemma. To prove Brouwer's fixed-point theorem for simplices (of any dimension), take the limit $\epsilon \to 0$ in the argument above and use the continuity of f. Because every compact convex subset of finite-dimensional Euclidean space is homeomorphic to a simplex of the same dimension (by scaling and radial projection, essentially), Brouwer's fixed-point theorem carries over to all compact convex subsets of Euclidean space.

Most of the ingredients in this road map were already present in a paper by Omri Weinstein and your lecturer [133], which was the first paper to define and study two-party versions of fixed-point computation problems, and to propose the use of simulation theorems in the context of equilibrium computation. One major innovation in Babichenko and Rubinstein [9] is the use of the generic EoL problem as the base of the reduction, thereby eluding the tricky interactions in [133] between simulation theorems (which seem inherently combinatorial) and fixed-point problems (which seem inherently geometric). Roughgarden and Weinstein [133] applied a simulation theorem directly to a fixed-point problem (relying on strong query complexity lower bounds for finding fixed points [74, 8]), which yielded a hard but unwieldy version of a two-party fixed-point problem. It is not clear how to reduce this version to the problem of computing an approximate Nash equilibrium. Babichenko and Rubinstein [9] instead apply a simulation theorem directly to the EoL problem, which results in a reasonably natural two-party version of the problem (see Section 2.7). There is significant flexibility in how to interpret this problem as a two-party fixed-point problem, and the interpretation in Babichenko and Rubinstein [9] (see Section 3.1) yields a version of the problem that is hard and yet structured enough to be solved using approximate Nash equilibrium computation. A second innovation in [9] is the reduction from ϵ-2BFP to ϵ-NE (see Section 3.2) which, while not difficult, is both new and clever.

2.6 Step 1: Query Lower Bound for EoL

We consider the following "oracle" version of the EoL problem. The vertex set V is fixed to be $\{0,1\}^n$. Let $N = |V| = 2^n$. Algorithms are allowed to access the graph only through vertex queries. A query to the vertex v reveals its predecessor $pred(v)$ (if any, otherwise $pred(v)$ is NULL) and its successor $succ(v)$ (or NULL if it has no successor). The interpretation is that the directed edge (v, w) belongs to the implicitly defined directed graph $G = (V, E)$ if and only if both $succ(v) = w$ and $pred(w) = v$. These semantics guarantee that the graph has in- and out-degree at most 1. We also assume $pred(0^n) = NULL$, and interpret the vertex 0^n as the a priori known source vertex of the graph.

The version of the EoL problem for this oracle model is:

The EoL Problem (Query Version)

given an oracle as above, find a vertex $v \in V$ that satisfies one of the following:

(i) $succ(v)$ is NULL;

(ii) $pred(v)$ is NULL and $v \neq 0^n$;

(iii) $v \neq pred(succ(v))$; or

(iv) $v \neq succ(pred(v))$ and $v \neq 0^n$.

According to our semantics, cases (iii) and (iv) imply that v is a sink and source vertex, respectively. A solution is guaranteed to exist—if nothing else, the other end of the path of G that originates with the vertex 0^n.

It will sometimes be convenient to restrict ourselves to a "promise" version of the EoL problem (which can only be easier), where the graph G is guaranteed to be a single Hamiltonian path. Even in this special case, because every vertex query reveals information about only three vertices or less, we have the following.

Proposition 2.6. *Every deterministic algorithm that solves the* EoL *problem requires* $\Omega(N)$ *queries in the worst case, even for instances that consist of a single Hamiltonian path.*

Slightly more formally, consider an adversary that always responds with values of $succ(v)$ and $pred(v)$ that are never-before-seen vertices (except as necessary to maintain the consistency of all of the adversary's answers, so that cases (iii) and (iv) never occur). After only $o(N)$ queries, the known parts of G constitute a bunch of vertex-disjoint paths, and G could still be (among other things) any Hamiltonian path of V consistent with these. The end of this Hamiltonian path could be any of $\Omega(N)$ different vertices, and the algorithm has no way of knowing which one.[11]

2.7 Step 2: Communication Complexity Lower Bound for 2EoL via a Simulation Theorem

Our next step is to use a "simulation theorem" to transfer our query lower bound for the EoL problem to a communication complexity lower bound for a two-party version of the problem, 2EoL.[12] The exact definition of the 2EoL problem will be determined by the output of the simulation theorem.

2.7.1 The Query Model

Consider an arbitrary function $f : \Sigma^N \to \Sigma$, where Σ denotes a finite alphabet. There is an input $\mathbf{z} = (z_1, \ldots, z_N) \in \Sigma^N$, initially unknown to an algorithm. The algorithm can query the input $\mathbf{z}$ adaptively, with each query revealing z_i for a coordinate i of the algorithm's choosing. It is trivial to evaluate $f(\mathbf{z})$ using N queries; the question is whether or not there is an algorithm that always does better (for some function f of interest). For example, the query version of the EoL problem in Proposition 2.6 can be viewed as a special case of this model, with $\Sigma = \{0,1\}^n \times \{0,1\}^n$ (to encode $pred(v)$ and $succ(v)$) and $f(z)$ encoding the (unique) vertex at the end of the Hamiltonian path.

2.7.2 Simulation Theorems

We now describe how a function $f : \Sigma^N \to \Sigma$ as above induces a two-party communication problem. The idea is to "factor" the input $\mathbf{z} = (z_1, \ldots, z_N)$ to the query version of the problem between Alice and Bob, so that neither player can unilaterally figure out any coordinate of $\mathbf{z}$. We use an Index gadget for this purpose, as follows. (See also Figure 2.4.)

Two-Party Problem Induced by $f : \Sigma^N \to \Sigma$

Alice's input: N "blocks" $A_1, \ldots, A_N$. Each block has $M = \text{poly}(N)$ entries (with each entry in Σ). (Say, $M = N^{20}$.)

Bob's input: N indices $y_1, \ldots, y_N \in [M]$.

Communication problem: compute $f(A_1[y_1], \ldots, A_N[y_N])$.

[11] A similar argument, based on choosing a Hamiltonian path of V at random, implies an $\Omega(N)$ lower bound for the randomized query complexity as well.

[12] These notes do not reflect a beautiful lecture given by Omri Weinstein on the history and applications of simulation theorems (e.g., to the first non-trivial lower bounds for the clique vs. independent set problem [64]). Contact him for his slides!

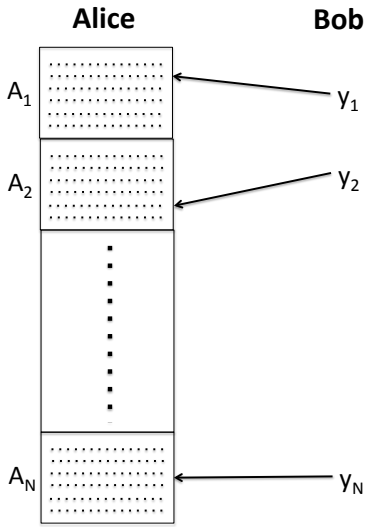

Figure 2.4: A query problem induces a two-party communication problem. Alice receives N blocks, each containing a list of possible values for a given coordinate of the input. Bob receives N indices, specifying where in Alice's blocks the actual coordinates of the input reside.

Note that the y_ith entry of A_i—Bob's index into Alice's block—is playing the role of z_i in the original problem. Thus each block A_i of Alice's input can be thought of as a "bag of garbage;" it tells Alice a huge number of possible values for the ith coordinate of the input, without any clue about which is the "real one." Meanwhile, Bob's indices tell him the locations of the real values, without any clues about what these values are.

If f can be evaluated with a query algorithm that always uses at most q queries, then the induced two-party problem can be solved using $O(q \log N)$ bits of communication. For Alice can just simulate the query algorithm; whenever it needs to query the ith coordinate of the input, Alice asks Bob for his index y_i and supplies the query algorithm with $A_i[y_i]$. Each of the at most q questions posed by Alice can be communicated with $\approx \log N$ bits, and each answer from Bob with $\approx \log M = O(\log N)$ bits.

There could also be communication protocols for the two-party problem that look nothing like the straightforward simulation. For example, Alice and Bob could send each other the exclusive-or of all of their input bits. While it may not clear why this would be useful, it's equally unclear how to prove that it *can't* be useful. The remarkable *Raz-McKenzie simulation theorem* asserts that there are no communication protocols for the two-party problem that improve over the straightforward simulation of a query algorithm.

Theorem 2.7 (Raz-McKenzie Simulation Theorem [120, 66])**.** *If every deterministic query algorithm for f requires at least q queries in the worst case, then every deterministic communication protocol for the induced two-party problem has cost $\Omega(q \log N)$.*

The proof, which is not easy but also not unreadable, shows how to extract a good query algorithm from an arbitrary low-cost communication protocol (essentially by a potential function argument).

The original Raz-McKenzie theorem [120] and the streamlined version by Göös et al. [66] are both restricted to deterministic algorithms and protocols, and this is the version we'll use in these notes. Very recently, Göös et al. [67] and Anshu et al. [5] proved the analog of Theorem 2.7 for randomized query algorithms and randomized communication protocols (with two-sided error).[13] This randomized simulation

[13] Open question: prove a simulation theorem for quantum computation.

theorem simplifies the original proof of Theorem 2.1 (which pre-dated [67, 5]) to the point that it's almost the same as the argument given here for the deterministic case.[14]

The Raz-McKenzie theorem provides a generic way to generate a hard communication problem from a hard query problem. We can apply it in particular to the EoL problem, and we call the induced two-party problem 2EoL.[15]

The EoL Problem (Two-Party Version, 2EoL)

- Let $V = \{0,1\}^n$ and $N = |V| = 2^n$.
- Alice's input consists of N blocks, one for each vertex of V, and each block A_v contains M entries, each encoding a possible predecessor-successor pair for v.
- Bob's input consists of one index $y_v \in \{1, 2, \ldots, M\}$ for each vertex $v \in V$, encoding the entry of the corresponding block holding the "real" predecessor-successor pair for v.
- The goal is to identify a vertex $v \in V$ that satisfies one of the following:
  - (i) the successor in $A_v[y_v]$ is NULL;
  - (ii) the predecessor in $A_v[y_v]$ is NULL and $v \neq 0^n$;
  - (iii) $A_v[y_v]$ encodes the successor w but $A_w[y_w]$ does not encode the predecessor v; or
  - (iv) $A_v[y_v]$ encodes the predecessor u but $A_u[y_u]$ does not encode the successor v, and $v \neq 0^n$.

The next statement is an immediate consequence of Claim 2.6 and Theorem 2.7.

Corollary 2.8. *The deterministic communication complexity of the* 2EoL *problem is* $\Omega(N \log N)$*, even for instances that consist of a single Hamiltonian path.*

A matching upper bound of $O(N \log N)$ is trivial, as Bob always has the option of sending Alice his entire input.

Corollary 2.8 concludes the second step of the proof of Theorem 2.1 and furnishes a generic hard total search problem. The next order of business is to transfer this communication complexity lower bound to the more natural ϵ-BFP and ϵ-NE problems via reductions.

[14] For typechecking reasons, the argument for randomized protocols needs to work with a decision version of the EoL problem, such as "is the least significant bit of the vertex at the end of the Hamiltonian path equal to 1?"

[15] Raz and McKenzie [120] stated their result for the binary alphabet and for total functions. Göös et al. [66] note that it applies more generally to arbitrary alphabets and partial functions, which is important for its application here. For further proof details of these extensions, see Roughgarden and Weinstein [133].

Solar Lecture 3

Communication Complexity Lower Bound for Computing an Approximate Nash Equilibrium of a Bimatrix Game (Part II)

Lecturer: Tim Roughgarden *Scribe: Michal Koucký and Pavel Pudlák*

This lecture completes the proof of Theorem 2.1. As a reminder, this result states that if Alice's and Bob's private inputs are the two payoff matrices of an $N \times N$ bimatrix game, and ϵ is a sufficiently small constant, then $N^{\Omega(1)}$ communication is required to compute an ϵ-approximate Nash equilibrium (Definition 1.12), even when randomization is allowed. In terms of the proof road map in Section 2.5, it remains to complete steps 3 and 4. This corresponds to implementing Goals #1 and #2 introduced in the last lecture—reversing the direction of the classical reductions from the ϵ-BFP problem to path-following and from the ϵ-NE problem to (a two-party version of) the ϵ-BFP problem.

3.1 Step 3: 2EoL $\leq \epsilon$-2BFP

3.1.1 Preliminaries

We know from Corollary 2.8 that 2EoL, the two-party version of the End-of-the-Line problem defined in Section 2.7, has large communication complexity. This section transfers this lower bound to a two-party version of an approximate fixed point problem, by reducing the 2EoL problem to it.

We next define our two-party version of the ϵ-BFP problem, the ϵ-2BFP problem. The problem is parameterized by the dimension d and an approximation parameter ϵ. The latter should be thought of as a sufficiently small constant (independent of d).

The ϵ-BFP Problem (Informal Two-Party Version)

- Let $H = [0,1]^d$ denote the d-dimensional hypercube.
- Alice and Bob possess private inputs that, taken together, implicitly define a continuous function $f : H \to H$.
- The goal is to identify an *ϵ-approximate fixed point*, meaning a point $x \in H$ such that $\|f(x) - x\| < \epsilon$, where $\|\cdot\|$ denotes the normalized ℓ_2 norm:

$$\|a\| = \sqrt{\frac{1}{d}\sum_{i=1}^{d} a_i^2}.$$

The normalized ℓ_2 norm of a point in the hypercube (or the difference between two such points) is always between 0 and 1. If a point $x \in H$ is *not* an ϵ-approximate fixed point with respect to this norm, then $f(x)$ and x differ by a constant amount in a constant fraction of the coordinates. This version of the problem can only be easier than the more traditional version, which uses the ℓ_∞ norm.

To finish the description of the ϵ-2BFP problem, we need to explain how Alice and Bob interpret their inputs as jointly defining a continuous function.

3.1.2 Geometric Intuition

Our reduction from 2EoL to ϵ-2BFP will use no communication—Alice and Bob will simply reinterpret their 2EoL inputs as ϵ-2BFP inputs in a specific way, and a solution to the 2EoL instance will be easy to recover from any approximate fixed point.

Figure 3.1 shows the key intuition: graphs of paths and cycles naturally lead to continuous functions, where the gradient of the function "follows the line" and fixed points correspond to sources and sinks of the graph.

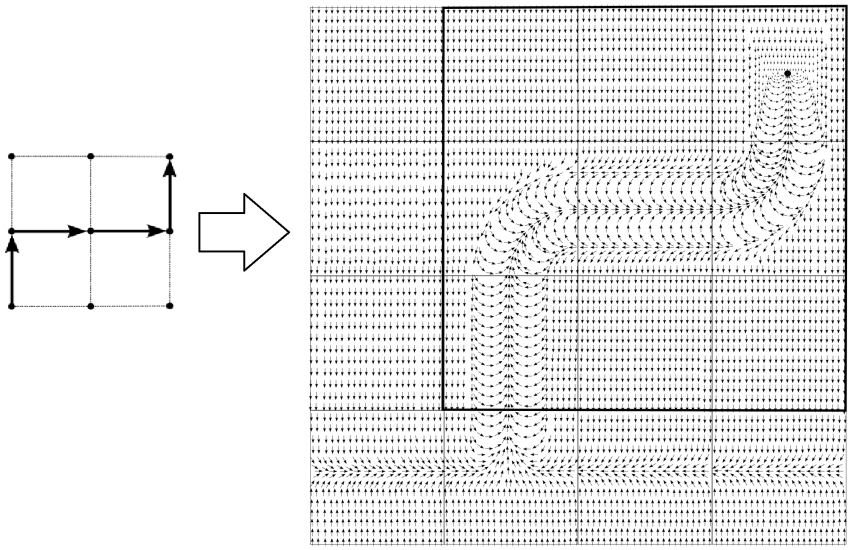

Figure 3.1: Directed paths and cycles can be converted to continuous functions that "follow the line." Points far from the path are moved by f is some canonical direction. (Figure courtesy of Yakov Babichenko.)

This idea originates in Hirsch et al. [74], who considered approximate fixed points in the ℓ_∞ norm. Rubinstein [135] showed how to modify the construction so that it works even for the normalized ℓ_2 norm. Babichenko and Rubinstein [9] used the construction from [135] in their proof of Theorem 2.1; our treatment here includes some simplifications.

3.1.3 Embedding a Graph in the Hypercube

Before explaining exactly how to interpret graphs as continuous functions, we need to set up an embedding of every possible graph on a given vertex set into the hypercube.

Let $V = \{0,1\}^n$ and $N = |V| = 2^n$. Let K denote the complete undirected graph with vertex set V—all edges that could conceivably be present in an EoL instance (ignoring their orientations). Decide once and

for all on an embedding σ of K into $H = [0,1]^d$, where $d = \Theta(n) = \Theta(\log N)$, with two properties:[1]

(P1) The images of the vertices are well separated: for every $v, w \in V$ (with $v \neq w$), $\|\sigma(v) - \sigma(w)\|$ is at least some constant (say $\frac{1}{10}$).

(P2) The images of the edges are well separated. More precisely, a point $x \in H$ is close (within distance 10^{-3}, say) to the images $\sigma(e)$ and $\sigma(e')$ of distinct edges e and e' only if x is close to the image of a shared endpoint of e and e'. (In particular, if e and e' have no endpoints in common, then no $x \in H$ is close to both $\sigma(e)$ and $\sigma(e')$.)

Property (P1) asserts that the images of two different vertices differ by a constant amount in a constant fraction of their coordinates.[2] One natural way to achieve this property is via an error-correcting code with constant rate. The simplest way to achieve both properties is to take a random straight-line embedding. Each vertex $v \in V$ is mapped to a point in $\{\frac{1}{4}, \frac{3}{4}\}^d$, with each coordinate set to $\frac{1}{4}$ or $\frac{3}{4}$ independently with 50/50 probability.[3] Each edge is mapped to a straight line between the images of its endpoints. Provided $d = cn$ for a sufficiently large constant c, properties (P1) and (P2) both hold with high probability.[4]

The point of properties (P1) and (P2) is to classify the points of H into three categories: (i) those close to the image of a (unique) vertex of K; (ii) those not close to the image of any vertex but close to the image of a (unique) edge of K; and (iii) points not close to the image of any vertex or edge of K. Accordingly, each point $x \in H$ can be "decoded" to a unique vertex v of K, a unique edge (v, w) of K, or $\perp$. Don't forget that this classification of points of H is made in advance of receiving any particular EoL input. In the ϵ-2BFP problem, since Alice and Bob both know the embedding in advance, they can decode points at will without any communication.[5]

3.1.4 Interpreting Paths as Continuous Functions

Given the embedding above, we can now describe how to interpret a directed graph $G = (V, E)$ induced by an instance of EoL as a continuous function on the hypercube, with approximate fixed points of the function corresponding only to sources and sinks of G. Write a function $f : H \to H$ as $f(x) = x + g(x)$ for the "displacement function" $g : H \to [-1, 1]^d$. (The final construction will take care to define g so that $x + g(x) \in H$ for every $x \in H$.) An ϵ-approximate fixed point is a point x with $\|g(x)\| < \epsilon$, so it's crucial for our reduction that our definition of g satisfies $\|g(x)\| \geq \epsilon$ whenever x is not close to the image of a source or sink of G.

Consider for simplicity a directed graph $G = (V, E)$ of an EoL instance that has no 2-cycles and no

[1] By an *embedding*, we mean a function σ that maps each edge (v, w) of K to a continuous path in H with endpoints $\sigma(v)$ and $\sigma(w)$.

[2] In the original construction of Hirsch et al. [74], vertices of K could potentially be mapped to points of H that differ significantly in only one coordinate. This construction is good enough to prevent spurious approximate fixed points in the ℓ_∞ norm, but not in the normalized ℓ_2 norm.

[3] For reasons related to the omitted technical details,, it's convenient to have a "buffer zone" between the embedding of the graph and the boundary of the hypercube.

[4] In the two-party communication model, we need not be concerned about efficiently constructing such an embedding. Since Alice and Bob have unbounded computational power, they can both compute the lexicographically first such embedding in advance of the protocol. When we consider computational lower bounds in Solar Lecture 5, we'll need an efficient construction.

[5] As suggested by Figure 3.1, in the final construction it's important to use a more nuanced classification that "interpolates" between points in the three different categories. It will still be the case that Alice and Bob can classify any point of $x \in H$ appropriately without any communication.

isolated vertices.[6] A rough description of the displacement function $g(x)$ induced by G is as follows, where $\delta > 0$ is a parameter (cf., Figure 3.1):

1. If x is close to $\sigma(e)$, where the endpoints of e are u and v, then

$$g(x) = \delta \cdot \begin{cases} \text{direction of the line segment } \sigma(e), \text{ oriented from } u \text{ to } v & \text{if directed edge } (u,v) \in E \\ \text{direction of the line segment } \sigma(e), \text{ oriented from } v \text{ to } u & \text{if directed edge } (v,u) \in E \\ \text{some default direction} & \text{otherwise.} \end{cases}$$

2. For x close to $\sigma(v)$, for $v \in G$,

   (a) if v has an incoming edge (u, v) and an outgoing edge (v, w), then define $g(x)$ by interpolating the displacement vectors corresponding to the edges (u, v) and (v, w) (i.e., "turn slowly" as in Figure 3.1);

   (b) otherwise (i.e., v is a source or sink of G), set $g(x)$ to interpolate the direction of the incoming or outgoing edge (whichever one exists) and the all-0 vector.

3. For x that are not close to any $\sigma(v)$ or $\sigma(e)$, set $g(x)$ to δ times the default direction.

For points x "in between" the three cases (e.g., almost but not quite close enough to the image of an edge), $g(x)$ is defined by interpolation (e.g., a weighted average of the direction of the edge and the default direction, with weights determined by just how close x is to the image of the edge).

The default direction can be implemented by doubling the number of dimensions to $2d$, and defining the displacement direction as the vector $(0, 0, \ldots, 0, 1, 1, \ldots, 1)$. Special handling (not detailed here) is then required at points x with value close to 1 in one of these extra coordinates, to ensure that $x + g(x)$ remains in H while also not introducing any unwanted approximate fixed points. Similarly, special handling is required for the source vertex 0^n, to prevent $\sigma(0^n)$ from being a fixed point. Roughly, this can be implemented by mapping the vertex 0^n to one corner of the hypercube and defining g to point in the opposite direction. The parameter δ is a constant, bigger than ϵ by a constant factor. (For example, one can assume that $\epsilon \leq 10^{-12}$ and take $\delta \approx 10^{-6}$.) This ensures that whenever the normalized ℓ_2 norm of a direction vector y is at least a sufficiently large constant, $\delta \cdot y$ has norm larger than ϵ. This completes our sketch of how to interpret an instance of EoL as a continuous function on the hypercube.

3.1.5 Properties of the Construction

Properly implemented, the construction in Sections 3.1.3 and 3.1.4 has the following properties:

1. Provided ϵ is at most a sufficiently small constant, a point $x \in H$ satisfies $\|g(x)\| < \epsilon$ only if it is close to the image of a source or sink of G different from the canonical source 0^n. (Intuitively, this should be true by construction.)

2. There is a constant λ, independent of d, such that the function $f(x) = x + g(x)$ is λ-Lipschitz. In particular, f is continuous. (Intuitively, this is because we take care to linearly interpolate between regions of H with different displacement vectors.)

[6] Recall from Corollary 2.8 that the 2EoL problem is already hard in the special case where the induced graph G is guaranteed to be a Hamiltonian path.

Sections 3.1.3 and 3.1.4, together with Figure 3.1, provide a plausibility argument that a construction with these two properties is possible along the proposed lines. Readers interested in further details are encouraged to start with the carefully written two-dimensional construction in [74, Section 4]—where many of these ideas originate—before proceeding to the general case in [74, Section 5] for the ℓ_∞ norm and finally Babichenko and Rubinstein [9] for the version tailored to the normalized ℓ_2 norm (which is needed here).

3.1.6 The ϵ-2BFP Problem and Its Communication Complexity

We can now formally define the two-party version of the ϵ-BFP problem that we consider, denoted ϵ-2BFP. The problem is parameterized by a positive integer n and a constant $\epsilon > 0$.

The ϵ-2BFP Problem

- Alice and Bob begin with private inputs to the 2EoL problem: Alice with $N = 2^n$ "blocks" $A_1, \ldots, A_N$, each with $M = \text{poly}(N)$ entries from the alphabet $\Sigma = \{0,1\}^n \times \{0,1\}^n$, and Bob with N indices $y_1, \ldots, y_N \in [M]$.
- Let G be the graph induced by these inputs (with $V = \{0,1\}^n$ and $A_v[y_v]$ encoding $(pred(v), succ(v))$.
- Let f denote the continuous function $f : H \to H$ induced by G, as per the construction in Sections 3.1.3 and 3.1.4, where $H = [0,1]^d$ is the d-dimensional hypercube with $d = \Theta(n)$.
- The goal is to compute a point $x \in H$ such that $\|f(x) - x\| < \epsilon$, where $\|\cdot\|$ denotes the normalized ℓ_2 norm.

The two properties in Section 3.1.5 imply a communication complexity lower bound for the ϵ-2BFP problem and implement step 3 of the road map in Section 2.5.

Theorem 3.1 ([9]). *For every sufficiently small constant $\epsilon > 0$, the deterministic communication complexity of the ϵ-2BFP problem is $\Omega(N \log N)$, even for $O(1)$-Lipschitz functions.*

Proof. If there is a deterministic communication protocol with cost c for the ϵ-2BFP problem, then there is also one for the 2EoL problem: Alice and Bob interpret their 2EoL inputs as inputs to the ϵ-2BFP problem, run the assumed protocol to compute an ϵ-approximate fixed point x, and (using no communication) decode x to a source or sink vertex v of G (that is different from 0^n). The theorem follows immediately from Corollary 2.8. □

3.1.7 Local Decodability of ϵ-2BFP Functions

There is one more important property of the functions f constructed in Sections 3.1.3 and 3.1.4: they are *locally decodable* in a certain sense. Suppose Alice and Bob want to compute the value of $f(x)$ at some commonly known point $x \in H$. If x decodes to $\perp$ (i.e., is not close to the image of any vertex or edge of the complete graph K on vertex set V), then Alice and Bob know the value of $f(x)$ without any communication whatsoever: $f(x)$ is x plus δ times the default direction (or a known customized displacement if x is too close to certain boundaries of H). If x decodes to the edge $e = (u, v)$ of the complete graph K, then Alice and Bob can compute $f(x)$ as soon as they know whether or not edge e belongs to the directed graph G induced by their inputs, along with its orientation. This requires Alice and Bob to exchange predecessor-successor information about only two vertices (u and v). Analogously, if x decodes to the vertex v of K, then Alice and Bob can compute $f(x)$ after exchanging information about at most three vertices (v, $pred(v)$, and $succ(v)$).

3.2 Step 4: ϵ-2BFP $\leq$ ϵ-NE

This section completes the proof of Theorem 2.1 by reducing the ϵ-2BFP problem to the ϵ-NE problem, where ϵ is a sufficiently small constant.

3.2.1 The McLennan-Tourky Analytic Reduction

The starting point for our reduction is a purely analytic reduction of McLennan and Tourky [104], which reduces the existence of (exact) Brouwer fixed points to the existence of (exact) Nash equilibria.[7] Subsequent sections explain the additional ideas needed to implement this reduction for approximate fixed points and Nash equilibria in the two-party communication model.

Theorem 3.2 (McLennan and Tourky [104])**.** *Nash's theorem (Theorem 1.14) implies Brouwer's fixed-point theorem (Theorem 2.2).*

Proof. Consider an arbitrary continuous function $f : H \to H$, where $H = [0,1]^d$ is the d-dimensional hypercube (for some positive integer d).[8] Define a two-player game as follows. The pure strategies of Alice and Bob both correspond to points of H. For pure strategies $x, z \in H$, Alice's payoff is defined as

$$1 - \|x - z\|^2 = 1 - \frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2 \tag{3.1}$$

and Bob's payoff as

$$1 - \|z - f(x)\|^2 = 1 - \frac{1}{d}\sum_{i=1}^{d}(z_i - f(x)_i)^2. \tag{3.2}$$

Alice wants to imitate Bob's strategy, while Bob wants to imitate the image of Alice's strategy under the function f.

For any mixed strategy σ of Bob (i.e., a distribution over points of the hypercube), Alice's unique best response is the corresponding center of gravity $\mathbf{E}_{z\sim\sigma}[z]$ (as you should check). Thus in any Nash equilibrium, Alice plays a pure strategy x. Bob's unique best response to such a pure strategy is the pure strategy $z = f(x)$. That is, every Nash equilibrium is pure, with $x = z = f(x)$ a fixed point of f. Since a Nash equilibrium exists, so does a fixed point of f.[9] □

An extension of the argument above shows that, for λ-Lipschitz functions f, an ϵ'-approximate fixed point (in the normalized ℓ_2 norm) can be extracted easily from any ϵ-approximate Nash equilibrium, where ϵ' is a function of ϵ and λ only.[10]

[7] This reduction was popularized in a Leisure of the Theory Class blog post by Eran Shmaya (https://theoryclass.wordpress.com/2012/01/05/brouwer-implies-nash-implies-brouwer/), who heard about the result from Rida Laraki.

[8] If fixed points are guaranteed for hypercubes in every dimension, then they are also guaranteed for all compact convex subsets of finite-dimensional Euclidean space; see footnote 10 in Solar Lecture 2.

[9] Strictly speaking, we're assuming a more general form of Nash's theorem that asserts the existence of a pure Nash equilibrium whenever every player has a convex compact strategy set (like H) and a continuous concave payoff function (like (3.1) and (3.2)). (The version in Theorem 1.14 corresponds to the special case where each strategy set corresponds to a finite-dimensional simplex of mixed strategies, and where all payoff functions are linear.) Most proofs of Nash's theorem—including the one outlined in Section 2.3.2—are straightforward to generalize in this way.

[10] It is not clear how to easily extract an approximate fixed point in the ℓ_∞ norm from an approximate Nash equilibrium without losing a super-constant factor in the parameters. The culprit is the "$\frac{1}{d}$" factor in (3.1) and (3.2)—needed to ensure that payoffs are bounded—which allows each player to behave in an arbitrarily crazy way in a few coordinates without violating the ϵ-approximate Nash equilibrium conditions. (Recall $\epsilon > 0$ is constant while $d \to \infty$.) This is one of the primary reasons why Rubinstein [135] and Babichenko and Rubinstein [9] needed to modify the construction in Hirsch et al. [74] to obtain their results.

3.2.2 The Two-Party Reduction: A Naive Attempt

We now discuss how to translate the McLennan-Tourky analytic reduction to an analogous reduction in the two-party model. First, we need to discretize the hypercube. Define H_ϵ as the set of $\approx \left(\frac{1}{\epsilon}\right)^n$ points of $[0,1]^d$ for which all coordinates are multiples of ϵ. Every $O(1)$-Lipschitz function f is guaranteed to have an $O(\epsilon)$-approximate fixed point at some point of the discretized hypercube (by rounding an exact fixed point to its nearest neighbor in H_ϵ). This also means that the corresponding game (with payoffs defined as in (3.1) and (3.2)) has an $O(\epsilon)$-approximate Nash equilibrium in which each player deterministically chooses a point of H_ϵ.

The obvious attempt at a two-party version of the McLennan-Tourky reduction is:

1. Alice and Bob start with inputs to the ϵ-2BFP problem.
2. The players interpret these inputs as a two-player game, with strategies corresponding to points of the discretized hypercube H_ϵ, and with Alice's payoffs given by (3.1) and Bob's payoffs by (3.2).
3. The players run the assumed low-cost communication protocol for computing an approximate Nash equilibrium.
4. The players extract an approximate fixed point of the ϵ-2BFP function from the approximate Nash equilibrium.

Just one problem: *this doesn't make sense*. The issue is that Bob needs to be able to compute $f(x)$ to evaluate his payoff function in (3.2), and his ϵ-2BFP input (just a bunch of indices into Alice's blocks) does not provide sufficient information to do this. Thus, the proposed reduction does not produce a well-defined input to the ϵ-NE problem.

3.2.3 Description of the Two-Party Reduction

The consolation prize is that Bob can compute the function f at a point x after a brief conversation with Alice. Recall from Section 3.1.7 that computing f at a point $x \in H$ requires information about at most three vertices of the 2EoL input that underlies the ϵ-2BFP input (in addition to x). Alice can send x to Bob, who can then send the relevant indices to Alice (after decoding x to some vertex or edge of K), and Alice can respond with the corresponding predecessor-successor pairs. This requires $O(\log N)$ bits of communication, where $N = 2^n$ is the number of vertices in the underlying 2EoL instance. (We are suppressing the dependence on the constant ϵ in the big-O notation.) Denote this communication protocol by P.

At this point, it's convenient to restrict the problem to the hard instances of 2EoL used to prove Corollary 2.8, where in particular, $succ(v) = w$ if and only if $v = pred(w)$. (I.e., cases (iii) and (iv) in the definition of the 2EoL problem in Section 2.7 never come up.) For this special case, P can be implemented as a two-round protocol where Alice and Bob exchange information about one relevant vertex v (if x decodes to v) or two relevant vertices u and v (if x decodes to the edge (u, v)).[11]

How can we exploit the local decodability of ϵ-2BFP functions? The idea is to enlarge the strategy sets of Alice and Bob, beyond the discretized hypercube H_ϵ, so that the players' strategies at equilibrium

[11] If x decodes to the edge (u, v), then Alice and Bob exchange information about u and v in two rounds. If x decodes to the vertex v, they exchange information about v in two rounds. This reveals v's opinion of its predecessor u and successor w. In the general case, Alice and Bob would still need to exchange information about u and w using two more rounds of communication to confirm that $succ(u) = pred(w) = v$. (Recall our semantics: directed edge (u, v) belongs to G if and only if both $succ(u) = v$ and $pred(v) = u$.) In the special case of instances where $succ(v) = w$ if and only if $v = pred(w)$, these two extra rounds of communication are redundant.

effectively simulate the protocol P. Alice's pure strategies are the pairs (x, α), where $x \in H_\epsilon$ is a point of the discretized hypercube and α is a possible transcript of Alice's communication in the protocol P. Thus α consists of at most two predecessor-successor pairs. Bob's pure strategies are the pairs (z, β), where $z \in H_\epsilon$ and β is a transcript that could be generated by Bob in P—a specification of at most two different vertices and his corresponding indices for them.[12] Crucially, because the protocol P has cost $O(\log N)$, there are only $N^{O(1)}$ possible α's and β's. There are also only $N^{O(1)}$ possible choices of x and z—since ϵ is a constant and $d = \Theta(n)$ in the ϵ-2BFP problem, $|H_\epsilon| \approx \left(\frac{1}{\epsilon}\right)^d$ is polynomial in $N = 2^n$. We conclude that the size of the resulting game is polynomial in the length of the given ϵ-2BFP (or 2EoL) inputs.

We still need to define the payoffs of the game. Let $A_1, \ldots, A_N$ and $y_1, \ldots, y_N$ denote Alice's and Bob's private inputs in the given ϵ-2BFP (equivalently, 2EoL) instance and f the corresponding function. Call an outcome (x, α, z, β) *consistent* if α and β are the transcripts generated by Alice and Bob when they honestly follow the protocol P to compute $f(x)$. Precisely, a consistent outcome is one that meets the following two conditions:

(i) for each of the (zero, one, or two) vertices v named in β, and corresponding index $\hat{y}_v$ announced by Bob in β, α contains the correct response $A_v[\hat{y}_v]$;

(ii) β specifies the names of the vertices relevant for Alice's announced point $x \in H_\epsilon$, and for each such vertex v, β specifies the correct index y_v.

Observe that Alice can privately check if condition (i) holds (using her private input $A_1, \ldots, A_N$ and the vertex names and indices in Bob's announced strategy β), and Bob can privately check condition (ii) (using his private input $y_1, \ldots, y_N$ and the point x announced by Alice).

For an outcome (x, z, α, β), we define Alice's payoffs by

$$\begin{cases} -1 - \frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2 & \text{if (i) fails} \\ 1 - \frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2 & \text{otherwise.} \end{cases} \tag{3.3}$$

(Compare (3.3) with (3.1).) This definition makes sense because Alice can privately check whether or not (i) holds and hence can privately compute her payoff.[13]

For Bob's payoffs, we need a preliminary definition. Let $f_\alpha(x)$ denote the value that the induced function f would take on if α was consistent with x and with Alice's and Bob's private inputs. That is, to compute $f_\alpha(x)$:

1. Decode x to a vertex or an edge (or $\perp$).

2. Interpret α as the predecessor-successor pairs for the vertices relevant for evaluating f at x.

3. Output x plus the displacement $g_\alpha(x)$ defined as in Sections 3.1.3 and 3.1.4 (with α supplying any predecessor-successor pairs that are necessary).

To review, f is the ϵ-2BFP function that Alice and Bob want to find a fixed point of, and $f(x)$ generally depends on the private inputs $A_1, \ldots, A_N$ and $y_1, \ldots, y_N$ of both Alice and Bob. The function f_α is a speculative version of f, predicated on Alice's announced predecessor-successor pairs in her strategy α. Crucially, the definition of f_α does not depend at all on Alice's private input, only on Alice's *announced*

[12] In the protocol P, Bob does not need to communicate the names of any vertices—Alice can decode x privately. But it's convenient for the reduction to include the names of the vertices relevant for x in the β part of Bob's strategy.

[13] If you want to be a stickler and insist on payoffs in $[0, 1]$, then shift and scale the payoffs in (3.3) appropriately.

strategy. Thus given α, Bob can privately execute the three steps above and evaluate $f_\alpha(x)$ for any $x \in H_\epsilon$. The other crucial property of f_α is that, if α happens to be the actual predecessor-successor pairs $\{A_v[y_v]\}$ for the vertices relevant for x (given Alice's and Bob's private inputs), then $f_\alpha(x)$ agrees with the value $f(x)$ of the true ϵ-2BFP function.

We can now define Bob's payoffs as follows (compare with (3.2)):

$$\begin{cases} -1 & \text{if (ii) fails} \\ 1 - \frac{1}{d}\sum_{i=1}^{d}(z_i - f_\alpha(x)_i)^2 & \text{otherwise.} \end{cases} \tag{3.4}$$

Because Bob can privately check condition (ii) and compute $f_\alpha(x)$ (given x and α), Bob can privately compute his payoff. This completes the description of the reduction from the ϵ-2BFP problem to the ϵ-NE problem.

Alice and Bob can carry out this reduction with no communication—by construction, their ϵ-2BFP inputs fully determine their payoff matrices. As noted earlier, because ϵ is a constant, the sizes of the produced ϵ-NE inputs are polynomial in those of the ϵ-2BFP inputs.

3.2.4 Analysis of the Two-Party Reduction

Finally, we need to show that the reduction "works," meaning that Alice and Bob can recover an approximate fixed point of the ϵ-2BFP function f from any approximate Nash equilibrium of the game produced by the reduction.

For intuition, let's think first about the case where Alice's and Bob's strategies are points of the hypercube H (rather than the discretized hypercube H_ϵ) and the case of exact fixed points and Nash equilibria. (Cf., Theorem 3.2.) What could a Nash equilibrium of the game look like? Consider mixed strategies by Alice and Bob.

1. Alice's payoff in (3.3) includes a term $-\frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2$ that is independent of her choice of α or Bob's choice of β, and the other term (either 1 or -1) is independent of her choice of x (since condition (i) depends only on α and β). Thus, analogous to the proof of Theorem 3.2, in every one of Alice's best responses, she deterministically chooses $x = \mathbf{E}_{z\sim\sigma}[z]$, where σ denotes the marginal distribution of z in Bob's mixed strategy.

2. Given that Alice is playing deterministically in her x-coordinate, in every one of Bob's best responses, he deterministically chooses β to name the vertices relevant for Alice's announced point x and his indices for these vertices (to land in the second case of (3.4) with probability 1).

3. Given that Bob is playing deterministically in his β-coordinate, Alice's unique best response is to choose x as before and also deterministically choose the (unique) message α that satisfies condition (i), so that she will be in the more favorable second case of (3.3) with probability 1.

4. Given that Alice is playing deterministically in both her x- and α-coordinates, Bob's unique best response is to choose β as before and set $z = f_\alpha(x)$ (to maximize his payoff in the second case of (3.4)).

These four steps imply that every (exact) Nash equilibrium (x, z, α, β) of the game is pure, with α and β consistent with x and Alice's and Bob's private information about the corresponding relevant vertices, and with $x = z = f_\alpha(x) = f(x)$ a fixed point of f.

As with Theorem 3.2, a more technical version of the same argument implies that an approximate fixed point—a point x satisfying $\|f(x) - x\| < \epsilon'$ with respect to the normalized ℓ_2 norm—can be easily extracted

by Alice and Bob from any ϵ-approximate Nash equilibrium, where ϵ' depends only on ϵ (e.g., $\epsilon' = O(\epsilon^{1/4})$ suffices). For example, the first step of the proof becomes: in an ϵ-approximate Nash equilibrium, Alice must choose a point $x \in H_\epsilon$ that is close to $\mathbf{E}[z]$ except with small probability (otherwise she could increase her expected payoff by more than ϵ by switching to the point of H_ϵ closest to $\mathbf{E}[z]$). And so on. Carrying out approximate versions of all four steps above, while keeping careful track of the epsilons, completes the proof of Theorem 2.1.

We conclude that computing an approximate Nash equilibrium of a general bimatrix game takes a polynomial amount of communication, and in particular there are no uncoupled dynamics guaranteed to converge to in a polylogarithmic number of iterations.

Solar Lecture 4

TFNP, PPAD *& All That*

Lecturer: Tim Roughgarden *Scribe: Amey Bhangale*

Having resolved the communication complexity of computing an approximate Nash equilibrium of a bimatrix game, we turn our attention to the *computational* complexity of the problem. Here, the goal will be to prove a super-polynomial lower bound on the amount of computation required, under appropriate complexity assumptions. The techniques developed in the last two lectures for our communication complexity lower bound will again prove useful for this goal, but we will also need several additional ideas.

This lecture identifies the appropriate complexity class for characterizing the computational complexity of computing an exact or approximate Nash equilibrium of a bimatrix game, namely PPAD. Solar Lecture 5 sketches some of the ideas in Rubinstein's recent proof [135] of a quasi-polynomial-time lower bound for the problem, assuming an analog of the Exponential Time Hypothesis for PPAD.

Section 4.1 explains why customized complexity classes are needed to reason about equilibrium computation and other total search problems. Section 4.2 defines the class TFNP and some of its syntactic subclasses, including PPAD.[1] Section 4.3 reviews a number of PPAD-complete problems. Section 4.4 discusses the existing evidence that TFNP and its important subclasses are hard, and proves that the class TFNP is hard on average assuming that NP is hard on average.

4.1 Preamble

We consider two-player (bimatrix) games, where each player has (at most) n strategies. The $n \times n$ payoff matrices for Alice and Bob A and B are described explicitly, with A_{ij} and B_{ij} indicating Alice's and Bob's payoffs when Alice plays her ith strategy and Bob his jth strategy. Recall from Definition 1.12 that an ϵ-NE is a pair $\hat{x}, \hat{y}$ of mixed strategies such that neither player can increase their payoff with a unilateral deviation by more than ϵ.

What do we know about the complexity of computing an ϵ-NE of a bimatrix game? Let's start with the exact case ($\epsilon = 0$), where no subexponential-time (let alone polynomial-time) algorithm is known for the problem. (This contrasts with the zero-sum case, see Corollary 1.5.) It is tempting to speculate that no such algorithm exists. How would we amass evidence that the problem is intractable? Since we're interested in super-polynomial lower bounds, communication complexity is of no direct help.

Could the problem be NP-complete?[2] The following theorem by Megiddo and Papadimitriou rules out this possibility (unless NP = co-NP).

[1] Some of the discussion in these two sections is drawn from [129, Lecture 20].

[2] Technically, we're referring to the *search* version of NP (sometimes called FNP, where the "F" stands for "functional"), where the goal is to either exhibit a witness or correctly deduce that no witness exists.

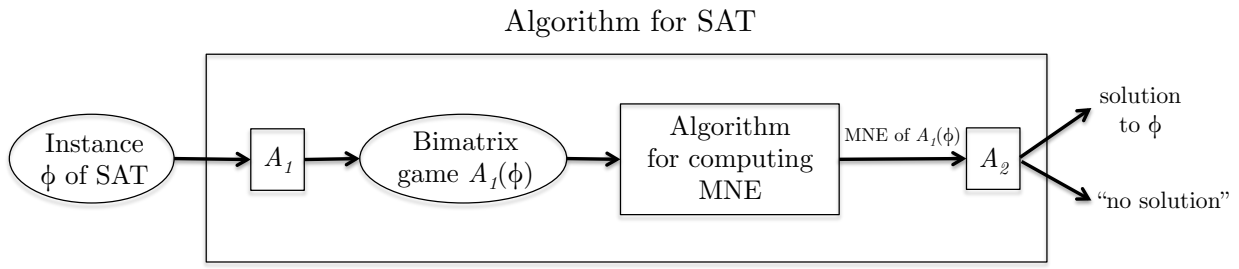

Figure 4.1: A reduction from the search version of the SAT problem to the problem of computing a Nash equilibrium of a bimatrix game would yield a polynomial-time verifier for the unsatisfiability problem.

Theorem 4.1 ([105]). *The problem of computing a Nash equilibrium of a bimatrix game is* NP*-hard only if* NP = co-NP.

Proof. The proof is short but a bit of a mind-bender, analogous to the argument back in Section 2.2. Suppose there is a reduction from, say, (the search version of) satisfiability to the problem of computing a Nash equilibrium of a bimatrix game. By definition, the reduction comprises two algorithms:

1. A polynomial-time algorithm $\mathcal{A}_1$ that maps every SAT formula ϕ to a bimatrix game $\mathcal{A}_1(\phi)$.

2. A polynomial-time algorithm $\mathcal{A}_2$ that maps every Nash equilibrium $(\hat{x}, \hat{y})$ of a game $\mathcal{A}_1(\phi)$ to a satisfying assignment $\mathcal{A}_2(\hat{x}, \hat{y})$ of ϕ, if one exists, and to the string "no" otherwise.

We claim that the existence of these algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ imply that NP = co-NP (see also Figure 4.1). In proof, consider an unsatisfiable SAT formula ϕ, and an arbitrary Nash equilibrium $(\hat{x}, \hat{y})$ of the game $\mathcal{A}_1(\phi)$.[3] We claim that $(\hat{x}, \hat{y})$ is a short, efficiently verifiable proof of the unsatisfiability of ϕ, implying that NP = co-NP. Given an alleged certificate $(\hat{x}, \hat{y})$ that ϕ is unsatisfiable, the verifier performs two checks: (1) compute the game $\mathcal{A}_1(\phi)$ using algorithm $\mathcal{A}_1$ and verify that $(\hat{x}, \hat{y})$ is a Nash equilibrium of $\mathcal{A}_1(\phi)$; (2) use the algorithm $\mathcal{A}_2$ to verify that $\mathcal{A}_2(\hat{x}, \hat{y})$ is the string "no." This verifier runs in time polynomial in the description lengths of ϕ and $(\hat{x}, \hat{y})$. If $(\hat{x}, \hat{y})$ passes both of these tests, then correctness of the algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ implies that ϕ is unsatisfiable. □

4.2 TFNP and Its Subclasses

4.2.1 TFNP

What's really going on in the proof of Theorem 4.1 is a mismatch between the search version of an NP-complete problem like SAT, where an instance may or may not have a witness, and a problem like computing a Nash equilibrium, where every instance has at least one witness. While the correct answer to a SAT instance might well be "no," a correct answer to an instance of Nash equilibrium computation is always a Nash equilibrium. It seems that if the problem of computing a Nash equilibrium is going to be complete for some complexity class, it must be a class smaller than NP.

[3] Crucially, $\mathcal{A}_1(\phi)$ has at least one Nash equilibrium (Theorem 1.14), including one whose description length is polynomial in that of the game (see the discussion following Theorem 1.14)).

The subset of NP (search) problems for which every instance has at least one witness is called TFNP, for "total functional NP." The proof of Theorem 4.1 shows more generally that if *any* TFNP problem is NP-complete, then NP = co-NP. Thus a fundamental barrier to NP-completeness is the guaranteed existence of a witness.

Since computing a Nash equilibrium does not seem to be NP-complete, the sensible refined goal is to prove that the problem is TFNP-complete—as hard as any other NP problem with a guaranteed witness.

4.2.2 Syntactic vs. Semantic Complexity Classes

Unfortunately, TFNP-completeness is also too ambitious a goal. The reason is that TFNP does not seem to have complete problems. Think about the complexity classes that *are* known to have complete problems—NP of course, and also classes like P and PSPACE. What do these complexity classes have in common? They are "syntactic," meaning that membership can be characterized via acceptance by some concrete computational model, such as polynomial-time or polynomial-space deterministic or nondeterministic Turing machines. In this sense, there is a generic reason for membership in these complexity classes.

Syntactically defined complexity classes always have a "generic" complete problem, where the input is a description of a problem in terms of the accepting machine and an instance of the problem, and the goal is to solve the given instance of the given problem. For example, the generic NP-complete problem takes as input a description of a verifier, a polynomial time bound, and an encoding of an instance, and the goal is to decide whether or not there is a witness, meaning a string that causes the given verifier to accept the given instance in at most the given number of steps.

TFNP has no obvious generic reason for membership, and as such is called a "semantic" class.[4] For example, the problem of computing a Nash equilibrium of a bimatrix game belongs to TFNP because of the topological arguments that guarantee the existence of a Nash equilibrium (see Section 2.3). Another problem in TFNP is factoring: given a positive integer, output its factorization. Here, membership in TFNP has a number-theoretic explanation.[5] Can the guaranteed existence of a Nash equilibrium of a game and of a factorization of an integer be regarded as separate instantiations of some "generic" TFNP argument? No one knows the answer.

4.2.3 Syntactic Subclasses of TFNP

Given that the problem of computing a Nash equilibrium appears too specific to be complete for TFNP, we must refine our goal again, and try to prove that the problem is complete for a still smaller complexity class. Papadimitriou [117] initiated the search for syntactic subclasses of TFNP that contain interesting problems not known to belong to P. His proposal was to categorize TFNP problems according to the type of mathematical proof used to prove the guaranteed existence of a witness. Interesting subclasses include the following:

- PPAD (for polynomial parity argument, directed version): Problems that can be solved by path-following in a (exponential-size) directed graph with in- and out-degree at most 1 and a known source vertex (specifically, the problem of identifying a sink or source vertex other than the given one).

- PPA (for polynomial parity argument, undirected version): Problems that can be solved by path-following in an undirected graph (specifically, given an odd-degree vertex, the problem of identifying

[4]There are many other interesting examples of classes that appear to be semantic in this sense, such as RP and NP ∩ co-NP.

[5]There are many more natural examples of TFNP problems, including computing a local minimum of a function, computing an approximate Brouwer fixed point, and inverting a one-way permutation.

a different odd-degree vertex).

- PLS (for polynomial local search): Problems that can be solved by path-following in a directed acyclic graph (specifically, given such a graph, the problem of identifying a sink vertex).[6]
- PPP (for polynomial pigeonhole principle): Problems that reduce to the following: given a function f mapping $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, n-1\}$, find $i \neq j$ such that $f(i) = f(j)$.

All of these complexity classes can be viewed as intermediate to P and NP. The conjecture, supported by oracle separations [10], is that all four of these classes are distinct (Figure 4.2).

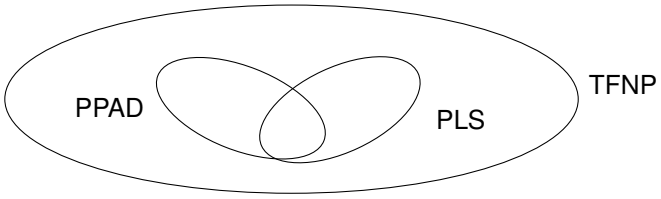

Figure 4.2: Conjectured relationships between subclasses of TFNP.

Section 2.3 outlined the argument that the guaranteed existence of Nash equilibria reduces to the guaranteed existence of Brouwer fixed points, and Section 2.4 showed (via Sperner's Lemma) that Brouwer's fixed-point theorem reduces to path-following in a directed graph with in- and out-degrees at most 1. Thus, PPAD would seem to be the subclass of TFNP with the best chance of capturing the complexity of computing a Nash equilibrium.

4.3 PPAD and Its Complete Problems

4.3.1 EoL: The Generic Problem for PPAD

We can formally define the class PPAD by defining its generic problem. (A problem is then in PPAD if it reduces to the generic problem.) Just as the End-of-the-Line (EoL) problem served as the starting point of our communication complexity lower bound (see Section 2.4), a succinct version of the problem will be the basis for our computational hardness results.

The EoL Problem (Succinct Version)

Given two circuits S and P (for "successor" and "predecessor"), each mapping $\{0,1\}^n$ to $\{0,1\}^n \cup \{NULL\}$ and with size polynomial in n, and with $P(0^n) = NULL$, find an input $v \in \{0,1\}^n$ that satisfies one of the following:

(i) $S(v)$ is NULL;

(ii) $P(v)$ is NULL and $v \neq 0^n$;

(iii) $v \neq P(S(v))$; or

[6] PLS was actually defined prior to TFNP, by Johnson et al. [81].

(iv) $v \neq S(P(v))$ and $v \neq 0^n$.

Analogous to Section 2.4, we can view the circuits S and P as defining a graph G with in- and out-degree at most 1 (with edge (u, v) in G if and only if $S(u) = v$ and $P(v) = u$), and with a given source vertex 0^n. The EoL problem then corresponds to identifying either a sink vertex of G or a source vertex other than 0^n.[7] A solution is guaranteed to exist—if nothing else, the other end of the path of G that originates with the vertex 0^n. Thus EoL does indeed belong to TFNP, and PPAD $\subseteq$ TFNP. Note also that the class is syntactic and by definition has a complete problem, namely the EoL problem.

4.3.2 Problems in PPAD

The class PPAD contains several natural problems (in addition to the EoL problem). For example, it contains a computational version of Sperner's Lemma—given a succinct description (e.g., polynomial-size circuits) of a legal coloring of an exponentially large triangulation of a simplex, find a sub-simplex such that its vertices showcase all possible colors. This problem can be regarded as a special case of the EoL problem (see Section 2.4), and hence belongs to PPAD.

Another example is the problem of computing an approximate fixed point. Here the input is a succinct description of a λ-Lipschitz function f (on the hypercube in d dimensions, say) and a parameter ϵ, and the goal is to compute a point x with $\|f(x) - x\| < \epsilon$ (with respect to some norm). The description length of x should be polynomial in that of the function f. Such a point is guaranteed to exist provided ϵ is not too small.[8] The reduction from Brouwer's fixed-point theorem to Sperner's Lemma (with colors corresponding to directions of movement, see Section 2.3) shows that this problem can also be regarded as a special case of the EoL problem, and hence belongs to PPAD.

The problem of computing an exact or approximate Nash equilibrium of a bimatrix game also belongs to PPAD. For the problem of computing an ϵ-approximate Nash equilibrium (with ϵ no smaller than inverse exponential in n), this follows from the proof of Nash's theorem outlined in Section 2.3.2. That proof shows that computing an ϵ-NE is a special case of computing an approximate fixed point (of the regularized best-response function defined in (2.1) and (2.2)), and hence the problem belongs to PPAD. The same argument shows that this is true more generally for any number of players (and not just for bimatrix games).

The problem of computing an exact Nash equilibrium ($\epsilon = 0$) also belongs to PPAD in the case of two-player (bimatrix) games.[9] One way to prove this is via the Lemke-Howson algorithm [96] (see also Section 1.4), which reduces the computation of an (exact) Nash equilibrium of a bimatrix game to a path-following problem, much in the way that the simplex algorithm reduces computing an optimal solution of a linear program to following a path of improving edges along the boundary of the feasible region. All known proofs of the Lemke-Howson algorithm's inevitable convergence use parity arguments akin to the one in the proof of Sperner's lemma. These convergence proofs show that the problem of computing a Nash equilibrium of a bimatrix game belongs to PPAD.

[7] The undirected version of the problem can be used to define the class PPA. The version of the problem where only sink vertices count as witnesses seems to give rise to a different (larger) complexity class called PPADS.

[8] For example, for the ℓ_∞ norm, ϵ can be as small as $\frac{(\lambda+1)}{2^n}$, where n is the description length of f. This follows from rounding each coordinate of an exact fixed point to its nearest multiple of 2^{-n}.

[9] Etessami and Yannakakis [49] proved that, with 3 or more players, the problem of computing an exact Nash equilibrium of a game appears to be strictly harder than any problem in PPAD.

4.3.3 PPAD-Complete Fixed-Point Problems

The EoL problem is PPAD-complete by construction. What about "more natural" problems? Papadimitriou [117] built evidence that PPAD is a fundamental complexity class by showing that fixed-point problems are complete for it.

To be precise, let $\textsc{Brouwer}(\|\cdot\|, d, \mathcal{F}, \epsilon)$ denote the following problem: given a (succinct description of a) function $f \in \mathcal{F}$, with $f : [0,1]^d \to [0,1]^d$, compute a point $x \in [0,1]^d$ such that $\|f(x) - x\| < \epsilon$. The original hardness result from [117] is the following.

Theorem 4.2 (Papadimitriou [117]). *The* $\textsc{Brouwer}(\|\cdot\|, d, \mathcal{F}, \epsilon)$ *problem is* PPAD*-complete, even when* $d = 3$*, the functions in* $\mathcal{F}$ *are* $O(1)$*-Lipschitz,* $\|\cdot\|$ *is the* ℓ_∞ *norm, and* ϵ *is exponentially small in the description length* n *of a function* $f \in \mathcal{F}$*.*

The high-level idea of the proof is similar to the construction in Section 3.1 that shows how to interpret EoL instances as implicitly defined Lipschitz functions on the hypercube. Given descriptions of the circuits S and P in an instance of the generic EoL problem, it is possible to define an (efficiently computable) function that "follows the line" of an embedding of the induced directed graph into the hypercube. Three dimensions are needed in the construction in [117] to ensure that the images of different edges do not intersect (except at a shared endpoint). Some time later, Chen and Deng [31] used a somewhat different approach to prove that Theorem 4.2 holds even when $d = 2$.[10]

Much more recently, with an eye toward hardness results for ϵ-approximate Nash equilibria with constant ϵ (see Solar Lecture 5), Rubinstein [135] proved the following.[11]

Theorem 4.3 (Rubinstein [135]). *The* $\textsc{Brouwer}(\|\cdot\|, d, \mathcal{F}, \epsilon)$ *problem is* PPAD*-complete even when the functions in* $\mathcal{F}$ *are* $O(1)$*-Lipschitz functions,* d *is linear in the description length* n *of a function in* $\mathcal{F}$*,* $\|\cdot\|$ *is the normalized* ℓ_2 *norm (with* $\|x\| = \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}$*), and* ϵ *is a sufficiently small constant.*

The proof of Theorem 4.3 is closely related to the third step of our communication complexity lower bound (Section 3.1), and in particular makes use of a similar embedding of graphs into the hypercube with the properties (P1) and (P2) described in Sections 3.1.3 and 3.1.4.[12] One major difference is that our proof of existence of the embedding in Section 3.1 used the probabilistic method and hence is not constructive (which is not an issue in the two-party communication model), while the computational lower bound in Theorem 4.3 requires a constructive version. In particular, the reduction from EoL to $\textsc{Brouwer}(\|\cdot\|, d, \mathcal{F}, \epsilon)$ must efficiently produce a succinct description of the function f induced by an instance of EoL, and it should be possible to efficiently evaluate f, presumably while using the given EoL circuits S and P only as black boxes. For example, it should be possible to efficiently decode points of the hypercube (to a vertex, edge, or $\perp$, see Section 3.1.3).

Conceptually, the fixes for these problems are relatively simple. First, rather than mapping the vertices randomly into the hypercube, the reduction in the proof of Theorem 4.3 embeds the vertices using an error-correcting code (with constant rate and efficient encoding and decoding algorithms). This enforces property (P1) of Section 3.1.3. Second, rather than using a straight-line embedding, the reduction is more proactive about making the images of different edges stay far apart (except for at shared endpoints).

[10] The one-dimensional case can be solved in polynomial time, essentially by binary search.

[11] Theorem 4.2 proves hardness in the regime where d and ϵ are both small, Theorem 4.3 when both are large. This is not an accident; if d is small (i.e., constant) and ϵ is large (i.e., constant), the problem can be solved in polynomial time by exhaustively checking a constant number of evenly spaced grid points.

[12] We have reversed the chronology; Theorem 2.1 was proved after Theorem 4.3 and used the construction in [135] more or less as a black box.

Specifically, an edge of the directed graph induced by the given EoL instance is now mapped to 5 straight line segments, and along each line segment, two-thirds of the coordinates stay fixed. (This requires blowing up the number of dimensions by a constant factor.) For example, the directed edge (u, v) can be mapped to the path

$$(\sigma(u), \sigma(u), \mathbf{\frac{1}{4}}) \mapsto (\sigma(u), \sigma(v), \mathbf{\frac{1}{4}}) \mapsto (\sigma(u), \sigma(v), \mathbf{\frac{3}{4}}) \mapsto (\sigma(v), \sigma(v), \mathbf{\frac{3}{4}}) \mapsto (\sigma(v), \sigma(v), \mathbf{\frac{1}{4}}),$$

where σ denotes the error-correcting code used to map the vertices to the hypercube and the boldface $\mathbf{\frac{1}{4}}$ and $\mathbf{\frac{3}{4}}$ indicate the value of the last third of the coordinates. This maneuver enforces property (P2) of Section 3.1.3. It also ensures that it is easy to decode points of the hypercube that are close to the image of an edge of the graph—at least one of the edge's endpoints can be recovered from the values of the frozen coordinates, and the other endpoint can be recovered using the given predecessor and successor circuits.[13]

4.3.4 PPAD-Complete Equilibrium Computation Problems

Papadimitriou [117] defined the class PPAD in large part to capture the complexity of computing a Nash equilibrium, conjecturing that the problem is in fact PPAD-complete. Over a decade later, a flurry of papers confirmed this conjecture. First, Daskalakis, Goldberg, and Papadimitriou [43, 63] proved that computing an ϵ-NE of a four-player game, with ϵ inverse exponential in the size of the game, is PPAD-complete. This approach was quickly refined [29, 42], culminating in the proof of Chen and Deng [30] that computing a Nash equilibrium (or even an ϵ-NE with exponentially small ϵ) of a bimatrix game is PPAD-complete. Thus the nice properties possessed by Nash equilibria of bimatrix games (see Section 1.4) are not enough to elude computational intractability. Chen et al. [32] strengthened this result to hold even for values of ϵ that are only inverse polynomial in the size of the game.[14] The papers by Daskalakis et al. [44] and Chen et al. [34] give a full account of this breakthrough sequence of results.

Theorem 4.4 (Daskalakis et al. [44], Chen et al. [34])**.** *The problem of computing an* ϵ-NE *of an* $n \times n$ *bimatrix game is* PPAD*-complete, even when* $\epsilon = 1/\text{poly}(n)$.

The proof of Theorem 4.4, which is a tour de force, is also outlined in the surveys by Johnson [80], Papadimitriou [118], Daskalakis et al. [45], and Roughgarden [125]. Fundamentally, the proof shows how to use a bimatrix game to perform a gate-by-gate simulation of the circuits of an EoL instance.

Theorem 4.4 left open the possibility that, for every constant $\epsilon > 0$, an ϵ-NE of a bimatrix game can be computed in polynomial time. (Recall from Corollary 1.17 that it can be computed in *quasi-polynomial* time.) A decade later, Rubinstein [135] ruled out this possibility (under suitable complexity assumptions) by proving a quasi-polynomial-time hardness result for the problem when ϵ is a sufficiently small constant. We will have much more to say about this result in Solar Lecture 5.

4.4 Evidence of Hardness

4.4.1 Basing the Hardness of TFNP on Cryptographic Assumptions

It's all fine and good to prove that a problem is as hard as any other problem in PPAD, but what makes us so sure that PPAD problems (or even TFNP problems) are hard? The initial evidence was exponential

[13] This embedding is defined only for the directed edges that are present in the given EoL instance, rather than for all possible edges (in contrast to the embedding in Sections 3.1.3 and 3.1.4).

[14] In particular, under standard complexity assumptions, this rules out an algorithm with smoothed polynomial complexity in the sense of Spielman and Teng [141]. Thus the parallels between the simplex method and the Lemke-Howson algorithm (see Section 1.4) only go so far.

lower bounds for functions given as "block boxes," or equivalently query complexity lower bounds, as in Proposition 2.6 for the EoL problem or Hirsch et al. [74] for the Brouwer problem.

Can we relate the hardness of TFNP and its subclasses to other standard complexity assumptions? Theorem 4.1 implies that we can't base hardness of TFNP on the assumption that $\mathsf{P} \neq \mathsf{NP}$, unless $\mathsf{NP} = \mathsf{co\text{-}NP}$. What about cryptographic assumptions? After all, the problem of inverting a one-way permutation belongs to TFNP (and even the subclass PPP). Thus, sufficiently strong cryptographic assumptions imply hardness of TFNP.

Can we prove hardness also for all of the interesting subclasses of TFNP, or can we establish the hardness of TFNP under weaker assumptions (like the existence of one-way functions)? Along the former lines, a recent sequence of papers (not discussed here) show that strong assumptions about indistinguishability obfuscation (i.o.) imply that PPAD is hard [13, 59, 123, 75]. The rest of this lecture covers a recent result in the second direction by Hubáček et al. [76], who show that the average-case hardness of TFNP can be based on the average-case hardness of NP. (Even though the worst-case hardness of TFNP *cannot* be based on that of NP, unless $\mathsf{NP} = \mathsf{co\text{-}NP}$!) Note that assuming that NP is hard on average is only weaker than assuming the existence of one-way functions.

Theorem 4.5 ([76])**.** *If there exists a hard-on-average language in* NP*, then there exists a hard-on-average search problem in* TFNP*.*

There is some fine print in the precise statement of the result (see Remarks 4.7 and 4.8), but the statement in Theorem 4.5 is the gist of it.

4.4.2 Proof Sketch of Theorem 4.5

Let L be a language in NP which is hard on average w.r.t. some family of distributions D_n on input strings of length n. Average-case hardness of (L, D_n) means that there is no polynomial-time algorithm with an advantage of $1/\mathrm{poly}(n)$ over random guessing when the input is sampled according to D_n (for any polynomial). Each D_n should be efficiently sampleable, so that hardness cannot be baked into the input distribution.

One natural candidate for a hard-on-average problem in NP is that of inverting a one-way function on a random range element. Since a one-way function is generally not surjective, this problem is not total and hence does not belong to TFNP. Can we convert it into a problem that *is* total and yet retains its average-case hardness?

Here's an initial attempt:

Attempt #1

Input: l independent samples $x_1, x_2, \ldots, x_l$ from D_n.
Output: a witness for some $x_i \in L$.

As $l \to \infty$, this problem is "almost total." Because (L, D_n) is hard-on-average, random instances are nearly equally likely to be "yes" or "no" instances (otherwise a constant response would beat random guessing). Thus, except with probability $\approx 2^{-l}$, at least one of the sampled instances x_i is a "yes" instance and has a witness. Taking l polynomial in n, we get a problem that is total except with exponentially small probability. How can we make it "totally total?"

The idea is to sample the x_i's in a correlated way, using a random shifting trick reminiscent of Lautemann's proof that $\mathsf{BPP} \subseteq \Sigma_2 \cap \Pi_2$ [94]. This will give a non-uniform version of Theorem 4.5; Remark 4.8 sketches the changes necessary to get a uniform version.

Fix n. Let $D_n(r)$ denote the output of the sampling algorithm for D_n, given the random seed $r \in \{0,1\}^n$. (By padding, we can assume that the input length and the random seed length both equal n.) Call a set containing the strings $s_1, s_2, \ldots, s_l \in \{0,1\}^n$ *good* if for every seed $r \in \{0,1\}^n$ there exists an index $i \in [l]$ such that $D(r \oplus s_i) \in L$. We can think of the s_i's as masks; goodness then means that there is always a mask whose application yields a "yes" instance.

Claim 4.6. *If $s_1, s_2, \ldots, s_{2n} \sim \{0,1\}^n$ are sampled uniformly and independently, then $\{s_1, \ldots, s_{2n}\}$ is good with high probability.*

Proof. Fix a seed $r \in \{0,1\}^n$. The distribution of $r \oplus s_i$ (over s_i) is uniform, so $D_n(r \oplus s_i)$ has a roughly 50% chance of being a "yes" instance (since (L, D_n) is hard on average). Thus the probability (over $s_1, \ldots, s_{2n}$) that $D_n(r \oplus s_i)$ is a "no" instance for *every* s_i is $\approx 2^{-2n}$. Taking a union bound over the 2^n choices for r completes the proof. □

Consider now the following reduction, from the assumed hard-on-average NP problem (L, D_n) to a hopefully hard-on-average TFNP problem.

Attempt 2 (non-uniform)

Chosen in advance: A good set of strings $\{s_1, s_2, \ldots, s_{2n}\}$.
Input: an instance x of (L, D_n), in the form of the random seed $\hat{r}$ used to generate $x = D_n(\hat{r})$.
Output: a witness for one of the instances $D(\hat{r} \oplus s_1), \ldots, D(\hat{r} \oplus s_{2n})$.

By the definition of a good set of strings, there is always at least one witness of the desired form, and so the output of this reduction is a TFNP problem (or more accurately, a TFNP/poly problem, with $s_1, \ldots, s_{2n}$ given as advice). Let D' denote the distribution over instances of this problem induced by the uniform distribution over $\hat{r}$. It remains to show how a (non-uniform) algorithm that solves this TFNP/poly problem (with respect to D') can be used to beat random guessing (with inverse polynomial advantage) for (L, D_n) in a comparable amount of time. Given an algorithm A for the former problem (and the corresponding good set of strings), consider the following algorithm B for (L, D_n).

Algorithm $B_{s_1, s_2, \ldots, s_{2n}}$

Input: A random instance x of (L, D_n) and the random seed $\hat{r}$ that generated it (so $x = D_n(\hat{r})$).

1. Choose $i \in [2n]$ uniformly at random.
2. Set $r^\star = \hat{r} \oplus s_i$.
3. Use the algorithm A to generate a witness w for one of the instances
$$D(r^\star \oplus s_1), D(r^\star \oplus s_2), \ldots, D(r^\star \oplus s_{2n}).$$
(Note that the ith problem is precisely the one we want to solve.)
4. If w is a witness for $D(r^\star \oplus s_i)$, then output "yes."
5. Otherwise, randomly answer "yes" or "no" (with 50/50 probability).

Consider a "yes" instance $D_n(\hat{r})$ of L. If algorithm A happens to output a witness to the ith instance $D_n(r^\star \oplus s_i) = D_n(\hat{r})$, then algorithm B correctly decides the problem. The worry is that the algorithm A somehow conspires to always output a witness for an instance other than the "real" one.

Suppose algorithm A, when presented with the instances $D(r^\star \oplus s_1), D(r^\star \oplus s_2), \ldots, D(r^\star \oplus s_{2n})$, exhibits a witness for the jth instance $D(r^\star \oplus s_j)$. This collection of instances could have been produced by the reduction in exactly $2n$ different ways: with $i = 1$ and $\hat{r} = r^\star \oplus s_1$, with $i = 2$ and $\hat{r} = r^\star \oplus s_2$, and so on. Since i and $\hat{r}$ were chosen independently and uniformly at random, each of these $2n$ outcomes is equally likely, and algorithm A has no way of distinguishing between them. Thus whatever j is, A's witness has at least a $1/2n$ chance of being a witness for the true problem $D_n(\hat{r})$ (where the probability is over both $\hat{r}$ and i). We conclude that, for "yes" instances of L, algorithm B has advantage $\frac{1}{2n}$ over random guessing. Since roughly 50% of the instances $D_n(\hat{r})$ are "yes" instances (since (L, D_n) is average-case hard), algorithm B has advantage roughly $\frac{1}{4n}$ over random guessing for (L, D_n). This contradicts our assumption that (L, D_n) is hard on average.

We have completed the proof of Theorem 4.5, modulo two caveats.

Remark 4.7 (Public vs. Private Coins)**.** The algorithm B used in the reduction above beats random guessing for (L, D_n), provided the algorithm receives as input the random seed $\hat{r}$ used to generate an instance of (L, D_n). That is, our current proof of Theorem 4.5 assumes that (L, D_n) is hard on average *even with public coins*. While there are problems in NP conjectured to be average-case hard in this sense (like randomized SAT near the phase transition), it would be preferable to have a version of Theorem 4.5 that allows for private coins. Happily, Hubáček et al. [76] prove that there exists a private-coin average-case hard problem in NP only if there is also a public-coin such problem. This implies that Theorem 4.5 holds also in the private-coin case.

Remark 4.8 (Uniform vs. Non-Uniform)**.** Our proof of Theorem 4.5 only proves hardness for the non-uniform class TFNP/poly. (The good set $\{s_1, \ldots, s_{2n}\}$ of strings is given as "advice" separately for each n.) It is possible to extend the argument to (uniform) TFNP, under some additional (reasonably standard) complexity assumptions. The idea is to use techniques from derandomization. We already know from Claim 4.6 that almost all sets of $2n$ strings from $\{0,1\}^n$ are good. Also, the problem of checking whether or not a set of strings is good is a Π_2 problem (for all $r \in \{0,1\}^n$ there exists $i \in [2n]$ such that $D_n(r \oplus s_i)$ has a witness). Assuming that there is a problem in E with exponential-size Π_2 circuit complexity, it is possible to derandomize the probabilistic argument and efficiently compute a good set $\{s_1, \ldots, s_l\}$ of strings (with l larger than $2n$ but still polynomial in n), à la Impagliazzo and Wigderson [77].

An important open research direction is to extend Theorem 4.5 to subclasses of TFNP, such as PPAD.

Open Problem: Does an analogous average-case hardness result hold for PPAD?

Solar Lecture 5

The Computational Complexity of Computing an Approximate Nash Equilibrium

Lecturer: Tim Roughgarden *Scribe: Salil Vadhan*

5.1 Introduction

Last lecture we stated without proof the result by Daskalakis et al. [44] and Chen et al. [34] that computing an ϵ-approximate Nash equilibrium of a bimatrix game is PPAD-complete, even when ϵ is an inverse polynomial function of the game size (Theorem 4.4). Thus, it would be surprising if there were a polynomial-time (or even subexponential-time) algorithm for this problem. Recall from Corollary 1.17 in Solar Lecture 1 that the story is different for constant values of ϵ, where an ϵ-approximate Nash equilibrium can be computed in quasi-polynomial (i.e., $n^{O(\log n)}$) time.

The Pavlovian response of a theoretical computer scientist to a quasi-polynomial-time algorithm is to conjecture that a polynomial-time algorithm must also exist. (There are only a few known natural problems that appear to have inherently quasi-polynomial time complexity.) But recall that the algorithm in the proof of Corollary 1.17 is just exhaustive search over all probability distributions that are uniform over a multi-set of logarithmically many strategies (which is good enough, by Theorem 1.15). Thus the algorithm reveals no structure of the problem other than the fact that the natural search space for it has quasi-polynomial size. It is easy to imagine that there are no "shortcuts" to searching this space, in which case a quasi-polynomial amount of time would indeed be necessary. How would we ever prove such a result? Presumably by a non-standard super-polynomial reduction from some PPAD-complete problem like succinct EoL (defined in Section 4.3.1). This might seem hard to come by, but in a recent breakthrough, Rubinstein [135] provided just such a reduction!

Theorem 5.1 ([135])**.** *For all sufficiently small constants $\epsilon > 0$, for every constant $\delta > 0$, there is no $n^{\log^{1-\delta} n}$-time algorithm for computing an ϵ-approximate Nash equilibrium of a bimatrix game, unless the succinct* EoL *problem has a $2^{n^{1-\delta'}}$-time algorithm for some constant $\delta' > 0$.*

In other words, assuming an analog of the Exponential Time Hypothesis (ETH) [78] for PPAD, the quasi-polynomial-time algorithm in Corollary 1.17 is essentially optimal![1,2]

Three previous papers that used an ETH assumption (for NP) along with PCP machinery to prove quasi-polynomial-time lower bounds for NP problems are:

[1]To obtain a quantitative lower bound like the conclusion of Theorem 5.1, it is necessary to make a quantitative complexity assumption (like an analog of ETH). This approach belongs to the tradition of "fine-grained" complexity theory.

[2]How plausible is the assumption that the ETH holds for PPAD, even after assuming that the ETH holds for NP and that PPAD has no polynomial-time algorithms? The answer is far from clear, although there are exponential query lower bounds for PPAD problems (e.g. [74]) and no known techniques that show promise for a subexponential-time algorithm for the succinct EoL problem.

1. Aaronson et al. [1], for the problem of computing the value of free games (i.e., two-prover proof systems with stochastically independent questions), up to additive error ϵ;

2. Braverman et al. [18], for the problem of computing the ϵ-approximate Nash equilibrium with the highest expected sum of player payoffs; and

3. Braverman et al. [19] for the problem of distinguishing graphs with a k-clique from those that only have k-vertex subgraphs with density at most $1 - \epsilon$.

In all three cases, the hardness results apply when $\epsilon > 0$ is a sufficiently small constant. Quasi-polynomial-time algorithms are known for all three problems.

The main goal of this lecture is to convey some of the ideas in the proof of Theorem 5.1. The proof is a tour de force and the paper [135] is 57 pages long, so our treatment will necessarily be impressionistic. We hope to explain the following:

1. What the reduction in Theorem 5.1 must look like. (Answer: a blow-up from size n to size $\approx 2^{\sqrt{n}}$.)

2. How a $n \mapsto\approx 2^{\sqrt{n}}$-type blowup can naturally arise in a reduction to the problem of computing an approximate Nash equilibrium.

3. Some of the tricks used in the reduction.

4. Why these tricks naturally lead to the development and application of PCP machinery.

5.2 Proof of Theorem 5.1: An Impressionistic Treatment

5.2.1 The Necessary Blow-Up

The goal is to reduce length-n instances of the succinct EoL problem to length-$f(n)$ instances of the problem of computing an ϵ-approximate Nash equilibrium with constant ϵ, so that a sub-quasi-polynomial-time algorithm for the latter implies a subexponential-time algorithm for the former. Thus the mapping $n \mapsto f(n)$ should satisfy $2^n \approx f(n)^{\log f(n)}$ and hence $f(n) \approx 2^{\sqrt{n}}$. That is, we should be looking to encode a length-n instance of succinct EoL as a $2^{\sqrt{n}} \times 2^{\sqrt{n}}$ bimatrix game. The $\sqrt{n}$ will essentially come from the "birthday paradox," with random subsets of $[n]$ of size s likely to intersect once s exceeds $\sqrt{n}$. The blow-up from n to $2^{\sqrt{n}}$ will come from PCP-like machinery, as well as a game-theoretic gadget ("Althöfer games," see Section 5.2.6) that forces players to randomize nearly uniformly over size-$\sqrt{n}$ subsets of $[n]$ in every approximate Nash equilibrium.

5.2.2 The Starting Point: ϵ-BFP

The starting point of the reduction is the PPAD-complete version of the ϵ-BFP problem in Theorem 4.3. We restate that result here.

Theorem 5.2 (Rubinstein [135]). *The* Brouwer$(\|\cdot\|, d, \mathcal{F}, \epsilon)$ *problem is* PPAD*-complete when the functions in $\mathcal{F}$ are $O(1)$-Lipschitz functions from the d-dimensional hypercube $H = [0,1]^d$ to itself, d is linear in the description length n of a function in $\mathcal{F}$, $\|\cdot\|$ is the normalized ℓ_2 norm (with $\|x\| = \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}$), and ϵ is a sufficiently small constant.*

The proof is closely related to the reduction from 2EoL to ϵ-2BFP outlined in Section 3.1, and Section 4.3.3 describes the additional ideas needed to prove Theorem 5.2. As long as the error-correcting code used to embed vertices into the hypercube (see Section 4.3.3) has linear-time encoding and decoding algorithms (as in [140], for example), the reduction can be implemented in linear time. In particular, our assumption that the succinct EoL problem has no subexponential-time algorithms automatically carries over to this version of the ϵ-BFP problem. In addition to the properties of the functions in $\mathcal{F}$ that are listed in the statement of Theorem 5.2, the proof of Theorem 5.1 crucially uses the "locally decodable" properties of these functions (see Section 5.2.8).

5.2.3 ϵ-BFP $\leq$ ϵ-NE (Attempt #1): Discretize McLennan-Tourky

One natural starting point for a reduction from ϵ-BFP to ϵ-NE is the McLennan-Tourky analytic reduction in Section 3.2.1. Given a description of an $O(1)$-Lipschitz function $f : [0,1]^d \to [0,1]^d$, with d linear in the length n of the function's description, the simplest reduction would proceed as follows. Alice and Bob each have a strategy set corresponding to the discretized hypercube H_ϵ (points of $[0,1]^d$ such that every coordinate is a multiple of ϵ). Alice's and Bob's payoffs are defined as in the proof of Theorem 3.2: for strategies $x, y \in H_\epsilon$, Alice's payoff is

$$1 - \|x - y\|^2 = 1 - \frac{1}{d}\sum_{i=1}^{d}(x_i - y_i)^2 \tag{5.1}$$

and Bob's payoff is

$$1 - \|y - f(x)\|^2 = 1 - \frac{1}{d}\sum_{j=1}^{d}(y_j - f(x)_j)^2. \tag{5.2}$$

(Here $\|\cdot\|$ denotes the normalized ℓ_2 norm.) Thus Alice wants to imitate Bob's strategy, while Bob wants to imitate the image of Alice's strategy under the function f.

This reduction is correct in that in every ϵ-approximate Nash equilibrium of this game, Alice's and Bob's strategies are concentrated around an $O(\epsilon)$-approximate fixed point of the given function f (in the normalized ℓ_2 norm). See also the discussion in Section 3.2.1.

The issue is that the reduction is not efficient enough. Alice and Bob each have $\Theta((1/\epsilon)^d)$ pure strategies; since $d = \Theta(n)$, this is exponential in the size n of the given ϵ-BFP instance, rather than exponential in $\sqrt{n}$. This exponential blow-up in size means that this reduction has no implications for the problem of computing an approximate Nash equilibria.

5.2.4 Separable Functions

How can we achieve a blow-up exponential in $\sqrt{n}$ rather than in n? We might guess that the birthday paradox is somehow involved. To build up our intuition, we'll discuss at length a trivial special case of the ϵ-BFP problem. It turns out that the hard functions used in Theorem 5.2 are in some sense surprisingly close to this trivial case.

For now, we consider only instances f of ϵ-BFP where f is *separable*. That is, f has the form

$$f(x_1, \ldots, x_d) = (f_1(x_1), \ldots, f_d(x_d)) \tag{5.3}$$

for efficiently computable functions $f_1, \ldots, f_d : [0,1] \to [0,1]$. Separable functions enjoy the ultimate form of "local decodability"—to compute the ith coordinate of $f(x)$, you only need to know the ith coordinate

of x. Finding a fixed point of a separable function is easy: the problem decomposes into d one-dimensional fixed point problems (one per coordinate), and each of these can be solved efficiently by a form of binary search. The hard functions used in Theorem 5.2 possess a less extreme form of "local decodability," in that each coordinate of $f(x)$ can be computed using only a small amount of "advice" about f and x (cf., the ϵ-2BFP $\leq \epsilon$-NE reduction in Section 3.2.3).

5.2.5 ϵ-BFP $\leq \epsilon$-NE (Attempt #2): Coordinatewise Play

Let's try to at least compute fixed points of separable functions with approximate Nash equilibria using a reduction with only subexponential blow-up. The key idea is, instead of Alice and Bob each picking one of the (exponentially many) points of the discretized hypercube H_ϵ, each will pick *only a single coordinate* of points x and y. Thus a pure strategy of Alice comprises an index $i \in [d]$ and a number $x_i \in [0,1]$ that is a multiple of ϵ, and similarly Bob chooses $j \in [d]$ and $y_j \in [0,1]$. Given choices (i, x_i) and (j, y_j), Alice's payoff is defined as

$$\begin{cases} 1 - (x_i - y_i)^2 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

and Bob's payoff is

$$\begin{cases} 1 - (y_i - f_i(x_i))^2 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases}$$

Thus Alice and Bob receive payoff 0 unless they "interact," meaning choose the same coordinate to play in, in which case their payoffs are analogous to (5.1) and (5.2). Note that Bob's payoff is well defined only because we have assumed that f is separable (Bob only knows the coordinate x_i proposed by Alice, but this is enough to compute the ith coordinate of the output of f and hence his payoff). Each player has only $\approx \frac{d}{\epsilon}$ strategies, so this is a polynomial-time reduction, with no blow-up.

The good news is that (approximate) fixed points give rise to (approximate) Nash equilibria of this game. Specifically, if $\hat{x} = \hat{y} = f(\hat{x})$ is a fixed point of f, then the following is a Nash equilibrium (as you should check): Alice and Bob pick their coordinates i, j uniformly at random and set $x_i = \hat{x}_i$ and $y_j = \hat{y}_j$. The problem is that the game also has equilibria other than the intended ones, for example where Alice and Bob choose pure strategies with $i = j$ and $x_i = y_i = f_i(x_i)$.

5.2.6 ϵ-BFP $\leq \epsilon$-NE (Attempt #3): Gluing Althöfer Games

Our second attempt failed because Alice and Bob were not forced to randomize their play over all d coordinates. We can address this issue with a game-theoretic gadget called an *Althöfer game* [4].[3] For a positive and even integer k, this $k \times \binom{k}{k/2}$ game is defined as follows.

- Alice chooses an index $i \in [k]$.
- Bob chooses a subset $S \subseteq [k]$ of size $k/2$.
- Alice's payoff is 1 if $i \in S$, and -1 otherwise.
- Bob's payoff is -1 if $i \in S$, and 1 otherwise.

[3] Similar ideas have been used previously, including in the proofs that computing an ϵ-approximate Nash equilibrium with ϵ inverse polynomial in n is a PPAD-complete problem [44, 34].

For example, here is the payoff matrix for the $k = 4$ case (with only Alice's payoffs shown):

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

Every Althöfer game is a zero-sum game with value 0: for both players, choosing a uniformly random strategy guarantees expected payoff 0. The following claim proves a robust converse for Alice's play. Intuitively, if Alice deviates much from the uniform distribution, Bob is well-positioned to punish her.[4]

Claim 5.3. *In every ϵ-approximate Nash equilibrium of an Althöfer game, Alice's strategy is ϵ-close to uniformly random in statistical distance (a.k.a. total variation distance).*

Proof. Suppose that Alice plays strategy $i \in [k]$ with probability p_i. After sorting the coordinates so that $p_{i_1} \leq p_{i_2} \leq \cdots \leq p_{i_k}$, Bob's best response is to play the subset $S = \{i_1, i_2, \ldots, i_{k/2}\}$. We must have either $p_{i_{k/2}} \leq 1/k$ or $p_{i_{k/2+1}} \geq 1/k$ (or both). Suppose that $p_{i_{k/2}} \leq 1/k$ (the other case can be handled symmetrically). Then Bob's expected payoff from playing S is:

$$\begin{aligned} \sum_{j>k/2} p_{i_j} - \sum_{j\leq k/2} p_{i_j} &= \sum_{j>k/2} (p_{i_j} - 1/k) + \sum_{j\leq k/2} (1/k - p_{i_j}) \\ &= \sum_{j:p_{i_j}>1/k} (p_{i_j} - 1/k) + \sum_{j>k/2:p_{i_j}\leq 1/k} (p_{i_j} - 1/k) + \sum_{j\leq k/2} (1/k - p_{i_j}) \\ &\geq \sum_{j:p_{i_j}>1/k} (p_{i_j} - 1/k), \end{aligned}$$

where the last inequality holds because the p_{i_j}'s are sorted in increasing order and $p_{i_{k/2}} \leq 1/k$. The final expression above equals the statistical distance between Alice's mixed strategy $\vec{p}$ and the uniform distribution. The claim now follows from that fact that Bob cannot achieve a payoff larger than ϵ in any ϵ-approximate Nash equilibrium (otherwise, Alice could increase her expected payoff by more than ϵ by switching to the uniform distribution). □

In Claim 5.3, it's important that the loss in statistical distance (as a function of ϵ) is independent of the size k of the game. For example, straightforward generalizations of rock-paper-scissors do not achieve the guarantee in Claim 5.3.

Gluing Games. We incorporate Althöfer games into our coordinatewise play game as follows. Let

- G_1 = the $\frac{d}{\epsilon} \times \frac{d}{\epsilon}$ coordinatewise game of Section 5.2.5,
- G_2 = a $d \times \binom{d}{d/2}$ Althöfer game, and
- G_3 = a $\binom{d}{d/2} \times d$ Althöfer game, with the roles of Alice and Bob reversed.

Consider the following game, where Alice and Bob effectively play all three games simultaneously:

- A pure strategy of Alice comprises an index $i \in [d]$, a multiple x_i of ϵ in $[0, 1]$, and a set $T \subseteq [d]$ of size $d/2$. The interpretation is that she plays (i, x_i) in G_1, i in G_2, and T in G_3.

[4]The statement and proof here include a constant-factor improvement, due to Salil Vadhan, over those in [135].

- A pure strategy of Bob comprises an index $j \in [d]$, a multiple y_j of ϵ in $[0, 1]$, and a set $S \subseteq [d]$ of size $d/2$, interpreted as playing (j, y_j) in G_1, S in G_2 and j in G_3.
- Each player's payoff is a weighted sum of their payoffs in the three games, $\frac{1}{100} \cdot G_1 + \frac{99}{200} \cdot G_2 + \frac{99}{200} \cdot G_3$.

The good news is that, in every exact Nash equilibrium of the combined game, Alice and Bob mix uniformly over their choices of i and j. Intuitively, because deviating from the uniform strategy can be punished by the other player at a rate linear in the deviation (Claim 5.3), it is never worth doing (no matter what happens in G_1). Given this, à la the McLennan-Tourky reduction (Theorem 3.2), the x_i's and y_j's must correspond to a fixed point of f (for each i, Alice must set x_i to the center of mass of Bob's distribution over y_i's, and then Bob must set $y_i = f_i(x_i)$).

The bad news is that this argument breaks down for ϵ-approximate Nash equilibria with constant ϵ. The reason is that, even when the distributions of i and j are perfectly uniform, the two players interact (i.e., choose $i = j$) only with probability $1/d$. This means that the contribution of the game G_1 to the expected payoffs is at most $1/d \ll \epsilon$, freeing the players to choose their x_i's and y_j's arbitrarily. Thus we need another idea to force Alice and Bob to interact more frequently.

A second problem is that the sizes of the Althöfer games are too big—exponential in d rather than in $\sqrt{d}$.

5.2.7 ϵ-BFP $\leq \epsilon$-NE (Attempt #4): Blockwise Play

To solve both of the problems with the third attempt, we force Alice and Bob to play larger sets of coordinates at a time. Specifically, we view $[d]$ as a $\sqrt{d} \times \sqrt{d}$ grid, and any $x, y \in [0, 1]^d$ as $\sqrt{d} \times \sqrt{d}$ matrices. Now Alice and Bob will play a row and column of their matrices, respectively, and their payoffs will be determined by the entry where the row and column intersect. That is, we replace the coordinatewise game of Section 5.2.5 with the following *blockwise game*:

- A pure strategy of Alice comprises an index $i \in \left[\sqrt{d}\right]$ and a row $x_{i*} \in [0, 1]^{\sqrt{d}}$. (As usual, every x_{ij} should be a multiple of ϵ.)
- A pure strategy of Bob comprises an index $j \in \left[\sqrt{d}\right]$ and a column $y_{*j} \in [0, 1]^{\sqrt{d}}$.
- Alice's payoff in the outcome (x_{i*}, y_{*j}) is

$$1 - (x_{ij} - y_{ij})^2.$$

- Bob's payoff in the outcome (x_{i*}, y_{*j}) is

$$1 - (y_{ij} - f_{ij}(x_{ij}))^2. \tag{5.4}$$

Now glue this game together with $k \times \binom{k}{k/2}$ and $\binom{k}{k/2} \times k$ Althöfer games with $k = \sqrt{d}$, as in Section 5.2.6. (For example, Alice's index $i \in \left[\sqrt{d}\right]$ is identified with a row in the first Althöfer game, and now Alice also picks a subset $S \subseteq \left[\sqrt{d}\right]$ in the second Althöfer game, in addition to i and x_{i*}.) This construction yields exactly what we want: a game of size $\exp(\tilde{O}(k)) = \exp(\tilde{O}(\sqrt{d}))$ in which every ϵ-approximate Nash equilibrium can be easily translated to a δ-approximate fixed point of f (in the normalized ℓ_2 norm), where δ depends only on ϵ.[5,6]

[5]The $\tilde{O}(\cdot)$ notation suppresses logarithmic factors.

[6]In more detail, in every ϵ-approximate Nash equilibrium of the game, Alice and Bob both randomize nearly uniformly over i and j; this is enforced by the Althöfer games as in Section 5.2.6. Now think of each player as choosing its strategy in two stages,

5.2.8 Beyond Separable Functions

We now know how to use an ϵ-approximate Nash equilibrium of a subexponential-size game (with constant ϵ) to compute a δ-approximate fixed point of a function that is separable in the sense of (5.3). This is not immediately interesting, because a fixed point of a separable function is easy to find by doing binary search independently in each coordinate. The hard Brouwer functions identified in Theorem 5.2 have lots of nice properties, but they certainly aren't separable.

Conceptually, the rest of the proof of Theorem 5.1 involves pushing in two directions: first, identifying hard Brouwer functions that are even "closer to separable" than the functions in Theorem 5.2; and second, extending the reduction in Section 5.2.7 to accommodate "close-to-separable" functions. We already have an intuitive feel for what the second step looks like, from Step 4 of our communication complexity lower bound (Section 3.2.3 in Lunar Lecture 3), where we enlarged the strategy sets of the players so that they could smuggle "advice" about how to decode a hard Brouwer function f at a given point. We conclude the lecture with one key idea for the further simplification of the hard Brouwer functions in Theorem 5.2.

5.2.9 Local EoL

Recall the hard Brouwer functions constructed in our communication complexity lower bound (see Section 3.1), which "follow the line" of an embedding of an EoL instance, as well as the additional tweaks needed to prove Theorem 5.2 (see Section 4.3.3). We are interested in the "local decodability" properties of these functions. That is, if Bob needs to compute the jth coordinate of $f(x)$ (to evaluate the jth term in his payoff in (5.2)), how much does he need to know about x? For a separable function $f = (f_1, \ldots, f_d)$, he only needs to know x_j. For the hard Brouwer functions in Theorem 5.2, Bob needs to know whether or not x is close to an edge (of the embedding of the succinct EoL instance into the hypercube) and, if so, which edge (or pair of edges, if x is close to a vertex). Ultimately, this requires evaluating the successor circuit S and predecessor circuit P of the succinct EoL instance that defines the hard Brouwer function. It is therefore in our interest to force S and P to be as simple as possible, subject to the succinct EoL problem remaining PPAD-complete. In a perfect world, minimal advice (say, $O(1)$ bits) would be enough to compute $S(v)$ and $P(v)$ from v.[7] The following lemma implements this idea. It shows that a variant of the succinct EoL problem, called Local EoL, remains PPAD-complete even when S and P are guaranteed to change only $O(1)$ bits of the input, and when S and P are NC^0 circuits (and hence each output bit depends on only $O(1)$ input bits).

Lemma 5.4 (Rubinstein [135]). *The following* Local EoL *problem is* PPAD*-complete:*

1. *the vertex set V is a subset of $\{0,1\}^n$, with membership in V specified by a given AC^0 circuit;*

first the index i or j and then the corresponding values x_{i*} or y_{*j} in the row or column. Whenever Alice plays i, her best response (conditioned on i) is to play $\mathbf{E}\left[y_{ij}\right]$ in every column j, where the expectation is over the distribution of y_{ij} conditioned on Bob choosing index j. In an ϵ-approximate Nash equilibrium, in most coordinates, Alice must usually choose x_{ij}'s that are close to this best response. Given this, for most indices $j \in \left[\sqrt{d}\right]$, whenever Bob chooses j, he must usually choose a value of y_{ij} that is close to $\mathbf{E}\left[x_{ij}\right]$ (for each i). It can be shown that this implies that Alice's strategy corresponds to a δ-approximate fixed point (in the normalized ℓ_2 norm), where δ is a function of ϵ only.

[7] It is also important that minimal advice suffices to translate between points x of the hypercube and vertices v of the underlying succinct EoL instance (as f is defined on the former, while S and P operate on the latter). This can be achieved by using a state-of-the-art locally decodable error-correcting code (with query complexity $d^{o(1)}$, similar to that in Kopparty et al. [91]) to embed the vertices into the hypercube (as described in Section 4.3.3). Incorporating the advice that corresponds to local decoding into the game produced by the reduction results in a further blow-up of $2^{d^{o(1)}}$. This is effectively absorbed by the $2^{\sqrt{d}}$ blow-up that is already present in the reduction in Section 5.2.7.

2. *the successor and predecessor circuits S, P are computable in* NC^0*;*

3. *for every vertex $v \in V$, $S(v)$ and $P(v)$ differ from v in $O(1)$ coordinates.*

The proof idea is to start from the original circuits S and P of a succinct EoL instance and form circuits S' and P' that operate on partial computation transcripts, carrying out the computations performed by the circuits S or P one gate/line at a time (with $O(1)$ bits changing in each step of the computation). The vertex set V then corresponds to the set of valid partial computation transcripts. The full proof is not overly difficult; see [135, Section 5] for the details. This reduction from succinct EoL to Local EoL can be implemented in linear time, so our assumption that the former problem has no subexponential-time algorithms carries over to the latter problem.

In the standard succinct EoL problem, every n-bit string $v \in \{0,1\}^n$ is a legitimate vertex. In the Local EoL problem, only elements of $\{0,1\}^n$ that satisfy the given AC^0 circuit are legitimate vertices. In our reduction, we need to produce a game that also incorporates checking membership in V, also with only a $d^{o(1)}$ blow-up in how much of x we need to access. This is the reason why Rubinstein [135] needs to develop customized PCP machinery in his proof of Theorem 5.1. These PCP proofs can then be incorporated into the blockwise play game (Section 5.2.7), analogous to how we incorporated the interactive protocol P into the game in our reduction from 2EoL to ϵ-NE in Section 3.2.3.

Part II

Lunar Lectures

Lunar Lecture 1

How Computer Science Has Influenced Real-World Auction Design.
Case Study: *The* 2016–2017 *FCC Incentive Auction*

Lecturer: Tim Roughgarden *Scribe: Dana Randall*

1.1 Preamble

Computer science is changing the way auctions are designed and implemented. For over 20 years, the US and other countries have used *spectrum auctions* to sell licenses for wireless spectrum to the highest bidder. What's different this decade, and what necessitated a new auction design, is that in the US the juiciest parts of the spectrum for next-generation wireless applications are already accounted for, owned by over-the-air television broadcasters. This led Congress to authorize the FCC in the fall of 2012 to design a novel auction (the *FCC Incentive Auction*) that would repurpose spectrum—procuring licenses from television broadcasters (a relatively low-value activity) and selling them to parties that put them to better use (e.g., telecommunication companies who want to roll out the next generation of wireless broadband services). Thus the new auction is really a *double auction*, comprising two stages: a *reverse auction*, where the government buys back licenses for spectrum from their current owners; and then a *forward auction*, where the government sells the procured licenses to the highest bidder. Computer science techniques played a crucial role in the design of the new reverse auction. The main aspects of the forward auction have been around a long time; here, theoretical computer science has contributed on the analysis side, and to understanding when and why such forward auctions work well. Sections 1.2 and 1.3 give more details on the reverse and forward parts of the auction, respectively.

The FCC Incentive Auction finished around the end of March 2017, and so the numbers are in. The government spent roughly 10 billion USD in the reverse part of the auction buying back licenses from television broadcasters, and earned 20 billion USD of revenue in the forward auction. Most of the 10 billion USD profit was used to reduce the US deficit![1]

1.2 Reverse Auction

1.2.1 Descending Clock Auctions

The reverse auction is the part of the FCC Incentive Auction that was totally new, and where computer science techniques played a crucial role in the design. The auction format, proposed by Milgrom and Segal [108],

[1]This was the plan all along, which is probably one of the reasons the bill didn't have trouble passing through a notoriously partisan Congress. Another reason might be the veto-proof title of the bill: "The Middle Class Tax Relief and Job Creation Act."

is what's called a *descending clock auction*. By design, the auction is very simple from the perspective of any one participant. The auction is iterative, and operates in rounds. In each round of the auction, each remaining broadcaster is asked a question of the form: "Would you or would you not be willing to sell your license for (say) 1 million dollars?" The broadcaster is allowed to say "no," with the consequence of getting kicked out of the auction forevermore (the station will keep its license and remain on the air, and will receive no compensation from the government). The broadcaster is also allowed to say "yes" and accept the buyout offer. In the latter case, the government will not necessarily buy the license for 1 million dollars—in the next round, the broadcaster might get asked the same question, with a lower buyout price (e.g., 950,000 USD). If a broadcaster is still in the auction when it ends (more on how it ends in a second), then the government does indeed buy their license, at the most recent (and hence lowest) buyout offer. Thus a broadcaster just has to answer a sequence of "yes/no" questions for some decreasing sequence of buyout offers. The obvious strategy for a broadcaster is to formulate the lowest acceptable offer for their license, and to drop out of the auction once the buyout price drops below this threshold.

The auction begins with very high buyout offers, so that every broadcaster would be ecstatic to sell their license at the initial price. Intuitively, the auction then tries to reduce the buyout prices as much as possible, subject to clearing a target amount of spectrum. Spectrum is divided into channels which are blocks of 6 MHz each. For example, one could target broadcasters assigned to channels 38–51, and insist on clearing 10 out of these 14 channels (60 MHz overall).[2] By "clearing a channel," we mean clearing it *nationwide*. Of course, in the descending clock auction, bidders will drop out in an uncoordinated way—perhaps the first station to drop out is channel 51 in Arizona, then channel 41 in western Massachusetts, and so on. To clear several channels nationwide without buying out essentially everybody, it was essential for the government to use its power to *reassign* the channels of the stations that remain on the air. Thus while a station that drops out of the auction is guaranteed to retain its license, it is not guaranteed to retain its channel—a station broadcasting on channel 51 before the auction might be forced to broadcast on channel 41 after the auction.

The upshot is that the auction maintains the invariant that the stations that have dropped out of the auction (and hence remain on the air) can be assigned channels so that at most a target number of channels get used (in our example, 4 channels). This is called the *repacking problem*. Naturally, two stations with overlapping broadcasting regions cannot be assigned the same channel (otherwise they would interfere with each other). See Figure 1.1.

1.2.2 Solving the Repacking Problem

Any properly trained computer scientist will recognize the repacking problem as the NP-complete graph coloring problem in disguise.[3] For the proposed auction format to be practically viable, it must quickly solve the repacking problem. Actually, make that thousands of repacking problems every round of the auction![4]

The responsibility of quickly solving repacking problems fell to a team led by Kevin Leyton-Brown (see [54, 97]). The FCC gave the team a budget of one minute per repacking problem, ideally with most instances solved within one second. The team's approach was to build on state-of-the-art solvers for the satisfiability (SAT) problem. As you can imagine, it's straightforward to translate an instance of the repacking

[2]The FCC Incentive Auction would up clearing 84 MHz of spectrum (14 channels).

[3]The actual repacking problem was more complicated—overlapping stations cannot even be assigned adjacent channels, and there are idiosyncratic constraints at the borders with Canada and Mexico. See Leyton-Brown et al. [97] for more details. But the essence of the repacking problem really is graph coloring.

[4]Before the auction makes a lower offer to some remaining broadcaster in the auction, it needs to check that it would be OK for the broadcaster to drop out of the auction. If a station's dropping out would render the repacking problem infeasible, then that station's buyout price remains frozen until the end of the auction.

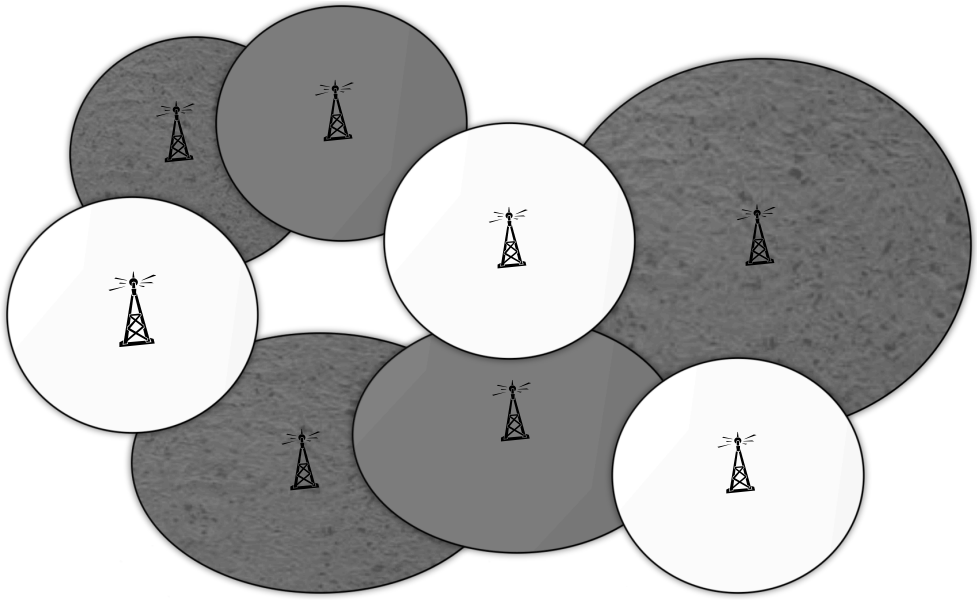

Figure 1.1: Different TV stations with overlapping broadcasting areas must be assigned different channels (indicated by shades of gray). Checking whether or not a given subset of stations can be assigned to a given number of channels without interference is an NP-hard problem.

problem into a SAT formula (even with the idiosyncratic constraints).[5] Off-the-shelf SAT solvers did pretty well, but still timed out on too many representative instances.[6] Leyton-Brown's team added several new innovations, including taking advantage of the advance knowledge of the interference constraints (as station's broadcasting regions are public), and implementing a number of effective caching techniques (reusing work done solving previous instances to quickly solve closely related new instances). In the end, they were able to solve more than 99% of the relevant repacking problems in under a minute.

Hopefully the high-level point is clear:

> without cutting-edge techniques for solving NP-complete problems, *the FCC would have had to use a different auction format*.

1.2.3 Reverse Greedy Algorithms

One final twist: the novel reverse auction format motivates some basic algorithmic questions (and thus ideas flow from computer science to auction theory and back). We can think of the auction as an algorithm, a heuristic that tries to maximize the value of the stations that remain on the air, subject to clearing the target amount of spectrum. Milgrom and Segal [108] prove that, ranging over all ways of implementing the auction (i.e., of choosing the sequences of descending prices), the corresponding algorithms are exactly the *reverse greedy algorithms*.[7] This result gives the first extrinsic reason to study the power and limitations of reverse

[5] A typical representative instance would have thousands of variables and tens of thousands of constraints.

[6] Every time the repacking algorithm fails to find a repacking when one exists, money is left on the table—the auction has to conservatively leave the current station's buyout offer frozen, even though it could have safely lowered it.

[7] For example, Kruskal's algorithm for the minimum spanning tree problem (start with the empty set, go through the edges of the graph from cheapest to most expensive, adding an edge as long as it doesn't create a cycle) is a standard (forward) greedy algorithm. The reverse version is: start with the entire edge set, go through the edges in reverse sorted order, and remove an edge whenever it

greedy algorithms, a research direction explored by Dütting et al. [48] and Gkatzelis et al. [62].

1.3 Forward Auction

Computer science did not have an opportunity to influence the design of the forward auction used in the FCC Incentive Auction, which resembles the formats used over the past 20+ years. Still, the theoretical computer science toolbox turns out to be ideally suited for explaining when and why these auctions work well.[8]

1.3.1 Bad Auction Formats Cost Billions

Spectrum auction design is stressful, because small mistakes can be extremely costly. One cautionary tale is provided by an auction run by the New Zealand government in 1990 (before governments had much experience with auctions). For sale were 10 essentially identical national licenses for television broadcasting. For some reason, lost to the sands of time, the government decided to sell these licenses by running 10 second-price auctions in parallel. Recall that a *second-price* or *Vickrey* auction for a single good is a sealed-bid auction that awards the item to the highest bidder and charges her the highest bid by someone else (the second-highest bid overall). When selling a single item, the Vickrey auction is often a good solution. In particular, each bidder has a dominant strategy (always at least as good as all alternatives), which is to bid her true maximum willingness-to-pay.[9]

The nice properties of a second-price auction evaporate if many of them are run simultaneously. It is no longer clear how a bidder should bid. For example, imagine you want one of the licenses, but only one. How should you bid? One legitimate strategy is to pick one of the licenses—at random, say—and go for it. Another strategy is to bid less aggressively on multiple licenses, hoping that you get one at a bargain price, and that you don't inadvertently win extra licenses that you don't want. The difficulty is trading off the risk of winning too many licenses with the risk of winning too few.

The challenge of bidding intelligently in simultaneous sealed-bid auctions makes the auction format prone to poor outcomes. The revenue in the 1990 New Zealand auction was only $36 million, a paltry fraction of the projected $250 million. On one license, the high bid was $100,000 while the second-highest bid (and selling price) was $6! On another, the high bid was $7 million and the second-highest was $5,000. To add insult to injury, the winning bids were made available to the public, who could then see just how much money was left on the table!

1.3.2 Simultaneous Ascending Auctions

Modern forward auctions are *simultaneous ascending auctions (SAAs)*, following 1993 proposals by McAfee and by Milgrom and Wilson. You've seen—in the movies, at least—the call-and-response format of an ascending single-item auction, where an auctioneer asks for takers at successively higher prices. Such an auction ends when there's only one person left accepting the currently proposed price (who then wins, at this price). Conceptually, SAAs are like a bunch of single-item English auctions being run in parallel in the same room, with one auctioneer per item.

---

doesn't disconnect the graph. For the minimum spanning tree problem (and more generally for finding the minimum-weight basis of a matroid), the reverse greedy algorithm is just as optimal as the forward one. In general (and even for e.g. bipartite matching), the reverse version of a good forward greedy algorithm can be bad [48].

[8] Much of the discussion in Sections 1.3.1–1.3.3 is from [129, Lecture 8], which in turn draws heavily from Milgrom [107].

[9] Intuitively, a second-price auction shades your bid optimally after the fact, so there's no reason to try to game it.

The primary reason that SAAs work better than sequential or sealed-bid auctions is *price discovery*. As a bidder acquires better information about the likely selling prices of licenses, she can implement mid-course corrections: abandoning licenses for which competition is fiercer than anticipated, snapping up unexpected bargains, and rethinking which packages of licenses to assemble. The format typically resolves the miscoordination problems that plague simultaneous sealed-bid auctions.

1.3.3 Inefficiency in SAAs

SAAs have two big vulnerabilities. The first problem is *demand reduction*, and this is relevant even when items are substitutes.[10] Demand reduction occurs when a bidder asks for fewer items than it really wants, to lower competition and therefore the prices paid for the items that it gets.

To illustrate, suppose there are two identical items and two bidders. The first bidder has valuation 10 for one of the items and valuation 20 for both. The second bidder has valuation 8 for one of the items and does not want both (i.e., her valuation remains 8 for both). The socially optimal outcome is to give both licenses to the first bidder. Now consider how things play out in an SAA. The second bidder would be happy to have either item at any price less than 8. Thus, the second bidder drops out only when both items have price at least 8. If the first bidder stubbornly insists on winning both items, her utility is $20 - 16 = 4$. An alternative strategy for the first bidder is to simply concede the second item and never bid on it. The second bidder takes the second item and (since it only wants one license) withdraws interest in the first, leaving it for the first bidder. Both bidders get their item essentially for free, and the utility of the first bidder has jumped to 10.

The second big problem with SAAs is relevant when items can be complements, and is called the *exposure problem*.[11] As an example, consider two bidders and two nonidentical items. The first bidder only wants both items—they are complementary items for the bidder—and her valuation is 100 for them (and 0 otherwise). The second bidder is willing to pay 75 for either item but only wants one item. The socially optimal outcome is to give both items to the first bidder. But in an SAA, the second bidder will not drop out until the price of both items reaches 75. The first bidder is in a no-win situation: to get both items it would have to pay 150, more than her value. The scenario of winning only one item for a nontrivial price could be even worse. Thus the exposure problem leads to economically inefficient allocations for two reasons. First, an overly aggressive bidder might acquire unwanted items. Second, an overly tentative bidder might fail to acquire items for which it has the highest valuation.

1.3.4 When Do SAAs Work Well?

If you ask experts who design or consult for bidders in real-world SAAs, a rough consensus emerges about when they are likely to work well.

Folklore Belief 1. *Without strong complements,* SAAs work pretty well. Demand reduction does happen, but it is not a deal-breaker because the loss of efficiency appears to be small.

Folklore Belief 2. *With strong complements,* simple auctions like SAAs are not good enough. The exposure problem is a deal-breaker because it can lead to very poor outcomes (in terms of both economic efficiency and revenue).

[10] Items are substitutes if they provide diminishing returns—having one item only makes others less valuable. For two items A and B, for example, the substitutes condition means that $v(AB) \leq v(A) + v(B)$. In a spectrum auction context, two licenses for the same area with equal-sized frequency ranges are usually substitute items.

[11] Items are complements if there are synergies between them, so that possessing one makes others more valuable. With two items A and B, this translates to the property $v(AB) > v(A) + v(B)$. Complements arise naturally in wireless spectrum auctions, as some bidders want a collection of licenses that are adjacent, either in their geographic areas or in their frequency ranges.

There are a number of beautiful and useful theoretical results about spectrum auctions in the economics literature, but none map cleanly to these two folklore beliefs. A possible explanation: translating these beliefs into theorems seems to fundamentally involve approximate optimality guarantees, a topic right in the wheelhouse of theoretical computer science.

In the standard model of *combinatorial auctions*, there are n bidders (e.g., telecoms) and m items (e.g., licenses). Bidder i has a nonnegative valuation $v_i(S)$ for each subset S of items it might receive. Note that each bidder has 2^m parameters. Each bidder wants to maximize its utility, which is the value of the items received minus the price it pays for them. From a social perspective, we'd like to award bundles of items $T_1, \ldots, T_n$ to the bidders to maximize the *social welfare* $\sum_{i=1}^{n} v_i(T_i)$.

To make the first folklore belief precise, we need to commit to a definition of "without strong complements" and to a specific auction format. We'll focus on simultaneous first-price auctions (S1As), where each bidder submits a separate bid for each item, for each item the winner is the highest bidder, and winning bidders pay their bid on each item won.[12] One relatively permissive definition of "complement-free" is to restrict bidders to have *subadditive valuations*. This means what it sounds like: if A and B are two bundles of items, then bidder i's valuation $v_i(A \cup B)$ of their union should be at most that of the sum $v_i(A) + v_i(B)$. Observe the subadditivity is violated in our exposure problem example above.

We also need to define what we mean by "the outcome of an auction" like S1As. Remember that bidders are strategic, and will bid to maximize their utility (value of items won minus the price paid). Thus we should prove approximation guarantees for the *equilibria* of auctions. Happily, computer scientists have been working hard since 1999 to prove approximation guarantees for game-theoretic equilibria, also known as bounds on *the price of anarchy* [92, 124, 132]. In the early days, price-of-anarchy bounds appeared somewhat ad hoc and problem-specific. Fast forwarding to the present, we now have a powerful and user-friendly theory for proving price-of-anarchy bounds, which combine "extension theorems" and "composition theorems" to build up bounds for complex settings (including S1As) from bounds for simple settings.[13] In particular, Feldman et al. [52] proved the following translation of Folklore Belief #1.[14]

Theorem 1.1 (Feldman et al. [52])**.** *When every bidder has a subadditive valuation, every equilibrium of an S1A has social welfare at least 50% of the maximum possible.*

One version of Theorem 1.1 concerns (mixed) Nash equilibria, as studied in the Solar Lectures. Even here, the bound in Theorem 1.1 is tight in the worst case [37]. The approximation guarantee in Theorem 1.1 holds more generally for *Bayes-Nash equilibria*, the standard equilibrium notion for games of incomplete information.[15]

Moving on to the second folklore belief, let's now drop the subadditivity restriction. S1As no longer work well.

Theorem 1.2 (Hassidim et al. [73])**.** *When bidders have arbitrary valuations, an S1A can have a mixed Nash equilibrium with social welfare arbitrarily smaller than the maximum possible.*

[12] Similar results hold for other auction formats, like simultaneous second-price auctions. Directly analyzing what happens in iterative auctions like SAAs when there are multiple items has proved difficult.

[13] We will say more about this theory in Lunar Lecture 5. See also Roughgarden et al. [134] for a recent survey.

[14] To better appreciate this result, we note that multi-item auctions like S1As are so strategically complex that they have historically been seen as unanalyzable. For example, we have no idea what their equilibria look like in general. Nevertheless, we can prove good approximation guarantees for them!

[15] In more detail, there is a commonly known prior distribution over bidders' valuations. In a Bayes-Nash equilibrium, every bidder bids to maximize its expected utility given its information at the time: its own valuation, its posterior belief about other bidders' valuations, and the bidding strategies (mapping valuations to bids) used by the other bidders. Theorem 1.1 continues to hold for every Bayes-Nash equilibrium of an S1A, as long as bidders' valuations are drawn independently (but not necessarily identically).

Thus for S1As, the perspective of worst-case approximation confirms the dichotomy between the cases of substitutes and complements. But the lower bound in Theorem 1.2 holds for just one auction format. Could we do better with a different (but still relatively simple) auction format? Folklore Belief #2 asserts the stronger statement that *no* "simple" auction works well with general valuations. This stronger statement can also be translated into a theorem (using nondeterministic communication complexity), and this will be the main subject of Lunar Lecture 2.

Theorem 1.3 (Roughgarden [126])**.** *With general valuations,* every *simple auction can have arbitrarily bad equilibria.*

The definition of "simple" used in Theorem 1.3 is quite generous: it requires only that the number of strategies available to each player is sub-*doubly*-exponential in the number of items m. For example, running separate single-item auctions provides each player with only an exponential number of strategies (assuming a bounded number of possible bid values). Thus Theorem 1.3 makes use of the theoretical computer science toolbox to provide solid footing for Folklore Belief #2.

Lunar Lecture 2

Communication Barriers to Near-Optimal Equilibria

Lecturer: Tim Roughgarden *Scribe: Omri Weinstein*

This lecture is about the communication complexity of the welfare-maximization problem in combinatorial auctions and its implications for the price of anarchy of simple auctions. After defining the model in Section 2.1, Section 2.2 proves lower bounds for nondeterministic communication protocols and Section 2.3 gives a black-box translation of these lower bounds to equilibria of simple auctions. In particular, Section 2.3 provides the proof of Theorem 1.3 from last lecture. Section 2.4 concludes with a juicy open problem on the topic.[1]

2.1 Welfare Maximization in Combinatorial Auctions

Recall from Section 1.3.4 the basic setup in the study of combinatorial auctions.

1. There are k players. (In a spectrum auction, these are the telecoms.)
2. There is a set M of m items. (In a spectrum auction, these are the licenses.)
3. Each player i has a *valuation* $v_i : 2^M \to \mathbb{R}_+$. The number $v_i(T)$ indicates i's value, or willingness to pay, for the items $T \subseteq M$. The valuation is the private input of player i — i knows v_i but none of the other v_j's. (I.e., this is a number-in-hand model.) We assume that $v_i(\emptyset) = 0$ and that the valuations are *monotone*, meaning $v_i(S) \leq v_i(T)$ whenever $S \subseteq T$. (The more items, the better.) To avoid bit complexity issues, we'll also assume that all of the $v_i(T)$'s are integers with description length polynomial in k and m. We sometimes impose additional restrictions on the valuations to study special cases of the general problem.

Note that we may have more than two players—more than just Alice and Bob. (For example, you might want to think of k as $\approx m^{1/3}$.) Also note that the description length of a player's valuation is exponential in the number of items m.

In the *welfare-maximization problem*, the goal is to partition the items M into sets $T_1, \ldots, T_k$ to maximize, at least approximately, the welfare

$$\sum_{i=1}^{k} v_i(T_i), \tag{2.1}$$

[1] Much of this lecture is drawn from [130, Lecture 7].

using communication polynomial in n and m. Note this amount of communication is logarithmic in the sizes of the private inputs. Maximizing social welfare (2.1) is the most commonly studied objective function in combinatorial auctions, and it is the one we will focus on in this lecture.

2.2 Communication Lower Bounds for Approximate Welfare Maximization

This section studies the communication complexity of computing an approximately welfare-maximizing allocation in a combinatorial auction. For reasons that will become clear in Section 2.3, we are particularly interested in the problem's nondeterministic communication complexity.[2]

2.2.1 Lower Bound for General Valuations

We begin with a result of Nisan [115] showing that, alas, computing even a very weak approximation of the welfare-maximizing allocation requires exponential communication. To make this precise, it is convenient to turn the optimization problem of welfare maximization into a decision problem. In the Welfare-Maximization(k) problem, the goal is to correctly identify inputs that fall into one of the following two cases:

(1) Every partition $(T_1, \ldots, T_k)$ of the items has welfare at most 1.

(0) There exists a partition $(T_1, \ldots, T_k)$ of the items with welfare at least k.

Arbitrary behavior is permitted on inputs that fail to satisfy either (1) or (0). Clearly, communication lower bounds for Welfare-Maximization(k) apply to the more general problem of obtaining a better-than-k-approximation of the maximum welfare.[3]

Theorem 2.1 ([115]). *The nondeterministic communication complexity of Welfare-Maximization(k) is* $\exp\{\Omega(m/k^2)\}$*, where k is the number of players and m is the number of items.*

This lower bound is exponential in m, provided that $m = \Omega(k^{2+\epsilon})$ for some $\epsilon > 0$. Since communication complexity lower bounds apply even to players who cooperate perfectly, this impossibility result holds even when all of the (tricky) incentive issues are ignored.

2.2.2 The Multi-Disjointness Problem

The plan for the proof of Theorem 2.1 is to reduce a multi-party version of the Disjointness problem to it. There is some ambiguity about how to define a version of Disjointness for three or more players. For example, suppose there are three players, and among the three possible pairings of them, two have disjoint sets while the third have intersecting sets. Should this count as a "yes" or "no" instance? We'll skirt this issue by worrying only about unambiguous inputs, that are either "totally disjoint" or "totally intersecting."

Formally, in the Multi-Disjointness problem, each of the k players i holds an input $\mathbf{x}_i \in \{0,1\}^n$. (Equivalently, a set $S_i \subseteq \{1, 2, \ldots, n\}$.) The task is to correctly identify inputs that fall into one of the following two cases:

[2] For basic background on nondeterministic multi-party communication protocols, see Kushilevitz and Nisan [93] or Roughgarden [130].

[3] Achieving a k-approximation is trivial: every player communicates their value $v_i(M)$ for the whole set of items, and the entire set of items is awarded to the bidder with the highest value for them.

(1) "Totally disjoint," with $S_i \cap S_{i'} = \emptyset$ for every $i \neq i'$.

(0) "Totally intersecting," with $\cap_{i=1}^{k} S_i \neq \emptyset$.

When $k = 2$, this is just Disjointness. When $k > 2$, there are inputs that are neither 1-inputs nor 0-inputs. We let protocols off the hook on such ambiguous inputs — they can answer "1" or "0" with impunity.

The following communication complexity lower bound for Multi-Disjointness is credited to Jaikumar Radhakrishnan and Venkatesh Srinivasan in [115]. (The proof is elementary, and for completeness is given in Section 2.5.)

Theorem 2.2. *The nondeterministic communication complexity of Multi-Disjointness, with k players with n-bit inputs, is $\Omega(n/k)$.*

This nondeterministic lower bound is for verifying a 1-input. (It is easy to verify a 0-input—the prover just suggests the index of an element r in $\cap_{i=1}^{k} S_i$.)[4]

2.2.3 Proof of Theorem 2.1

The proof of Theorem 2.1 relies on Theorem 2.2 and a combinatorial gadget. We construct this gadget using the probabilistic method. As a thought experiment, consider t random partitions $P^1, \ldots, P^t$ of M, where t is a parameter to be defined later. By a random partition $P^j = (P^j_1, \ldots, P^j_k)$, we just mean that each of the m items is assigned to exactly one of the k players, independently and uniformly at random.

We are interested in the probability that two classes of different partitions intersect: for all $i \neq i'$ and $j \neq \ell$, since the probability that a given item is assigned to i in P^j and also to i' in P^ℓ is $\frac{1}{k^2}$, we have

$$\mathbf{Pr}\left[P^j_i \cap P^\ell_{i'} = \emptyset\right] = \left(1 - \frac{1}{k^2}\right)^m \leq e^{-m/k^2}.$$

Taking a Union Bound over the k choices for i and i' and the t choices for j and ℓ, we have

$$\mathbf{Pr}\left[\exists i \neq i', j \neq \ell \text{ s.t. } P^j_i \cap P^\ell_{i'} = \emptyset\right] \leq k^2 t^2 e^{-m/k^2}. \tag{2.2}$$

Call $P^1, \ldots, P^t$ an *intersecting family* if $P^j_i \cap P^\ell_{i'} \neq \emptyset$ whenever $i \neq i'$, $j \neq \ell$. By (2.2), the probability that our random experiment fails to produce an intersecting family is less than 1 provided $t < \frac{1}{k} e^{m/2k^2}$. The following lemma is immediate.

Lemma 2.3. *For every $m, k \geq 1$, there exists an intersecting family of partitions $P^1, \ldots, P^t$ with $t = \exp\{\Omega(m/k^2)\}$.*

A simple combination of Theorem 2.2 and Lemma 2.3 now proves Theorem 2.1.

Proof. (of Theorem 2.1) The proof is a reduction from Multi-Disjointness. Fix k and m. (To be interesting, m should be significantly bigger than k^2.) Let $(S_1, \ldots, S_k)$ denote an input to Multi-Disjointness with t-bit inputs, where $t = \exp\{\Omega(m/k^2)\}$ is the same value as in Lemma 2.3. We can assume that the players have

[4] To prove Theorem 2.1, we'll be interested in the case where k is much smaller than n, such as $k = \Theta(\log n)$. Intuition might suggest that the lower bound should be $\Omega(n)$ rather than $\Omega(n/k)$, but this is incorrect — a slightly non-trivial argument shows that Theorem 2.2 is tight for nondeterministic protocols (for all small enough k, like $k = O(\sqrt{n})$). This factor-k difference won't matter for our applications, however.

coordinated in advance on an intersecting family of t partitions of a set M of m items. Each player i uses this family and its input S_i to form the following valuation:

$$v_i(T) = \begin{cases} 1 & \text{if } T \supseteq P_i^j \text{ for some } j \in S_i \\ 0 & \text{otherwise.} \end{cases}$$

That is, player i is either happy (value 1) or unhappy (value 0), and is happy if and only if it receives all of the items in the corresponding class P_i^j of some partition P^j with index j belonging to its input to Multi-Disjointness. The valuations $v_1, \ldots, v_k$ define an input to Welfare-Maximization(k). Forming this input requires no communication between the players.

Consider the case where the input to Multi-Disjointness is a 1-input, with $S_i \cap S_{i'} = \emptyset$ for every $i \neq i'$. We claim that the induced input to Welfare-Maximization(k) is a 1-input, with maximum welfare at most 1. To see this, consider a partition $(T_1, \ldots, T_k)$ in which some player i is happy (with $v_i(T_i) = 1$). For some $j \in S_i$, player i receives all the items in P_i^j. Since $j \notin S_{i'}$ for every $i' \neq i$, the only way to make a second player i' happy is to give it all the items in $P_{i'}^\ell$ in some other partition P^ℓ with $\ell \in S_{i'}$ (and hence $\ell \neq j$). Since $P^1, \ldots, P^t$ is an intersecting family, this is impossible — P_i^j and $P_{i'}^\ell$ overlap for every $\ell \neq j$.

When the input to Multi-Disjointness is a 0-input, with an element r in the mutual intersection $\cap_{i=1}^k S_i$, we claim that the induced input to Welfare-Maximization(k) is a 0-input, with maximum welfare at least k. This is easy to see: for $i = 1, 2, \ldots, k$, assign the items of P_i^r to player i. Since $r \in S_i$ for every i, this makes all k players happy.

This reduction shows that a (deterministic, nondeterministic, or randomized) protocol for Welfare-Maximization(k) yields one for Multi-Disjointness (with t-bit inputs) with the same communication. We conclude that the nondeterministic communication complexity of Welfare-Maximization(k) is $\Omega(t/k) = \exp\{\Omega(m/k^2)\}$. □

2.2.4 Subadditive Valuations

To an algorithms person, Theorem 2.1 is depressing, as it rules out any non-trivial positive results. A natural idea is to seek positive results by imposing additional structure on players' valuations. Many such restrictions have been studied. We consider here the case of *subadditive* valuations (see also Section 1.3.4 from last lecture), where each v_i satisfies $v_i(S \cup T) \leq v_i(S) + v_i(T)$ for every pair $S, T \subseteq M$.

Our reduction in Theorem 2.1 immediately yields a weaker inapproximability result for welfare maximization with subadditive valuations. Formally, define the Welfare-Maximization(2) problem as that of identifying inputs that fall into one of the following two cases:

(1) Every partition $(T_1, \ldots, T_k)$ of the items has welfare at most $k + 1$.

(0) There exists a partition $(T_1, \ldots, T_k)$ of the items with welfare at least $2k$.

Communication lower bounds for Welfare-Maximization(2) apply also to the more general problem of obtaining a better-than-2-approximation of the maximum welfare.

Theorem 2.4 (Dobzinski et al. [47]). *The nondeterministic communication complexity of Welfare-Maximization(2) is* $\exp\{\Omega(m/k^2)\}$*, even when all players have subadditive valuations.*

This theorem follows from a modification of the proof of Theorem 2.1. The 0-1 valuations used in that proof are not subadditive, but they can be made subadditive by adding 1 to each bidder's valuation $v_i(T)$ of each non-empty set T. The social welfare obtained in inputs corresponding to 1- and 0-inputs of Multi-Disjointness become $k + 1$ and $2k$, respectively, and this completes the proof of Theorem 2.4.

There is also a quite non-trivial matching upper bound of 2 for deterministic, polynomial-communication protocols [50].

2.3 Lower Bounds on the Price of Anarchy of Simple Auctions

The lower bounds of the previous section show that every protocol for the welfare-maximization problem that interacts with the players and then explicitly computes an allocation has either a bad approximation ratio or high communication cost. Over the past decade, many researchers have aimed to shift the work from the protocol to the players, by analyzing the equilibria of simple auctions. Can such equilibria bypass the communication complexity lower bounds proved in Section 2.2? The answer is not obvious, because equilibria are defined non-constructively, and not through a low-cost communication protocol.

2.3.1 Auctions as Games

What do we mean by a "simple" auction? For example, recall the *simultaneous first-price auctions (S1As)* introduced in Section 1.3.4 of the preceding lecture. Each player i chooses a strategy $b_{i1}, \ldots, b_{im}$, with one bid per item.[5] Each item is sold separately in parallel using a "first-price auction"—the item is awarded to the highest bidder, and the price is whatever that player bid.[6] The payoff of a player in a given outcome (i.e., given a choice of strategy for each player) is then her utility:

$$\underbrace{v_i(T_i)}_{\text{value of items won}} - \underbrace{\sum_{j \in S_i} b_{ij}}_{\text{price paid for them}},$$

where T_i denotes the items on which i is the highest bidder (given the bids of the others).

Bidders strategize already in a first-price auction for a single item—a bidder certainly doesn't want to bid her actual valuation (this would guarantee utility 0), and instead will "shade" her down to a lower value. (How much to shade is a tricky question, and depends on what the other bidders are doing.) Thus it makes sense to assess the performance of an auction by its equilibria. As usual, a Nash equilibrium comprises a (randomized) strategy for each player, so that no player can unilaterally increase her expected payoff through a unilateral deviation to some other strategy (given how the other players are bidding).

2.3.2 The Price of Anarchy

So how good are the equilibria of various games, such as S1As? To answer this question, we use an analog of the approximation ratio, adapted for equilibria. Given a game (like an S1A) and a nonnegative maximization objective function on the outcomes (like the social welfare), the *price of anarchy (POA)* Koutsoupias and Papadimitriou [92] is defined as the ratio between the objective function value of an optimal solution, and that of the worst equilibrium:

$$\mathsf{PoA}(G) := \frac{f(OPT(G))}{\min_{\mu \text{ is an equilibrium of } G} f(\mu)},$$

[5] To keep the game finite, let's agree that each bid has to be an integer between 0 and some known upper bound B.

[6] You may have also heard of the *Vickrey* or *second-price* auction, where the winner does not pay their own bid, but rather the highest bid by someone else (the second-highest overall). We'll stick with S1As for simplicity, but similar results are known for simultaneous second-price auctions, as well.

where G denotes a game, f denotes a (maximization) objective function, and $OPT(G)$ is the optimal outcome of G (with respect to f).[7] Thus the price of anarchy of a game quantifies the inefficiency of selfish behavior.[8] The POA of a game and a maximization objective function is always at least 1. It is common to identify "good performance" of a system with strategic participants as having a POA close to 1.[9]

The POA depends on the choice of equilibrium concept. For example, the POA with respect to approximate Nash equilibria can only be worse (i.e., bigger) than for exact Nash equilibria (since there are only more of the former).

2.3.3 The Price of Anarchy of S1As

As we saw in Theorem 1.1 of the preceding lecture, the equilibria of simple auctions like S1As can be surprisingly good.[10] We restate that result here.[11]

Theorem 2.5 (Feldman et al. [52]). *In every S1A with subadditive bidder valuations, the POA is at most 2.*

This result is particularly impressive because achieving an approximation factor of 2 for the welfare-maximization problem with subadditive bidder valuations by any means (other than brute-force search) is not easy (see [50]).

As mentioned last lecture, a recent result shows that the analysis of [52] is tight.

Theorem 2.6 (Christodoulou et al. [37]). *The worst-case POA of S1As with subadditive bidder valuations is at least 2.*

The proof of Theorem 2.6 is an ingenious explicit construction—the authors exhibit a choice of subadditive bidder valuations and a Nash equilibrium of the corresponding S1A so that the welfare of this equilibrium is only half of the maximum possible. One reason that proving results like Theorem 2.6 is challenging is that it can be difficult to solve for a (bad) equilibrium of a complex game like a S1A.

2.3.4 Price-of-Anarchy Lower Bounds from Communication Complexity

Theorem 2.5 motivates an obvious question: can we do better? Theorem 2.6 implies that the analysis in [52] cannot be improved, but can we reduce the POA by considering a different auction? Ideally, the auction would still be "reasonably simple" in some sense. Alternatively, perhaps no "simple" auction could be better than S1As? If this is the case, it's not clear how to prove it directly—proving lower bounds via explicit constructions auction-by-auction does not seem feasible.

Perhaps it's a clue that the POA upper bound of 2 for S1As (Theorem 2.5) gets stuck at the same threshold for which there is a lower bound for protocols that use polynomial communication (Theorem 2.4). It's not clear, however, that a lower bound for low-communication protocols has anything to do with equilibria. Can we extract a low-communication protocol from an equilibrium?

[7] If μ is a probability distribution over outcomes, as in a mixed Nash equilibrium, then $f(\mu)$ denotes the expected value of f w.r.t. μ.

[8] Games generally have multiple equilibria. Ideally, we'd like an approximation guarantee that applies to *all* equilibria, so that we don't need to worry about which one is reached—this is the point of the POA.

[9] One caveat is that it's not always clear that a system will reach an equilibrium in a reasonable amount of time. A natural solution to this is to relax the notion of equilibrium enough so that it become relatively easy to reach an equilibrium. See Lunar Lecture 5 for more on this point.

[10] The first result of this type, for simultaneous second-price auctions and bidders with submodular valuations, is due to Christodoulou et al. [36].

[11] For a proof, see the original paper [52] or course notes by your lecturer [127, Lecture 17.5].

Theorem 2.7 (Roughgarden [126]). *Fix a class $\mathcal{V}$ of possible bidder valuations. Suppose there exists no nondeterministic protocol with subexponential (in m) communication for the 1-inputs of the following promise version of the welfare-maximization problem with bidder valuations in $\mathcal{V}$:*

(1) Every allocation has welfare at most W^*/α.

(0) There exists an allocation with welfare at least W^*.

Let ϵ be bounded below by some inverse polynomial function of n and m. Then, for every auction with sub-doubly-exponential (in m) strategies per player, the worst-case POA of ϵ-approximate Nash equilibria with bidder valuations in $\mathcal{V}$ is at least α.

Theorem 2.7 says that lower bounds for nondeterministic protocols carry over to all "sufficiently simple" auctions, where "simplicity" is measured by the number of strategies available to each player. These POA lower bounds follow automatically from communication complexity lower bounds, and do not require any new explicit constructions.

To get a feel for the simplicity constraint, note that S1As with integral bids between 0 and B have $(B+1)^m$ strategies per player—singly exponential in m. On the other hand, in a "direct-revelation" auction, where each bidder is allowed to submit a bid on each bundle $S \subseteq M$ of items, each player has a doubly-exponential (in m) number of strategies.[12]

The POA lower bound promised by Theorem 2.7 is only for approximate Nash equilibria; since the POA is a worst-case measure and the set of ϵ-NE is nondecreasing with ϵ, this is weaker than a lower bound for exact Nash equilibria. It is an open question whether or not Theorem 2.7 holds also for the POA of exact Nash equilibria.[13]

Theorem 2.7 has a number of interesting corollaries. First, consider the case where $\mathcal{V}$ is the set of subadditive valuations. Since S1As have only a singly-exponential (in m) number of strategies per player, Theorem 2.7 applies to them. Thus, combining it with Theorem 2.4 recovers the POA lower bound of Theorem 2.6—modulo the exact vs. approximate Nash equilibria issue—and shows the optimality of the upper bound in Theorem 2.5 without an explicit construction. Even more interestingly, this POA lower bound of 2 applies not only to S1As, but more generally to all auctions in which each player has a sub-doubly-exponential number of strategies. Thus, S1As are in fact *optimal* among the class of all such auctions when bidders have subadditive valuations (w.r.t. the worst-case POA of ϵ-approximate Nash equilibria).

We can also take $\mathcal{V}$ to be the set of all (monotone) valuations, and then combine Theorem 2.7 with Theorem 2.1 to deduce that no "simple" auction gives a non-trivial (i.e., better-than-k) approximation for general bidder valuations. We conclude that with general valuations, complexity is essential to any auction format that offers good equilibrium guarantees. This completes the proof of Theorem 1.3 from the preceding lecture and formalizes the second folklore belief in Section 1.3.4; we restate that result here.

Theorem 2.8 ([126]). *With general valuations,* every *simple auction can have equilibria with social welfare arbitrarily worse than the maximum possible.*

2.3.5 Proof of Theorem 2.7

Presumably, the proof of Theorem 2.7 extracts a low-communication protocol from a good POA bound. The hypothesis of Theorem 2.7 offers the clue that we should be looking to construct a nondeterministic protocol.

[12] Equilibria can achieve the optimal welfare in a direct-revelation auction, so the bound in Theorem 2.7 on the number of strategies is necessary.

[13] Arguably, Theorem 2.7 is good enough for all practical purposes—a POA upper bound that holds for exact Nash equilibria and does not hold (at least approximately) for approximate Nash equilibria with very small ϵ is too brittle to be meaningful.

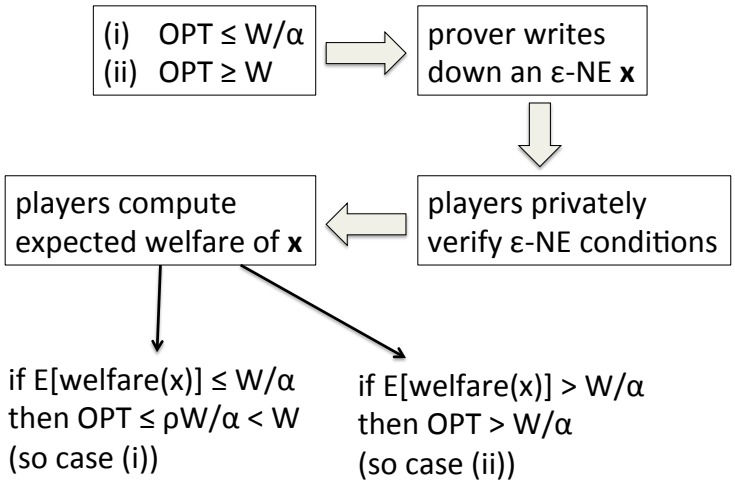

Figure 2.1: Proof of Theorem 2.7. How to extract a low-communication nondeterministic protocol from a good price-of-anarchy bound.

So what could we use an all-powerful prover for? We'll see that a good role for the prover is to suggest a Nash equilibrium to the players.

Unfortunately, it's too expensive for the prover to even write down the description of a Nash equilibrium, even in S1As. Recall that a mixed strategy is a distribution over pure strategies, and that each player has an exponential (in m) number of pure strategies available in a S1A. Specifying a Nash equilibrium thus requires an exponential number of probabilities. To circumvent this issue, we resort to approximate Nash equilibria, which are guaranteed to exist even if we restrict ourselves to distributions with small descriptions. We proved this for two-player games in Solar Lecture 1 (Theorem 1.15); the same argument works with games with any number of players.

Lemma 2.9 (Lipton et al. [99]). *For every $\epsilon > 0$ and every game with k players with strategy sets $A_1, \ldots, A_k$, there exists an approximate Nash equilibrium with description length polynomial in k, $\log(\max_{i=1}^{k} |A_i|)$, and $\frac{1}{\epsilon}$.*

In particular, every game with a sub-doubly-exponential number of strategies admits an approximate Nash equilibrium with subexponential description length.

We now proceed to the proof of Theorem 2.7.

Proof. (of Theorem 2.7) Fix an auction with at most A strategies per player, and a value for $\epsilon = \Omega(1/\text{poly}(k, m))$. Assume that, no matter what the bidder valuations $v_1, \ldots, v_k \in \mathcal{V}$ are, the POA of ϵ-approximate Nash equilibria of the auction is at most $\rho < \alpha$. We will show that A must be doubly-exponential in m.

Consider the following nondeterministic protocol for verifying a 1-input of the welfare-maximization problem—for convincing the k players that every allocation has welfare at most W^*/α. See also Figure 2.1. The prover writes on a publicly visible blackboard an ϵ-approximate Nash equilibrium $(\sigma_1, \ldots, \sigma_k)$ of the auction, with description length polynomial in k, $\log A$, and $\frac{1}{\epsilon} = O(\text{poly}(k, m))$ as guaranteed by Lemma 2.9. The prover also writes down the expected welfare contribution $\mathbf{E}[v_i(S)]$ of each bidder i in this equilibrium.

Given this advice, each player i verifies that σ_i is indeed an ϵ-approximate best response to the other σ_j's and that its expected welfare is as claimed when all players play the mixed strategies $\sigma_1, \ldots, \sigma_k$. Crucially, player i is fully equipped to perform both of these checks without any communication—it knows

its valuation v_i (and hence its utility in each outcome of the game) and the mixed strategies used by all players, and this is all that is needed to verify its ϵ-approximate Nash equilibrium conditions and compute its expected contribution to the welfare.[14] Player i accepts if and only if the prover's advice passes these two tests, and if the expected welfare of the equilibrium is at most W^*/α.

For the protocol correctness, consider first the case of a 1-input, where every allocation has welfare at most W^*/α. If the prover writes down the description of an arbitrary ϵ-approximate Nash equilibrium and the appropriate expected contributions to the social welfare, then all of the players will accept (the expected welfare is obviously at most W^*/α). We also need to argue that, for the case of a 0-input—where some allocation has welfare at least W^*—there is no proof that causes all of the players to accept. We can assume that the prover writes down an ϵ-approximate Nash equilibrium and its correct expected welfare W, since otherwise at least one player will reject. Since the maximum-possible welfare is at least W^* and (by assumption) the POA of ϵ-approximate Nash equilibria is at most $\rho < \alpha$, the expected welfare of the given ϵ-approximate Nash equilibrium must satisfy $W \geq W^*/\rho > W/\alpha$. Since the players will reject such a proof, we conclude that the protocol is correct. Our assumption then implies that the protocol has communication cost exponential in m. Since the cost of the protocol is polynomial in k, m, and $\log A$, A must be doubly exponential in m. □

Conceptually, the proof of Theorem 2.7 argues that, when the POA of ϵ-approximate Nash equilibria is small, every ϵ-approximate Nash equilibrium provides a privately verifiable proof of a good upper bound on the maximum-possible welfare. When such upper bounds require large communication, the equilibrium description length (and hence the number of available strategies) must be large.

2.4 An Open Question

While Theorems 2.4, 2.5, and 2.7 pin down the best-possible POA achievable by simple auctions with subadditive bidder valuations, there are still open questions for other valuation classes. For example, a valuation v_i is *submodular* if it satisfies

$$v_i(T \cup \{j\}) - v_i(T) \leq v_i(S \cup \{j\}) - v_i(S)$$

for every $S \subseteq T \subset M$ and $j \notin T$. This is a "diminishing returns" condition for set functions. Every submodular function is also subadditive, so welfare-maximization with the former valuations is only easier than with the latter.

The worst-case POA of S1As is exactly $\frac{e}{e-1} \approx 1.58$ when bidders have submodular valuations. The upper bound was proved in [143], the lower bound in [37]. It is an open question whether or not there is a simple auction with a smaller worst-case POA. The best lower bound known—for nondeterministic protocols and hence, by Theorem 1.3, for the POA of ϵ-approximate Nash equilibria of simple auctions—is $\frac{2e}{2e-1} \approx 1.23$ [46]. Intriguingly, there is an upper bound (very slightly) better than $\frac{e}{e-1}$ for polynomial-communication protocols [51]—can this better upper bound also be realized as the POA of a simple auction? What is the best-possible approximation guarantee, either for polynomial-communication protocols or for the POA of simple auctions? Resolving this question would require either a novel auction format (better than S1As), a novel lower bound technique (better than Theorem 2.7), or both.

[14] These computations may take a super-polynomial amount of time, but they do not contribute to the protocol's cost.

2.5 Appendix: Proof of Theorem 2.2

The proof of Theorem 2.2 proceeds in three easy steps.

Step 1: *Every nondeterministic protocol with communication cost c induces a cover of the 1-inputs of $M(f)$ by at most 2^c monochromatic boxes.* By "$M(f)$," we mean the k-dimensional array in which the ith dimension is indexed by the possible inputs of player i, and an array entry contains the value of the function f on the corresponding joint input. By a "box," we mean the k-dimensional generalization of a rectangle—a subset of inputs that can be written as a product $A_1 \times A_2 \times \cdots \times A_k$. By "monochromatic," we mean a box that does not contain both a 1-input and a 0-input. (Recall that for the Multi-Disjointness problem there are also inputs that are neither 1 nor 0—a monochromatic box can contain any number of these.) The proof of this step is the same as the standard one for the two-party case (see e.g. [93]).

Step 2: *The number of 1-inputs in $M(f)$ is $(k+1)^n$.* In a 1-input $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$, for every coordinate ℓ, at most one of the k inputs has a 1 in the ℓth coordinate. This yields $k+1$ options for each of the n coordinates, thereby generating a total of $(k+1)^n$ 1-inputs.

Step 3: *The number of 1-inputs in a monochromatic box is at most k^n.* Let $B = A_1 \times A_2 \times \cdots \times A_k$ be a 1-box. The key claim here is: for each coordinate $\ell = 1, \ldots, n$, there is a player $i \in \{1, \ldots, k\}$ such that, for every input $\mathbf{x}_i \in A_i$, the ℓth coordinate of $\mathbf{x}_i$ is 0. That is, to each coordinate we can associate an "ineligible player" that, in this box, never has a 1 in that coordinate. This is easily seen by contradiction: otherwise, there exists a coordinate ℓ such that, for every player i, there is an input $\mathbf{x}_i \in A_i$ with a 1 in the ℓth coordinate. As a box, B contains the input $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$. But this is a 0-input, contradicting the assumption that B is a 1-box.

The claim implies the stated upper bound. Every 1-input of B can be generated by choosing, for each coordinate ℓ, an assignment of at most one "1" in this coordinate to one of the $k-1$ eligible players for this coordinate. With only k choices per coordinate, there are at most k^n 1-inputs in the box B.

Conclusion: Steps 2 and 3 imply that covering the 1s of the k-dimensional array of the Multi-Disjointness function requires at least $(1+\frac{1}{k})^n$ 1-boxes. By the discussion in Step 1, this implies a lower bound of $n \log_2(1+\frac{1}{k}) = \Theta(n/k)$ on the nondeterministic communication complexity of the Multi-Disjointness function (and output 1). This concludes the proof of Theorem 2.2.

Lunar Lecture 3

Why Prices Need Algorithms

Lecturer: Tim Roughgarden *Scribe: Sumegha Garg & Joshua R. Wang*

You've probably heard about "market-clearing prices," which equate the supply and demand in a market. When are such prices guaranteed to exist? In the classical setting with divisible goods (milk, wheat, etc.), market-clearing prices exist under reasonably weak conditions [6]. But with indivisible goods (houses, spectrum licenses, etc.), such prices may or may not exist. As you can imagine, many papers in the economics and operations research literatures study necessary and sufficient conditions for existence. The punchline of today's lecture, based on joint work with Inbal Talgam-Cohen [131], is that computational complexity considerations in large part govern whether or not market-clearing prices exist in a market of indivisible goods. This is cool and surprising because the question (existence of equilibria) seems to have nothing to do with computation (cf., the questions studied in the Solar Lectures).

3.1 Markets with Indivisible Items

The basic setup is the same as in the preceding lecture, when we were studying price-of-anarchy bounds for simple combinatorial auctions (Section 2.1). To review, there are k players, a set M of m items, and each player i has a valuation $v_i : 2^M \to \mathbb{R}_+$ describing its maximum willingness to pay for each bundle of items. For simplicity, we also assume that $v_i(\emptyset) = 0$ and that v_i is monotone (with $v_i(S) \leq v_i(T)$ whenever $S \subseteq T$). As in last lecture, we will often vary the class $\mathcal{V}$ of allowable valuations to make the setting more or less complex.

3.1.1 Walrasian Equilibria

Next is the standard definition of "market-clearing prices" in a market with multiple indivisible items.

Definition 3.1 (Walrasian Equilibrium). A *Walrasian equilibrium* is an allocation $S_1, \ldots, S_k$ of the items of M to the players and nonnegative prices $p_1, p_2, ..., p_m$ for the items such that:

(W1) All buyers are as happy as possible with their respective allocations, given the prices: for every $i = 1, 2, \ldots, k$, $S_i \in \text{argmax}_T (v_i(T) - \sum_{j \in T} p_j)$.

(W2) Feasibility: $S_i \cap S_j = \emptyset$ for $i \neq j$.

(W3) The market clears: for every $j \in M$, $j \in S_i$ for some i.[1]

[1]The most common definition of a Walrasian equilibrium asserts instead that an item j is not awarded to any player only if $p_j = 0$. With monotone valuations, there is no harm in insisting that every item is allocated.

Note that S_i might be the empty set, if the prices are high enough for (W1) to hold for player i. Also, property (P3) is crucial for the definition to be non-trivial (otherwise set $p_j = +\infty$ for every j).

Walrasian equilibria are remarkable: even though each player optimizes independently (modulo tie-breaking) and gets exactly what it wants, somehow the global feasibility constraint is respected.

3.1.2 The First Welfare Theorem

Recall from last lecture that the *social welfare* of an allocation $S_1, \ldots, S_k$ is defined as $\sum_{i=1}^{k} v_i(S)$. Walrasian equilibria automatically maximize the social welfare, a result known as the "First Welfare Theorem."

Theorem 3.2 (First Welfare Theorem). *If the prices $p_1, p_2, \ldots, p_m$ and allocation $S_1, S_2, \ldots, S_k$ of items constitute a Walrasian equilibrium, then*

$$(S_1, S_2, ..., S_k) \in \text{argmax}_{(T_1, T_2, \ldots, T_k)} \sum_{i=1}^{k} v_i(T_i),$$

where $(T_1, \ldots, T_k)$ ranges over all feasible allocations (with $T_i \cap T_j = \emptyset$ for $i \neq j$).

If one thinks of a Walrasian equilibrium as the natural outcome of a market, then Theorem 3.2 can be interpreted as saying "markets are efficient."[2] There are many "First Welfare Theorems," and all have this flavor.

Proof. Let $(S_1^*, \ldots, S_k^*)$ denote a welfare-maximizing feasible allocation. We can apply property (W1) of Walrasian equilibria to obtain

$$v_i(S_i) - \sum_{j \in S_i} p_j \geq v_i(S_i^*) - \sum_{j \in S_i^*} p_j$$

for each player $i = 1, 2, \ldots, k$. Summing over i, we have

$$\sum_{i=1}^{k} v_i(S_i) - \sum_{i=1}^{k} \left(\sum_{j \in S_i} p_j \right) \geq \sum_{i=1}^{k} v_i(S_i^*) - \sum_{i=1}^{k} \left(\sum_{j \in S_i^*} p_j \right). \tag{3.1}$$

Properties (W2) and (W3) imply that the second term on the left-hand side of (3.1) equals the sum $\sum_{j=1}^{m} p_j$ of all the item prices. Since $(S_1^*, \ldots, S_n^*)$ is a feasible allocation, each item is awarded at most once and hence the second term on the right-hand side is at most $\sum_{j=1}^{m} p_j$. Adding $\sum_{j=1}^{m} p_j$ to both sides gives

$$\sum_{i=1}^{k} v_i(S_i) \geq \sum_{i=1}^{k} v_i(S_i^*),$$

which proves that $(S_1, \ldots, S_k)$ is also welfare-maximizing. □

3.1.3 Existence of Walrasian Equilibria

The First Welfare Theorem says that Walrasian equilibria are great when they exist. But when do they exist?

[2] Needless to say, much blood and ink has been spilled over this interpretation over the past couple of centuries.

Example 3.3. Suppose M contains only one item. Consider the allocation that awards the item to the player i with the highest value for it, and a price that is between the i's value and the highest value of some other player (the second-highest overall). This is a Walrasian equilibrium: the price is low enough that bidder i prefers receiving the item to receiving nothing, and high enough that all the other bidders prefer the opposite. A simple case analysis shows that these are all of the Walrasian equilibria.

Example 3.4. Consider a market with two items, A and B. Suppose the valuation of the first player is

$$v_1(T) = \begin{cases} 3 & \text{for } T = \{A, B\} \\ 0 & \text{otherwise} \end{cases}$$

and that of the second player is

$$v_2(T) = \begin{cases} 2 & \text{when } T \text{ is } A \text{ or } B \\ 0 & \text{otherwise.} \end{cases}$$

The first bidder is called a "single-minded" or "AND" bidder, who is only happy if she gets both items. The second bidder is called a "unit-demand" or "OR" bidder, and only wants one of the items.

We claim that there is no Walrasian equilibrium in this market. From the First Welfare Theorem, we know what such an equilibrium must allocate the items to maximize the social welfare, which in this case means awarding both items to the first player. For the second player to be happy getting neither item, the price of each item must be at least 2. But then the first player pays 4 and has negative utility, and would prefer to receive nothing.

These examples suggest a natural question: under what conditions is a Walrasian equilibrium guaranteed to exist? There is a well-known literature on this question in economics (e.g. [87, 69, 106]); here are the highlights.

1. If every player's valuation v_i satisfies the "gross substitutes (GS)" condition, then a Walrasian equilibrium is guaranteed to exist. We won't need the precise definition of the GS condition in this lecture. GS valuations are closely related to weighted matroid rank functions, and hence are a subclass of the submodular valuations defined at the end of last lecture in Section 2.4.[3] A unit-demand (a.k.a. "OR") valuation, like that of the second player in Example 3.4, satisfies the GS condition (corresponding to the 1-uniform matroid). It follows that single-minded (a.k.a. "AND") valuations, like that of the first player in Example 3.4, do not in general satisfy the GS condition (otherwise the market in Example 3.4 would have a Walrasian equilibrium).

2. If $\mathcal{V}$ is a class of valuations that contains all unit-demand valuations and also some valuation that violates the GS condition, then there is a market with valuations in $\mathcal{V}$ that does not possess a Walrasian equilibrium.

These results imply that GS valuations are a maximal class of valuations subject to the guaranteed existence of Walrasian equilibria. These results do, however, leave open the possibility of guaranteed existence for classes $\mathcal{V}$ that contain non-GS valuations but not all unit-demand valuations, and a number of recent papers in economics and operations research have pursued this direction (e.g. [11, 24, 25, 142]). All of the non-existence results in this line of work use explicit constructions, like in Example 3.4.

[3] A weighted matroid rank function f is defined using a matroid $(E, \mathcal{I})$ and nonnegative weights on the elements E, with $f(S)$ defined as the maximum weight of an independent set (i.e., a member of $\mathcal{I}$) that lies entirely in S.

3.2 Complexity Separations Imply Non-Existence of Walrasian Equilibria

3.2.1 Statement of Main Result

Next we describe a completely different approach to ruling out the existence of Walrasian equilibria, based on complexity theory rather than explicit constructions. The main result is the following.

Theorem 3.5 (Roughgarden and Talgam-Cohen [131]). *Let $\mathcal{V}$ denote a class of valuations. Suppose the welfare-maximization problem for $\mathcal{V}$ does not reduce to the utility-maximization problem for $\mathcal{V}$. Then, there exists a market with all player valuations in $\mathcal{V}$ that has no Walrasian equilibrium.*

In other words, a necessary condition for the guaranteed existence of Walrasian equilibria is that welfare-maximization is no harder than utility-maximization. This connects a purely economic question (when do equilibria exist?) to a purely algorithmic one.

To fill in some of the details in the statement of Theorem 3.5, by "does not reduce to," we mean that there is no polynomial-time Turing reduction from the former problem to the latter. By "the welfare-maximization problem for $\mathcal{V}$," we mean the problem of, given player valuations $v_1, \ldots, v_k \in \mathcal{V}$, computing an allocation that maximizes the social welfare $\sum_{i=1}^{k} v_i(S_i)$.[4] By "the utility-maximization problem for $\mathcal{V}$," we mean the problem of, given a valuation $v \in \mathcal{V}$ and nonnegative prices $p_1, \ldots, p_m$, computing a utility-maximizing bundle $S \in \text{argmax}_{T \subseteq M} v(T) - \sum_{j \in T} p_j$.

The utility-maximization problem, which involves only one player, can generally only be easier than the multi-player welfare-maximization problem. Thus the two problems either have the same computational complexity, or welfare-maximization is strictly harder. Theorem 3.5 asserts that whenever the second case holds, Walrasian equilibria need not exist.

3.2.2 Examples

Before proving Theorem 3.5, let's see how to apply it. For most natural valuation classes $\mathcal{V}$, a properly trained theoretical computer scientist can identify the complexity of the utility- and welfare-maximization problems in a matter of minutes.

Example 3.6 (AND Valuations). Let $\mathcal{V}_m$ denote the class of "AND" valuations for markets where $|M| = m$. That is, each $v \in \mathcal{V}_m$ has the following form, for some $\alpha \geq 0$ and $T \subseteq M$:

$$v(S) = \begin{cases} \alpha & \text{if } S \supseteq T \\ 0 & \text{otherwise.} \end{cases}$$

The utility-maximization problem for $\mathcal{V}_m$ is trivial: for a single player with an AND valuation with parameters α and T, the better of $\emptyset$ or T is a utility-maximizing bundle. The welfare-maximization problem for $\mathcal{V}_m$ is essentially set packing and is NP-hard (with $m \to \infty$).[5] We conclude that the welfare-maximization problem for $\mathcal{V}$ does not reduce to the utility-maximization problem for $\mathcal{V}$ (unless P = NP). Theorem 3.5 then implies that, assuming P ≠ NP, there are markets with AND valuations that do not have any Walrasian equilibria.[6]

[4] For concreteness, think about the case where every valuation v_i has a succinct description and can be evaluated in polynomial time. Analogous results hold when an algorithm has only oracle access to the valuations.

[5] For example, given an instance $G = (V, E)$ of the Independent Set problem, take $M = E$, make one player for each vertex $i \in V$, and give player i an AND valuation with parameters $\alpha = 1$ and T equal to the edges that are incident to i in G.

[6] It probably seems weird to have a conditional result ruling out equilibrium existence. A conditional non-existence result can of course be made unconditional through an explicit example. A proof that the welfare-maximization problem for $\mathcal{V}$ is NP-hard will

Of course, Example 3.4 already shows, without any complexity assumptions, that markets with AND bidders do not generally have Walrasian equilibria.[7] Our next example addresses a class of valuations for which the status of Walrasian equilibria was not previously known.

Example 3.7 (Capped Additive Valuations)**.** A *capped additive* valuation v is parameterized by $m + 1$ numbers $c, \alpha_1, \alpha_2, \ldots, \alpha_m$ and is defined as

$$v(S) = \min \left\{ c, \sum_{j \in S} \alpha_j \right\}.$$

The α_j's indicate each item's value, and c the "cap" on the maximum value that can be attained. Capped additive valuations were proposed in Lehmann et al. [95] as a natural subclass of submodular valuations, and have been studied previously from a welfare-maximization standpoint.

Let $\mathcal{V}_{m,d}$ denote the class of capped additive valuations in markets with $|M| = m$ and with c and $\alpha_1, \ldots, \alpha_m$ restricted to be positive integers between 1 and m^d. (Think of d as fixed and $m \to \infty$.) A Knapsack-type dynamic programming algorithm shows that the utility-maximization problem for $\mathcal{V}_{m,d}$ can be solved in polynomial time (using that c and the α_j's are polynomially bounded). For d a sufficiently large constant, however, the welfare-maximization problem for $\mathcal{V}_{m,d}$ is NP-hard (it includes the strongly NP-hard Bin Packing problem). Theorem 3.5 then implies that, assuming P $\neq$ NP, there are markets with valuations in $\mathcal{V}_{m,d}$ with no Walrasian equilibrium.

3.3 Proof of Theorem 3.5

3.3.1 The Plan

Here's the plan for proving Theorem 3.5. Fix a class $\mathcal{V}$ of valuations, and assume that a Walrasian equilibrium exists in every market with player valuations in $\mathcal{V}$. We will show, in two steps, that the welfare-maximization problem for $\mathcal{V}$ (polynomial-time Turing) reduces to the utility-maximization problem for $\mathcal{V}$.

Step 1: The "fractional" version of the welfare-maximization problem for $\mathcal{V}$ reduces to the utility-maximization problem for $\mathcal{V}$.

Step 2: A market admits a Walrasian equilibrium if and only if the fractional welfare-maximization problem has an optimal integral solution. (We'll only need the "only if" direction.)

Since every market with valuations in $\mathcal{V}$ admits a Walrasian equilibrium (by assumption), these two steps imply that the integral welfare-maximization problem reduces to utility-maximization.

---

generally suggest candidate markets to check for non-existence.

The following analogy may help: consider computationally tractable linear programming relaxations of NP-hard optimization problems. Conditional on P $\neq$ NP, such relaxations cannot be exact (i.e., have no integrality gap) for all instances. NP-hardness proofs generally suggest instances that can be used to prove directly (and unconditionally) that a particular linear programming relaxation has an integrality gap.

[7] Replacing the OR bidder in Example 3.4 with an appropriate pair of AND bidders extends the example to markets with only AND bidders.

3.3.2 Step 1: Fractional Welfare-Maximization Reduces to Utility-Maximization

This step is folklore, and appears for example in Nisan and Segal [116]. Consider the following linear program (often called the *configuration LP*), with one variable x_{iS} for each player i and bundle $S \subseteq 2^M$:

$$\begin{aligned} \max & \sum_{i=1}^{k} \sum_{S \subseteq M} v_i(S) x_{iS} \\ \text{s.t.} & \sum_{i=1}^{k} \sum_{S \subseteq M \,:\, j \in S} x_{iS} \leq 1 \qquad \text{for } j = 1, 2, \ldots, m \\ & \sum_{S \subseteq M} x_{iS} = 1 \qquad \text{for } i = 1, 2, \ldots, k. \end{aligned}$$

The intended semantics are

$$x_{iS} = \begin{cases} 1 & \text{if } i \text{ gets the bundle } S \\ 0 & \text{otherwise.} \end{cases}$$

The first set of constraints enforces that each item is awarded only once (perhaps fractionally), and the second set enforces that every player receives one bundle (perhaps fractionally). Every feasible allocation induces a 0-1 feasible solution to this linear program according to the intended semantics, and the objective function value of this solution is exactly the social welfare of the allocation.

This linear program has an exponential (in m) number of variables. The good news is that it has only a polynomial number of constraints. This means that the dual linear program will have a polynomial number of variables and an exponential number of constraints, which is right in the wheelhouse of the ellipsoid method.

Precisely, the dual linear program is:

$$\begin{aligned} \min & \sum_{i=1}^{k} u_i + \sum_{j=1}^{m} p_j \\ \text{s.t.} \quad & u_i + \sum_{j \in S} p_j \geq v_i(S) \qquad \text{for all } i = 1, 2, \ldots, k \text{ and } S \subseteq M \\ & p_j \geq 0 \qquad \text{for } j = 1, 2, \ldots, m, \end{aligned}$$

where u_i and p_j corresponds to the primal constraints that bidder i receives one bundle and item j is allocated at most once, respectively.

Recall that the ellipsoid method [88] can solve a linear program in time polynomial in the number of variables, as long as there is a polynomial-time *separation oracle* that can verify whether or not a given point is feasible and, if not, produce a violated constraint. For the dual linear program above, this separation oracle boils down to solving the following problem: for each player $i = 1, 2, \ldots, k$, check that

$$u_i \geq \max_{S \subseteq M} \left[v_i(S) - \sum_{j \in S} p_j \right].$$

But this reduces immediately to the utility-maximization problem for $\mathcal{V}$! Thus the ellipsoid method can be used to solve the dual linear program to optimality, using a polynomial number of calls to a utility-maximization oracle. The optimal solution to the original fractional welfare-maximization problem can then be efficiently extracted from the optimal dual solution.[8]

3.3.3 Step 2: Walrasian Equilibria and Exact Linear Relaxations

We now proceed with the second step, which is based on Bikhchandani and Mamer [12] and follows from strong linear programming duality. Recall from linear programming theory (see e.g. [38]) that a pair of primal and dual feasible solutions are both optimal if and only if the "complementary slackness" conditions hold.[9] These conditions assert that every non-zero decision variable in one of the linear programs corresponds to a tight constraint in the other. For our primal-dual pair of linear programs, these conditions are:

(i) $x_{iS} > 0$ implies that $u_i = v_i(S) - \sum_{j \in S} p_j$ (i.e., only utility-maximizing bundles are used);

(ii) $p_j > 0$ implies that $\sum_i \sum_{S: j \in S} x_{iS} = 1$ (i.e., item j is not fully sold only if it is worthless).

Comparing the definition of Walrasian equilibria (Definition 3.1) with conditions (i) and (ii), we see that a 0-1 primal feasible solution $\mathbf{x}$ (corresponding to an allocation) and a dual solution $\mathbf{p}$ (corresponding to item prices) constitute a Walrasian equilibrium if and only if the complementary slackness conditions hold (where u_i is understood to be set to $\max_{S \subseteq M} v_i(S) - \sum_{j \in S} p_j$). Thus a Walrasian equilibrium exists if and only if there is a feasible 0-1 solution to the primal linear program and a feasible solution to the dual linear problem that satisfy the complementary slackness conditions, which in turn holds if and only if the primal linear program has an optimal 0-1 feasible solution.[10] We conclude that a Walrasian equilibrium exists if and only if the fractional welfare-maximization problem has an optimal integral solution.[11]

3.4 Beyond Walrasian Equilibria

For valuation classes $\mathcal{V}$ that do not always possess Walrasian equilibria, is it possible to define a more general notion of "market-clearing prices" so that existence is guaranteed? For example, what if we use prices that are more complex than item prices? This section shows that complexity considerations provide an explanation of why interesting generalizations of Walrasian equilibria have been so hard to come by.

Consider a class $\mathcal{V}$ of valuations, and a class $\mathcal{P}$ of *pricing functions*. A pricing function, just like a valuation, is a function $p : 2^M \to \mathbb{R}_+$ from bundles to nonnegative numbers. The item prices $p_1, \ldots, p_m$ used to define Walrasian equilibria correspond to additive pricing functions, with $p(S) = \sum_{j \in S} p_j$. The next definition articulates the appropriate generalization of Walrasian equilibria to more general classes of pricing functions.

[8] In more detail, consider the (polynomial number of) dual constraints generated by the ellipsoid method when solving the dual linear program. Form a reduced version of the original primal problem, retaining only the (polynomial number of) variables that correspond to this subset of dual constraints. Solve this polynomial-size reduced version of the primal linear program using your favorite polynomial-time linear programming algorithm.

[9] If you've never seen or have forgotten about complementary slackness, there's no need to be afraid. To derive them, just write down the usual proof of weak LP duality (which is a chain of inequalities), and back out the conditions under which all the inequalities hold with equality.

[10] This argument re-proves the First Welfare Theorem (Theorem 3.2). It also proves the Second Welfare Theorem, which states that for every welfare-maximizing allocation, there exist prices that render it a Walrasian equilibrium—any optimal solution to the dual linear program furnishes such prices.

[11] Recall our analogy with integrality gaps of linear programs (footnote 6). NP-hardness implies an integrality gap (assuming P $\neq$ NP), and we now see that integrality gaps and non-existence of Walrasian equilibria go together.

Definition 3.8 (Price Equilibrium)**.** A *price equilibrium* (w.r.t. pricing functions $\mathcal{P}$) is an allocation $S_1, \ldots, S_k$ of the items of M to the players and a pricing function $p \in \mathcal{P}$ such that:

(P1) All buyers are as happy as possible with their respective allocations, given the prices: for every $i = 1, 2, \ldots, k$, $S_i \in \text{argmax}_T (v_i(T) - p(T))$.

(P2) Feasibility: $S_i \cap S_j = \emptyset$ for $i \neq j$.

(P3) Revenue maximizing, given the prices: $(S_1, S_2, ..., S_k) \in \text{argmax}_{(T_1, T_2, \ldots, T_k)} \sum_{i=1}^{k} p(T_i)$.

Condition (P3) is the analog of the market-clearing condition (W3) in Definition 3.1. It is not enough to assert that all items are sold, because with a general pricing function, different ways of selling all of the items can lead to different amounts of revenue. Under conditions (P1)–(P3), the First Welfare Theorem (Theorem 3.2) still holds, with essentially the same proof, and so every price equilibrium maximizes the social welfare.

For which choices of valuations $\mathcal{V}$ and pricing functions $\mathcal{P}$ is Definition 3.8 interesting? Ideally, the following properties should hold.

1. Guaranteed existence: for every set M of items and valuations $v_1, \ldots, v_k \in \mathcal{V}$, there exists a price equilibrium with respect to $\mathcal{P}$.

2. Efficient recognition: there is a polynomial-time algorithm for checking whether or not a given allocation and pricing function constitute a price equilibrium. This boils down to assuming that utility-maximization (with respect to $\mathcal{V}$ and $\mathcal{P}$) and revenue-maximization (with respect to $\mathcal{P}$) are polynomial-time solvable problems (to check (W1) and (W3), respectively).[12]

3. Markets with valuations in $\mathcal{V}$ do not always have a Walrasian equilibrium. (Otherwise, why bother generalizing item prices?)

We can now see why there are no known natural choices of $\mathcal{V}$ and $\mathcal{P}$ that meet these three requirements. The first two requirements imply that the welfare-maximization problem belongs to $\mathsf{NP} \cap \mathsf{co\text{-}NP}$. To certify a lower bound of W^* on the maximum social welfare, one can exhibit an allocation with social welfare at least W^*. To certify an upper bound of W^*, one can exhibit a price equilibrium that has welfare at most W^*—this is well defined by the first condition, efficiently verifiable by the second condition, and correct by the First Welfare Theorem.

Problems in $(\mathsf{NP} \cap \mathsf{co\text{-}NP}) \setminus \mathsf{P}$ appear to be rare, especially in combinatorial optimization. The preceding paragraph gives a heuristic argument that interesting generalizations of Walrasian equilibria are possible only for valuation classes for which welfare-maximization is polynomial-time solvable. For every natural such class known, the linear programming relaxation in Section 3.3 has an optimal integral solution; in this sense, solving the configuration LP appears to be a "universal algorithm" for polynomial-time welfare-maximization. But the third requirement asserts that a Walrasian equilibrium does not always exist in markets with valuations in $\mathcal{V}$ and so, by the second step of the proof of Theorem 3.5 (in Section 3.3.3), there are markets for which the configuration LP sometimes has only fractional optimal solutions.

The upshot is that interesting generalizations of Walrasian equilibria appear possible only for valuation classes where a non-standard algorithm is necessary and sufficient to solve the welfare-maximization problem in polynomial time. It is not clear if there are any natural valuation classes for which this algorithmic barrier can be overcome.[13]

[12] One could require the stronger property that a price equilibrium can be computed (not just verified) in polynomial time. Using the weaker requirement makes our negative results stronger.

[13] See [131, Section 5.3.2] for an unnatural such class.

Lunar Lecture 4

The Borders of Border's Theorem

Lecturer: Tim Roughgarden *Scribe: Cristopher Moore*

Border's theorem [16] is a famous result in auction theory about the design space of single-item auctions, and it provides an explicit linear description of the single-item auctions that are "feasible" in a certain sense. Despite the theorem's fame, there have been few generalizations of it. This lecture, based on joint work with Parikshit Gopalan and Noam Nisan [68], uses complexity theory to explain why: if there *were* significant generalizations of Border's theorem, the polynomial hierarchy would collapse!

4.1 Optimal Single-Item Auctions

4.1.1 The Basics of Single-Item Auctions

Single-item auctions have made brief appearances in previous lectures; let's now study the classic model, due to Vickrey [146], in earnest. There is a single seller of a single item. There are n bidders, and each bidder i has a valuation v_i for the item (its maximum willingness to pay). Valuations are *private*, meaning that v_i is known a priori to bidder i but not to the seller or the other bidders. Each bidder wants to maximize the value obtained from the auction (v_i if it wins, 0 otherwise) minus the price it has to pay. In the presence of randomization (either in the input or internal to the auction), we assume that bidders are risk-neutral, meaning they act to maximize their expected utility.

This lecture is our only one on the classical *Bayesian* model of auctions, which can be viewed as a form of average-case analysis. The key assumption is that each valuation v_i is drawn from a distribution F_i that is known to the seller and possibly the other bidders. The actual realization v_i remains unknown to everybody other than bidder i. For simplicity we'll work with discrete distributions, and let V_i denote the support of F_i and $f_i(v_i)$ the probability that bidder i's valuation is $v_i \in V_i$. Typical examples include (discretized versions of) the uniform distribution, the lognormal distribution, the exponential distribution, and power-law distributions. We also assume that bidders' valuations are stochastically independent.

When an economist speaks of an "optimal auction," they usually mean the auction that maximizes the seller's expected revenue with respect to a known prior distribution.[1] Before identifying optimal auctions, we need to formally define the design space. The auction designer needs to decide who wins and how much they pay. Thus they must define two (possibly randomized) functions of the bid vector $\vec{b}$: an *allocation*

[1] One advantage of assuming a distribution over inputs is that there is an unequivocal way to compare the performance of different auctions (by their expected revenues), and hence an unequivocal way to define an optimal auction. One auction generally earns more revenue than another on some inputs and less on others, so in the absence of a prior distribution, it's not clear which one to prefer.

rule $\vec{x}(\vec{b})$ which determines which bidder wins the item, where $x_i = 1$ and if i wins and $x_i = 0$ otherwise, and a *payment rule* $\vec{p}(\vec{b})$ where p_i is how much i pays. We impose the constraint that whenever bidder i bids b_i, the expected payment $\mathbf{E}\left[p_i(\vec{b})\right]$ of the bidder is at most b_i times the probability $x_i(\vec{b})$ that it wins. (The randomization is over the bids by the other bidders and any randomness internal to the auction.) This participation constraint ensures that a bidder who does not overbid will obtain nonnegative expected utility from the auction. (Without it, an auction could just charge $+\infty$ to every bidder.) The *revenue* of an auction on the bid vector $\vec{b}$ is $\sum_{i=1}^{n} p_i(\vec{b})$.

For example, in the *Vickrey* or *second-price auction*, the allocation rule awards the item to the highest bidder, and the payment rule charges the second-highest bid. This auction is *truthful*, meaning that for each bidder, truthful bidding (i.e., setting $b_i = v_i$) is a *dominant strategy* that maximizes its utility no matter what the other bidders do. With a truthful auction, there is no need to assume that the distributions $F_1, \ldots, F_n$ are known to the bidders. The beauty of the Vickrey auction is that it delegates underbidding to the auctioneer, who determines the optimal bid for the winner on their behalf.

A *first-price auction* has the same allocation rule as a second-price auction (give the item to the highest bidder), but the payment rule charges the winner its bid. Bidding truthfully in a first-price auction guarantees zero utility, so strategic bidders will underbid. Because bidders do not have dominant strategies—the optimal amount to underbid depends on the bids of the others—it is non-trivial to reason about the outcome of first-price auctions. The traditional solution is to assume that the distributions $F_1, \ldots, F_n$ are known in advance to the bidders, and to consider Bayes-Nash equilibria. Formally, a *strategy* of a bidder i in a first-price auction is a predetermined plan for bidding—a function $b_i(\cdot)$ that maps its valuation v_i to a bid $b_i(v_i)$ (or a distribution over bids). The semantics are: "when my valuation is v_i, I will bid $b_i(v_i)$." We assume that bidders' strategies are common knowledge, with bidders' valuations (and hence induced bids) private as usual. A strategy profile $b_1(\cdot), \cdots, b_n(\cdot)$ is a *Bayes-Nash equilibrium* if every bidder always bids optimally given its information—if for every bidder i and every valuation v_i, the bid $b_i(v_i)$ maximizes i's expected utility, where the expectation is with respect to the distribution over the bids of other bidders induced by $F_1, \ldots, F_n$ and their bidding strategies.[2] Note that the set of Bayes-Nash equilibria of an auction generally depends on the prior distributions $F_1, \ldots, F_n$.

An auction is called *Bayesian incentive compatible (BIC)* if truthful bidding (with $b_i(v_i) = v_i$ for all i and v_i) is a Bayes-Nash equilibrium. That is, as a bidder, if all other bidders bid truthfully, then you also want to bid truthfully. A second-price auction is BIC, while a first-price auction is not.[3] However, for every choice of $F_1, \ldots, F_n$, there is a BIC auction that is equivalent to the first-price auction. Specifically: given bids $a_1, \ldots, a_n$, implement the outcome of the first-price auction with bids $b_1(a_1), \ldots, b_n(a_n)$, where $b_1(\cdot), \ldots, b_n(\cdot)$ denotes a Bayes-Nash equilibrium of the first-price auction (with prior distributions $F_1, \ldots, F_n$). Intuitively, this auction makes the following pact with each bidder: "you promise to tell me your true valuation, and I promise to bid on your behalf as you would in a Bayes-Nash equilibrium." More generally, this simulation argument shows that for *every* auction A, distributions $F_1, \ldots, F_n$, and Bayes-Nash equilibrium of A (w.r.t. $F_1, \ldots, F_n$), there is a BIC auction A' whose (truthful) outcome (and hence expected revenue) matches that of the chosen Bayes-Nash equilibrium of A. This result is known as the *Revelation Principle*. This principle implies that, to identify an optimal auction, there is no loss of generality in restricting to BIC auctions.[4]

[2] Straightforward exercise: if there are n bidders with valuations drawn i.i.d. from the uniform distribution on $[0, 1]$, then setting $b_i(v_i) = \frac{n-1}{n} \cdot v_i$ for every i and v_i yields a Bayes-Nash equilibrium.

[3] The second-price auction is in fact *dominant-strategy incentive compatible (DSIC)*—truthful bidding is a dominant strategy for every bidder, not merely a Bayes-Nash equilibrium.

[4] Of course, non-BIC auctions like first-price auctions are still useful in practice. For example, the description of the first-price auction does not depend on bidders' valuation distributions $F_1, \ldots, F_n$ and can be deployed without knowledge of them. This is not

4.1.2 Optimal Auctions

In optimal auction design, the goal is to identify an expected revenue-maximizing auction, as a function of the prior distributions $F_1, \ldots, F_n$. For example, suppose that $n = 1$, and we restrict attention to truthful auctions. The only truthful auctions are take-it-or-leave-it offers (or a randomization over such offers). That is, the selling price must be independent of the bidder's bid, as any dependence would result in opportunities for the bidder to game the auction. The optimal truthful auction is then the take-it-or-leave-it offer at the price r that maximizes

$$\underbrace{r}_{\text{revenue of a sale}} \cdot \underbrace{(1 - F(r))}_{\text{probability of a sale}},$$

where F denotes the bidder's valuation distribution. Given a distribution F, it is usually a simple matter to solve for the best r. An optimal offer price is called a *monopoly price* of the distribution F. For example, if F is the uniform distribution on $[0, 1]$, then the monopoly price is $\frac{1}{2}$.

Myerson [111] gave a complete solution to the optimal single-item auction design problem, in the form of a generic compiler that takes as input prior distributions $F_1, \ldots, F_n$ and outputs a closed-form description of the optimal auction for $F_1, \ldots, F_n$. The optimal auction is particularly easy to interpret in the symmetric case, when bidders' valuations are drawn i.i.d. from a common distribution F. Here, the optimal auction is just a second-price auction with a reserve price r equal to the monopoly price of F (i.e., an eBay auction with a suitably chosen opening bid).[5,6] For example, with any number n of bidders with valuations drawn i.i.d. from the uniform distribution on $[0, 1]$, the optimal single-item auction is a second-price auction with a reserve price of $\frac{1}{2}$. This is a pretty amazing confluence of theory and practice—we optimized over the space of all imaginable auctions (which includes some very strange specimens), and discovered that the theoretically optimal auction format is one that is already in widespread use![7]

Myerson's theory of optimal auctions extends to the asymmetric case where bidders have different distributions (where the optimal auction is no longer so simple), and also well beyond single-item auctions.[8] The books by Hartline [72] and your lecturer [129, Lectures 3 and 5] describe this theory from a computer science perspective.

4.2 Border's Theorem

4.2.1 Context

Border's theorem identifies a tractable description of *all* BIC single-item auctions, in the form of a polytope in polynomially many variables. (See Section 4.1.1 for the definition of a BIC auction.) This goal is in some sense more ambitious than merely identifying the optimal auction, since with this tractable description in hand, one can efficiently compute the optimal auction for any given set $F_1, \ldots, F_n$ of prior distributions.

Economists are interested in Border's theorem because it can be used to extend the reach of Myerson's optimal auction theory (Section 4.1.2) to more general settings, such as the case of risk-adverse bidders

the case for the simulating auction.

[5]The winner is the highest bidder who clears the reserve (if any). The winner (if any) pays either the reserve price or the second-highest bid, whichever is higher.

[6]Technically, this requires a mild "regularity" condition on the distribution F, which holds for all of the most common parametric distributions.

[7]In particular, there is always an optimal auction in which truthful bidding is a dominant strategy (as opposed to merely being a BIC auction). This is also true in the asymmetric case.

[8]The theory applies more generally to "single-parameter problems." These include problems where in each outcome a bidder is either a "winner" or a "loser," with private valuation v_i for winning and 0 for losing (and with multiple winners allowed).

studied by Maskin and Riley [101]. Matthews [102] conjectured the precise result that was proved by Border [16]. Computer scientists have used Border's theorem for orthogonal extensions to Myerson's theory, like computationally tractable descriptions of the expected-revenue maximizing auction in settings with multiple non-identical items [3, 21]. While there is no hope of deriving a closed-form solution to the optimal auction design problem with risk-adverse bidders or with multiple items, Border's theorem at least enables an efficient algorithm for computing a description of an optimal auction (given descriptions of the prior distributions).

4.2.2 An Exponential-Size Linear Program

As a lead-in to Border's theorem, we show how to formulate the space of BIC single-item auctions as an (extremely big) linear program. The decision variables of the linear program encode the allocation and payment rules of the auction (assuming truthful bidding, as appropriate for BIC auctions). There is one variable $x_i(\vec{v}) \in [0,1]$ that describes the probability (over any randomization in the auction) that bidder i wins the item when bidders' valuations (and hence bids) are $\vec{v}$. Similarly, $p_i(\vec{v}) \in \mathbb{R}_+$ denotes the expected payment made by bidder i when bidders' valuations are $\vec{v}$.

Before describing the linear program, we need some odd but useful notation (which is standard in game theory and microeconomics).

Some Notation

For an n-vector $\vec{z}$ and a coordinate $i \in [n]$, let $\vec{z}_{-i}$ denote the $(n-1)$-vector obtained by removing the ith component from $\vec{z}$. We also identify $(z_i, \vec{z}_{-i})$ with $\vec{z}$.

Also, recall that V_i denotes the possible valuations of bidder i, and that we assume that this set is finite.

Our linear program will have three sets of constraints. The first set enforces the property that truthful bidding is in fact a Bayes-Nash equilibrium (as required for a BIC mechanism). For every bidder i, possible valuation $v_i \in V_i$ for i, and possible false bid $v_i' \in V_i$,

$$\underbrace{v_i \cdot \mathbf{E}_{\vec{v}_{-i} \sim \vec{F}_{-i}}[x_i(\vec{v})] - \mathbf{E}_{\vec{v}_{-i} \sim \vec{F}_{-i}}[p_i(\vec{v})]}_{\text{expected utility of truthful bid } v_i} \geq \underbrace{v_i \cdot \mathbf{E}_{\vec{v}_{-i} \sim \vec{F}_{-i}}\left[x_i(v_i', \vec{v}_{-i})\right] - \mathbf{E}_{\vec{v}_{-i} \sim \vec{F}_{-i}}\left[p_i(v_i', \vec{v}_{-i})\right]}_{\text{expected utility of false bid } v_i'}. \tag{4.1}$$

The expectation is over both the randomness in $\vec{v}_{-i}$ and internal to the auction. Each of the expectations in (4.1) expands to a sum over all possible $\vec{v}_{-i} \in \vec{V}_{-i}$, weighted by the probability $\prod_{j \neq i} f_j(v_j)$. Since all of the $f_j(v_j)$'s are numbers known in advance, each of these constraints is linear (in the $x_i(\vec{v})$'s and $p_i(\vec{v})$'s).

The second set of constraints encode the participation constraints from Section 4.1.1, also known as the *interim individually rational (IIR)* constraints. For every bidder i and possible valuation $v_i \in V_i$,

$$v_i \cdot \mathbf{E}_{\vec{v}_{-i} \sim \vec{F}_{-i}}[x_i(\vec{v})] - \mathbf{E}_{\vec{v}_{-i} \sim \vec{F}_{-i}}[p_i(\vec{v})] \geq 0. \tag{4.2}$$

The final set of constraints assert that, with probability 1, the item is sold to at most one bidder: for every $\vec{v} \in \vec{V}$,

$$\sum_{i=1}^{n} x_i(\vec{v}) \leq 1. \tag{4.3}$$

By construction, feasible solutions to the linear system (4.1)–(4.3) correspond to the allocation and payment rules of BIC auctions with respect to the distributions $F_1, \ldots, F_n$. This linear program has an exponential number of variables and constraints, and is not immediately useful.

4.2.3 Reducing the Dimension with Interim Allocation Rules

Is it possible to re-express the allocation and payment rules of BIC auctions with a small number of decision variables? Looking at the constraints (4.1) and (4.2), a natural idea is use only the decision variables $\{y_i(v_i)\}_{i\in[n],v_i\in V_i}$ and $\{q_i(v_i)\}_{i\in[n],v_i\in V_i}$, with the intended semantics that

$$y_i(v_i) = \mathbf{E}_{\vec{v}_{-i}}\left[x_i(v_i,\vec{v}_{-i})\right] \quad \text{and} \quad q_i(v_i) = \mathbf{E}_{\vec{v}_{-i}}\left[p_i(v_i,\vec{v}_{-i})\right].$$

In other words, $y_i(v_i)$ is the probability that bidder i wins when it bids v_i, and $q_i(v_i)$ is the expected amount that it pays; these were the only quantities that actually mattered in (4.1) and (4.2). (As usual, the expectation is over both the randomness in $\vec{v}_{-i}$ and internal to the auction.) In auction theory, the $y_i(v_i)$'s are called an *interim allocation rule*, the $q_i(v_i)$'s an *interim payment rule*.[9]

There are only $2\sum_{i=1}^{n}|V_i|$ such decision variables, far fewer than the $2\prod_{i=1}^{n}|V_i|$ variables in (4.1)–(4.3). We'll think of the $|V_i|$'s (and hence the number of decision variables) as polynomially bounded. For example, V_i could be the multiples of some small ϵ that lie in some bounded range like $[0,1]$.

We can then express the BIC constraints (4.1) in terms of this smaller set of variables by

$$\underbrace{v_i\cdot y_i(v_i)-q_i(v_i)}_{\text{expected utility of truthful bid } v_i} \geq \underbrace{v_i\cdot y_i(v_i')-q_i(v_i')}_{\text{expected utility of false bid } v_i'} \tag{4.4}$$

for every bidder i and $v_i, v_i' \in V_i$. Similarly, the IIR constraints (4.2) become

$$v_i\cdot y_i(v_i)-q_i(v_i)\geq 0 \tag{4.5}$$

for every bidder i and $v_i\in V_i$.

Just one problem. What about the feasibility constraints (4.3), which reference the individual $x_i(\vec{v})$'s and not just their expectations? The next definition articulates what feasibility means for an interim allocation rule.

Definition 4.1 (Feasible Interim Allocation Rule)**.** An interim allocation rule $\{y_i(v_i)\}_{i\in[n],v_i\in V_i}$ is *feasible* if there exist nonnegative values for $\{x_i(\vec{v})\}_{i\in[n],\vec{v}\in\vec{V}}$ such that

$$\sum_{i=1}^{n} x_i(\vec{v}) \leq 1$$

for every $\vec{v}$ (i.e., the $x_i(\vec{v})$'s constitute a feasible allocation rule), and

$$y_i(v_i) = \sum_{\vec{v}_{-i}\in\vec{V}_{-i}}\left(\prod_{j\neq i} f_j(v_j)\right)\cdot x_i(v_i,\vec{v}_{-i})$$

for every $i\in[n]$ and $v_i\in V_i$ (i.e., the intended semantics are respected).

In other words, the feasible interim allocation rules are exactly the projections (onto the $y_i(v_i)$'s) of the feasible (ex post) allocation rules.

The big question is: how can we translate interim feasibility into our new, more economical vocabulary?[10] As we'll see, Border's theorem [16] provides a crisp and computationally useful solution.

[9] Auction theory generally thinks about three informational scenarios: *ex ante*, where each bidder knows the prior distributions but not even its own valuation; *ex interim*, where each bidder knows its own valuation but not those of the others; and *ex post*, where all of the bidders know everybody's valuation. Bidders typically choose their bids at the interim stage.

[10] In principle, we know this is possible. The feasible (ex post) allocation rules form a polytope, the projection of a polytope is again a polytope, and every polytope can be described by a finite number of linear inequalities. So the real question is whether or not there's a *computationally useful* description of interim feasibility.

(v_1, v_2)	$x_1(v_1, v_2)$	$x_2(v_1, v_2)$
$(1, 1)$		
$(1, 2)$		
$(2, 1)$		
$(2, 2)$		

Table 4.1: Certifying feasibility of an interim allocation rule is analogous to filling in the table entries while respecting constraints on the sums of certain subsets of entries.

4.2.4 Examples

To get a better feel for the issue of checking the feasibility of an interim allocation rule, let's consider a couple of examples. A necessary condition for interim feasibility is that the item is awarded to at most one bidder in expectation (over the randomness in the valuations and internal to the auction):

$$\sum_{i=1}^{n} \underbrace{\sum_{v_i \in V_i} f_i(v_i) y_i(v_i)}_{\mathbf{Pr}[i \text{ wins}]} \leq 1. \tag{4.6}$$

Could this also be a sufficient condition? That is, is every interim allocation rule $\{y_i(v_i)\}_{i \in [n], v_i \in V_i}$ that satisfies (4.6) induced by a bone fide (ex post) allocation rule?

Example 4.2. Suppose there are $n = 2$ bidders. Assume that v_1, v_2 are independent and each is equally likely to be 1 or 2. Consider the interim allocation rule given by

$$y_1(1) = \tfrac{1}{2}, y_1(2) = \tfrac{7}{8}, y_2(1) = \tfrac{1}{8}, \text{ and } y_2(2) = \tfrac{1}{2}. \tag{4.7}$$

Since $f_i(v) = \frac{1}{2}$ for all $i = 1, 2$ and $v = 1, 2$, the necessary condition in (4.6) is satisfied. Can you find an (ex post) allocation rule that induces this interim rule? Answering this question is much like solving a Sudoko or KenKen puzzle—the goal is to fill in the table entries in Table 4.1 so that each row sums to at most 1 (for feasibility) and that the constraints (4.7) are satisfied. For example, the average of the top two entries in the first column of Table 4.1 should be $y_1(1) = \frac{1}{2}$. In this example, there are a number of such solutions; one is shown in Table 4.2. Thus, the given interim allocation rule is feasible.

Example 4.3. Suppose we change the interim allocation rule to

$$y_1(1) = \tfrac{1}{4}, y_1(2) = \tfrac{7}{8}, y_2(1) = \tfrac{1}{8}, \text{ and } y_2(2) = \tfrac{3}{4}.$$

The necessary condition (4.6) remains satisfied. Now, however, the interim rule is not feasible. One way to see this is to note that $y_1(2) = \frac{7}{8}$ implies that $x_1(2, 2) \geq \frac{3}{4}$ and hence $x_2(2, 2) \leq \frac{1}{4}$. Similarly, $y_2(2) = \frac{3}{4}$ implies that $x_2(2, 2) \geq \frac{1}{2}$, a contradictory constraint.

(v_1, v_2)	$x_1(v_1, v_2)$	$x_2(v_1, v_2)$
$(1, 1)$	1	0
$(1, 2)$	0	1
$(2, 1)$	3/4	1/4
$(2, 2)$	1	0

Table 4.2: One solution to Example 4.2.

The first point of Examples 4.2 and 4.3 is that it is not trivial to check whether or not a given interim allocation rule is feasible—the problem corresponds to solving a big linear system of equations and inequalities. The second point is that (4.6) is not a sufficient condition for feasibility. In hindsight, trying to summarize the exponentially many ex post feasibility constraints (4.3) with a single interim constraint (4.6) seems naive. Is there a larger set of linear constraints—possibly an exponential number—that characterizes interim feasibility?

4.2.5 Border's Theorem

Border's theorem states that a collection of "obvious" necessary conditions for interim feasibility are also sufficient. To state these conditions, assume for notational convenience that the valuation sets $V_1, \ldots, V_n$ are disjoint.[11] Let $\{x_i(\vec{v})\}_{i \in [n], \vec{v} \in \vec{V}}$ be a feasible (ex post) allocation rule and $\{y_i(v_i)\}_{i \in [n], v_i \in V_i}$ the induced (feasible) interim allocation rule. Fix for each bidder i a set $S_i \subseteq V_i$ of valuations. Call the valuations $\cup_{i=1}^n S_i$ the *distinguished* valuations. Consider first the probability, over the random valuation profile $\vec{v} \sim \vec{F}$ and any coin flips of the ex post allocation rule, that the winner of the auction (if any) has a distinguished valuation. By linearity of expectations, this probability can be expressed in terms of the interim allocation rule:

$$\sum_{i=1}^{n} \sum_{v_i \in S_i} f_i(v_i) y_i(v_i). \tag{4.8}$$

The expression (4.8) is linear in the $y_i(v_i)$'s.

The second quantity we study is the probability, over $\vec{v} \sim \vec{F}$, that there is a bidder with a distinguished valuation. This has nothing to do with the allocation rule, and is a function of the prior distributions only:

$$1 - \prod_{i=1}^{n} \left(1 - \sum_{v_i \in S_i} f_i(v_i) \right). \tag{4.9}$$

Since there can only be a winner with a distinguished valuation if there is a bidder with a distinguished valuation, the quantity in (4.8) can only be less than (4.9). Border's theorem asserts that these conditions, ranging over all choices of $S_1 \subseteq V_1, \ldots, S_n \subseteq V_n$, are also sufficient for the feasibility of an interim allocation rule.

Theorem 4.4 (Border's theorem [16])**.** *An interim allocation rule $\{y_i(v_i)\}_{i \in [n], v_i \in V_i}$ is feasible if and only if for every choice $S_1 \subseteq V_1, \ldots, S_n \subseteq V_n$ of distinguished valuations,*

$$\sum_{i=1}^{n} \sum_{v_i \in S_i} f_i(v_i) y_i(v_i) \leq 1 - \prod_{i=1}^{n} \left(1 - \sum_{v_i \in S_i} f_i(v_i) \right). \tag{4.10}$$

Border's theorem can be derived from the max-flow/min-cut theorem (following [17, 28]); we include the proof in Section 4.4 for completeness.

[11] This is without loss of generality, since we can simply "tag" each valuation $v_i \in V_i$ with the "name" i (i.e., view each $v_i \in V_i$ as the set $\{v_i, i\}$).

Border's theorem yields an explicit description as a linear system of the feasible interim allocation rules induced by BIC single-item auctions. To review, this linear system is

$$v_i \cdot y_i(v_i) - q_i(v_i) \geq v_i \cdot y_i(v_i') - q_i(v_i') \qquad \forall i \text{ and } v_i, v_i' \in V_i \tag{4.11}$$

$$v_i \cdot y_i(v_i) - q_i(v_i) \geq 0 \qquad \forall i \text{ and } v_i \in V_i \tag{4.12}$$

$$\sum_{i=1}^{n} \sum_{v_i \in S_i} f_i(v_i) y_i(v_i) \leq 1 - \prod_{i=1}^{n} \left(1 - \sum_{v_i \in S_i} f_i(v_i)\right) \qquad \forall S_1 \subseteq V_1, \ldots, S_n \subseteq V_n. \tag{4.13}$$

For example, optimizing the objective function

$$\max \sum_{i=1}^{n} f_i(v_i) \cdot q_i(v_i) \tag{4.14}$$

over the linear system (4.11)–(4.13) computes the expected revenue of an optimal BIC single-item auction for the distributions $F_1, \ldots, F_n$.

The linear system (4.11)–(4.13) has only a polynomial number of variables (assuming the $|V_i|$'s are polynomially bounded), but it does have an exponential number of constraints of the form (4.13). One solution is to use the ellipsoid method, as the linear system does admit a polynomial-time separation oracle [3, 21].[12] Alternatively, Alaei et al. [3] provide a polynomial-size extended formulation of the polytope of feasible interim allocation rules (with a polynomial number of additional decision variables and only polynomially many constraints). In any case, we conclude that there is a computationally tractable description of the feasible interim allocation rules of BIC single-item auctions.

4.3 Beyond Single-Item Auctions: A Complexity-Theoretic Barrier

Myerson's theory of optimal auctions (Section 4.1.2) extends beyond single-item auctions to all "single-parameter" settings (see Section 4.3.1 for two examples). Can Border's theorem be likewise extended? There are analogs of Border's theorem in settings modestly more general than single-item auctions, including k-unit auctions with unit-demand bidders [3, 21, 28], and approximate versions of Border's theorem exist fairly generally [21, 22]. Can this state-of-the-art be improved upon? We next use complexity theory to develop evidence for a negative answer.

Theorem 4.5 (Gopalan et al. [68]). *(Informal) There is no exact Border's-type theorem for settings significantly more general than the known special cases (unless* PH *collapses).*

We proceed to defining what we mean by "significantly more general" and a "Border's-type theorem."

4.3.1 Two Example Settings

The formal version of Theorem 4.5 conditionally rules out "Border's-type theorems" for several specific settings that are representative of what a more general version of Border's theorem might cover. We mention two of these here (more are in [68]).

In a *public project* problem, there is a binary decision to make: whether or not to undertake a costly project (like building a new school). Each bidder i has a private valuation v_i for the outcome where the project

[12] This is not immediately obvious, as the max-flow/min-cut argument in Section 4.4 involves an exponential-size graph.

is built, and valuation 0 for the outcome where it is not. If the project is built, then everyone can use it. In this setting, feasibility means that all bidders receive the same allocation: $x_1(\vec{v}) = x_2(\vec{v}) = \cdots = x_n(\vec{v}) \in [0, 1]$ for every valuation profile $\vec{v}$.

In a *matching* problem, there is a set M of items, and each bidder is only interested in receiving a specific pair $j, \ell \in M$ of items. (Cf., the AND bidders of the preceding lecture.) For each bidder, the corresponding pair of items is known in advance, and the bidder's valuation for the pair is private as usual. Feasible outcomes correspond to (distributions over) matchings in the graph with vertices M and edges given by bidders' desired pairs.

Public project and matching problems are both "single-parameter" problems (i.e., each bidder has only one private parameter). As such, Myerson's optimal auction theory (Section 4.1.2) can be used to characterize the expected revenue-maximizing auction. Do these settings also admit analogs of Border's theorem?

4.3.2 Border's-Type Theorems

What do we actually mean by a "Border's-type theorem?" Since we aim to prove impossibility results, we should adopt a definition that is as permissive as possible. Border's theorem (Theorem 4.4) gives a characterization of the feasible interim allocation rules of a single-item auction as the solutions to a finite system of linear inequalities. This by itself is not impressive—since the set is a polytope, it is guaranteed to have such a characterization. The appeal of Border's theorem is that the characterization uses only the "nice" linear inequalities in (4.10). Our "niceness" requirement is that the characterization use only linear inequalities that can be efficiently recognized and tested. This is a weak necessary condition for such a characterization to be computationally useful.

Definition 4.6 (Border's-Type Theorem)**.** A *Border's-type theorem* holds for an auction design setting if, for every instance of the setting (specifying the number of bidders and their prior distributions, etc.), there is a system of linear inequalities such that the following properties hold.

1. (Characterization) The feasible solutions of the linear system are precisely the feasible interim allocation rules of the instance.

2. (Efficient recognition) There is a polynomial-time algorithm that can decide whether or not a given linear inequality belongs to the linear system.

3. (Efficient testing) The bit complexity of each linear inequality is polynomial in the description of the instance. (The number of inequalities can be exponential.)

For example, consider the original Border's theorem, for single-item auctions (Theorem 4.4). The recognition problem is straightforward: the left-side of (4.10) encodes the S_i's, from which the right-hand side can be computed and checked in polynomial time. It is also evident that every inequality in (4.10) has a polynomial-length description.[13]

4.3.3 Consequences of a Border's-Type Theorem

The high-level idea behind the proof of Theorem 4.5 is to show that a Border's-type theorem puts a certain computational problem low in the polynomial hierarchy, and then to show that this problem is #P-hard for

[13]The characterization in Theorem 4.4 and the extensions in [3, 21, 28] have additional features not required or implied by Definition 4.6, such as polynomial-time separation oracles (and even a compact extended formulation in the single-item case [3]). The impossibility results in Section 4.3.4 rule out analogs of Border's theorem that merely satisfy Definition 4.6, let alone these stronger properties.

the public project and matching settings defined in Section 4.3.1.[14] The computational problem is: given a description of an instance (including the prior distributions), compute the maximum-possible expected revenue that can be obtained by a feasible and BIC auction.[15]

What use is a Border's-type theorem? For starters, it implies that the problem of testing the feasibility of an interim allocation rule is in co-NP. To prove the infeasibility of such a rule, one simply exhibits an inequality of the characterizing linear system that the rule fails to satisfy. Verifying this failure reduces to the recognition and testing problems, which by Definition 4.6 are polynomial-time solvable.

Proposition 4.7. *If a Border's-type theorem holds for an auction design setting, then the membership problem for the polytope of feasible interim allocation rules belongs to* co-NP.

Combining Proposition 4.7 with the ellipsoid method puts the problem of computing the maximum-possible expected revenue in P^{NP}.

Theorem 4.8. *If a Border's-type theorem holds for an auction design setting, then the maximum expected revenue of a feasible BIC auction can be computed in* P^{NP}.

Proof. We compute the optimal expected revenue of a BIC auction via linear programming, as follows. The decision variables are the same $y_i(v_i)$'s and $q_i(v_i)$'s as in (4.11)–(4.13), and we retain the BIC constraints (4.11) and the IIR constraints (4.12). By assumption, we can replace the single-item interim feasibility constraints (4.13) with a linear system that satisfies the properties of Definition 4.6. The maximum expected revenue of a feasible BIC auction can then be computed by optimizing a linear objective function (in the $q_i(v_i)$'s, as in (4.14)) subject to these constraints. Using the ellipsoid method [88], this can be accomplished with a polynomial number of invocations of a separation oracle (which either verifies feasibility or exhibits a violated constraint). Proposition 4.7 implies that we can implement this separation oracle in co-NP, and thus compute the maximum expected revenue of a BIC auction in P^{NP}.[16] □

4.3.4 Impossibility Results from Computational Intractability

Theorem 4.8 concerns the problem of computing the maximum expected revenue of a feasible BIC auction, given a description of an instance. It is easy to classify the complexity of this problem in the public project and matching settings introduced in Section 4.3.1 (and several other settings, see [68]).

Proposition 4.9. *Computing the maximum expected revenue of a feasible BIC auction of a public project instance is a* #P*-hard problem.*

Proposition 4.9 is a straightforward reduction from the #P-hard problem of computing the number of feasible solutions to an instance of the Knapsack problem.[17]

Proposition 4.10. *Computing the maximum expected revenue of a feasible BIC auction of a matching instance is a* #P*-hard problem.*

[14] Recall that Toda's theorem [144] implies that a #P-hard problem is contained in the polynomial hierarchy only if PH collapses.

[15] Sanity check: this problem turns out to be polynomial-time solvable in the setting of single-item auctions [68].

[16] One detail: Proposition 4.7 only promises solutions to the "yes/no" question of feasibility, while a separation oracle needs to produce a violated constraint when given an infeasible point. But under mild conditions (easily satisfied here), an algorithm for the former problem can be used to solve the latter problem as well [137, P.189].

[17] An aside for aficionados of the analysis of Boolean functions: Proposition 4.9 is essentially equivalent to the #P-hardness of checking whether or not given Chow parameters can be realized by some bounded function on the hypercube. See [68] for more details on the surprisingly strong correspondence between Myerson's optimal auction theory (in the context of public projects) and the analysis of Boolean functions.

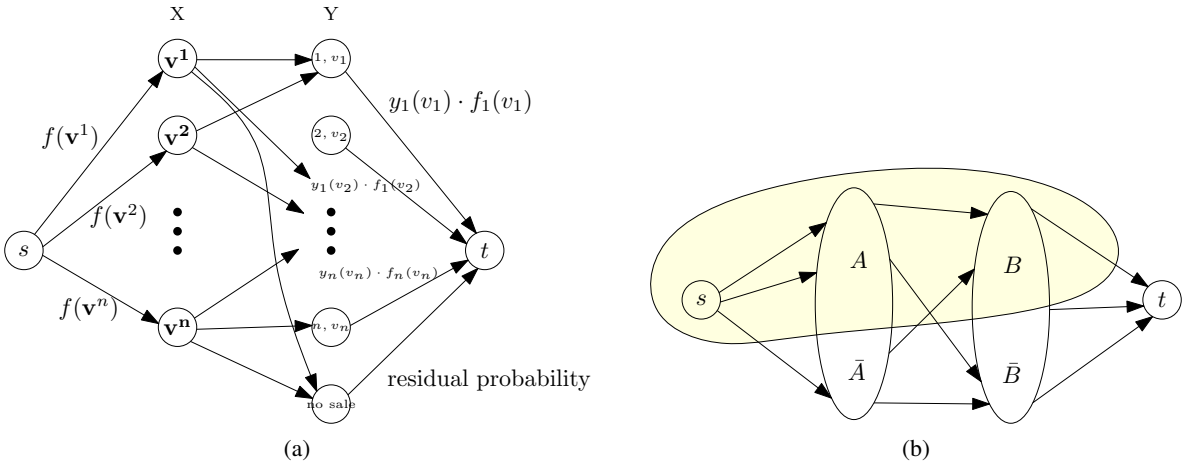

Figure 4.1: The max-flow/min-cut proof of Border's theorem.

Proposition 4.10 is a straightforward reduction from the #P-hard Permanent problem.

We reiterate that Myerson's optimal auction theory applies to the public project and matching settings, and in particular gives a polynomial-time algorithm that outputs a description of an optimal auction (for given prior distributions). Moreover, the optimal auction can be implemented as a polynomial-time algorithm. Thus it's not hard to figure out what the optimal auction is, nor to implement it—what's hard is figuring out exactly how much revenue it makes on average!

Combining Theorem 4.8 with Propositions 4.9 and 4.10 gives the following corollaries, which indicate that there is no Border's-type theorem significantly more general than the ones already known.

Corollary 4.11. *If* #P $\not\subseteq$ PH*, then there is no Border's-type theorem for the setting of public projects.*

Corollary 4.12. *If* #P $\not\subseteq$ PH*, then there is no Border's-type theorem for the matching setting.*

4.4 Appendix: A Combinatorial Proof of Border's Theorem

Proof. (of Theorem 4.4) We have already argued the "only if" direction, and now prove the converse. The proof is by the max-flow/min-cut theorem—given the statement of the theorem and this hint, the proof writes itself.

Suppose the interim allocation rule $\{y_i(v_i)\}_{i\in[n],v_i\in V_i}$ satisfies (4.10) for every $S_1 \subseteq V_1, \ldots, S_n \subseteq V_n$. Form a four-layer s-t directed flow network G as follows (Figure 4.1(a)). The first layer is the source s, the last the sink t. In the second layer X, vertices correspond to valuation profiles $\vec{v}$. We abuse notation and refer to vertices of X by the corresponding valuation profiles. There is an arc $(s, \vec{v})$ for every $\vec{v} \in X$, with capacity $\prod_{i=1}^n f_i(v_i)$. Note that the total capacity of these edges is 1.

In the third layer Y, vertices correspond to winner-valuation pairs; there is also one additional "no winner" vertex. We use (i, v_i) to denote the vertex representing the event that bidder i wins the item and also has valuation v_i. For each i and $v_i \in V_i$, there is an arc $((i, v_i), t)$ with capacity $f_i(v_i)y_i(v_i)$. There is also an arc from the "no winner" vertex to t, with capacity $1 - \sum_{i=1}^n \sum_{v_i\in V_i} f_i(v_i)y_i(v_i)$.[18]

[18] If $\sum_{i=1}^n \sum_{v_i\in V_i} f_i(v_i)y_i(v_i) > 1$, then the interim allocation rule is clearly infeasible (recall (4.6)). Alternatively, this would violate Border's condition for the choice $S_i = V_i$ for all i.

Finally, each vertex $\vec{v} \in X$ has $n+1$ outgoing arcs, all with infinite capacity, to the vertices $(1, v_1)$, $(2, v_2)$, $\ldots$, (n, v_n) of Y and also to the "no winner" vertex.

By construction, s-t flows of G with value 1 correspond to ex post allocation rules with induced interim allocation rule $\{y_i(v_i)\}_{i \in [n], v_i \in V_i}$, with $x_i(\vec{v})$ equal to the amount of flow on the arc $(\vec{v}, (i, v_i))$ times $(\prod_{i=1}^{n} f_i(v_i))^{-1}$.

To show that there exists a flow with value 1, it suffices to show that every s-t cut has value at least 1 (by the max-flow/min-cut theorem). So fix an s-t cut. Let this cut include the vertices A from X and B from Y. Note that all arcs from s to $X \setminus A$ and from B to t are cut (Figure 4.1(b)). For each bidder i, define $S_i \subseteq V_i$ as the possible valuations of i that are *not* represented among the valuation profiles in A. Then, for every valuation profile $\vec{v}$ containing at least one distinguished valuation, the arc $(s, \vec{v})$ is cut. The total capacity of these arcs is the right-hand side (4.9) of Border's condition.

Next, we can assume that every vertex of the form (i, v_i) with $v_i \notin S_i$ is in B, since otherwise an (infinite-capacity) arc from A to $Y \setminus B$ is cut. Similarly, unless $A = \emptyset$—in which case the cut has value at least 1 and we're done—we can assume that the "no winner" vertex lies in B. Thus, the only edges of the form $((i, v_i), t)$ that are not cut involve a distinguished valuation $v_i \in S_i$. It follows that the total capacity of the cut edges incident to t is at least 1 minus the left-hand size (4.8) of Border's condition. Given our assumption that (4.8) is at most (4.9), this s-t cut has value at least 1. This completes the proof of Border's theorem. □

Lunar Lecture 5

Tractable Relaxations of Nash Equilibria

Lecturer: Tim Roughgarden *Scribe: Jacobo Torán*

5.1 Preamble

We've spent much of this week proving several types of impossibility results for the efficient computation of exact and approximate Nash equilibria. How should we respond to such rampant computational intractability? What should be the message to economists—should they change the way they do economic analysis in some way?[1]

One approach, familiar from coping with NP-hard problems, is to look for tractable special cases. For example, Solar Lecture 1 proved tractability results for two-player zero-sum games. Some interesting tractable generalizations of zero-sum games have been identified (see [23] for a recent example), and polynomial-time algorithms are also known for some relatively narrow classes of games (see e.g. [85]). Still, for the lion's share of games that we might care about, no polynomial-time algorithms for computing exact or approximate Nash equilibria are known.

A different approach, which has been more fruitful, is to continue to work with general games and look for an *equilibrium concept* that is more computationally tractable than exact or approximate Nash equilibria. The equilibrium concepts that we'll consider—the correlated equilibrium and the coarse correlated equilibrium—were originally invented by game theorists, but computational complexity considerations are now shining a much brighter spotlight on them.

Where do these alternative equilibrium concepts come from? They arise quite naturally from the study of uncoupled dynamics, which we last saw in Solar Lecture 1.

5.2 Uncoupled Dynamics Revisited

Section 1.3 of Solar Lecture 1 introduced uncoupled dynamics in the context of two-player games. In this lecture we work with the analogous setup for a general number k of players. We use S_i to denote the (pure) strategies of player i, $s_i \in S_i$ a specific strategy, σ_i a mixed strategy, $\vec{s}$ and $\vec{\sigma}$ for profiles (i.e., k-vectors) of pure and mixed strategies, and $u_i(\vec{s})$ for player i's payoff in the outcome $\vec{s}$.

[1] Recall the discussion in Section 1.2.7 of Solar Lecture 1: a critique of a widely used concept like the Nash equilibrium is not particularly helpful unless accompanied by a proposed alternative.

Uncoupled Dynamics (k-Player Version)

At each time step $t = 1, 2, 3, \ldots$:

1. Each player $i = 1, 2, \ldots, k$ simultaneously chooses a mixed strategy σ_i^t over S_i as a function only of her own payoffs and the strategies chosen by players in the first $t - 1$ time steps.

2. Every player learns all of the strategies $\vec{\sigma}^t$ chosen at time t.

"Uncoupled" refers to the fact that each player initially knows only her own payoff function $u_i(\cdot)$, while "dynamics" means a process by which players learn how to play in a game.

One of the only positive algorithmic results that we've seen this week concerned *smooth fictitious play (SFP)*. The k-player version of SFP is as follows.

Smooth Fictitious Play (k-Player Version)

Given: parameter family $\{\eta^t \in [0, \infty) : t = 1, 2, 3, \ldots\}$.

At each time step $t = 1, 2, 3, \ldots$:

1. Every player i simultaneously chooses the mixed strategy σ_i^t by playing each strategy s_i with probability proportional to $e^{\eta^t \pi_i^t}$, where π_i^t is the time-averaged expected payoff player i would have earned by playing s_i at every previous time step. Equivalently, π_i^t is the expected payoff of strategy s_i when the other players' strategies $\vec{s}_{-i}$ are drawn from the joint distribution $\frac{1}{t-1} \sum_{h=1}^{t-1} \vec{\sigma}_{-i}^h$.[2]

2. Every player learns all of the strategies $\vec{\sigma}^t$ chosen at time t.

A typical choice for the η_t's is $\eta_t \approx \sqrt{t}$.

In Theorem 1.8 in Solar Lecture 1 we proved that, in an $m \times n$ two-player zero-sum game, after $O(\log(m + n)/\epsilon^2)$ time steps, the empirical distributions of the two players constitute an ϵ-approximate Nash equilibrium.[3] An obvious question is: what is the outcome of a logarithmic number of rounds of smooth fictitious play in a non-zero-sum game? Our communication complexity lower bound in Solar Lectures 2 and 3 implies that it cannot in general be an ϵ-approximate Nash equilibrium. Does it have some alternative economic meaning? The answer to this question turns out to be closely related to some classical game-theoretic equilibrium concepts, which we discuss next.

[2] Recall from last lecture that for an n-vector $\vec{z}$ and a coordinate $i \in [k]$, $\vec{z}_{-i}$ denotes the $(k-1)$-vector obtained by removing the ith component from $\vec{z}$, and we identify $(z_i, \vec{z}_{-i})$ with $\vec{z}$.

[3] Recall the proof idea: smooth fictitious play corresponds to running the vanishing-regret "multiplicative weights" algorithm (with reward vectors induced by the play of others), and in a two-player zero-sum game, the vanishing-regret guarantee (i.e., with time-averaged payoff at least that of the best fixed action in hindsight, up to $o(1)$ error) implies the ϵ-approximate Nash equilibrium condition.

5.3 Correlated and Coarse Correlated Equilibria

5.3.1 Correlated Equilibria

The correlated equilibrium is a well-known equilibrium concept defined by Aumann [7]. We define it, then explain the standard semantics, and then offer an example.[4]

Definition 5.1 (Correlated Equilibrium)**.** A joint distribution ρ on the set $S_1 \times \cdots \times S_k$ of outcomes of a game is a *correlated equilibrium* if for every player $i \in \{1, 2, \ldots, k\}$, strategy $s_i \in S_i$, and deviation $s_i' \in S_i$,

$$\mathbf{E}_{\vec{s} \sim \rho}\left[u_i(\vec{s}) \mid s_i\right] \geq \mathbf{E}_{\vec{s} \sim \rho}\left[u_i(s_i', \vec{s}_{-i}) \mid s_i\right]. \tag{5.1}$$

Importantly, the distribution ρ in Definition 5.1 need not be a product distribution; in this sense, the strategies chosen by the players are correlated. The Nash equilibria of a game correspond to the correlated equilibria that are product distributions.

The usual interpretation of a correlated equilibrium involves a trusted third party. The distribution ρ over outcomes is publicly known. The trusted third party samples an outcome $\vec{s}$ according to ρ. For each player $i = 1, 2, \ldots, k$, the trusted third party privately suggests the strategy s_i to i. The player i can follow the suggestion s_i, or not. At the time of decision making, a player i knows the distribution ρ and one component s_i of the realization $\vec{s}$, and accordingly has a posterior distribution on others' suggested strategies $\vec{s}_{-i}$. With these semantics, the correlated equilibrium condition (5.1) requires that every player minimizes her expected cost by playing the suggested strategy s_i. The expectation is conditioned on i's information—ρ and s_i—and assumes that other players play their recommended strategies $\vec{s}_{-i}$.

Definition 5.1 is a bit of a mouthful. But you are intimately familiar with a good example of a correlated equilibrium that is not a mixed Nash equilibrium—a traffic light! Consider the following two-player game, with each matrix entry listing the payoffs of the row and column players in the corresponding outcome:

	Stop	Go
Stop	0,0	0,1
Go	1,0	-5,-5

This game has two pure Nash equilibria, the outcomes (Stop, Go) and (Go, Stop). Define ρ by randomizing uniformly between these two Nash equilibria. This is not a product distribution over the game's four outcomes, so it cannot correspond to a Nash equilibrium of the game. It is, however, a correlated equilibrium.[5]

5.3.2 Coarse Correlated Equilibria

The outcome of smooth fictitious play in non-zero-sum games relates to a still more permissive equilibrium concept, the *coarse correlated equilibrium*, which was first studied by Moulin and Vial [110].

Definition 5.2 (Coarse Correlated Equilibrium)**.** A joint distribution ρ on the set $S_1 \times \cdots \times S_k$ of outcomes of a game is a *coarse correlated equilibrium* if for every player $i \in \{1, 2, \ldots, k\}$ and every unilateral deviation $s_i' \in S_i$,

$$\mathbf{E}_{\vec{s} \sim \rho}\left[u_i(\vec{s})\right] \geq \mathbf{E}_{\vec{s} \sim \rho}\left[u_i(s_i', \vec{s}_{-i})\right]. \tag{5.2}$$

[4] This section draws from [129, Lecture 13].

[5] For example, consider the row player. If the trusted third party (i.e., the traffic light) recommends the strategy "Go" (i.e., is green), then the row player knows that the column player was recommended "Stop" (i.e., has a red light). Assuming the column player plays her recommended strategy and stops at the red light, the best strategy for the row player is to follow her recommendation and to go.

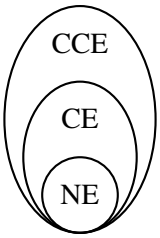

Figure 5.1: The relationship between Nash equilibria (NE), correlated equilibria (CE), and coarse correlated equilibria (CCE). Enlarging the set of equilibria increases computational tractability but decreases predictive power.

The condition (5.2) is the same as that for the Nash equilibrium (Definition 1.3), except without the restriction that ρ is a product distribution. In this condition, when a player i contemplates a deviation s_i', she knows only the distribution ρ and *not* the component s_i of the realization. That is, a coarse correlated equilibrium only protects against unconditional unilateral deviations, as opposed to the unilateral deviations conditioned on s_i that are addressed in Definition 5.1. It follows that every correlated equilibrium is also a coarse correlated equilibrium (Figure 5.1).

As you would expect, *ϵ-approximate* correlated and coarse correlated equilibria are defined by adding a "$-\epsilon$" to the right-hand sides of (5.1) and (5.2), respectively. We can now answer the question about smooth fictitious play in general games: the time-averaged history of joint play under smooth fictitious play converges to the set of coarse correlated equilibria.

Proposition 5.3. *For every k-player game in which every player has at most m strategies, after $T = O((\log m)/\epsilon^2)$ time steps of smooth fictitious play, the time-averaged history of play $\frac{1}{T}\sum_{t=1}^{T}\vec{\sigma}^t$ is an ϵ-approximate coarse correlated equilibrium.*

Proposition 5.3 follows straightforwardly from the definition of ϵ-approximate coarse correlated equilibria and the vanishing regret guarantee of smooth fictitious play that we proved in Solar Lecture 1. Precisely, by Corollary 1.11 of that lecture, after $O((\log m)/\epsilon^2)$ time steps of smooth fictitious play, every player has at most ϵ regret (with respect to the best fixed strategy in hindsight, see Definition 1.9 in Solar Lecture 1). This regret guarantee is equivalent to the conclusion of Proposition 5.3 (as you should check).

What about correlated equilibria? While the time-averaged history of play in smooth fictitious play does not in general converge to the set of correlated equilibria, Foster and Vohra [53] and Hart and Mas-Colell [71] show that the time-averaged play of other reasonably simple types of uncoupled dynamics is guaranteed to be an ϵ-correlated equilibrium after a polynomial (rather than logarithmic) number of time steps.

5.4 Computing an Exact Correlated or Coarse Correlated Equilibrium

5.4.1 Normal-Form Games

Solar Lecture 1 showed that approximate Nash equilibria of two-player zero-sum games can be learned (and hence computed) efficiently (Theorem 1.8). Proposition 5.3 and the extensions in [53, 71] show analogs of this result for approximate correlated and coarse correlated equilibria of general games. Solar Lecture 1 also showed that an exact Nash equilibrium of a two-player zero-sum game can be computed in polynomial

time by linear programming (Corollary 1.5). Is the same true for an exact correlated or coarse correlated equilibrium of a general game?

Consider first the case of coarse correlated equilibria, and introduce one decision variable $x_{\vec{s}}$ per outcome $\vec{s}$ of the game, representing the probability assigned to $\vec{s}$ in a joint distribution ρ. The feasible solutions to the following linear system are then precisely the coarse correlated equilibria of the game:

$$\sum_{\vec{s}} u_i(\vec{s}) x_{\vec{s}} \geq \sum_{\vec{s}} u_i(s_i', \vec{s}_{-i}) x_{\vec{s}} \qquad \text{for every } i \in [k] \text{ and } s_i' \in S_i \tag{5.3}$$

$$\sum_{\vec{s} \in \vec{S}} x_{\vec{s}} = 1 \tag{5.4}$$

$$x_{\vec{s}} \geq 0 \qquad \text{for every } \vec{s} \in \vec{S}. \tag{5.5}$$

Similarly, correlated equilibria are captured by the following linear system:

$$\sum_{\vec{s}\,:\, s_i = j} u_i(\vec{s}) x_{\vec{s}} \geq \sum_{\vec{s}\,:\, s_i = j} u_i(s_i', \vec{s}_{-i}) x_{\vec{s}} \qquad \text{for every } i \in [k] \text{ and } j, s_i' \in S_i \tag{5.6}$$

$$\sum_{\vec{s} \in \vec{S}} x_{\vec{s}} = 1 \tag{5.7}$$

$$x_{\vec{s}} \geq 0 \qquad \text{for every } \vec{s} \in \vec{S}. \tag{5.8}$$

The following proposition is immediate.

Proposition 5.4 (Gilboa and Zemel [61])**.** *An exact correlated or coarse correlated equilibrium of a game can be computed in time polynomial in the number of outcomes of the game.*

More generally, any linear function (such as the sum of players' expected payoffs) can be optimized over the set of correlated or coarse correlated equilibria in time polynomial in the number of outcomes.

For games described in *normal form*, with each player i's payoffs $\{u_i(\vec{s})\}_{\vec{s} \in \vec{S}}$ given explicitly in the input, Proposition 5.4 provides an algorithm with running time polynomial in the input size. However, the number of outcomes of a game scales exponentially with the number k of players.[6] The computationally interesting multi-player games, and the multi-player games that naturally arise in computer science applications, are those with a *succinct description*. Can we compute an exact correlated or coarse correlated equilibrium in time polynomial in the size of a game's description?

5.4.2 Succinctly Represented Games

For concreteness, let's look at one concrete example of a class of succinctly represented games: *graphical games* [86, 90]. A graphical game is described by an undirected graph $G = (V, E)$, with players corresponding to vertices, and a local payoff matrix for each vertex. The local payoff matrix for vertex i specifies i's payoff for each possible choice of its strategy and the strategies chosen by its neighbors in G. By assumption, the payoff of a player is independent of the strategies chosen by non-neighboring players. When the graph G has maximum degree Δ, the size of the game description is exponential in Δ but polynomial in the number k of players. The most interesting cases are when $\Delta = O(1)$ or perhaps $\Delta = O(\log k)$. In these cases, the

[6] This fact should provide newfound appreciation for the distributed learning algorithms that compute an approximate coarse correlated equilibrium (in Proposition 5.3) and an approximate correlated equilibrium (in [53, 71]), where the total amount of computation is only *polynomial* in k (and in m and $\frac{1}{\epsilon}$).

number of outcomes (and hence the size of the game's normal-form description) is exponential in the size of the succinct description of the game, and solving the linear system (5.3)–(5.5) or (5.6)–(5.8) does not result in a polynomial-time algorithm.

We next state a result showing that, quite generally, an exact correlated (and hence coarse correlated) equilibrium of a succinctly represented game can be computed in polynomial time. The key assumption is that the following Expected Utility problem can be solved in time polynomial in the size of the game's description.[7]

The Expected Utility Problem

Given a succinct description of a player's payoff function u_i and mixed strategies $\sigma_1, \ldots, \sigma_k$ for all of the players, compute the player's expected utility:

$$\mathbf{E}_{\vec{s} \sim \vec{\sigma}}\left[u_i(\vec{s})\right].$$

For most of the succinctly represented multi-player games that come up in computer science applications, the Expected Utility problem can be solved in polynomial time. For example, in a graphical game it can be solved by brute force—summing over the entries in player i's local payoff matrix, weighted by the probabilities in the given mixed strategies. This algorithm takes time exponential in Δ but polynomial in k, and hence is polynomial in the size of the game's succinct representation.

Tractability of solving the Expected Utility problem is a sufficient condition for the tractability of computing an exact correlated equilibrium.

Theorem 5.5 (Papadimitriou and Roughgarden [119], Jiang and Leyton-Brown [79])**.** *There is a polynomial-time Turing reduction from the problem of computing a correlated equilibrium of a succinctly described game to the* Expected Utility *problem.*

Theorem 5.5 applies to a long list of succinctly described games that have been studied in the computer science literature, with graphical games serving as one example.[8]

The starting point of the proof of Theorem 5.5 is the exponential-size linear system (5.6)–(5.8). We know that this linear system is feasible (by Nash's Theorem, since the system includes all Nash equilibria). With exponentially many variables, however, it's not clear how to efficiently compute a feasible solution. The dual linear system, meanwhile, has a polynomial number of variables (corresponding to the constraints in (5.6)) and an exponential number of inequalities (corresponding to game outcomes). By Farkas's Lemma—or, equivalently, strong linear programming duality (see e.g. [38])—we know that this dual linear system is infeasible.

The key idea is to run the ellipsoid algorithm [88] on the infeasible dual linear system—called the "ellipsoid against hope" in [119]. A polynomial-time separation oracle must produce, given an alleged solution (which we know is infeasible), a violated inequality. It turns out that this separation oracle reduces to solving a polynomial number of instances of the Expected Utility problem (which is polynomial-time solvable by assumption) and computing the stationary distribution of a polynomial number of polynomial-size Markov chains (also polynomial-time solvable, e.g. by linear programming). The ellipsoid against hope terminates after a polynomial number of invocations of its separation oracle, necessarily with a proof that the dual linear system is infeasible. To recover a primal feasible solution (i.e., a correlated equilibrium), one

[7] Some kind of assumption is necessary to preclude baking an NP-complete problem into the game's description.

[8] For the specific case of graphical games, Kakade et al. [82] were the first to develop a polynomial-time algorithm for computing an exact correlated equilibrium.

can retain only the primal decision variables corresponding to the (polynomial number of) dual constraints generated by the separation oracle, and solve directly this polynomial-size reduced version of the primal linear system.[9]

5.5 The Price of Anarchy of Coarse Correlated Equilibria

5.5.1 Balancing Computational Tractability with Predictive Power

We now understand senses in which Nash equilibria are computationally intractable (Solar Lectures 2–5) while correlated equilibria are computationally tractable (Sections 5.3 and 5.4). From an economic perspective, these results suggest that it could be prudent to study the correlated equilibria of a game, rather than just its Nash equilibria.[10]

Passing from Nash equilibria to the larger set of correlated equilibria is a two-edged sword. Computational tractability increases, and with it the plausibility that actual game play will conform to the equilibrium notion. But whatever criticisms we had about the Nash equilibrium's predictive power (recall Section 1.2.7 in Solar Lecture 1), they are even more severe for the correlated equilibrium (since there are only more of them). The worry is that games typically have far too many correlated equilibria to say anything interesting about them. Our final order of business for the week is to dispel this worry, at least in the context of price-of-anarchy analyses.

Recall from Lunar Lecture 2 that the *price of anarchy (POA)* is defined as the ratio between the objective function value of an optimal solution, and that of the worst equilibrium:

$$\mathsf{PoA}(G) := \frac{f(OPT(G))}{\min_{\mu \text{ is an equilibrium of } G} f(\mu)},$$

where G denotes a game, f denotes a maximization objective function (with $f(\mu) = \mathbf{E}_{\vec{s}\sim\mu}[f(\vec{s})]$ when μ is a probability distribution), and $OPT(G)$ is the optimal outcome of G with respect to f. Thus the POA of a game is always at least 1, and the closer to 1, the better.

The POA of a game depends on the choice of equilibrium concept. Since it is defined with respect to the worst equilibrium, the POA only degrades as the set of equilibria grows larger. Thus, the POA with respect to coarse correlated equilibria can only be worse (i.e., larger) than that with respect to correlated equilibria, which can in turn only be worse than the POA with respect to Nash equilibria (recall Figure 5.1).

The hope is that there's a "sweet spot" equilibrium concept—permissive enough to be computationally tractable, yet stringent enough to allow good worse-case approximation guarantees. Happily, the coarse correlated equilibrium is just such a sweet spot!

5.5.2 Smooth Games and Extension Theorems

After the first ten years of price-of-anarchy analyses (roughly 1999-2008), it was clear to researchers in the area that many such analyses across different application domains share a common architecture (in routing games, facility location games, scheduling games, auctions, etc.). The concept of "proofs of POA bounds

[9] As a bonus, this means that the algorithm will output a "sparse" correlated equilibrium, with support size polynomial in the size of the game description.

[10] This is not a totally unfamiliar idea to economists. According to Solan and Vohra [139], Roger Myerson, winner of the 2007 Nobel Prize in Economics, asserted that "if there is intelligent life on other planets, in a majority of them, they would have discovered correlated equilibrium before Nash equilibrium."

that follow the standard template" was made precise in the theory of smooth games [128].[11,12] One can then define the *robust price of anarchy* as the best (i.e., smallest) bound on a game's POA that can be proved by following the standard template.

The proof template formalized by smooth games superficially appears to be relevant only for the POA with respect to *pure* Nash equilibria, as the definition involves no randomness (let alone correlation). The good news is that the template's simplicity makes it relatively easy to use. One would expect the bad news to be that bounds on the POA of more permissive equilibrium concepts require different proof techniques, and that the corresponding POA bounds would be much worse. This is not the case—every POA bound proved using the canonical template automatically applies not only to the pure Nash equilibria of a game, but more generally to all of the game's coarse correlated equilibria (and hence all of its correlated and mixed Nash equilibria).[13]

Theorem 5.6 (Roughgarden [128])**.** *In every game, the POA with respect to coarse correlated equilibria is bounded above by its robust POA.*

For ϵ-approximate coarse correlated equilibria—as guaranteed by a logarithmic number of rounds of smooth fictitious play (Proposition 5.3)—the POA bound in Theorem 5.6 degrades by an additive $O(\epsilon)$ term.

[11]The formal definition is a bit technical, and we won't need it here. Roughly, it requires that the best-response condition is invoked in an equilibrium-independent way and that a certain restricted type of charging argument is used.

[12]There are several important precursors to this theory, including Blum et al. [14], Christodoulou and Koutsoupias [35], and Vetta [145]. See [128] for a detailed history.

[13]Smooth games and the "extension theorem" in Theorem 5.6 are the starting point for the modular and user-friendly toolbox for proving POA bounds in complex settings mentioned in Section 1.3.4 of Lunar Lecture 1. Generalizations of this theory to incomplete-information games (like auctions) and to the composition of smooth games (like simultaneous single-item auctions) lead to good POA bounds for simple auctions [143]. (These generalizations also brought together two historically separate subfields of algorithmic game theory, namely algorithmic mechanism design and price-of-anarchy analyses.) See [134] for a user's guide to this toolbox.

Bibliography

[1] S. Aaronson, R. Impagliazzo, and D. Moshkovitz. AM with multiple Merlins. In *Proceedings of the 29th IEEE Conference on Computational Complexity (CCC)*, pages 44–55, 2014. 54

[2] I. Adler. The equivalence of linear programs and zero-sum games. *International Journal of Game Theory*, 42(1):165–177, 2013. 9

[3] S. Alaei, H. Fu, N. Haghpanah, J. D. Hartline, and A. Malekian. Bayesian optimal auctions via multi- to single-agent reduction. In *Proceedings of the 13th Annual ACM Conference on Economics and Computation (EC)*, page 17, 2012. 90, 94, 95

[4] I. Althöfer. On sparse approximations to randomized strategies and convex combinations. *Linear Algebra and Its Applications*, 199(1):339–355, 1994. 19, 56

[5] A. Anshu, N. Goud, R. Jain, S. Kundu, and P. Mukhopadhyay. Lifting randomized query complexity to randomized communication complexity. Technical Report TR17-054, ECCC, 2017. 23, 31, 32

[6] K. J. Arrow and G. Debreu. Existence of an equilibrium for a competitive economy. *Econometrica*, 22:265–290, 1954. 79

[7] R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974. 101

[8] Y. Babichenko. Query complexity of approximate Nash equilibria. *Journal of the ACM*, 63(4):36, 2016. 29

[9] Y. Babichenko and A. Rubinstein. Communication complexity of approximate Nash equilibria. In *Proceedings of 49th Annual ACM Symposium on Theory of Computing (STOC)*, pages 878–889, 2017. 6, 22, 23, 29, 34, 37, 38

[10] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998. 46

[11] O. Ben-Zwi, R. Lavi, and I. Newman. Ascending auctions and Walrasian equilibrium. Working paper, 2013. 81

[12] S. Bikhchandani and J. W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74:385–413, 1997. 85

[13] N. Bitansky, O. Paneth, and A. Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1480–1498, 2015. 50

[14] A. Blum, M. T. Hajiaghayi, K. Ligett, and A. Roth. Regret minimization and the price of total anarchy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 373–382, 2008. 106

[15] K. C. Border. *Fixed point theorems with applications to economics and game theory*. Cambridge University Press, 1985. 24

[16] K. C. Border. Implementation of reduced form auctions: A geometric approach. *Econometrica*, 59 (4):1175–1187, 1991. URL http://www.jstor.org/stable/2938181. 87, 90, 91, 93

[17] K. C. Border. Reduced form auctions revisited. *Economic Theory*, 31:167–181, 2007. 93

[18] M. Braverman, Y. Kun Ko, and O. Weinstein. Approximating the best Nash equilibrium in $n^{o(\log n)}$-time breaks the exponential time hypothesis. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 970–982. SIAM, 2015. doi: 10.1137/1.9781611973730.66. 54

[19] M. Braverman, Y. Kun Ko, A. Rubinstein, and O. Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1326–1341, 2017. 54

[20] G. W. Brown. Iterative solutions of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13, chapter XXIV, pages 374–376. Wiley, 1951. 14

[21] Y. Cai, C. Daskalakis, and S. M. Weinberg. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 459–478, 2012. 90, 94, 95

[22] Y. Cai, C. Daskalakis, and S. M. Weinberg. Optimal multi-dimensional mechanism design: Reducing revenue to welfare maximization. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 130–139, 2012. 94

[23] Y. Cai, O. Candogan, C. Daskalakis, and C. H. Papadimitriou. Zero-sum polymatrix games: A generalization of minmax. *Mathematics of Operations Research*, 41(2):648–655, 2016. 99

[24] O. Candogan, A. Ozdaglar, and P. Parrilo. Iterative auction design for tree valuations. *Operations Research*, 63(4):751–771, 2015. 81

[25] O. Candogan, A. Ozdaglar, and P. Parrilo. Pricing equilibria and graphical valuations. *ACM Transactions on Economics and Computation*, 2017. To appear. 81

[26] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. 17

[27] N. Cesa-Bianchi, Y. Mansour, and G. Stolz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2–3):321–352, 2007. 16

[28] Y.-K. Che, J. Kim, and K. Mierendorff. Generalized reduced form auctions: A network flow approach. *Econometrica*, 81:2487–2520, 2013. 93, 94, 95

[29] X. Chen and X. Deng. 3-Nash is PPAD-complete. Technical Report TR05-134, ECCC, 2005. 49, 109

[30] X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–270, 2006. 49, 109

[31] X. Chen and X. Deng. On the complexity of 2D discrete fixed point problem. *Theoretical Computer Science*, 410(44):4448–4456, Oct. 2009. doi: 10.1016/j.tcs.2009.07.052. 48

[32] X. Chen, X. Deng, and S.-H. Teng. Computing Nash equilibria: Approximation and smoothed complexity. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–612, 2006. 49, 109

[33] X. Chen, X. Deng, and S.-H. Teng. Sparse games are hard. In *Proceedings of the Second Annual International Workshop on Internet and Network Economics (WINE)*, volume 4286 of *Lecture Notes in Computer Science*, pages 262–273, 2006. 109

[34] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3):14, 2009. doi: 10.1145/1516512.1516516. Journal version of [29], [30], [32], and [33]. 26, 49, 53, 56

[35] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, pages 59–70, 2005. 106

[36] G. Christodoulou, A. Kovács, and M. Schapira. Bayesian combinatorial auctions. *Journal of the ACM*, 63(2):11, 2016. 74

[37] G. Christodoulou, A. Kovács, A. Sgouritsa, and B. Tang. Tight bounds for the price of anarchy of simultaneous first price auctions. *ACM Transactions on Economics and Computation*, 4(2):9, 2016. 67, 74, 77

[38] V. Chvátal. *Linear Programming*. Freeman, 1983. 10, 85, 104

[39] G. B. Dantzig. A proof of the equivalence of the programming problem and the game problem. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13, chapter XX, pages 330–335. Wiley, 1951. 9

[40] G. B. Dantzig. Reminiscences about the origins of linear programming. Technical Report SOL 81-5, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1981. 9

[41] C. Daskalakis and Q. Pan. A counter-example to Karlin's strong conjecture for fictitious play. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 11–20. IEEE, IEEE Computer Society, 2014. 14

[42] C. Daskalakis and C. H. Papadimitriou. Three-player games are hard. Technical Report TR05-139, ECCC, 2005. 49, 110

[43] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 71–78, 2006. 49, 110

[44] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009. doi: 10.1137/070699652. Journal version of [42], [43], and [63]. 26, 49, 53, 56

[45] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009. 49

[46] S. Dobzinski and J. Vondrak. Communication complexity of combinatorial auctions with submodular valuations. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1205–1215, 2013. 77

[47] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. *Mathematics of Operations Research*, 35(1):1–13, 2010. 72

[48] P. Dütting, V. Gkatzelis, and T. Roughgarden. The performance of deferred-acceptance auctions. *Mathematics of Operations Research*, 42(4):897–914, 2017. 65

[49] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010. 47

[50] U. Feige. On maximizing welfare where the utility functions are subadditive. *SIAM Journal on Computing*, 39(1):122–142, 2009. 73, 74

[51] U. Feige and J. Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6(1):247–290, 2010. 77

[52] M. Feldman, H. Fu, N. Gravin, and B. Lucier. Simultaneous auctions are (almost) efficient. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 201–210, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2029-0. doi: 10.1145/2488608.2488634. URL http://doi.acm.org/10.1145/2488608.2488634. 67, 74

[53] D. P. Foster and R. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1–2):40–55, 1997. 102, 103

[54] A. Fréchette, N. Newman, and K. Leyton-Brown. Solving the station repacking problem. In *Handbook of Spectrum Auction Design*, chapter 38, pages 813–827. Cambridge University Press, 2017. 63

[55] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. 16

[56] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1–2):79–103, 1999. 15

[57] D. Fudenberg and D. K. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19(5):1065–1089, 1995. 15

[58] D. Gale, H. W. Kuhn, and A. W. Tucker. Linear programming and the theory of games. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13, chapter XIX, pages 317–329. Wiley, 1951. 9

[59] S. Garg, O. Pandey, and A. Srinivasan. Revisiting the cryptographic hardness of finding a Nash equilibrium. In *Proceedings of the 36th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO)*, pages 579–604, 2016. 50

[60] J. Geanakoplos. Nash and Walras equilibrium via Brouwer. *Economic Theory*, 21(2/3):585–603, 2003. 25

[61] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989. 103

[62] V. Gkatzelis, E. Markakis, and T. Roughgarden. Deferred-acceptance auctions for multiple levels of service. In *Proceedings of the 18th Annual ACM Conference on Economics and Computation (EC)*, pages 21–38, 2017. 65

[63] P. W. Goldberg and C. H. Papadimitriou. Reducibility among equilibrium problems. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 61–70, 2006. 49, 110

[64] M. Göös. Lower bounds for clique vs. independent set. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1066–1076, 2015. 30

[65] M. Göös, S. Lovett, R. Meka, T. Watson, and D. Zuckerman. Rectangles are nonnegative juntas. In *Proceedings of 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 257–266, 2015. 23

[66] M. Göös, T. Pitassi, and T. Watson. Deterministic communication vs. partition number. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1077–1088, 2015. doi: 10.1109/FOCS.2015.70. URL http://dx.doi.org/10.1109/FOCS.2015.70. 31, 32

[67] M. Göös, T. Pitassi, and T. Watson. Query-to-communication lifting for BPP. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science*, pages 132–143, 2017. 23, 31, 32

[68] P. Gopalan, N. Nisan, and T. Roughgarden. Public projects, Boolean functions, and the borders of Border's theorem. In *Proceedings of the 16th ACM Conference on Economics and Computation (EC)*, page 395. ACM, 2015. doi: 10.1145/2764468.2764538. 7, 87, 94, 96

[69] F. Gul and E. Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87:95–124, 1999. 81

[70] J. Hannan. Approximation to Bayes risk in repeated play. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957. 14

[71] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000. 102, 103

[72] J. D. Hartline. Mechanism design and approximation. Book draft, July 2017. 89

[73] A. Hassidim, H. Kaplan, Y. Mansour, and N. Nisan. Non-price equilibria in markets of discrete goods. In *Proceedings of the 12th Annual ACM Conference on Economics and Computation (EC)*, pages 295–296, 2011. 67

[74] M. D. Hirsch, C. H. Papadimitriou, and S. A. Vavasis. Exponential lower bounds for finding Brouwer fix points. *Journal of Complexity*, 5(4):379–416, 1989. 24, 29, 34, 35, 37, 38, 50, 53

[75] P. Hubáček and E. Yogev. Hardness of continuous local search: Query complexity and cryptographic lower bounds. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1352–1371, 2017. 50

[76] P. Hubáček, M. Naor, and E. Yogev. The journey from NP to TFNP hardness. In *Proceedings of the 8th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2017. Article 60. 50, 52

[77] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 220–229, 1997. 52

[78] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. 53

[79] A. X. Jiang and K. Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. *Games and Economic Behavior*, 91:347–359, 2015. 104

[80] D. S. Johnson. The NP-completeness column: Finding needles in haystacks. *ACM Transactions on Algorithms*, 3(2):24, 2007. 49

[81] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. 46

[82] S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 42–47, 2003. 104

[83] B. Kalyanasundaram and G. Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992. 23

[84] S. Karlin. *Mathematical Methods and Theory in Games, Programming, and Economics*. Addison-Wesley, 1959. 14

[85] M. Kearns. Graphical games. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 7, pages 159–180. Cambridge University Press, 2007. 99

[86] M. Kearns, M. L. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 253–260, 2001. 103

[87] A. S. Kelso and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica*, 50(6):1483–1504, 1982. 81

[88] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20 (1):191–194, 1979. 84, 96, 104

[89] T. H. Kjeldsen. John von Neumann's conception of the Minimax theorem: A journey through different mathematical contexts. *Archive for History of Exact Sciences*, 56:39–68, 2001. 25

[90] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45:181–221, 2003. 103

[91] S. Kopparty, O. Meir, N. Ron-Zewi, and S. Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *Journal of the ACM*, 64(2):11, 2017. 59

[92] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS)*, pages 404–413, Berlin, Heidelberg, 1999. Springer-Verlag. URL http://dl.acm.org/citation.cfm?id=1764891.1764944. 67, 73

[93] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1996. 23, 70, 78

[94] C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215–217, 1983. 50

[95] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55:270–296, 2006. 83

[96] C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *SIAM Journal*, 12(2): 413–423, 1964. 19, 47

[97] K. Leyton-Brown, P. Milgrom, and I. Segal. Economics and computer science of a radio spectrum reallocation. *Proceedings of the National Academy of Sciences (PNAS)*, 114(28):7202–7209, 2017. 63

[98] R. J. Lipton and N. E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *Proceedings of 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 734–740, 1994. 19

[99] R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce (EC)*, pages 36–41. ACM, 2003. doi: 10.1145/779928.779933. 20, 76

[100] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994. 16

[101] E. Maskin and J. Riley. Optimal auctions with risk-adverse buyers. *Econometrica*, 52:1473–1518, 1984. 90

[102] S. A. Matthews. On the implementability of reduced form auctions. *Econometrica*, 52:1519–1522, 1984. 90

[103] A. McLennan. Advanced fixed point theory for economics. Book in preparation, 2015. 24

[104] A. McLennan and R. Tourky. From imitation games to Kakutani. Unpublished manuscript, 2006. 38

[105] N. Megiddo and C. H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991. 44

[106] P. Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108(2):245–272, 2000. 81

[107] P. Milgrom. *Putting Auction Theory to Work*. Churchill Lectures in Economics. Cambridge University Press, 2004. 65

[108] P. Milgrom and I. Segal. Deferred-acceptance auctions and radio spectrum reallocation. In *Proceedings of the 15th ACM Conference on Economics and Computation (EC)*, pages 185–186, New York, NY, USA, 2014. ACM. doi: 10.1145/2600057.2602834. 62, 64

[109] W. D. Morris, Jr. Lemke paths on simple polytopes. *Mathematics of Operations Research*, 19: 780–789, 1994. 19

[110] H. Moulin and J. P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3–4):201–221, 1978. 101

[111] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981. 89

[112] S. Nasar. *A Beautiful Mind: a Biography of John Forbes Nash, Jr., Winner of the Nobel Prize in Economics, 1994*. Simon & Schuster, 1998. 19

[113] J. F. Nash, Jr. Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. 19

[114] J. F. Nash, Jr. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951. 19

[115] N. Nisan. The communication complexity of approximate set packing and covering. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 868–875. Springer-Verlag, 2002. URL http://dl.acm.org/citation.cfm?id=646255.684594. 70, 71

[116] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129:192–224, 2006. 84

[117] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. 45, 48, 49

[118] C. H. Papadimitriou. The complexity of finding Nash equilibria. In N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 2, pages 29–51. Cambridge, 2007. 49

[119] C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14, 2008. 104

[120] R. Raz and P. McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi: 10.1007/s004930050062. URL http://dx.doi.org/10.1007/s004930050062. 6, 31, 32

[121] A. A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106 (2):385–390, 1992. 23

[122] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951. 14

[123] A. Rosen, G. Segev, and I. Shahaf. Can PPAD hardness be based on standard cryptographic assumptions? In *Proceedings of the 15th International Conference on Theory of Cryptography (TCC)*, pages 173–205, 2017. 50

[124] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005. 67

[125] T. Roughgarden. Computing equilibria: A computational complexity perspective. *Economic Theory*, 42(1):193–236, 2010. 49

[126] T. Roughgarden. Barriers to near-optimal equilibria. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 71–80. IEEE Computer Society, 2014. doi: 10.1109/FOCS.2014.16. 7, 68, 75

[127] T. Roughgarden. CS364B lecture notes. Stanford University, 2014. 74

[128] T. Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM*, 62(5):32, 2015. 106

[129] T. Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, 2016. 25, 43, 65, 89, 101

[130] T. Roughgarden. Communication complexity (for algorithm designers). *Foundations and Trends in Theoretical Computer Science*, 11(3-4):217–404, 2016. 23, 69, 70

[131] T. Roughgarden and I. Talgam-Cohen. Why prices need algorithms. In *Proceedings of the 16th Annual ACM Conference on Economics and Computation (EC)*, pages 19–36, 2015. 7, 79, 82, 86

[132] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002. 67

[133] T. Roughgarden and O. Weinstein. On the communication complexity of approximate fixed points. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 229–238, 2016. 29, 32

[134] T. Roughgarden, V. Syrgkanis, and É.. Tardos. The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, 59:59–101, 2017. 67, 106

[135] A. Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 258–265, 2016. 7, 24, 34, 38, 43, 48, 49, 53, 54, 57, 59, 60

[136] R. Savani and B. von Stengel. Hard-to-solve bimatrix games. *Econometrica*, 74(2):397–429, 2006. 19

[137] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986. 96

[138] L. S. Shapley. Some topics in two-person games. In M. Dresher, L. S. Shapley, and A. W. Tucker, editors, *Advances in Game Theory*, pages 1–28. Princeton University Press, 1964. 14, 18

[139] E. Solan and R. Vohra. Correlated equilibrium payoffs and public signalling in absorbing games. *International Journal of Game Theory*, 31:91–121, 2002. 105

[140] D. A. Spielman. The complexity of error-correcting codes. In *Proceedings of the 11th International Symposium on Fundamentals of Computation Theory*, pages 67–84, 1997. 55

[141] D. A. Spielman and S.-H. Teng. Smoothed analysis: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004. 49

[142] N. Sun and Z. Yang. Equilibria and indivisibilities: Gross substitutes and complements. *Econometrica*, 74(5):1385–1402, 2006. 81

[143] V. Syrgkanis and É. Tardos. Composable and efficient mechanisms. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC)*, pages 211–220, 2013. 77, 106

[144] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991. 96

[145] A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–425, 2002. 106

[146] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1): 8–37, 1961. 87

[147] J. Ville. Sur la theorie générale des jeux ou intervient l'habileté des joueurs. Fascicule 2 in Volume 4 of É. Borel, *Traité du Calcul des probabilités et de ses applications*, pages 105–113. Gauthier-Villars, 1938. 9

[148] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928. 9

[149] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 9

[150] B. von Stengel. Equilibrium computation for two-player games in strategic and extensive form. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 3, pages 53–78. Cambridge University Press, 2007. 19