Title
Hardness of Approximation Between P and NP

Author
Rubinstein, Aviad

Publication Date
2017

Peer reviewed|Thesis/dissertation

Hardness of Approximation Between P and NP

by

Aviad Rubinstein

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Christos Papadimitriou, Chair
Professor Ilan Adler
Associate Professor Prasad Raghavendra
Professor Satish Rao

Summer 2017

Hardness of Approximation Between P and NP

Abstract

Hardness of Approximation Between P and NP

by

Aviad Rubinstein

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Christos Papadimitriou, Chair

Nash equilibrium is the central solution concept in Game Theory. Since Nash's original paper in 1951, it has found countless applications in modeling strategic behavior of traders in markets, (human) drivers and (electronic) routers in congested networks, nations in nuclear disarmament negotiations, and more. A decade ago, the relevance of this solution concept was called into question by computer scientists [DGP09; CDT09], who proved (under appropriate complexity assumptions) that *computing* a Nash equilibrium is an intractable problem. And if centralized, specially designed algorithms cannot find Nash equilibria, why should we expect distributed, selfish agents to converge to one? The remaining hope was that at least approximate Nash equilibria can be efficiently computed.

Understanding whether there is an efficient algorithm for *approximate Nash equilibrium* has been the central open problem in this field for the past decade. In this thesis, we provide strong evidence that even finding an approximate Nash equilibrium is intractable. We prove several intractability theorems for different settings (two-player games and many-player games) and models (computational complexity, query complexity, and communication complexity). In particular, our main result is that under a plausible and natural complexity assumption ("Exponential Time Hypothesis for PPAD"), there is no polynomial-time algorithm for finding an approximate Nash equilibrium in two-player games.

The problem of approximate Nash equilibrium in a two-player game poses a unique technical challenge: it is a member of the class PPAD, which captures the complexity of several fundamental total problems, i.e. problems that always have a solution; and it also admits a quasipolynomial ($\approx n^{\log n}$) time algorithm. Either property alone is believed to place this problem far below NP-hard problems in the complexity hierarchy; having both simultaneously places it just above P, at what can

be called the frontier of intractability. Indeed, the tools we develop in this thesis to advance on this frontier are useful for proving hardness of approximation of several other important problems whose complexity lies between P and NP:

Brouwer's fixed point Given a continuous function f mapping a compact convex set to itself, Brouwer's fixed point theorem guarantees that f has a fixed point, i.e. x such that $f(x) = x$. Our intractability result holds for the relaxed problem of finding an approximate fixed point, i.e. x such that $f(x) \approx x$.

Market equilibrium Market equilibrium is a vector of prices and allocations where the supply meets the demand for each good. Our intractability result holds for the relaxed problem of finding an approximate market equilibrium, where the supply of each good approximately meets the demand.

CourseMatch (A-CEEI) Approximate Competitive Equilibrium from Equal Income (A-CEEI) is the economic principle underlying CourseMatch, a system for fair allocation of classes to students (currently in use at Wharton, University of Pennsylvania).

Densest k-subgraph Our intractability result holds for the following relaxation of the k-Clique problem: given a graph containing a k-clique, the algorithm has to find a subgraph over k vertices that is "almost a clique", i.e. most of the edges are present.

Community detection We consider a well-studied model of communities in social networks, where each member of the community is friends with a large fraction of the community, and each non-member is only friends with a small fraction of the community.

VC dimension and Littlestone dimension The Vapnik-Chervonenkis (VC) dimension is a fundamental measure in learning theory that captures the complexity of a binary concept class. Similarly, the Littlestone dimension is a measure of complexity of online learning.

Signaling in zero-sum games We consider a fundamental problem in signaling, where an informed signaler reveals private information about the payoffs in a two-player zero-sum game, with the goal of helping one of the players.

Contents

List of Figures

List of Tables

Acknowledgments

I am incredibly lucky to have Christos as my advisor. I cannot compete with the praise written in dozens of Acknowledgment Sections in the dissertations of previous students of Christos. I can only attest that it's all true. Christos, thank you for giving me beautiful problems to think about, for sprinkling your magic over my introductions, for the pre-deadline nights, and for showing me around Greece (in particular, Ikaria). Most importantly, thanks for all the advice!

I also thank my thesis committee, Ilan Adler, Prasad Raghavendra, Satish Rao, and Christos. Their feedback during my qualifying exam was already tremendously helpful, and was my starting point for the last two parts of my thesis.

For the past few years it has been a great pleasure to belong to the theory group at Berkeley. The lunches, retreats, basketball and soccer games, and occasional talks were an excellent inspiration. In particular, I was extremely fortunate to be part of *the office*: Jonah Brown-Cohen, Rishi Gupta, Alex Psomas, Tselil Schramm, Jarett Schwartz, Ning Tan, Ben Weitz, thanks!

The latest perk of being a theorist at Berkeley is the Simons Institute. Each semester brings an influx of fascinating new and old visitors and learning opportunities. Indeed, much of this thesis was first written on the white boards in the collaboration area. Thanks Dick Karp, Christos, Alistair Sinclair, Luca Trevisan, program organizers, and the Simons Foundation for spoiling me.

Before coming to Berkeley, I was particularly influenced by long conversations with Elad Haramaty and my M.Sc. advisor Muli Safra. I don't believe I would be doing theory if it weren't for Muli; I probably would not be in academia at all if it weren't for Elad.

I am grateful to Microsoft Research for the MSR PhD Fellowship, as well as wonderful summer internships, where I had the fortune to learn from Moshe Babaioff, Siu On Chan, Wei Chen, Jason Hartline, Bobby Kleinberg, Nicole Immorlica, Pinyan Lu, Brendan Lucier, Yishay Mansour, Noam Nisan, Moshe Tennenholtz, and many others. I also thank Mark Braverman, Michal Feldman, Noam Nisan, and Yaron Singer for hosting me for shorter visits during my PhD.

I am also grateful to all my coauthors: Amir Abboud, Ilan Adler, Yakov Babichenko, Ashwin Badanidiyuru, Eric Balkanski, Mark Braverman, Siu On Chan, Wei Chen, Michal Feldman, Ophir Friedler, Young Kun Ko, Fu Li, Tian Lin, Adi Livnat, Yishay Mansour, Pasin Manurangsi, Abe Othman, Christos, Dimitris Papailiopoulos, George Pierrakos, Alex, Tselil, Lior Seeman, Yaron Singer, Sahil Singla, Moshe Tennenholtz, Greg Valiant, Shai Vardi, Andrew Wan, Matt Weinberg, Omri Weinstein, and Ryan Williams.

Thanks also to Boaz Barak, Shai Ben-David, Jonah, Karthik C.S., Yang Cai, Alessandro Chiesa, Paul Christiano, Constantinos Daskalakis, Shaddin Dughmi, Mika Goos, Rishi, Elad, Noam, Prasad, Tselil, Madhu Sudan, Luca Trevisan, Michael Viderman, and anonymous reviewers, for commenting on, correcting, and inspiring much of the content of this thesis.

This has been an amazing journey. I am most grateful for the opportunities I had to share it with my family and friends.

Part I

Overview

Chapter 1

The frontier of intractability

The combination of vast amounts of data, unprecedented computing power, and clever algorithms allows today's computer systems to drive autonomous cars, beat the best human players at Chess and Go, and live stream videos of cute cats across the world. Yet computers can fail miserably at predicting outcomes of social processes and interactions, elections being only the most salient example. And this is hardly surprising, as there are two potent reasons for such unpredictability: One reason is that human agents are driven by emotions and hormones and billions of neurons interacting with a complex environment, and as a result their behavior is extremely hard to model mathematically. The second reason is perhaps a bit surprising, and certainly closer to the concerns of this thesis: *Even very simple and idealized models of social interaction, in which agents have extremely clear-cut objectives, can be intractable.*

To have any hope of reasoning about human behavior on national and global scales, we need a mathematically sound theory. Game Theory is the mathematical discipline that models and analyzes the interaction between agents (e.g. voters) who have different goals and are affected by each other's actions. The central solution concept in game theory is the *Nash equilibrium*. It has an endless list of applications in economics, politics, biology, etc. (e.g. [Aum87]).

By Nash's theorem [Nas51], an equilibrium always exists. Furthermore, once at a Nash equilibrium, players have no incentive to deviate. The main missing piece in the puzzle is:

> *How do players arrive at an equilibrium in the first place?*

After many attempts by economists for more than six decades[1] (e.g. [Bro51; Rob51;

[1] In fact, "more than six decades" is an understatement: Irving Fisher's thesis from 1891 dealt

LH64; Sha64; Sca67; KL93; HMC03; FY06]), we still don't have a satisfying explanation.

About ten years ago, the study of this fundamental question found a surprising answer in computer science: Chen et al [CDT09] and Daskalakis et al [DGP09] proved that finding a Nash equilibrium is computationally intractable[2]. And if no centralized, specialized algorithm can find an equilibrium, it is even less likely that distributed, selfish agents will naturally converge to one. This casts doubt over the entire solution concept.

For the past decade, the main remaining hope for Nash equilibrium has been *approximation*. The central open question in this field has been:

> *Is there an efficient algorithm for finding an approximate Nash equilibrium?*

In this thesis we give a strong negative resolution to this question: our main result rules out efficient approximation algorithms for finding Nash equilibria[3].

Our theorem is the latest development in a long and intriguing technical story. The first question we computer scientists ask when encountering a new algorithmic challenge is: is it in P, the class of polynomial time (tractable) problems; or is it NP-hard, like Satisfiability and the Traveling Salesperson Problem (where the best known algorithms require exponential time)? Approximate Nash equilibrium falls into neither category; its complexity lies between P and NP-hard — hence the title of our thesis. Let us introduce two (completely orthogonal) reasons why we do not expect it to be NP-hard.

The first obstacle for NP-hardness is the *totality* of Nash equilibrium. When we say that Satisfiability or the Traveling Salesperson Person problems are NP-hard, we formally mean that it is NP-hard to determine *whether* a given formula has a satisfying assignment, or *whether* a given network allows the salesperson to complete her travels within a certain budget. In contrast, by Nash's theorem an equilibrium always exists. Deciding whether an equilibrium exists is is trivial, but we still don't know how to find one. This is formally captured by an intermediate complexity class called PPAD.

The second obstacle for NP-hardness is an algorithm (the worst kind of obstacle for intractability). The best known algorithms for solving NP-hard or PPAD-hard problems require exponential ($\approx 2^n$) time, and there is a common belief (formulated

with the closely related question of convergence to market equilibria [BS00].

[2]Assuming P $\neq$ PPAD; see the discussion in the next section about this assumption.

[3]Under a complexity assumption stronger than P $\neq$ PPAD, that we call the Exponential Time Hypothesis (ETH) for PPAD, see Section 1.2 for details.

a few years ago as the "Exponential Time Hypothesis" [IPZ01]) that much faster algorithms do not exist. In contrast, an approximate Nash equilibrium can be found in *quasi-polynomial* ($\approx n^{\log n}$) time. Notice that this again places the complexity of approximate Nash equilibrium between the polynomial time solvable problems in P and the exponential time required by NP-hard problems. Therefore, approximate Nash equilibrium is unlikely to be NP-hard — or even PPAD-hard, since we know of no quasi-polynomial algorithm for any other PPAD-hard problem.

As illustrated in the last two paragraphs, approximate Nash equilibrium is very far from our standard notion of intractability, NP-hardness. In some sense, it is one of the computationally easiest problems which we can still prove to be intractable — it lies at the *frontier of our understanding of intractability*. Unsurprisingly, the techniques we had to master to prove the intractability of approximate Nash equilibrium are useful for many other problems. In particular, we also prove hardness of approximation for several other interesting problems that either belong to the class PPAD or have quasi-polynomial time algorithms.

1.1 PPAD: Finding a needle you *know* is in the haystack

Consider a directed graph G. Each edge contributes to the degree of the two vertices incident on it. Hence the sum of the degrees is twice the number of edges, and in particular it is even. Now, given an odd-degree vertex v, there must exist another odd-degree vertex. But *can you find one?* There is, of course, the trivial algorithm which simply brute-force enumerates over all the graph vertices.

Now suppose G further has the property[4] that every vertex has in- and out-degree at most 1. Thus, G is a disjoint unions of lines and cycles. If v has an outgoing edge but no incoming edge, it is the beginning of a line. Now the algorithm for finding another odd degree node is a bit more clever — follow the path to the end — but its worst case is still linear in the number of nodes.

In the worst case, both algorithms run in time linear in the number of vertices. When the graph is given explicitly, e.g. as an adjacency matrix, this is quite efficient compared to the size of the input. But what if the graph is given as black-box oracles S and P that return, for each vertex, its successor and predecessor in the graph? The

[4]This guarantee is actually without loss of generality [Pap94], but this is not so important for our purposes.

amount of work[5] required for finding another odd-degree vertex by the exhaustive algorithm, or the path-following algorithm, is still linear in the number of vertices, but now this number is exponential in the natural parameter of the problem, the length of the input to the oracle, which equals the *logarithm* of the number of nodes. In the oracle model, it is not hard to prove that there are no better algorithms.

Let us consider the computational analog of the black-box oracle model: the oracles S and P are implemented as explicit ("white-box") circuits, which are given as the input to the algorithm. This is the End-of-a-Line problem, which is the starting point for most of our reductions. The computational class PPAD (which stands for Polynomial Parity Argument in Directed graphs) is the class of search problems reducible to End-of-a-Line.

Definition 1.1.1 (End-of-a-Line [DGP09])**.** Given two circuits S and P, with m input bits and m output bits each, such that $P(0^m) = 0^m \neq S(0^m)$, find an input $x \in \{0,1\}^m$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^m$.

How hard is End-of-a-Line? We believe that it is likely to be very hard, almost as hard as the black-box variant. This belief is backed by relativized separations [Mor03] and cryptographic hardness [BPR15; GPS16; HY17], as well as zero algorithmic progress, even on special cases, since it was first introduced in [Pap94]. More importantly, we are embarrassingly bad at gaining insight to a circuit functionality by looking at its structure (with some notable exceptions in cryptography [Bar04]) — hence it seems reasonable to conjecture that the white-box variant is easier than the black-box problem.

Even more embarrassing is our inability to prove computational intractability. Essentially all "proofs" of computational intractability (including the ones in this thesis) are *conditional*; i.e. we assume that some problem is hard (e.g. End-of-a-Line or Satisfiability), and reduce it to a new problem we want to "prove" is hard. The intractability of the new problem only holds conditioned on the intractability of the original hard problem. Without first resolving whether P = NP, we cannot prove that End-of-a-Line is indeed hard. Furthermore, even merely proving that End-of-a-Line is NP-hard would already imply unlikely consequences like NP = coNP [MP91]. The ultimate reason we believe that End-of-a-Line is intractable is that we have no idea how to prove it.

Once we believe that End-of-a-Line is indeed intractable, any problem that is PPAD-complete, i.e. "at least as hard as End-of-a-Line" is also intractable. The celebrated works of [DGP09; CDT09] prove that finding an exact Nash equilibrium

[5]Here *work* is measured by the number of oracle calls rather than running time; indeed this model will be the starting point of our reductions for query and communication complexity.

is PPAD-complete. In Section 1.3 we describe some variants of approximate Nash equilibrium that are also PPAD-complete.

We conclude this section with a few problems (other than variants of Nash equilibrium) that we show are also PPAD-hard:

- Finding an approximate fixed point of an implicit function; Brouwer's fixed point theorem, which guarantees the existence of a fixed point, lays the mathematical foundation for the rest of our PPAD-complete problems.
- Finding an approximate market equilibrium, which is the conceptual foundation of neoclassical economics.
- Finding an Approximate Competitive Equilibrium from Equal Incomes (A-CEEI), which is an algorithmic problem of practical interest due to its use for allocating classes to students (CourseMatch).

1.1.1 Brouwer's fixed point

Brouwer's fixed point theorem, together with its generalizations (in particular, Kakutani's fixed point theorem), is the basis for many of the equilibrium concepts in game theory and economics. It states that any continuous function f from a compact convex set (in particular, the n-dimensional hypercube) to itself has a fixed point, i.e. a point x^* such that $f(x^*) = x^*$. Just like in the case of a Nash equilibrium and the odd-degree vertex, the existence does not come with an algorithm for finding the fixed point. Brouwer eventually became so unsatisfied with the non-constructive nature of his proof that he founded intuitionism, a formalism of mathematics mandating constructive proofs [Iem16].

Before we discuss the computational tractability of finding a fixed point, there are two subtle but important technicalities that we have to clear. The first is that finding an exact fixed point may be "obviously" intractable when all the fixed points are irrational. Fortunately there is a natural notion[6] of approximate fixed point: find x such that $f(x) \approx x$. The second issue is that we want a finite description of the input to our algorithm. Naively, one can define the value of the function on a grid, but then the set is no longer convex and a fixed point might not exist at all (see e.g. [RW16]). It turns out that a better way to define the input is as an arithmetic circuit: this gives a local and easy-to-verify guarantee that the function is indeed continuous.

[6]There is also a stricter notion that requires x to be close to an x^* for which $f(x^*) = x^*$ exactly. See e.g. [EY10].

Admittedly, there is something dissatisfying about proving computational hardness of "circuit problems": we are going to prove that this problem is PPAD-hard, i.e. no easier than End-of-a-Line — but the reason we believe in the first place that End-of-a-Line is intractable is that we don't know how to gain insight to the functionality of the circuit from staring at its structure. Nevertheless, we begin with the complexity of Brouwer's fixed point because: this is a fundamental question; it is the starting point of many of our reductions; and, as we discuss later, even the black-box hardness in this case is highly non-trivial.

We know of two algorithms for computing approximate Brouwer fixed point: there is Scarf's algorithm [Sca67], but its running time may be exponential in the description of the function; the same holds for brute-force search.

On the complexity side, [DGP09] proved that even in three dimensions, finding an exponentially-close approximation (x such that $\|f(x) - x\|_\infty \leq 2^{-n}$, where n is the size of the input) is PPAD-complete; [CD09] proved the same problem continues to be hard even with only two dimensions. Notice that with a constant number of dimensions, any weaker approximation desideratum becomes trivial for brute-force search. [CDT09] showed that in n dimensions, polynomial approximations are also PPAD-complete. We will later prove that even constant approximation are PPAD-complete, i.e. there exists some absolute constant $\varepsilon > 0$ such that it's hard to find an x for which $\|f(x) - x\|_\infty \leq \varepsilon$; this is already a significant improvement over the existing state of the art.

We can prove an even stronger form of inapproximability for finding a Brouwer fixed point: so far, we characterized the approximate fixed point with ℓ_∞-norm, which means that $f(x)$ and x must be close in every coordinates. Our main result in this regard (Theorem 4.2.1) is that even if we only require that $f(x)$ and x are close in *most* of the coordinates, finding such x and $f(x)$ remains PPAD-complete.

1.1.2 Market equilibrium

Supply and demand is central to our modern understanding of economics: when demand exceeds supply, raising the price makes it less attractive to consumers and more attractive to producers, thereby reducing demand and increasing supply. Vice versa if the supply exceeds the demand. This simple idea marks the birth of neoclassical economics [Jev66; Men71; Wal74], and continues to inspire us today when we reason about free market economy. Since the consumption and production of one good depends on other goods, we are interested in a *market equilibrium*: a vector of prices where the supply and demand of every good matches.

The supply and demand argument has many weak spots, one of which is Giffen goods [Mar95]: Consider a poor 19th century English family consuming a 10,000

calories a day from meat and bread. They prefer to eat meat, but the bread is cheaper - so they spend their daily income on 6 loafs of bread and 4 pounds of meat (each provides 1,000 calories). What happens the price of bread increases? They have less free money to spend on the luxury good (meat), which *increases* their demand for bread. The existence of Giffen goods in practice has been contentiously debated for over a century [Edg09; Sti47], with evidence ranging from econometric analasys of historical market data to experiments in lab rats [BKK91].

Despite counter-intuitive issues like Giffen goods, Arrow and Debreu proved that, under quite general conditions, a market equilibrium always exists [DA54]. In particular, in the Arrow-Debreu model, agents sell the goods they own and use the revenue to buy other goods they want; this is in contrast to Fisher markets where agents with a monetary budget purchase from a centralized sellers. Market equilibria in both models exist, but can we find them? As in the case of Nash equilibrium, this question is of particular importance because if a centralized, omniscient algorithm cannot compute an equilibrium, it is hard to expect a distributed market with selfish agents to converge to one. In the words of Kamal Jain [Jai04; Nis09b]:

> "If your laptop cannot find it, neither can the market."

In the same paper, Jain also gave an algorithm for computing Arrow-Debreu's equilibrium when consumers utilities are linear. Since then, there has been a long sequence of algorithms for computing or approximating market equilibria (e.g. [Jai04; CMV05; GK06; Dev+08; CF08; JV10; BDX11; Gar+15]) for certain models and utility functions, as well as intractability results in other settings [DD08; Che+09; VY11; Gar+17]. In particular, Chen, Paparas, and Yannakakis [CPY13] consider a setting of markets which exhibit *non-monotonicity*: when the price of one product increases, its demand increases. Giffen goods described above are one example of non-monotone markets; [CPY13] construct examples in Arrow-Debreu markets when the increased price of a product increases the revenue of the seller, who now has more available money and demands more of the same good. Chen, Paparas, and Yannakakis [CPY13] show that for essentially any class of utility function that exhibits some non-monotonicity, computing the Arrow-Debreu market equilibrium is PPAD-hard.

We extend the PPAD-hardness proof of [CPY13] and show that even approximate market equilibrium is PPAD-hard (Theorem 9.0.4). We note that although our inapproximability factor is stronger than that showed by Chen et al, the results are incomparable as ours only holds for the stronger notion of "tight" approximate equilibrium, by which we mean the more standard definition which bounds the two-sided error of the market equilibrium. Chen et al, in contrast, prove that even if we allow arbitrary excess supply, finding a $(1/n)$-approximate equilibrium is PPAD-hard.

Furthermore, for the interesting case of CES utilities with parameter $\rho < 0$, they show that there exist markets where every (1/2)-tight equilibrium requires prices that are doubly-exponentially large (and thus require an exponential-size representation). Indeed, for a general non-monotone family of utility functions, the problem of computing a (tight or not) approximate equilibrium may not belong to PPAD. Nevertheless, the important family of additively separable, concave piecewise-linear utilities is known to satisfy the non-monotone condition [CPY13], and yet the computation of (exact) market equilibrium is in PPAD [VY11]. Therefore, we obtain as a corollary that computing an ε-tight approximate equilibrium for Arrow-Debreu market with additively separable, concave piecewise-linear utilities is PPAD-complete.

1.1.3 A-CEEI (CourseMatch)

University courses have limited capacity, and some are more popular than others. This creates an interesting allocation problem. Imagine that each student has ordered all the possible schedules—bundles of courses—from most desirable to least desirable, and the capacities of the classes are known. What is the best way to allocate seats in courses to students? There are several desiderata for a course allocation mechanism:

Fairness In what sense is the mechanism "fair"?

Efficiency Are seats in courses allocated to the students who want them the most?

Feasibility Are any courses oversubscribed?

Truthfulness Are students motivated to honestly report their preferences to the mechanism?

Computational efficiency Can the allocation be computed from the data in polynomial time?

Competitive Equilibrium from Equal Incomes (CEEI) [Fol67; Var74; TV85] is a venerable mechanism with many attractive properties: In CEEI all agents are allocated the same amount of "funny money", next they declare their preferences, and then a price equilibrium is found that clears the market. The market clearing guarantees Pareto efficiency and feasibility. The mechanism has a strong, albeit technical, *ex post* fairness guarantee that emerges from the notion that agents who miss out on a valuable, competitive item will have extra funny money to spend on other items at equilibrium. Truthfulness is problematic—as usual with market mechanisms—but potential incentives for any individual agent to deviate are mitigated by the large number of agents. However, CEEI only works when the resources to be allocated are

divisible and the utilities are relatively benign. This restriction has both benefits and drawbacks. It ensures computational feasibility, because CEEI can be computed in polynomial time with a linear or convex program, depending on the utilities involved [Var74; Dev+08; Gho+11]; on the other hand, it is easy to construct examples in which a CEEI does not exist when preferences are complex or the resources being allocated are not divisible. Indeed, both issues arise in practice in a variety of allocation problems, including shifts to workers, landing slots to airplanes, and the setting that we focus on here, courses to students [Var74; Bud11].

It was shown in [Bud11] that an approximation to a CEEI solution, called A-CEEI, exists even when the resources are indivisible and agent preferences are arbitrarily complex, as required by the course allocation problems one sees in practice. The approximate solution guaranteed to exist is approximately fair (in that the students are given almost the same budget), and approximately Pareto efficient and feasible (in that all courses are filled close to capacity, with the possible exception of courses with more capacity than popularity). This result seems to be wonderful news for the course allocation problem. However, there is a catch: Budish's proof is non-constructive as it relies on Kakutani's fixed-point theorem.

A heuristic search algorithm for solving A-CEEI was introduced in [OSB10]. The algorithm resembles a traditional t\^atonnement process, in which the prices of courses that are oversubscribed are increased and the prices of courses that are undersubscribed are decreased. A modified version of this algorithm that guarantees courses are not oversubscribed is currently used by the Wharton School (University of Pennsylvania) to assign their MBA students to courses [Bud+14]. While it has been documented that the heuristic algorithm often produces much tighter approximations than the theoretical bound, on some instances it fails to find even the guaranteed approximation [Bud11, Section 9].

Thus A-CEEI is a problem where practical interest motivates theoretical inquiry. We have a theorem that guarantees the existence of an approximate equilibrium—the issue is finding it. Can the heuristic algorithms currently used to assign Wharton MBAs to their courses be replaced by a fast and rigorous algorithm for finding an approximate CEEI? In this thesis, we answer this question on the negative (Theorem 10.1.5), showing that computing an A-CEEI is PPAD-complete.

1.2 Quasi-polynomial time and the birthday paradox

The following bilinear optimization meta-problem captures a wide range of applications, from areas like statistics (Sparse PCA), graph theory (Clique), and game theory (Nash equilibrium):

$$\max_{(x,y)\in\mathcal{X}} x^\top A y. \tag{1.1}$$

For all the applications we consider, once we fix some y^*, finding the best feasible x that maximizes $x^\top A y^*$ is a tractable problem. (Similarly, if we were given a good x^*, finding a matching y is easy.) But optimizing x and y simultaneously is NP-hard. What about approximations?

Caratheodory's theorem states that a point v in the convex hull of n points in $\mathbb{R}^d$ can be written as a convex combination of $d+1$ points. In general, $d+1$ points are necessary, but this number can be reduced drastically if we are willing to settle for approximation. In particular, Barman[7] [Bar15] proves an approximate Caratheodory's theorem that requires only $r = O(p/\varepsilon^2)$ points to express a point $\hat{v}$ such that $\|\hat{v} - v\|_p < \varepsilon$, assuming the n points belong to a unit ℓ_p-ball. In particular, $\hat{v}$ can be written as an average over a multi-set of r out of the n points.

Viewing the columns of A as n vectors in $\mathbb{R}^n$, Barman observes that (after proper normalization), the point $v = Ay^*$ is in their convex hull. If only want to approximately solve the bilinear optimization (1.1), we drastically reduce the dimension of the problem by enumerating over all choices of $\hat{v}$, and for each $\hat{v}$ solving the optimization problem $A\hat{v}$. It turns out that in order to guarantee a good additive approximation of (1.1), we need to set $p \approx \log n$. The dominant term in the running time of this meta-algorithm is the enumeration over all choices of $\hat{v}$ (all multi-sets of the columns of A), which takes approximately $n^r = n^{O\left(\frac{\log n}{\varepsilon^2}\right)}$, i.e. quasi-polynomial time.

The above meta-algorithm provides evidence that the approximate variant of all those problems is much easier than solving them exactly: in particular we believe that NP-hard (respectively, PPAD-hard) problems like 3-SAT (resp. End-of-a-Line) require approximately 2^n time. This belief is formulated by the *Exponential Time Hypothesis*, or ETH [IPZ01] (resp. ETH for PPAD [BPR16]). The reasons we believe in ETH are similar to the ones outlined in the previous section for our belief that

[7] Both the approximate Caratheodory's theorem and the resulting algorithm have been discovered been described by many researchers (e.g. [LMM03; Aro+12]); however, the presentation in [Bar15] is our favorite.

End-of-a-Line is intractable: on one hand, we have a combination of unconditional lower bounds in analogous models and little or no progress on algorithms; on the other hand, we are "forced" to believe it because we have no idea how to prove it (in particular, ETH implies the much weaker $\mathsf{P} \neq \mathsf{NP}$).

However, quasi-polynomial time is still both unrealistic to implement in practice, and does not meet our gold standard of polynomial time in theory. The main question we address in this section is:

Can the quasi-polynomial time be improved to polynomial time?

For Sparse PCA this is indeed possible [Alo+13; Ast+15; CPR16]. But for several other problems we can prove that, assuming ETH, quasi-polynomial time is actually necessary:

- Finding a k-subgraph which is almost a clique; this is one of the most fundamental approximation problems in theoretical computer science.
- Finding and counting stable communities in a friendship graph; this problem received a lot of attention in recent years with the emergence online social networks.
- Finding an approximately optimal signaling scheme; this resolves a recent open question by Dughmi .
- Computing the VC and Littlestone's dimensions of a binary concept class, two of the most fundamental quantities in learning theory.

A common approach to all our proofs is the *birthday repetition* framework due to Aaronson, Impagliazzo, and Moshkovitz [AIM14]: construct a reduction from 3-SAT to any of the above problems, with reduction size $N \approx 2^{\sqrt{n}}$. Then, assuming ETH, one needs approximately $N^{\log N} \approx 2^n$ time to solve the problem on the larger instance. A key step in the reduction is to consider subsets of $\sqrt{n}$ variables; then by the *birthday paradox* any two subsets are likely to share at least one variable (hence the name "birthday repetition").

1.2.1 Densest k-Subgraph

k-Clique is one of the most fundamental problems in computer science: given a graph, decide whether it has a fully connected induced subgraph on k vertices. Since it was proven NP-complete by Karp [Kar72], extensive research has investigated the complexity of its relaxations.

We consider two natural relaxations of k-Clique which have received significant attention from both algorithmic and complexity communities: The first one is to relax the "k" requirement, i.e. looking for a smaller subgraph: Given an n-vertex graph G containing a clique of size k, find a clique of size at least δk for some parameter $0 < \delta < 1$.

The second natural relaxation is to relax the "Clique" requirement, replacing it with the more modest goal of finding a subgraph that is almost a clique: Given an n-vertex graph G containing a clique of size k, find an induced subgraphs of G of size k with (edge) density at least $1 - \varepsilon$, for some parameter $0 < \varepsilon < 1$.

The first relaxation has been a motivating example throughout a long line of research that laid the foundations for NP-hardness of approximation [Fei+96; AS98; Aro+98; Hås99; Kho01; Zuc07]. In particular, we now know that it is NP-hard to distinguish between a graph that has a clique of size k, and a graph whose largest induced clique is of size at most $k' = \delta k$, where $\delta = 1/n^{1-\varepsilon}$ [Zuc07]. Until our work, the computational complexity of the second relaxation remained largely open. There are a couple of (very different) quasi-polynomial algorithms that guarantee finding a $(1-\varepsilon)$-dense k-subgraph in every graph containing a k-clique: the meta-algorithm by Barman, which we outlined above, and an older algorithm due to Feige and Seltser [FS97], but nothing non-trivial was known about hardness of approximation.

In this thesis we prove that, assuming ETH, even if one makes *both relaxations* the problem remains intractable. In particular, even if the graph contains a clique of size k, it takes quasi-polynomial time to find an $(1 - \varepsilon)$-dense δk-subgraph, for constant $\varepsilon > 0$ and $\delta = o(1)$.

1.2.2 Community Detection

Identifying communities is a central graph-theoretic problem with important applications to sociology and marketing (when applied to social networks), biology and bioinformatics (when applied to protein interaction networks), and more (see e.g. Fortunato's classic survey [For10]). Defining what exactly is a *community* remains an interesting problem on its own (see Arora et al [Aro+12] and Borgs et al [Bor+16] for excellent treatment from a theoretical perspective). Ultimately, there is no single "right" definition, and the precise meaning of community should be different for social networks and protein interaction networks.

In this thesis we focus on the algorithmic questions arising from one of the simplest and most canonical definitions, which has been considered by several theoretical computer scientists [Mis+08; Aro+12; Bal+13; Bra+17].

Definition 1.2.1 ((α, β)-Community)**.** Given an undirected graph $G = (V, E)$ an (α, β)-community is a subset $S \subseteq V$ that satisfies:

Strong ties inside the community For every $v \in S$, $|\{v\} \times S| \cap E \geq \alpha \cdot |S|$; and

Weak ties to nodes outside the community For every $u \notin S$, $|\{u\} \times S| \cap E \leq \beta \cdot |S|$.

This problem has considered by several researchers before: Mishra, Schreiber, Stanton, and Tarjan [Mis+08] gave a polynomial-time algorithm for finding (α, β)-communities that contain a vertex with very few neighbors outside the community. Balcan et al [Bal+13] give a polynomial-time algorithm for enumerating (α, β)-communities in the special case where the degree of every node is $\Omega(n)$. Arora, Ge, Sachdeva, and Schoenebeck [Aro+12] consider several semi-random models where the edges inside the community are generated at random, according to the expected degree model. For the general case, the latter paper by Arora et al gave a simple quasi-polynomial ($n^{O(\log n)}$) time for detecting (α, β)-communities whenever $\alpha - \beta$ is at least some positive constant. (Their algorithm is essentially identical to the meta-algorithm for bilinear optimization that we outlined above.)

We show that, for *every* constants $\alpha > \beta \in (0, 1]$, community detection requires quasi-polynomial time (assuming ETH). For example, when $\alpha = 1$ and $\beta = 0.01$, this means that we can hide a clique C, such that every single vertex not in C is connected to at most 1% of C. Our main result is actually a much stronger inapproximability: even in the presence of a $(1, o(1))$-community, finding any $(\beta + o(1), \beta)$-community is hard.

Unlike all quasi-polynomial approximation schemes mentioned above, Arora et al's algorithm has the unique property that it can also *exactly count* all the (α, β)-communities. Our second result is that counting even the number of $(1, o(1))$-communities requires quasi-polynomial time. A nice feature of this result is that we can base it on the much weaker #ETH assumption, which asserts that counting the satisfying assignment for a 3SAT instance requires time $2^{\Omega(n)}$. (Note, for example, that #ETH is likely to be true even if $\mathsf{P} = \mathsf{NP}$.)

1.2.3 VC and Littlestone's dimensions

A common and essential assumption in learning theory is that the concepts we want to learn come from a nice, simple concept class, or (in the agnostic case) they can at least be approximated by a concept from a simple class. When the concept class is sufficiently simple, there is hope for good (i.e. sample-efficient and low-error) learning algorithms.

There are many different ways to measure the *simplicity* of a concept class. The most influential measure of simplicity is the VC Dimension [VC71], which captures learning in the PAC model. We also consider Littlestone's Dimension [Lit87], which corresponds to minimizing mistakes in online learning (see Section 2.5 for definitions). When either dimension is small, there are algorithms that exploit the simplicity of the class, to obtain good learning guarantees.

In this thesis we consider the algorithmic challenge of computing either dimension. In particular, we study the most optimistic setting, where the entire universe and concept class are given as explicit input (a binary matrix whose (x, c)-th entry is 1 iff element x belongs to concept c) In this model, both dimensions can be computed in quasi-polynomial time. Interestingly, the algorithm does not go through the bilinear optimization problem; instead, it exploits the fact that for concept class $\mathcal{C}$, both dimensions are bounded by $\log|\mathcal{C}|$. Two decades ago, it was shown that quasi-polynomial time is indeed necessary for both dimensions [PY96; FL98]. The computational intractability of computing the (VC, Littlestone's) dimension of a concept class suggests that even in cases where a simple structure exists, it may be inaccessible to computationally bounded algorithms.

In this thesis we extend the results of [PY86; FL98] to show that the VC and Littlestone's Dimensions cannot even be *approximately* computed in polynomial time.

1.2.4 Signaling

Many classical questions in economics involve extracting information from strategic agents. Lately, there has been growing interest within algorithmic game theory in *signaling*: the study of how to *reveal information* to strategic agents (see e.g. [MS12; DIR13; Eme+14; Dug14; Che+15b] and references therein). Signaling has been studied in many interesting economic and game theoretic settings. Among them, Zero-Sum Signaling proposed by Dughmi [Dug14] stands out as a canonical problem that cleanly captures the computational nature of signaling. In particular, focusing on zero-sum games clears away issues of equilibrium selection and computational tractability of finding an equilibrium.

Definition (Zero-Sum Signaling [Dug14])**.** Alice and Bob play a Bayesian zero-sum game where the payoff matrix M is drawn from a publicly known prior. The signaler Sam privately observes the state of nature (i.e. the payoff matrix), and then publicly broadcasts a signal $\varphi(M)$ to both Alice and Bob. Alice and Bob Bayesian-update their priors according to $\varphi(M)$'s and play the Nash equilibrium of the expected game; but they receive payoffs according to the true M. Sam's goal is

to design an efficient signaling scheme φ (a function from payoff matrices to strings) that maximizes Alice's expected payoff.

Dughmi's [Dug14] main result proves that assuming the hardness of the Planted Clique problem, there is no additive FPTAS for Zero-Sum Signaling. The main open question left by [Dug14] is whether there exists an additive PTAS. Here we answer this question in the negative: we prove that assuming the Exponential Time Hypothesis (ETH) [IPZ01], obtaining an additive-ϵ-approximation (for some constant $\epsilon > 0$) requires quasi-polynomial time $(n^{\tilde{\Omega}(\lg n)})$. This result is tight thanks to a recent quasi-polynomial $(n^{\frac{\lg n}{\mathrm{poly}(\epsilon)}})$ time algorithm by Cheng et al. [Che+15b]. Another important advantage of our result is that it replaces the hardness of Planted Clique with a more believable worst-case hardness assumption (see e.g. the discussion in [BKW15]).

1.3 Approximate Nash equilibrium

The main result in this thesis rules out the PTAS (polynomial time approximation schemes) for two-player Nash equilibrium. Consider a game between two players, each choosing between a large number (n) of actions. The input to the algorithm are two $n \times n$ matrices with entries in $[0, 1]$. The goal is to find, for every constant ε, an ε-approximate Nash equilibrium; i.e. a mixed strategy for each player, such that either player can gain at most ε by deviating to a different strategy.

This has been the central open question in equilibrium computation for the past decade. There were good reasons to be hopeful: there was a quasi-polynomial time [LMM03], a series of improved approximation ratios [KPS09; DMP09; DMP07; BBM10; TS08] and several approximation schemes for special cases [KT07; DP09; Alo+13; Bar15]. Our main result settles this question on the negative:

Theorem 1.3.1 (Main Theorem)**.** *There exists a constant $\epsilon > 0$ such that, assuming ETH for* PPAD*, finding an ϵ-Approximate Nash Equilibrium in a two-player $n \times n$ game requires time $T(n) = n^{\log^{1-o(1)} n}$.*

We supplement Theorem 1.3.1 with a series of other hardness of approximation results for Nash equilibrium in related settings, further establishing the point that even approximate Nash equilibria are intractable. First, we consider different sources of complexity. Our main result shows intractability of approximate Nash equilibrium when the complexity of the game arises from each player choosing among many actions. In Theorems 5.0.2 and 5.0.4, we prove in games where each player only has

two actions, the complexity can arise form a large number of players; finding an approximate Nash equilibrium in n-player, binary action games is PPAD-hard (settling another open question from Daskalakis's thesis [Das08]). Alternatively, even if there are only two player and the number of actions is constant, a different source of complexity can be the players uncertainty; finding an approximate Bayesian Nash equilibrium in such incomplete information games is also PPAD-hard (Corollary 8.0.3).

We also prove intractability in different models: query complexity, communication complexity, and uncoupled dynamics (settling a long list of open questions from [HM10; Nis09a; Fea+13; Bab12; HN13; Bab16; CCT17; RW16]). The main advantage of these results is that they are *unconditional*, i.e. they do not rely on complexity assumptions such as ETH for PPAD, or $\mathsf{P} \neq \mathsf{NP}$. In particular, in the setting where each player knows her own utility function, even computationally-unbounded players have to communicate almost their entire private information in order to reach even an approximate equilibrium.

Chapter 2

Preliminaries

2.0.1 Notation

We use $\mathbf{0}_n$ (respectively $\mathbf{1}_n$) to denote the length-n vectors whose value is 0 (1) in every coordinate. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we let

$$\|\mathbf{x} - \mathbf{y}\|_2 \triangleq \sqrt{\mathsf{E}_{i \in [n]} (x_i - y_i)^2}$$

denote the *normalized* 2-norm. Unless specified otherwise, when we say that $\mathbf{x}$ and $\mathbf{y}$ are Δ-close (or Δ-far), we mean Δ-close in normalized 2-norm. Similarly, for a binary string $\pi \in \{0,1\}^n$, we denote

$$|\pi| \triangleq \mathsf{E}_{i \in [n]} [\pi_i] .$$

We use $\mathrm{den}(S) \in [0,1]$ to denote the density of subgraph S,

$$\mathrm{den}(S) := \frac{\left|\left(S \times S\right) \cap E\right|}{|S \times S|}.$$

2.1 Nash equilibrium and relaxations

A mixed strategy of player i is a distribution x_i over i's set of actions, A_i. We say that a vector of mixed strategies $\mathbf{x} \in \times_j \Delta A_j$ is a *Nash equilibrium* if every strategy a_i in the support of every x_i is a best response to the actions of the mixed strategies of the rest of the players, x_{-i}. Formally, for every $a_i \in \mathrm{supp}(x_i)$

$$\mathsf{E}_{a_{-i} \sim x_{-i}} [u_i (a_i, a_{-i})] = \max_{a' \in A_i} \mathsf{E}_{a_{-i} \sim x_{-i}} [u_i (a', a_{-i})] .$$

Equivalently, $\mathbf{x}$ is a Nash equilibrium if each mixed strategy x_i is a best response to x_{-i}:

$$\mathsf{E}_{\mathbf{a}\sim\mathbf{x}}\left[u_i\left(\mathbf{a}\right)\right] = \max_{x_i'\in\Delta A_i} \mathsf{E}_{\mathbf{a}\sim\left(x_i';x_{-i}\right)}\left[u_i\left(\mathbf{a}\right)\right].$$

Each of those equivalent definitions can be generalized to include approximation in a different way.

Definition 2.1.1 (ϵ-Approximate Nash Equilibrium)**.** We say that $\mathbf{x}$ is an *ϵ-Approximate Nash Equilibrium* (*ϵ-ANE*) if each x_i is an ϵ-best response to x_{-i}:

$$\mathsf{E}_{\mathbf{a}\sim\mathbf{x}}\left[u_i\left(\mathbf{a}\right)\right] \geq \max_{x_i'\in\Delta A_i} \mathsf{E}_{\mathbf{a}\sim\left(x_i';x_{-i}\right)}\left[u_i\left(\mathbf{a}\right)\right] - \epsilon.$$

On the other hand, we generalize the first definition of Nash equilibrium in the following stricter definition:

Definition 2.1.2 (ϵ-Well-Supported Nash Equilibrium)**.** $\mathbf{x}$ is a *ϵ-Well-Supported Nash Equilibrium* (*ϵ-WSNE*) if every a_i in the support of x_i is an ϵ-best response to x_{-i}: for every $a_i \in \operatorname{supp}\left(x_i\right)$

$$\mathsf{E}_{a_{-i}\sim x_{-i}}\left[u_i\left(a_i, a_{-i}\right)\right] \geq \max_{a'\in A_i} \mathsf{E}_{a_{-i}\sim x_{-i}}\left[u_i\left(a', a_{-i}\right)\right] - \epsilon.$$

WeakNash

We can further relax the (already more lenient) notion of ϵ-ANE by requiring that the ϵ-best response condition only hold for most of the players (rather than all of them).

Definition 2.1.3 ((ϵ,δ)-WeakNash [BPR16])**.** We say that $\mathbf{x}$ is an *(ϵ,δ)-WeakNash* if for a $(1-\delta)$-fraction of i's, x_i is an ϵ-best mixed response to x_{-i}:

$$\Pr_i\left[\mathsf{E}_{\mathbf{a}\sim\mathbf{x}}\left[u_i\left(\mathbf{a}\right)\right] \geq \max_{x_i'\in\Delta A_i} \mathsf{E}_{\mathbf{a}\sim\left(x_i';x_{-i}\right)}\left[u_i\left(\mathbf{a}\right)\right] - \epsilon\right] \geq 1-\delta.$$

Definition 2.1.4 ((ϵ,δ)-Well-Supported WeakNash)**.** $\mathbf{x}$ is a *(ϵ,δ)-Well-Supported WeakNash* if for a $(1-\delta)$-fraction of i's, every a_i in the support of x_i is an ϵ-best response to x_{-i}:

$$\Pr_i\left[\forall a_i \in \operatorname{supp}\left(x_i\right)\ \mathsf{E}_{a_{-i}\sim x_{-i}}\left[u_i\left(a_i, a_{-i}\right)\right] \geq \max_{a'\in A_i} \mathsf{E}_{a_{-i}\sim x_{-i}}\left[u_i\left(a', a_{-i}\right)\right] - \epsilon\right] \geq 1-\delta.$$

2.2 PPAD and End-of-a-Line

The End-of-a-Line of problem considers an implicitly-represented, exponential size, directed graph whose vertices have in- and out-degree at most 1 (this is without loss of generality). The special vertex $\mathbf{0}_n$ has in-degree 0, and the goal is to find another odd-degree vertex. The graph is a union of lines and cycles, so in particular the line starting at $\mathbf{0}_n$ ends with another odd-degree vertex.

The graph is implicitly defined with functions S, P that, for each vertex, give its *S*uccessor (out-going neighbor) and *P*redecessor (incoming neighbor). In the computational variant of End-of-a-Line, S, P are given as explicit circuits, whereas in the query complexity variant they are given as oracles. There is also a communication complexity variant, whose definition is more involved and is deferred to Section 3.1.

In the formal definition of the computational variant we have to also consider the case that S, P are inconsistent, i.e. for some $u \neq v$ we have $S(u) = v$, but $P(v) \neq u$; we also allow the algorithm to succeed by finding such an inconsistency. (In the oracle model we can explicitly require that there are no inconsistencies.)

Definition 2.2.1 (End-of-a-Line)**.** The input to the problem is functions $S, P : \{0,1\}^n \to \{0,1\}^n$, such that $S(\mathbf{0}_n) = \mathbf{0}_n \neq P(\mathbf{0}_n)$. The output is another odd degree vertex $\mathbf{0}_n \neq v \in \{0,1\}^n$ such that $P(S(v)) \neq S(P(v))$.

The computational complexity class PPAD is defined as the class of all total search problems reducible to End-of-a-Line.

Membership End-of-a-Line

The following variant of End-of-a-Line is equivalent and more convenient for some of our reductions. In particular, the problem is restricted to a subset of the vertices. The restricted vertex-set is defined implicitly via a membership function $T : \{0,1\}^n \to \{0,1\}$. Abusing notation, let T also denote the restricted set of vertices whose T-value is 1. We think of S and P as only applying to vertices in T, and connecting them to other vertices in T. Formally, we also allow the algorithm to return any violations.

Definition 2.2.2 (Membership End-of-a-Line)**.** The input to the problem is functions $S, P : \{0,1\}^n \to \{0,1\}^n$ and $T : \{0,1\}^n \to \{0,1\}$, such that $S(\mathbf{0}_n) = \mathbf{0}_n \neq P(\mathbf{0}_n)$ and $T(\mathbf{0}_n), T(S(\mathbf{0}_n)) = 1$. The output is another odd degree vertex $\mathbf{0}_n \neq v \in \{0,1\}^n$ such that $T(v) = 1$ and v satisfies at least one of the following:

End-of-a-line $P(S(v)) \neq S(P(v))$; or

Boundary condition $T(S(v)) = 0$ or $T(P(v)) = 0$.

Lemma 2.2.3. End-of-a-Line *is equivalent to* Membership End-of-a-Line.

Proof. Given an instance of End-of-a-Line, we can construct an equivalent instance of Membership End-of-a-Line by setting $T \equiv 1$. In the other direction, we can add self-loops to every vertex v such that $T(v) = 0$ (i.e. $P(v) = S(v) = v$); this guarantees that v is never a solution to the new End-of-a-Line instance. □

We will be interested with restricted (but equally hard) variants of Membership End-of-a-Line. For example, in Section 16.2 we define Local End-of-a-Line where, among other restrictions, T, S, P are AC^0 circuits. In particular, in Chapter 3, we will consider a variant where each vertex is a-priori restricted to have at most two potential incoming/outgoing neighbors, and the functions S, P merely specify which neighbor is chosen. We then abuse notation and let S, P output just a single bit.

2.3 Exponential Time Hypotheses

Our quasi-polynomial hardness results are conditional on the following hypotheses. We begin with the "plain" ETH:

Hypothesis 1 (Exponential Time Hypothesis (ETH) [IPZ01]). 3SAT takes time $2^{\Omega(n)}$.

Since a Nash equilibrium always exists, we are unlikely to have a reduction (even of subexponential size) from 3SAT to Nash equilibrium. Instead, we need to assume the following analogue of ETH for the total class PPAD:

Hypothesis 2 (ETH for PPAD [BPR16]). Solving EndOfALine requires time $2^{\tilde{\Omega}(n)}$.[1]

In Section 13.1 we will prove a quasi-polynomial lower bound on the running time for counting the number of communities in a social network. This result is also conditional, but requires the following much weaker #ETH assumption:

Hypothesis 3 (#ETH [Del+14]). Given a 3SAT formula, counting the number of satisfying assignments takes time $2^{\Omega(n)}$.

2.4 PCP theorems

2.4.1 2CSP and the PCP Theorem

In the **2CSP** problem, we are given a graph $G = (V, E)$ on $|V| = n$ vertices, where each of the edges $(u, v) \in E$ is associated with some constraint function $\psi_{u,v} : \Sigma \times \Sigma \to \{0, 1\}$

[1] As usual, n is the size of the description of the instance, i.e. the size of the circuits S and P.

which specifies a set of legal "colorings" of u and v, from some finite alphabet Σ (2 in the term "**2CSP** " stands for the "arity" of each constraint, which always involves two variables). Let us denote by ψ the entire **2CSP** instance, and define by $\mathrm{OPT}(\psi)$ the maximum fraction of satisfied constraints in the associated graph G, over all possible assignments (colorings) of V.

The starting point of some of our reductions is the following version of the PCP theorem, which asserts that it is NP-hard to distinguish a **2CSP** instance whose value is 1, and one whose value is $1-\eta$, where η is some small constant:

Theorem 2.4.1 (PCP Theorem [Din07])**.** *Given a* **3SAT** *instance φ of size n, there is a polynomial time reduction that produces a* **2CSP** *instance ψ, with size $|\psi| = n \cdot \mathrm{polylog} n$ variables and constraints, and constant alphabet size such that*

- *(Completeness) If $\mathrm{OPT}(\varphi) = 1$ then $\mathrm{OPT}(\psi) = 1$.*
- *(Soundness) If $\mathrm{OPT}(\varphi) < 1$ then $\mathrm{OPT}(\psi) < 1-\eta$, for some constant $\eta = \Omega(1)$*
- *(Graph) The constraint graph is d-regular, for some constant d, and bipartite.*

See e.g. the full version of [Bra+17] or [AIM14] for derivations of this formulation of the PCP theorem.

Notice that since the size of the reduction is near linear, ETH implies that solving the above problem requires near exponential time.

Corollary 2.4.2. *Let ψ be as in Theorem 2.4.1. Then assuming ETH, distinguishing between $\mathrm{OPT}(\psi) = 1$ and $\mathrm{OPT}(\psi) < 1-\eta$ requires time $2^{\tilde{\Omega}(|\psi|)}$.*

Label Cover

Definition 2.4.3 (Label Cover)**.** Label Cover is a maximization problem, and a special case of **2CSP**. The input is a bipartite graph $G = (A, B, E)$, alphabets Σ_A, Σ_B, and a projection $\pi_e : \Sigma_A \to \Sigma_B$ for every $e \in E$.

The output is a labeling $\varphi_A : A \to \Sigma_A$, $\varphi_B : B \to \Sigma_B$. Given a labeling, we say that a constraint (or edge) $(a, b) \in E$ is *satisfied* if $\pi_{(a,b)}(\varphi_A(a)) = \varphi_B(b)$. The *value of a labeling* is the fraction of $e \in E$ that are satisfied by the labeling. The value of the instance is the maximum fraction of constraints satisfied by any assignment.

We often encounter an assignment that only labels a subset of $A \cup B$ but leaves the rest unlabeled. We refer to such assignment as a *partial assignment* to an instance; more specifically, for any $V \subseteq A \cup B$, a V-partial assignment (or partial assignment on V) is a function $\phi : V \to \Sigma$. For notational convenience, we sometimes write Σ^V to denote the set of all functions from V to Σ.

Theorem 2.4.4 (Moshkovitz-Raz PCP [MR10, Theorem 11])**.** *For every n and every $\epsilon > 0$ (in particular, ϵ may be a function of n), solving* 3SAT *on inputs of size n can be reduced to distinguishing between the case that a (d_A, d_B)-bi-regular instance of* Label Cover*, with parameters $|A| + |B| = n^{1+o(1)} \cdot \mathsf{poly}(1/\epsilon)$, $|\Sigma_A| = 2^{\mathsf{poly}(1/\epsilon)}$, and $d_A, d_B, |\Sigma_B| = \mathsf{poly}(1/\epsilon)$, is completely satisfiable, versus the case that it has value at most ϵ.*

Counting the number of satisfying assignments is even harder. The following hardness is well-known, and we sketch its proof only for completeness:

Fact 2.4.5. There is a linear-time reduction from #3SAT to counting the number of satisfying assignments of a Label Cover instance.

Proof. Construct a vertex in A for each variable and a vertex in B for each clause. Set $\Sigma_A \triangleq \{0,1\}$ and let $\Sigma_B \triangleq \{0,1\}^3 \setminus (000)$ (i.e. Σ_B is the set of satisfying assignments for a 3SAT clause, after applying negations). Now if variable x appears in clause C, add a constraint that the assignments to x and C are consistent (taking into account the sign of x in C). Notice that any assignment to A: (i) corresponds to a unique assignment to the 3SAT formula; and (ii) if the 3SAT formula is satisfied, this assignment uniquely defines a satisfying assignment to B. Therefore there is a one-to-one correspondence between satisfying assignments to the 3SAT formula and to the instance of Label Cover. □

2.5 Learning Theory

For a universe (or ground set) $\mathcal{U}$, a concept C is simply a subset of $\mathcal{U}$ and a concept class $\mathcal{C}$ is a collection of concepts. For convenience, we sometimes relax the definition and allow the concepts to not be subsets of $\mathcal{U}$; all definitions here extend naturally to this case.

The VC and Littlestone's Dimensions can be defined as follows.

Definition 2.5.1 (VC Dimension [VC71])**.** A subset $S \subseteq \mathcal{U}$ is said to be *shattered* by a concept class $\mathcal{C}$ if, for every $T \subseteq S$, there exists a concept $C \in \mathcal{C}$ such that $T = S \cap C$.

The VC Dimension VC-dim$(\mathcal{C}, \mathcal{U})$ of a concept class $\mathcal{C}$ with respect to the universe $\mathcal{U}$ is the largest d such that there exists a subset $S \subseteq \mathcal{U}$ of size d that is shattered by $\mathcal{C}$.

Definition 2.5.2 (Mistake Tree and Littlestone's Dimension [Lit87])**.** A depth-d instance-labeled tree of $\mathcal{U}$ is a full binary tree of depth d such that every internal

node of the tree is assigned an element of $\mathcal{U}$. For convenience, we will identify each node in the tree canonically by a binary string s of length at most d.

A depth-d mistake tree (aka shattered tree [BPS09]) for a universe $\mathcal{U}$ and a concept class $\mathcal{C}$ is a depth-d instance-labeled tree of $\mathcal{U}$ such that, if we let $v_s \in \mathcal{U}$ denote the element assigned to the vertex s for every $s \in \{0,1\}^{<d}$, then, for every leaf $\ell \in \{0,1\}^d$, there exists a concept $C \in \mathcal{C}$ that agrees with the path from root to it, i.e., that, for every $i < d$, $v_{\ell_{\le i}} \in C$ iff $\ell_{i+1} = 1$ where $\ell_{\le i}$ denote the prefix of ℓ of length i.

The Littlestone's Dimension L-dim$(\mathcal{C},\mathcal{U})$ of a concept class $\mathcal{C}$ with respect to the universe $\mathcal{U}$ is defined as the maximum d such that there exists a depth-d mistake tree for $\mathcal{U},\mathcal{C}$.

An equivalent formulation of Littlestone's Dimension is through mistakes made in online learning, as stated below. This interpretation will be useful in the proof of Theorem 14.2.1.

Definition 2.5.3 (Mistake Bound)**.** An online algorithm $\mathcal{A}$ is an algorithm that, at time step i, is given an element $x_i \in \mathcal{U}$ and the algorithm outputs a prediction $p_i \in \{0,1\}$ whether x is in the class. After the prediction, the algorithm is told the correct answer $h_i \in \{0,1\}$. For a sequence $(x_1,h_1),\dots,(x_n,h_n)$, *prediction mistake* of $\mathcal{A}$ is defined as the number of incorect predictions, i.e., $\sum_{i\in n} \mathbb{1}[p_i \neq h_i]$. The *mistake bound* of $\mathcal{A}$ for a concept class $\mathcal{C}$ is defined as the maximum prediction mistake of $\mathcal{A}$ over all the sequences $(x_1,h_1),\dots,(x_n,h_n)$ which corresponds to a concept $C \in \mathcal{C}$ (i.e. $h_i = \mathbb{1}[x_i \in C]$ for all $i \in [n]$).

Theorem 2.5.4 ([Lit87])**.** *For any universe $\mathcal{U}$ and any concept class $\mathcal{C}$,* L-dim$(\mathcal{C},\mathcal{U})$ *is equal to the minimum mistake bound of $\mathcal{C},\mathcal{U}$ over all online algorithms.*

The following facts are well-know and follow easily from the above definitions.

Fact 2.5.5. For any universe $\mathcal{U}$ and concept class $\mathcal{C}$, we have

$$\text{VC-dim}(\mathcal{C},\mathcal{U}) \le \text{L-dim}(\mathcal{C},\mathcal{U}) \le \log|\mathcal{C}|.$$

Fact 2.5.6. For any two universes $\mathcal{U}_1,\mathcal{U}_2$ and any concept class $\mathcal{C}$,

$$\text{L-dim}(\mathcal{C},\mathcal{U}_1 \cup \mathcal{U}_2) \le \text{L-dim}(\mathcal{C},\mathcal{U}_1) + \text{L-dim}(\mathcal{C},\mathcal{U}_2).$$

2.6 Information Theory

In this section, we introduce information-theoretic quantities used in this paper. For a more thorough introduction, the reader should refer to [CT12]. Unless stated otherwise, all log's in this paper are base-2.

Definition 2.6.1. Let μ be a probability distribution on sample space Ω. The *Shannon entropy* (or just *entropy*) of μ, denoted by $H(\mu)$, is defined as $H(\mu) := \sum_{\omega\in\Omega} \mu(\omega) \log \frac{1}{\mu(\omega)}$.

Definition 2.6.2 (Binary Entropy Function)**.** For $p \in [0,1]$, the binary entropy function is defined as follows (with a slight abuse of notation) $H(p) := -p \log p - (1-p)\log(1-p)$.

Fact 2.6.3 (Concavity of Binary Entropy). Let μ be a distribution on $[0,1]$, and let $p \sim \mu$. Then $H(\mathsf{E}_\mu[p]) \geq \mathsf{E}_\mu[H(p)]$.

For a random variable A we shall write $H(A)$ to denote the entropy of the induced distribution on the support of A. We use the same abuse of notation for other information-theoretic quantities appearing later in this section.

Definition 2.6.4. The *Conditional entropy* of a random variable A conditioned on B is defined as

$$H(A|B) = \mathbb{E}_b(H(A|B=b)).$$

Fact 2.6.5 (Chain Rule).

$$H(AB) = H(A) + H(B|A).$$

Fact 2.6.6 (Conditioning Decreases Entropy). $H(A|B) \geq H(A|BC)$.

Another measure we will use (briefly) in our proof is that of *Mutual Information*, which informally captures the correlation between two random variables.

Definition 2.6.7 (Conditional Mutual Information)**.** The *mutual information* between two random variable A and B, denoted by $I(A;B)$ is defined as

$$I(A;B) := H(A) - H(A|B) = H(B) - H(B|A).$$

The *conditional mutual information* between A and B given C, denoted by $I(A;B|C)$, is defined as

$$I(A;B|C) := H(A|C) - H(A|BC) = H(B|C) - H(B|AC).$$

The following is a well-known fact on mutual information.

Fact 2.6.8 (Data processing inequality). Suppose we have the following Markov Chain:

$$X \to Y \to Z$$

where $X \perp Z|Y$. Then $I(X;Y) \geq I(X;Z)$ or equivalently, $H(X|Y) \leq H(X|Z)$.

Mutual Information is related to the following distance measure.

Definition 2.6.9 (Kullback-Leiber Divergence)**.** Given two probability distributions μ_1 and μ_2 on the same sample space Ω such that $(\forall \omega \in \Omega)(\mu_2(\omega) = 0 \Rightarrow \mu_1(\omega) = 0)$, the *Kullback-Leibler Divergence* between is defined as (also known as relative entropy)

$$\mathsf{D}_{\mathsf{KL}}\left(\mu_1 \middle\| \mu_2\right) = \sum_{\omega \in \Omega} \mu_1(\omega) \log \frac{\mu_1(\omega)}{\mu_2(\omega)}.$$

The connection between the mutual information and the Kullback-Leibler divergence is provided by the following fact.

Fact 2.6.10. For random variables A, B, and C we have

$$I(A; B|C) = \mathbb{E}_{b,c}\left[\mathsf{D}_{\mathsf{KL}}\left(A_{bc} \middle\| A_c\right)\right].$$

2.7 Useful lemmata

2.7.1 Concentration

Lemma 2.7.1 (Chernoff Bound)**.** *Let $X_1, \ldots, X_n$ be i.i.d. random variables taking value from $\{0, 1\}$ and let p be the probability that $X_i = 1$, then, for any $\delta > 0$, we have*

$$\Pr\left[\sum_{i=1}^{n} X_i \geq (1+\delta)np\right] \leq \begin{cases} 2^{-\delta^2 np/3} & \text{if } \delta < 1, \\ 2^{-\delta np/3} & \text{otherwise.} \end{cases}$$

2.7.2 Pseudorandomness

Theorem 2.7.2 (k-wise independence Chernoff bound [SSS95, Theorem 5.I])**.** *Let $x_1 \ldots x_n \in [0, 1]$ be k-wise independent random variables, and let $\mu \triangleq \mathsf{E}\left[\sum_{i=1}^{n} x_i\right]$ and $\delta \leq 1$. Then*

$$\Pr\left[\left|\sum_{i=1}^{n} x_i - \mu\right| > \delta\mu\right] \leq \mathrm{e}^{-\Omega\left(\min\{k, \delta^2\mu\}\right)}.$$

2.7.3 λ-biased sets

Definition 2.7.3 (λ-biased sets)**.** Let $\mathcal{G}$ be a finite field, and $t > 0$ an integer. A multiset $S \subseteq \mathcal{G}^t$ is λ-biased if for every nontrivial character χ of $\mathcal{G}^t$,

$$\left|\mathsf{E}_{\mathbf{y} \sim S}\left[\chi\left(\mathbf{y}\right)\right]\right| \leq \lambda.$$

Lemma 2.7.4 ([AMN98, Theorem 3.2]). *A randomly chosen multiset $S \subseteq \mathcal{G}^t$ of cardinality $\Theta\left(t \log |\mathcal{G}| / \lambda^2\right)$ is λ-biased with high probability.*

For many applications, an explicit construction is necessary. In our case, however, we can enumerate over all sets S of sufficient cardinality in quasi-polynomial time[2]. The following Sampling Lemma due to Ben-Sasson et al. [Ben+03] allows us to estimate the average of any function over $\mathcal{G}^t$ using only one line and $(1 + o(1)) \log_2 |\mathcal{G}^t|$ randomness:

Lemma 2.7.5 (Sampling Lemma: [Ben+03, Lemma 4.3]). *Let $B : \mathcal{G}^t \to [0, 1]$. Then, for any $\epsilon > 0$,*

$$\Pr_{\substack{\mathbf{x} \in \mathcal{G}^t, \\ \mathbf{y} \in S}} \left[\left| \mathsf{E}_{\beta \in \mathcal{G}} \left[B\left(\mathbf{x} + \beta \mathbf{y}\right) \right] - \mathsf{E}_{\mathbf{z} \in \mathcal{G}^t} \left[B\left(\mathbf{z}\right) \right] \right| > \epsilon \right] \leq \left(\frac{1}{|\mathcal{G}|} + \lambda \right) \frac{\mathsf{E}_{\mathbf{z} \in \mathcal{G}^t} \left[B\left(\mathbf{z}\right) \right]}{\epsilon^2}.$$

2.7.4 Partitions

Given a 2CSP formula, we provide a few techniques to deterministically partition n variables to approximately $\sqrt{n}$ subsets of approximately $\sqrt{n}$ variables each, so that number of constraints between every pair of partitions is approximately as expected.

Greedy partition

Lemma 2.7.6. *Let $G = (V, E)$ be a d-regular graph and $n \triangleq |V|$. We can partition V into n/k disjoint subsets $\left\{S_1, \ldots, S_{n/k}\right\}$ of size at most $2k$ such that:*

$$\forall i, j \quad |(S_i \times S_j) \cap E| \leq 8d^2k^2/n \tag{2.1}$$

Proof. We assign vertices to subsets iteratively, and show by induction that we can always maintain (2.1) and the bound on the subset size. Since the average set size is less than k, we have by Markov's inequality that at each step less than half of the subsets are full. The next vertex we want to assign, v, has neighbors in at most d subsets. By our induction hypothesis, each S_i is of size at most $2k$, so on average over $j \in [n/k]$, it has less than $4dk^2/n$ neighbors in each S_j. Applying Markov's

[2]Note that we need an ϵ-biased set for a large field $\mathcal{G} = \mathbb{F}_{2^\ell}$. Such constructions are not as common in the literature which mostly focuses on the field $\mathbb{F}_2$. To the best of our knowledge, existing explicit constructions for larger fields require much larger cardinality. Nevertheless, for our modest pseudorandomness desiderata, we could actually use the explicit construction from [Alo+92]. For ease of presentation, we prefer to brute-force derandomize the construction from [AMN98].

inequality again, S_i has at least $8d^2k^2/n$ neighbors in less than a $(1/2d)$-fraction of subsets S_j. In total, we ruled out less than half of the subsets for being full, and less than half of the subsets for having too many neighbors with subsets that contain neighbors of v. Therefore there always exists some subset S_i to which we can add v while maintaining the induction hypothesis. □

Derandomized partition

We use Chernoff bound with $\Theta(\log n)$-wise independent variables to deterministically partition variables into subsets of cardinality $\approx \sqrt{n}$. Our (somewhat naive) deterministic algorithm for finding a good partition takes quasi-polynomial time $(n^{O(\log n)})$, which is negligible with respect to the sub-exponential size $(N = 2^{\tilde{O}(\sqrt{n})})$ of our reduction[3].

Lemma 2.7.7. *Let $G = (A, B, E)$ be a bipartite (d_A, d_B)-bi-regular graph, and let $n_A \triangleq |A|$, $n_B \triangleq |B|$; set also $n \triangleq n_B + n_A$ and $\rho \triangleq \sqrt{n}\log n$. Let $T_1, \ldots, T_{n_B/\rho}$ be an arbitrary partition of B into disjoint subsets of size ρ. There is a quasi-polynomial deterministic algorithm (alternatively, linear-time randomized algorithm) that finds a partition of A into $S_1, \ldots, S_{n_A/\rho}$, such that:*

$$\forall i \quad \left| |S_i| - \rho \right| < \rho/2, \tag{2.2}$$

and

$$\forall i, j \quad \left| |(S_i \times T_j) \cap E| - \frac{d_A \rho^2}{n_B} \right| < \frac{d_A \rho^2}{2n_B}. \tag{2.3}$$

Proof. Suppose that we place each $a \in A$ into a uniformly random S_i. By Chernoff bound and union bound, (2.2) and (2.3) hold with high probability. Now, by Chernoff Bound for k-wise independent variables (Theorem 2.7.2), it suffices to partition A using a $\Theta(\log n)$-wise independent distribution. Such distribution can be generated with a sample space of $n^{O(\log n)}$ (e.g. [ABI86]). Therefore, we can enumerate over all possibilities in quasi-polynomial time. By the probabilistic argument, we will find at least one partition that satisfies (2.2) and (2.3). □

2.7.5 How to catch a far-from-uniform distribution

The following lemma due to [DP09] implies that

[3]Do not confuse this with the quasi-polynomial lower bound $(N^{\tilde{O}(\log N)})$ we obtain for the running time of the community detection problem.

Lemma 2.7.8 (Lemma 3 in the full version of [DP09])**.** *Let $\{a_i\}_{i=1}^n$ be real numbers satisfying the following properties for some $\theta > 0$: (1) $a_1 \geq a_2 \geq \cdots \geq a_n$; (2) $\sum a_i = 0$; (3) $\sum_{i=1}^{n/2} a_i \leq \theta$. Then $\sum_{i=1}^n |a_i| \leq 4\theta$.*

2.7.6 Simulation theorem

Let $D : \{0,1\}^N \to \{0,1\}$ be a decision problem. We consider the following query complexity models (called also *decision tree*). Each query is an index $k \in [N]$ and the answer is the k-th bit of the input. The randomized query complexity of D, denoted by $\mathbf{BPP}_\delta^{\mathrm{dt}}(D)$ where δ is the allowed probability of error.

We also consider the following communication complexity models. Here, for every $k \in [N]$ Alice holds a vector $\alpha_k \in \{0,1\}^M$ and Bob holds an index $\beta_k \in [M]$, for some $M = \mathsf{poly}(N)$. Their goal is to compute D for the input $(\alpha_1(\beta_1), \ldots, \alpha_N(\beta_N))$. The standard bounded error two-party probabilistic communication complexity of the simulated problem D, denoted by $\mathsf{BPP}_\delta^{\mathrm{cc}}(\textsc{Sim}\text{-}D)$.

To "lift" from query complexity hardness to communication complexity, we use the following recent *simulation theorem* for BPP, due to Goos et al [GPW17], and independently due to Anshu et al [Ans+17].

Theorem 2.7.9 (BPP Simulation Theorem, [GPW17; Ans+17, Theorem 2])**.** *There exists $M = \mathsf{poly}(N)$ such that for any constants $0 < \delta < 1/2$,*

$$\mathsf{BPP}_\delta^{\mathrm{cc}}(\textsc{Sim}\text{-}D) = \Omega\left(\mathsf{BPP}_\delta^{\mathrm{dt}}(D)(\log N)\right).$$

Part II

Communication Complexity

Chapter 3

Communication Complexity of approximate Nash equilibrium

The main motivation for studying the complexity of approximate Nash equilibrium is the insight about the relevance of Nash equilibrium as a predictive solution concept: if specialized algorithms cannot compute an (approximate) equilibrium, it is unreasonable to expect selfish agents to "naturally" converge to one. (See also discussions in the introduction, as well as [DGP09; Nis09b; HM10].) Although extremely useful and the main focus of this thesis, lower bounds on computational complexity suffer from obvious an caveat: we actually don't know how to really *prove* any computational lower bounds: All our computational lower bounds inherently rely on complexity assumptions (such as $\mathsf{NP} \neq \mathbf{P}$ or $\mathsf{PPAD} \neq \mathbf{P}$); even though these assumptions are widely accepted by computer scientists, they make these theorems less accessible to game theorists and economists. For example, it is not clear how they relate to the uncoupled dynamics model studied by game theorists [HMC03; HMC06; Bab12].

In this part of the thesis, we prove *unconditional* lower bounds on the communication complexity of approximate Nash equilibrium. In the *communication complexity* model, each player knows her own utility function, and we restrict the amount of information exchanged between players in order to agree on an approximate Nash equilibrium. The players in this model are unreasonably powerful beings with unlimited computational power. In this sense, obtaining lower bounds even in this setting is more convincing than our computational complexity results. Furthermore, our lower bounds on communication complexity are translate immediately to the *uncoupled dynamics* model mentioned above (see also Subsection 3.0.1). The tradeoff is that the computational complexity lower bounds we can prove are stronger. Take two-player games with N actions, for example. The main result in this thesis

is that no polynomial time algorithm can find an approximate equilibrium. In the communication complexity model, per contra, it is trivial for both players to send their entire $N \times N$ utility matrix; hence the most we can hope for is a polynomial lower bound.

Indeed, our communication complexity results do not directly fit into the title of "between P and NP" theme of this thesis. However we chose to include them because they provide further evidence that real players may not converge to a Nash equilibrium. More importantly, en route to obtaining our communication complexity lower bounds, we develop a construction of a hard-to-find Brouwer fixed point. This construction will be useful in Chapters 6. TBD

Our results

We study both two-player games with a large number (N) of actions, and two-action games with a large number (n) of players. The trivial solution of communicating every player's entire utility function in normal form requires $O(N^2)$ and $O(n2^n)$ communication, respectively[1]. For constant approximation, no non-trivial lower bounds were previously known for the general model, and even for the restricted case of randomized query complexity (see Section 3.0.3), both settings were stated as open questions in [Fea+13] and [HN13; Bab16; CCT17], respectively. For n-player Hart and Mansour gave an $\exp(n)$ lower bound on the communication complexity of exact Nash equilibrium in n-player games is also $\exp(n)$ [HM10][2], and even for an approximate parameter of $1/\mathsf{poly}(n)$ the problem was open [Nis09a].

For two-player games, we prove a polynomial lower bound on the communication complexity:

Theorem 3.0.1. *There exists a constant $\epsilon > 0$ such that the randomized communication complexity ($\mathsf{BPP}^{\mathrm{cc}}$) of ϵ-Nash equilibrium in two-player $N \times N$ games is at least N^{ϵ}.*

For n-player games, we consider a two-party communication problem where the set of players $[n]$ is divided into two disjoint subsets $[n] = n^A \uplus n^B$. Alice holds the utilities of the players in n^A, and Bob holds the utilities of the players in n^B. In particular, this communication problem is easier than the n-parties communication problem where each player holds his own utility function. Furthermore, our negative

[1] Since we care about ϵ-approximate equilibrium for constant ϵ, it suffices to represent the utility with constant precision

[2] The unique Nash equilibrium in the game considered by [HM10] requires probabilities that are *doubly-exponentially* small. Hence their lower bound is exponential in the number of players, but only polynomial in the size of the description of the equilibrium; see [Nis09a].

result holds for the notion of *weak* approximate Nash equilibrium [BPR16], which in particular implies the same negative result for the standard notion of approximate Nash equilibrium (see also Definition 2.1.3).

Theorem 3.0.2. *There exists a constant $\epsilon > 0$ such that the randomized communication complexity ($\mathsf{BPP}^{\mathrm{cc}}$) of (ϵ, ϵ)-weak approximate Nash equilibrium in n-player binary-action games is at least $2^{\epsilon n}$.*

3.0.1 Uncoupled dynamics

An underling assumption of the Nash equilibrium solution is that players predict correctly the (mixed) action of their opponents (or alternatively predict correctly their expected payoff at each action). One justification for this problematic assumption, which appears in the seminal work of John Nash [Nas51], is that *in some scenarios* players may *learn* the behavior of their opponents in cases where the game is played repeatedly. This idea led to an extensive study of learning dynamics and their convergence to Nash equilibrium, see e.g. [You04; HMC13; KL93]. One natural, and general, class of adaptive dynamics is that of *uncoupled dynamics* [HMC03; HMC06] where it is assumed that players do not know the utilities of their opponents (but observe their past behavior). The question on the *existence* of uncoupled dynamics that lead to Nash equilibrium is quite well understood [FY06; HMC06; GL07; Bab12]. Several uncoupled dynamics that converge to approximate Nash equilibrium (or pure Nash equilibrium [You09]) are known. All these dynamics are based on an exhaustive search principle, where at the moment a player realizes she is acting suboptimally she updates her action to a random one (rather than to an optimal one or a better one). One criticism of these dynamics is that convergence to equilibrium may take an unreasonably long time in large games where the exhaustive search is done over a large space. This led to the study of the *rate of convergence* of uncoupled dynamics. As pointed out by [CS04] for every solution concept (in particular equilibria solutions), the (randomized) communication complexity of a solution is identical (up to a logarithmic factor) to the rate of convergence by any (randomized) uncoupled dynamics to the solution. This observation initiated the communication complexity study in games. As was mentioned above, the communication complexity, and thus also the rate of convergence of uncoupled dynamics, was known only for exact or pure Nash equilibrium. The question on the rate of convergence of uncoupled dynamics to approximate Nash equilibrium was an open question. Given the fact that all known positive results introduce dynamics that converge to *approximate* Nash equilibrium, this question is central. Our results for communication complexity resolve this open question, yielding the following negative results for uncoupled dynamics:

Corollary 3.0.3 (Uncoupled Dynamics)**.** *There exists a constant $\epsilon > 0$ such that any uncoupled dynamics requires:*

2-player *at least* $\mathsf{poly}(N)$ *rounds to converge to an ϵ-Nash equilibrium in two-player $N \times N$ games.*

n-player *at least $2^{\Omega(n)}$ rounds to converge to an ϵ-Nash equilibrium (or even (ϵ, ϵ)-weak approximate Nash equilibrium) in n-player binary-action games.*

3.0.2 Techniques

Proving communication complexity lower bounds for Nash equilibrium is notoriously challenging for two reasons. The first reason, as is common in hardness of Nash equilibrium in other models, is *totality*: there always exists at least one (exact) equilibrium, and the proof of existence induces a non-trivial (yet inefficient) algorithm for finding it. In order to construct hard instances we must carefully hide the equilibrium (we can't just remove it), and make sure that the above algorithm is indeed inefficient for our instances.

Another reason for the communication complexity of approximate equilibrium being an open question for a long time is the fact that there exist efficient *non-deterministic* communication protocols ($\mathsf{polylog}(N)$ for two-player, $\mathsf{poly}(n)$ for n-player): verification of equilibrium (exact or approximate) requires only constant communication, and small-representation approximate equilibria always exist (e.g. by [LMM03]). Therefore, the communication complexity lower bounds for approximate equilibria, as we prove in the present paper, show an exponential gap between the non-deterministic and randomized (or even deterministic) communication complexity of a total search problem. We are certainly not the first to show such separations, see e.g. [RW90; KRW95; RM99][3]. But such separations are still not very common in communication complexity, and for a good reason: for decision problems, they are impossible! The deterministic communication complexity is upper-bounded by the product of the non-deterministic and co-non-deterministic communication complexities [AUY83].

The first ingredient in our proof is a construction of a special continuous function $f : [0,1]^n \to [0,1]^n$ whose approximate fixed points are hard to find. The construction is is inspired by that of Hirsch, Papadimitriou, and Vavasis [HPV89], and the main new ingredient is the use of error correcting codes to replace the ℓ_∞ inapproximability

[3]It is interesting to remark that our result is arguably the first example of a *natural* problem which exhibits such a gap: To the best of our knowledge, approximate Nash equilibrium is the first problem that is not defined in order to exhibit a gap, but rather happens to have one.

with ℓ_2 inapproximability. The construction appears in Chapter 4. The second ingredient in our proof is the *simulation theorems* for randomized communication complexity due to [Ans+17; GPW17].

The main steps in our proofs are as follows. First, we prove a randomized *query complexity* hardness result for the problem of finding the end of a line in a particular constant-degree graph. Then we use a *simulation theorem* of [Ans+17; GPW17] to "lift" this query complexity hardness result to randomized communication complexity hardness. We use the construction in Chapter 4 to embed this line as a continuous Lipschitz function $f : [0,1]^n \to [0,1]^n$. Finally, we build on ideas from [MT05; Shm12; Bab16] to construct a two-player (respectively n-player) "imitation game" that simulates both the short communication protocol for the computation of f, as well as a fixed-point verification. In particular, every (approximate) Nash equilibrium of the game corresponds to an approximate fixed-point of f, which in turn corresponds to an end of a line.

Since in this chapter we are proving unconditional intractability results, we have the privilege of reasoning about an explicit distribution of hard instances. In particular, it suffices to begin with the End-of-*the*-Line special case of the End-of-a-Line problem, where the graph consists of just one line — and we want to find *the* end of that line. This hardly changes the proof, but it makes the notation a little simpler. For example, it suffices to prove that a *decision* problem (find the most significant bit of the end of the line) is hard. Furthermore, our hardness now holds for the interesting case where the game has a unique Nash equilibrium.

3.0.3 Additional related literature

Pure Nash equilibrium The communication complexity of *pure* Nash equilibrium has been studied before: in two-player $N \times N$ games it is $\mathsf{poly}(N)$ [CS04], and in n-player games it is $\exp(n)$ [HM10].

Approximation Protocols For two-player $N \times N$ games and $\epsilon \approx 0.382$, [Czu+15] show that $\mathsf{polylog}(N)$ communication is sufficient for computing an ϵ-approximate Nash equilibrium (improving over a protocol for $\epsilon \approx 0.438$ due to [GP14]).

Query complexity There are several interesting results on the more restricted model of query complexity of approximate Nash equilibria, where the algorithm is assumed to have black-box access to the normal form representation of the utility function. Note that our communication complexity lower bounds immediately extend to this model as well.

Hart and Nisan [HN13] prove that any deterministic algorithm needs to query at least an exponential number of queries to compute any ϵ-Well Supported Nash Equilibrium - and even for any ϵ-correlated equilibrium. Babichenko [Bab16] showed that any randomized algorithm requires an exponential number of queries to find an ϵ-Well Supported Nash Equilibrium. Chen et al [CCT17] extended Babichenko's result to an almost-exponential ($2^{\Omega(n/\log n)}$) lower bound on the query complexity of ϵ-Approximate Nash Equilibrium. Note that our lower bound here is both bigger (saving the $\log n$ factor), holds for the more general notion of weak approximate Nash equilibrium, and in the more general model of communication complexity.

Goldberg and Roth [GR16] give a polynomial upper bound on the query complexity of ϵ-WSNE for any family of games that have *any* concise representation. This result is to be contrasted with (a) Babichenko's query complexity lower bound, which uses a larger family of games, and (b) our lower bounds on the *computational complexity* of succinctly-represented games (Theorem 5.0.2).

A much older yet very interesting and closely related result is that of Hirsch, Papadimitriou, and Vavasis [HPV89]. [HPV89] show that any deterministic algorithm for computing a Brouwer fixed point in the oracle model must make an exponential -in the dimension n and the approximation ϵ- number of queries for values of the function. Our construction here builds upon and improves over [HPV89] by working with the ℓ_2-norm instead of ℓ_∞-norm.

Correlated equilibrium For the related notion of correlated equilibrium, in n-player games with a constant number of actions, it is known that even exact correlated equilibrium can be computed using only $\mathsf{poly}(n)$-communication, see [HM10; PR08; JLB15]. Interestingly, for exact correlated equilibria, there is an exponential gap between the above communication protocol and the *query complexity* lower bound of [HN13; BB15]. [GR16] characterize the query complexity of approximate *coarse* correlated equilibrium in games with many players. Further discussion on correlated equilibria appears in Section 3.3.

Communication complexity of finding fixed points For the related problem of finding a fixed point, [RW16] study the communication complexity of approximate fixed point of the decomposition. Namely, Alice holds a Lipschitz function $f : A \to B$ Bob holds a Lipschitz function $g : B \to A$, where A and B are compact convex sets, and their goal is to compute a fixed point of the decomposition $g \circ f$. [RW16] prove that the following version of this problem is communicationally hard: find an approximate fixed point of $g \circ f$ on a grid of A, when it is promised that such an approximate fixed point *on the grid* exists (the problem is not total).

Complexity of equilibrium and Price of Anarchy As discussed earlier, the main motivation for studying the (communication) complexity of Nash equilibrium is understanding its relevance as a predictive solution concept. This is a good place to mention a recent work of Roughgarden [Rou14], which highlights another important motivation for studying the complexity of Nash equilibrium: understanding the *quality* of equilibria. The *Price of Anarchy (POA)* of a game is the ratio between the *social welfare* (sum of players' utilities) in an optimum strategy profile, and the social welfare in the worst Nash equilibrium of that game. Roughgarden [Rou14] provides the following widely applicable recipe for lower bounds on PoA: if a Nash equilibrium can be found efficiently (in particular, via the non-deterministic protocol due to [LMM03]), but approximating the optimal social welfare requires a higher communication complexity (even for non-deterministic protocols, e.g. by reduction from set disjointness), then clearly not all Nash equilibria yield high social welfare.

3.1 Proof overview

The formal proofs appear in Section 3.2. Below we present the main ideas of the proof. As mentioned in the Introduction, the proof consists of four main steps. Below we present the ideas of each step.

Query Complexity of End-of-the-Line

Our proof starts with the following *query complexity* hardness result (Lemma 3.2.2): There exists a constant degree graph $G = (V, E)$ with $2^{\Theta(n)}$ vertices, such that finding the end of a line in G requires $2^{\Omega(n)}$ queries. In fact, we prove the hardness result for directed graph G where each vertex has outgoing and incoming degree 2. Therefore, the successor and predecessor of each vertex are binary variables. In particular, for each $v \in V$, the information about its role in the line can be represented using only three bits, which we denote $I(v) \triangleq (T(v), P(v), S(v)) \in \{0,1\}^3$:

(a) Whether the line goes trough v, which is denoted by $T(v)$,

(b) Who is the successor of v (if v in on the line), which is denoted by $S(v)$,

(c) Who is the predecessor of v (if v in on the line), which is denoted by $P(v)$.

Lemma (Query End-of-a-Line; informal)**.** *Finding an end of a line with high probability, requires $2^{\Omega(n)}$ queries to I.*

From Query complexity to Communication Complexity

We use a recent *simulation theorem* to "lift" our randomized query complexity lower bound to a randomized communication complexity bound.

The simulated communicationally hard problem has the following form. For each $v \in V$, Alice holds a triplet of vectors $\alpha_{T,v}, \alpha_{S,v}, \alpha_{P,v} \in \{0,1\}^M$ where $M = 2^{O(n)}$, and Bob holds a reasonably small input which is just a triplet of indexes $\beta_{T,v}, \beta_{S,v}, \beta_{P,v} \in [M]$. $T(v)$ is given by the decomposition $T(v) = \alpha_{T,v}(\beta_{T,v})$ (similarly for the successor and predecessor). The simulation theorem of [GPW17; Ans+17] now implies:

Corollary (CC(Simulation End-of-a-Line); informal)**.** *Finding an end of a line requires $2^{\Omega(n)}$ bits of communication.*

Embedding as a continuous function

Our next step is to reduce the problem of finding an end of a line to that of finding a Brouwer fixed point. Here, we use the construction from Chapter 4

We embed the vertices of the discrete graph G in the continuous space $[-1,2]^{\Theta(n)}$. Specifically, we embed each vertex v of G into a point $\mathbf{x}_v$ in $[-1,2]^{\Theta(n)}$ and each edge (v,w) in G into a (continuous) path in $[-1,2]^{\Theta(n)}$ that connects the corresponding points $\mathbf{x}_v$ and $\mathbf{x}_w$. In particular, we construct a continuous, Lipschitz function $f : [-1,2]^{\Theta(n)} \to [-1,2]^{\Theta(n)}$ such that:

1. The computation of f can be done using *local information about I*. Namely, for points that are close to $\mathbf{x}_v$ it is sufficient to know $I(v)$ to compute f. For points that are close to a path that corresponds to the edge (v,w) but far from the points $\mathbf{x}_v, \mathbf{x}_w$ it is sufficient to know whether (v,w) is an edge in the line (in particular, knowing either $I(u)$ or $I(v)$ suffices). For points that are far from all paths (v,w), f does not depend on I at all (thus can be computed without any communication).

2. Any (normalized) $\|\cdot\|_2$-approximate fixed point of f can be mapped (efficiently) back to an end of some line in I.

Property 1 induces the following efficient communication protocol for computing $f(\mathbf{x})$: Bob finds v such that $\mathbf{x}$ is close to $\mathbf{x}_v$, and sends $\beta_{T,v}, \beta_{S,v}, \beta_{T,v}$; Alice replies with $I(v) = \Big(\alpha_{T,v}(\beta_{T,v}), \alpha_{T,v}(\beta_{T,v}), \alpha_{T,v}(\beta_{T,v})\Big)$, and they each use $I(v)$ to locally compute $f(\mathbf{x})$. (Similarly, if $\mathbf{x}$ is close to the path corresponding to edge (v,w), they use a similar protocol to compute $I(v)$ and $I(w)$.)

By Property 2, we have:

Corollary (CC(Brouwer); informal)**.** *Finding a (normalized) $\|\cdot\|_2$-approximate fixed point of f requires $2^{\Omega(n)}$ bits of communication.*

Two-player game

Naively thinking, we would like to design a game where Alice chooses a point $\mathbf{x} \in [-1,2]^{\Theta(n)}$ (on the ε-grid) and Bob chooses a point $\mathbf{y} \in [-1,2]^{\Theta(n)}$ (on the ε-grid). Alice's utility will be given by $-\|\mathbf{x}-\mathbf{y}\|_2^2$, and Bob's utility will be given by[4] $-\|\mathbf{y}-f(\mathbf{x})\|_2^2$. Then, by applying similar arguments to those in [MT05; Shm12; Bab16; Rub16] we can deduce that every approximate Nash equilibrium corresponds to an approximate fixed point, and thus also to an end of a line.

However, the above idea is obviously incorrect because Bob's utility depends on f, whereas in the communication problem his utility should depend on the βs only. Our key idea is to use the fact that f can be computed locally to design a somewhat similar game where similar phenomena to those in the "naive" game will occur in approximate equilibria.

Bob doesn't know f, but to compute $f(\mathbf{x})$ he should only know the local information about the vertex (or vertices) that correspond to $\mathbf{x}$. We incentivize Alice and Bob to combine their private information about the corresponding vertex (or vertices) by the following utilities structure.

- Alice's first component of utility is given by $-\|\mathbf{x}-\mathbf{y}\|_2^2$. As in the "naive" game, in any approximate Nash equilibrium Alice will play points in the ϵ-cube of the ϵ-grid that contains $\mathsf{E}[\mathbf{y}]$ with probability close to one.

- Bob tries to guess the vertex v (or the vertices v, w) that correspond to the point $\mathbf{x}$. Since $\mathbf{x}$ (almost always) belongs to the same ϵ-cube, in any (approximate) Nash equilibrium, his guess should be correct (with high probability). In addition, Bob should announce the β indexes β_T, β_S and β_P of v (of v and w). Again, we incentivize him to do so by defining that he should "guess" also these β indexes, and in an (approximate) equilibrium his guess should be correct (w.h.p).

- We want Alice to announce $I(v)$ (similarly for w in case of two vertices). Thus, we ask her to guess the decomposition $\alpha_{v^B}(\beta^B)$ where v^B and β^B are the announced v and β by Bob. In (approximate) equilibrium, since Bob announces the correct v and β (w.h.p), this incentivizes her to announce the correct $I(v)$ (w.h.p).

[4] Note that here it is crucial that we use the normalized $\|\cdot\|_2$ to obtain payoffs bounded in $[-9,0]$; using the non-normalized $\|\cdot\|_2$ we get payoffs in $[-\sqrt{n},0]$.

- Now Bob uses the local information of $I(v)$ (and $I(w)$) to compute f. Namely, his last utility component is defined by $-\left\|\mathbf{y} - f_{I^A(v),I^A(w)}(\mathbf{x})\right\|_2^2$ where $f_{I^A(v),I^A(w)}$ is Bob's "estimation" of f under the relevant local information announced by Alice. In (approximate) equilibrium Alice announces the correct local information (w.h.p), thus Bob computes f correctly (w.h.p).

Summarizing, the (approximate) equilibrium analysis of the presented game is similar to the analysis of the naive game, because in (approximate) equilibrium f is computed correctly (w.h.p). But unlike the naive game, here Alice's utility depends only on the αs and Bob's utility only on the βs.

n-player game: ϵ-WSNE

The n-player game reduction is based on the same ideas as the two-player reduction. For clarity, we present first the idea of a reduction that proves the following weaker result:

There exists a constant $\epsilon > 0$ such that the communication complexity of ϵ-well supported approximate Nash equilibrium in n-player games with constant number of actions for each player is at least 2^{cn} for some constant c.

After that, we explain how we can strengthen this result in two aspects: first to improve the constant-number-of-action to binary-action, second to improve the ϵ-well supported Nash equilibrium to (ϵ, ϵ)-weak approximate equilibrium.

The idea is to replace a single player- Alice- who chooses $\mathbf{x}$ in the ϵ-grid of $[-1, 2]^{\Theta(n)}$ by a population of $\Theta(n)$ players $\{p_{x_i}\}_{i \in \Theta(n)}$; each player p_{x_i} in the population is responsible for the ith coordinate of $\mathbf{x}$. The payoff of player p_{x_i} is given by $-|x_i - y_i|^2$. This incentivizes player p_{x_i} to play either a single, or two adjacent actions, in the ϵ-grid of the segment $[-1, 2]$ (in every WSNE). By looking at the action profile of all p_{x_i} players we get the same phenomenon as in the two-player case: every point x in the support of Alice's players belongs to the same ϵ-cube of the ϵ-grid.

Now, we replace the guess of $v \in \{0, 1\}^{\Theta(n)}$, that is done by Bob, by population of size $\Theta(n)$ where again each player is responsible to a single coordinate of v. Again in a WSNE all players will guess correctly.

Similarly for the guess of β: we think of $\beta \in [M]^3$ as an element of $\{0, 1\}^{3 \log M}$ and we construct a population of $3 \log M$ players, each controls a single bit.

Similarly for Alice's guesses of $I^A(v)$ and $I^A(v)$: we construct 6 players, each chooses a bit.

Finally, we again replace the choice of $\mathbf{y} \in [-1, 2]^{\Theta(n)}$ by a population of $\Theta(n)$ players p_{y_i}. Each is responsible to a single coordinate. The payoff of player p_{y_i} is

given by $-|y_i - [f_{I^A(v),I^A(w)}(\mathbf{x})]_i|^2$. The analysis of this game is very similar to the two-player game analysis.

n-player game: (ϵ,ϵ)-Weak ANE and binary actions

In the above reduction, the $\mathbf{x}$-type (and $\mathbf{y}$-type players) have $3/\epsilon$ actions each. To construct a binary action game we use the technique of [Bab16]. We replace each such player by a population of $3/\epsilon$ players, each is located at a point in the ϵ-grid of the segment $[-1,2]$. Player that is located at $j \in [-1,2]$ (on the ϵ-grid) has to choose between the two points j or $j+\epsilon$. In a WSNE all players are located from the left of y_i will choose $j+\epsilon$, and all players are located from the right of y_i will choose j.

More tricky, is to generalize this reduction to weak approximate equilibria. Recall that in weak approximate equilibria, a constant fraction of players may play an arbitrary suboptimal action. Here we take into account both,

1. Players that are not ϵ-best replying, and
2. Players that are ϵ-best replying, but put small positive weight on the inferior action (among the two) and the realization of their mixed action turned out to be the inferior action.

In order to be immune from these, irrational, small constant fraction of players, we use error correcting codes[5]. Let $E_\beta{:}\{0,1\}^{3\log M} \to \{0,1\}^{\Theta(3\log M)}$ be a good binary error correcting code. Instead of having a population of size $3\log M$ which tries to guess β, we replace it by a population of size $\Theta(3\log M)$ where each player tries to guess his bit in the *encoding* of β. Now even if a small constant fraction of players will act irrationally, the *decoding* of the action profile of the β-type players will turn out to be β. We use the same idea for all types of populations ($\mathbf{x}$-type, $\mathbf{y}$-type, v-type and I-type). This idea completes the reduction for weak approximate equilibria.

3.2 Proofs

In Section 3.2.1 we prove a randomized query lower bound for the end-of-the-line problem. In Section 3.2.2 we show how the lower bounds of Sections 3.2.1 can be "lifted" to get a hard problem in the randomized communication complexity models. In Sections 3.2.3, 3.2.4, and 3.2.5 we reduce the communicationally hard end-of-any-line problem to the approximate Nash equilibrium problem.

[5]In fact, we use error correcting codes even earlier, in [Rub16]'s modification construction of hard Brouwer function.

3.2.1 A randomized query complexity lower bound

Let G be a *directed* graph with the vertices $V = \{0,1\}^n \times \{0,1\}^n \times [n+1]$. Each vertex (v_1, v_2, k), where $v_1, v_2 \in \{0,1\}^n$ and $k \in [n]$, has two outgoing edges to the vertices $(v_1, v_2^{k+1}(0), k+1)$ and $(v_1, v_2^{k_1}(1), k+1)$, where $v^j(0) = (v_1, \ldots, v_{j-1}, 0, v_{j+1}, \ldots, v_n)$. We call $(v_1, v_2^{k+1}(0), k+1)$ the 0-*successor of* v, and $(v_1, v_2^{k+1}(1), k+1)$ the 1-*successor of* v. Each vertex $v = (v_1, v_2, n+1)$ has a single outgoing edge to the vertex $(v_2, v_1, 0)$. Note that the incoming degree of each vertex $v = (v_1, v_2, k) \in V$ is at most two. For $k = 1$ there is a single incoming edge from $(v_2, v_1, n+1)$. For $k > 1$ there are two incoming edges from $(v_1, v_2^k(0), k-1)$ and from $(v_1, v_2^k(1), k-1)$. We call $(v_1, v_2^k(0), k-1)$ the 0-*predecessor of* v, and $(v_1, v_2^k(1), k-1)$ the 1-*predecessor of* v.

We define the Query End-of-the-Line to be the problem of finding the end of a line in G that starts at the point $\mathbf{0}_{2n+1}$. More formally, we represent a line in G by a triple $I(v) \triangleq (T(v), S(v), P(v))$ where $T(v) \in \{0,1\}$ indicates whether the line goes through v, $S(v) \in \{0,1\}$ indicates who is the successor of v, and $P(v) \in \{0,1\}$ indicates who is the predecessor of v (here we use the fact that each vertex has outgoing and incoming degree of at most two). Throughout the paper, we slightly abuse notation and use $S(v)/P(v)$ to refer both to the bits, and to the corresponding vertices (i.e. the $S(v)/P(v)$-successor/predecessor of v). The end of the line is the vertex v^* such that $T(v^*) = 1$ but $T(S(v^*)) = 0$.

Definition 3.2.1 (Query End-of-the-Line)**. Input:** A line $I = (T, S, P)$ over the graph G that starts at the point $\mathbf{0}_{2n+1}$.
Output: The first bit $([v^*]_1)$ of the end of the line vertex.
Queries: Each query is a vertex $v \in V$. The answer is the triplet of bits $I(v) = (T(v), S(v), P(v)) \in \{0,1\}^3$.

Lemma 3.2.2 (Randomized query complexity)**.** *For every constant* $\delta < \frac{1}{2}$, $\mathsf{BPP}^{\mathrm{dt}}_{\delta}$(Query End-of-the-Lin
$\Omega(2^n)$.

Proof. By Yao's Minmax Theorem it is sufficient to introduce a distribution over paths such every deterministic query algorithm requires $\Omega(2^n)$ queries to determine the first bit of the end of line vertex with probability of at least $1 - \delta$. We choose a permutation π over $\{0,1\}^n \setminus \{\mathbf{0}_n\}$ uniformly at random, and set $\pi(0) \triangleq \mathbf{0}_n$. π induces a line of length $\Theta(2^n \cdot n)$ over G, starting at $\mathbf{0}_{2n+1}$, ending at $(\pi(2^n - 1), \pi(2^n - 1), 0)$, and where two consecutive vertices $v = \pi(i)$ and $w = \pi(i+1)$ are mapped to the following line of $n+1$ edges:

$$(v, v, 0) \to \cdots \to (v, (w_{[1,k]}, v_{[k+1,n]}), k) \to \\ \to \cdots \to (v, w, n) \to (w, w, 0).$$

Here $(w_{[1,k]}, v_{[k+1,n]})$ denotes the vector with first k coordinates as in w and the last $n-k$ coordinates as in v.

The information of a single query of Query End-of-the-Line (for the above class of lines) can be extracted from $\pi(i-1), \pi(i)$ and $\pi(i+1)$. Therefore Query End-of-the-Line is at least as hard as the problem of finding the first bit of the last element in a random permutation, where each query returns the previous, the current, and the next vertices. Conditioning on the answers to k queries $\pi(q_1-1), \pi(q_1), \pi(q_1+1), \ldots, \pi(q_k-1), \pi(q_k), \pi(q_k+1)$, the last element of the permutation is still uniformly random across all vertices that are not $\pi(q_1), \ldots, \pi(q_k), \pi(q_1-1), \ldots, \pi(q_k-1), \pi(q_1+1), \ldots, \pi(q_k+1)$. This proves that the latter problem requires $\Omega(2^n)$ queries. □

3.2.2 Communicationally hard, discrete end-of-any-line problem

In order to use a simulation theorem (Theorem 2.7.9) for randomized communication complexity), we define the following simulation variant of Query End-of-the-Line:

Definition 3.2.3 (Simulation End-of-the-Line)**.** We let $N = 2^n \cdot 2^n \cdot (n+1) \cdot 3$.
Input: For each $v \in \{0,1\}^n \times \{0,1\}^n \times [n+1]$, Alice receives three vectors $\alpha_v^T, \alpha_v^S, \alpha_v^P \in \{0,1\}^M$, and Bob receives three indices $\beta_v^T, \beta_v^S, \beta_v^P \in [M]$.

We define

$$T(v) = \alpha_v^T(\beta_v^T), S(v) = \alpha_v^S(\beta_v^S), \text{ and } P(v) = \alpha_v^P(\beta_v^P). \tag{3.1}$$

We simulate the problem Query End-of-the-Line, therefore we restrict attention to inputs such that (T, S, P) that are defined in (3.1) meet all the requirements of Query End-of-the-Line.

Output: The first bit $([v^*]_1)$ of a non-trivial end or start of a line $(v^*, v^*, 0) \neq \mathbf{0}_{2n+1}$.

Applying the randomized Simulation Theorem (Theorem 2.7.9) to the query complexity lower bound (Lemma 3.2.2) gives a lower bound on the randomized communication complexity of a discrete end of line problem Simulation End-of-the-Line.

Corollary 3.2.4. $\mathsf{BPP}^{cc}_{0.3}$ (Simulation End-of-the-Line) $= \Omega(2^n)$.

3.2.3 Embedding a line as a local Lipschitz function

We embed I as a Euclidean-norm hard continuous function a-la Section 4.2. Below, we recall some of the properties of the construction that will be useful for our reduction.

It will be more convenient to think of G as a graph over $\{0,1\}^{2n+\log(n+1)}$.

Let $m = \Theta(2n + \log(n+1)) = \Theta(n)$ and let $E\colon\{0,1\}^{2n+\log(n+1)} \to \{0,1\}^m$ denote the encoding function of a good binary error correcting code. We embed the discrete graph G into the continuous cube $[-1,2]^{4m}$.

The vertex v is embedded to the point $(E(v), E(v), \mathbf{0}_m, \mathbf{0}_m) \in [-1,2]^{4m}$, which is called *the embedded vertex*.

For two vertices $v, w \in V$ such that (v,w) is an edge in the graph G, we define five vertices:

$$\begin{aligned}
\mathbf{x}^1(v,w) &\triangleq (E(v), E(v), \mathbf{0}_m, \mathbf{0}_m)\\
\mathbf{x}^2(v,w) &\triangleq (E(v), E(v), \mathbf{1}_m, \mathbf{0}_m)\\
\mathbf{x}^3(v,w) &\triangleq (E(v), E(w), \mathbf{1}_m, \mathbf{0}_m)\\
\mathbf{x}^4(v,w) &\triangleq (E(v), E(w), \mathbf{0}_m, \mathbf{0}_m)\\
\mathbf{x}^5(v,w) &\triangleq (E(w), E(w), \mathbf{0}_m, \mathbf{0}_m)\,.
\end{aligned}$$

Note that $\mathbf{x}^1(v,w)$ is the embedded vertex v, $\mathbf{x}^5(v,w)$ is the embedded vertex w.

The line that connects the points $\mathbf{x}^i(v,w)$ and $\mathbf{x}^{i+1}(v,w)$ is called a *Brouwer line segment*. The union of these four Brouwer line segments is called the *embedded edge* (v,w). It is not hard to check that non-consecutive Brouwer line segments are $\Omega(1)$-far one from the other, and in particular it implies that non-consecutive embedded edges are sufficiently far one from the other.

The following Proposition shows that the End-of-the-Line problem can be reduced to the problem of finding an approximate fixed point of a continuous Lipschitz function, when the function is "local" in the following sense: every edge in G is embedded as a path in the continuous hypercube (as described above). For points close to the embedding of an edge, f depends only on the "local behaviour" of the lines I at the endpoints of this edge; for all other points, f is independent of the lines I.

Proposition 3.2.5 (Theorem 4.2.1 and Fact 4.2.2)**.** *There exist constants $\delta, h > 0$ such that given a line $I = (T,S,P)$ over G there exists a function $f = f(I) = :[-1,2]^{4m} \to [-1,2]^{4m}$ with the following properties:*

1. *$\|f(\mathbf{x}) - \mathbf{x}\|_2 > \delta$ for every $\mathbf{x}$ that is not h-close to the embedded edge of the end of the line (i.e., the embedding of the edge $(P(v^*), v^*)$.*

2. *f is $O(1)$-Lipschitz in $\|\cdot\|_2$ norm.*

3. *f is* local *in the sense that it can be defined as an interpolation between a few (in fact, 64) functions, $\{f_{I_1,I_2} : [-1,2]^{4m} \to [-1,2]^{4m}\}_{I_i \in \{0,1\}^3}$, that do not depend on the lines I and such that:*

   a) *If the first m-tuple of coordinates of $\mathbf{x}$ is $6h$-close to the encoded vertex $E(v)$, but the second m-tuple of coordinates of $\mathbf{x}$ is $6h$-far from any encoded vertex $E(w)$ then $f_{I(v),I_2}(\mathbf{x}) = f(\mathbf{x})$ for every $I_2 \in \{0,1\}^3$.*

   b) *If the second m-tuple of coordinates of $\mathbf{x}$ is $6h$-close to the encoded vertex $E(w)$, but the first m-tuple of coordinates of $\mathbf{x}$ is $6h$-far from any encoded vertex $E(v)$ then $f_{I_1,I(w)}(\mathbf{x}) = f(\mathbf{x})$ for every $I_1 \in \{0,1\}^3$.*

   c) *If the first m-tuple of coordinates of $\mathbf{x}$ is $6h$-close to the encoded vertex $E(v)$, and the second m-tuple of coordinates of $\mathbf{x}$ is $6h$-close to the encoded vertex $E(w)$ then $f_{(I(v),I(w)}(\mathbf{x}) = f(\mathbf{x})$.*

   d) *If none of the above conditions are satisfied, then $f_{I_1,I_2}(\mathbf{x}) = f(\mathbf{x})$ for every $I_1, I_2 \in \{0,1\}^3$.*

3.2.4 Two-Player game

Theorem (Theorem 3.0.1, restated)**.** *There exists a constant $\epsilon > 0$ such that the communication complexity of ϵ-Nash equilibrium in two-player $N \times N$ games is at least N^ϵ.*

We construct a two-player game between Alice and Bob of size $N_A \times N_B$ for

$$N_A \triangleq (3/\epsilon)^{4m} \cdot 2^3 = 2^{\Theta(n)}$$
$$N_B \triangleq (3/\epsilon)^{4m} \cdot (2^{2n+\log(n+1)})^2 \cdot M^3 = 2^{\Theta(n)}.$$

such that Alice's utility depends on $\{\alpha_v^T, \alpha_v^S, \alpha_v^P\}_v$ only, Bob's utility depends on $\{\beta_v^T, \beta_v^S, \beta_v^P\}_v$ only, and all ϵ^4-approximate Nash equilibria of the game correspond to a δ-fixed point of f from Proposition 3.2.5. By property 1 in Proposition 3.2.5, any fixed point of f corresponds to a non-trivial end or start of a line in I.

3.2.4.1 The game

In this subsection we construct our reduction from Simulation End-of-the-Line to the problem of finding an ϵ-WSNE.

Strategies

Recall that δ is the desired approximation parameter for Brouwer fixed point in the construction of Proposition 3.2.5. We let ϵ be a sufficiently small constant; in particular, $\epsilon = O(\delta)$ (this will be important later for Inequality (3.10)).

Each of Alice's actions corresponds to an ordered tuple $(\mathbf{x}, I_v^A, I_w^A)$, where:

- $\mathbf{x} \in [-1,2]^{4m}$, where the interval $[-1,2]$ is discretized into $\{-1, -1+\epsilon, \ldots, 2-\epsilon, 2\}$;
- $I_v^A \triangleq (t_v^A, s_v^A, p_v^A) \in \{0,1\}^3$ and $I_w^A \triangleq (t_w^A, s_w^A, p_w^A) \in \{0,1\}^3$.

Each of Bob's actions corresponds to an ordered tuple $(\mathbf{y}, v^B, w^B, \beta_v^B, \beta_w^B)$, where:

- $\mathbf{y} \in [-1,2]^{4m}$, where the interval $[-1,2]$ is discretized into $\{-1, -1+\epsilon, \ldots, 2-\epsilon, 2\}$;
- $v^B, w^B \in \{0,1\}^{2n+\log(n+1)}$ are vertices in the graph G.
- $\beta_v^B = (\beta_v^{B,T}, \beta_v^{B,S}, \beta_v^{B,P}) \in [M]^3$ and $\beta_w^B = (\beta_w^{B,T}, \beta_w^{B,S}, \beta_w^{B,P}) \in [M]^3$ are triples of indexes.

Utilities

Alice's and Bob's utilities decompose as

$$\begin{aligned} U^A &\triangleq U^A_{\textsc{Imitation}} + U^A_{\textsc{GuessV}} + U^A_{\textsc{GuessW}}. \\ U^B &\triangleq U^B_{\textsc{Brouwer}} + U^B_{\textsc{GuessV}} + U^B_{\textsc{GuessW}}. \end{aligned}$$

The first component of Alice's utility depends only on the first components of her and Bob's strategies; it is given by:

$$U^A_{\textsc{imitation}}(\mathbf{x};\mathbf{y}) \triangleq -\|\mathbf{x}-\mathbf{y}\|_2^2.$$

Given the first component $\mathbf{x} \in [-1,2]^{4m}$ of Alice's strategy, we define two decoding functions $D_v, D_w : [-1,2]^{4m} \to \{0,1\}^n$ as follows. Let $R_v(\mathbf{x}) \in \{0,1\}^m$ be the rounding of the first m-tuple of coordinates of $\mathbf{x}$ to $\{0,1\}^m$; let $D_v(\mathbf{x}) = E^{-1}(R_v(\mathbf{x})) \in \{0,1\}^{2n+\log(n+1)}$, where E^{-1} denote the decoding of the error correcting code from Section 3.2.3. We define $D_w(\mathbf{x}) \in \{0,1\}^{2n+\log(n+1)}$ analogously with respect to the second m-tuple of coordinates of $\mathbf{x}$. The second components of Bob's utility is now given by $U^B_{\textsc{GuessV}} = 1$ iff he guesses correctly the vertex $D_v(\mathbf{x})$, and the corresponding β operation on this vertex. Namely, $U^B_{\textsc{GuessV}}(v^B, \beta_v^B; \mathbf{x}) = 1$ if $v^B = D_v(\mathbf{x})$ and $\beta_v^B = (\beta^T_{D_v(\mathbf{x})}, \beta^S_{D_v(\mathbf{x})}, \beta^P_{D_v(\mathbf{x})})$, and $U^B_{\textsc{GuessV}}(v^B, \beta_v^B; \mathbf{x}) = 0$ otherwise. Similarly we define Bob's third component $U^B_{\textsc{GuessW}}$ with respect to $D_w(\mathbf{x})$.

Note that Bob knows the indexes $\beta_v^T, \beta_v^S, \beta_v^P$ (for every v), thus to achieve $U^B_{\text{GUESS}} = 1$ Bob needs to guess correctly only the vertices $D_v(\mathbf{x}), D_w(\mathbf{x})$ and announce the corresponding triplet of β indexes.

Going back to Alice, the second component of her utility is given by $U^A_{\text{GUESSV}} = 1$ iff she guesses correctly the triplet $I(v^B) = (T(v^B), S(v^B), P(v^B))$ when the calculation of T, S, P is done by the decomposition of $\alpha(\beta^B)$. Namely, $U^A_{\text{GUESSV}}(I_v^A; v^B, \beta^B) = 1$ if $I_v^A = (\alpha^T_{v^B}(\beta_v^{B,T}), \alpha^S_{v^B}(\beta_v^{B,S}), \alpha^P_{v^B}(\beta_v^{B,P}))$, and $U^A_{\text{GUESSV}}(I_v^A; v^B, \beta^B) = 0$ otherwise. Similarly we define Alice's third component U^B_{GUESSW}.

Finally, the first component of Bob's utility is given by:

$$U^B_{\text{BROUWER}}(\mathbf{y}; \mathbf{x}, e^A) \triangleq -\left\| f_{I_v^A, I_w^A}(\mathbf{x}) - \mathbf{y} \right\|_2^2.$$

where the function f_{I_1,I_2} is defined in Proposition 3.2.5.

3.2.4.2 Analysis of game

In this subsection, we prove the reduction from SIMULATION END-OF-THE-LINE to finding an ϵ^4-ANE. The proof proceeds via a sequence of lemmas that establish the structure of any ϵ^4-ANE.

Lemma 3.2.6. *In every ϵ^4-ANE $(\mathcal{A}; \mathcal{B})$, it holds that $\|\mathbf{x} - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2^2 = O(\epsilon^2)$ with probability of at least $1 - \epsilon^2$ (where the probability is taken over $\mathcal{A}$).*

Proof. We denote $\mathsf{E}_i(\mathcal{B}) = \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[y_i]$, $\mathsf{E}(\mathcal{B}) = (\mathsf{E}_1(\mathcal{B}), \ldots, \mathsf{E}_n(\mathcal{B}))$ is the vector of expectations, and $\text{Var}(\mathcal{B}) = (\text{Var}_{\mathbf{y}\sim\mathcal{B}}[y_1], \ldots, \text{Var}_{\mathbf{y}\sim\mathcal{B}}[y_n])$ is the vector of variances. For every $\mathbf{x}$ we can rewrite

$$\begin{aligned} U^A_{\text{IMITATION}}(\mathbf{x}, \mathcal{B}) &= -\mathsf{E}_{\mathbf{y}\sim\mathcal{B}} \|\mathbf{x} - \mathbf{y}\|_2^2 \\ &= -\frac{1}{4m} \sum_{i\in[4m]} \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}\left[(x_i - y_i)^2\right] \\ &= -\frac{1}{4m} \sum_{i\in[4m]} \left[(x_i - y_i(\mathcal{B}))^2 + \text{Var}_{\mathbf{y}\sim\mathcal{B}}[y_i]\right] \\ &= -\|\mathbf{x} - \mathsf{E}(\mathcal{B})\|_2^2 - \|\text{Var}(\mathcal{B})\|_2^2. \end{aligned} \tag{3.2}$$

Since the variance of the y_i's, as well as U^A_{GUESSV} and U^A_{GUESSW}, do not depend on $\mathbf{x}$, Alice's best response to $\mathcal{B}$ is

$$\mathbf{x}^* = ([\mathsf{E}_1(\mathcal{B})]_\epsilon, \ldots, [\mathsf{E}_n(\mathcal{B})]_\epsilon)$$

where $[\cdot]_\epsilon$ denotes the rounding to the closest ϵ integer multiplication. $\mathbf{x}^*$ yields a payoff of at least

$$U^A_{\text{IMITATION}}(\mathbf{x}^*, \mathcal{B}) \geq -\frac{\epsilon^2}{4} - \|\text{Var}(\mathcal{B})\|_2^2.$$

Note that in every ϵ^4-ANE Alice assigns a probability of at most $1-\epsilon^2$ to actions that are ϵ^2-far from optimal. By Equation (3.2) this implies that the probability of Alice to choose a vector $\mathbf{x}$ that satisfies $\|\mathbf{x} - \mathsf{E}(\mathcal{B})\|_2^2 \geq \epsilon^2 + \frac{\epsilon^2}{4}$ is at most ϵ^2. □

Lemma 3.2.7. *In every ϵ^4-ANE $(\mathcal{A};\mathcal{B})$, if the first m-tuple of coordinates of $\mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]$ is $6h$-close to the binary encoding $E(v)$ of a vertex v, then*

$$v^B = v, \text{ and } \beta_v^B = (\beta_v^T, \beta_v^S, \beta_v^P) \tag{3.3}$$

with probability of at least $1-O(\epsilon^4)$ (where the probability is taken over $\mathcal{B}$).

Proof. By Lemma 3.2.6 and the triangle inequality, with probability of at least $1-\epsilon^2$, the first m-tuple of $\mathbf{x}$ is $O(h)$-close to $E(v)$. Rounding to $R_v(\mathbf{x}) \in \{0,1\}^m$ can at most double the distance to $E(v)$ in each coordinate. Therefore, the Hamming distance of $R_v(\mathbf{x})$ and $E(v)$ is $O(h)$. Hence $R_v(\mathbf{x})$ is correctly decoded as $D_v(\mathbf{x}) = v$, with probability of at least $1-\epsilon^2$.

Since v^B, β_v^B do not affect $U^B_{\text{BROUWER}} + U^B_{\text{GUESSW}}$, Bob's utility from guessing $v^B = v$, and $\beta_v^B = (\beta_v^T, \beta_v^S, \beta_v^P)$ is at least $1-\epsilon^2$. Whereas his utility from guessing any other guess is at most ϵ^2. Therefore, Bob assigns probability at least $1-\epsilon^4/(1-2\epsilon^2)$ to actions that satisfy (3.3). □

A similar lemma holds for the second m-tuple of $\mathbf{x}$ and the vertex w:

Lemma 3.2.8. *In every ϵ^4-ANE $(\mathcal{A};\mathcal{B})$, if the second m-tuple of coordinates of $\mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]$ is $6h$-close to the binary encoding $E(W)$ of a vertex w, then*

$$w^B = w, \text{ and } \beta_w^B = (\beta_w^T, \beta_w^S, \beta_w^P)$$

with probability of at least $1-O(\epsilon^4)$ (where the probability is taken over $\mathcal{B}$).

Since Alice receives the correct v^B and β^B, we also have:

Lemma 3.2.9. *In every ϵ^4-ANE $(\mathcal{A};\mathcal{B})$, if the first m-tuple of coordinates of $\mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]$ is $6h$-close to the binary encoding $E(v)$ of a vertex v, then*

$$I_v^A = (\alpha_v^T(\beta_v^T), \alpha_v^S(\beta_v^S), \alpha_v^P(\beta_v^P))$$

with probability $1-O(\epsilon^4)$ (where the probability is taken over $\mathcal{A}$ and $\mathcal{B}$).

Proof. Follows immediately from Lemma 3.2.7 and the fact that I_v^A does not affect $U^A_{\text{Imitation}} + U^A_{\text{GuessW}}$. □

A similar lemma holds for the second m-tuple of $\mathbf{x}$ and the vertex w:

Lemma 3.2.10. *In every ϵ^4-ANE $(\mathcal{A};\mathcal{B})$, if the second m-tuple of coordinates of $\mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]$ is $6h$-close to the binary encoding $E(W)$ of a vertex w, then*

$$I_w^A = (\alpha_w^T(\beta_w^T), \alpha_w^S(\beta_w^S), \alpha_w^P(\beta_w^P))$$

with probability $1 - O(\epsilon^4)$ (where the probability is taken over $\mathcal{A}$ and $\mathcal{B}$).

Lemma 3.2.11. *In every ϵ^4-ANE $(\mathcal{A};\mathcal{B})$, $f_{I_v^A, I_w^A}(\mathbf{x}) = f(\mathbf{x})$ with probability $1-O(\epsilon^2)$.*

Proof. Follows immediately from Lemmas 3.2.9 and 3.2.10 and the "locality" condition in Proposition 3.2.5. □

The following corollary completes the analysis of the 2-player game.

Corollary 3.2.12. *In every ϵ^4-ANE $(\mathcal{A};\mathcal{B})$, $\|\mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}[\mathbf{x}'] - f(\mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}[\mathbf{x}'])\|_2 < \delta$.*

Proof. We recall that in Lemma 3.2.6 we have proved that

$$\|\mathbf{x} - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2^2 = O(\epsilon^2) \tag{3.4}$$

with probability $1 - O(\epsilon^2)$. This also implies that $\mathbf{x}$ is, with high probability, close to its expectation:

$$\begin{aligned}
\|\mathbf{x} - \mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}[\mathbf{x}']\|_2^2 &\le \left(\|\mathbf{x} - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2 + \|\mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}[\mathbf{x}'] - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2 \right)^2 \\
&\le 2\|\mathbf{x} - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2^2 + 2\|\mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}[\mathbf{x}'] - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2^2 \\
&\le 2\|\mathbf{x} - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2^2 + 2\mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}\left[\|\mathbf{x}' - \mathsf{E}_{\mathbf{y}\sim\mathcal{B}}[\mathbf{y}]\|_2^2\right] \\
&= O(\epsilon^2),
\end{aligned} \tag{3.5}$$

with probability $1 - O(\epsilon^2)$. Where the first inequality follows form the Triangle ineqaultiy, the second follows from the Arithmetic-Mean Geometric-Mean inequality AM-GM ineqaulty, the third follows from convexity of $\|\cdot\|_2^2$, and the last follows from Lemma 3.2.6.

Using that f is $O(1)$-Lipschitz together with Equation (3.5), we get that

$$\|f(\mathbf{x}) - f(\mathsf{E}_{\mathbf{x}'\sim\mathcal{A}}[\mathbf{x}'])\|_2^2 = O(\epsilon^2) \tag{3.6}$$

with probability $1 - O(\epsilon^2)$.

By Lemma 3.2.11 we know that $f_{I_v^A, I_w^A}(\mathbf{x}) = f(\mathbf{x})$ with probability $1-O(\epsilon^2)$, which implies that

$$\left\| \mathsf{E}_{\mathbf{x}' \sim \mathcal{A}}[f_{I_v^A, I_w^A}(\mathbf{x}')] - \mathsf{E}_{\mathbf{x}' \sim \mathcal{A}}[f(\mathbf{x}')] \right\|_2^2 = O(\epsilon^2). \tag{3.7}$$

Using similar arguments to those of Lemma 3.2.6 we can show that

$$\left\| \mathbf{y} - \mathsf{E}_{\mathbf{x}' \sim \mathcal{A}}[f_{I_v^A, I_w^A}(\mathbf{x}')] \right\|_2^2 = O(\epsilon^2) \tag{3.8}$$

with probability $1 - O(\epsilon^2)$. As in the derivation of Equation (3.5), this implies:

$$\left\| \mathbf{y} - \mathsf{E}_{\mathbf{y}' \sim \mathcal{B}}[\mathbf{y}'] \right\|_2^2 = O(\epsilon^2) \tag{3.9}$$

with probability $1 - O(\epsilon^2)$.

With probability $1 - O(\epsilon^2)$ Inequalities (3.5),(3.4),(3.9),(3.8),(3.7),(3.6) hold simultaneously. In such a case, by the triangle inequality and by applying the inequalities in the exact above order, we have

$$\left\| \mathsf{E}_{\mathbf{x}' \sim \mathcal{A}}[\mathbf{x}'] - f(\mathsf{E}_{\mathbf{x}' \sim \mathcal{A}}[\mathbf{x}']) \right\|_2^2 = O(\epsilon^2) < \delta^2. \tag{3.10}$$

□

Proof of Theorem 3.0.1. Any communication protocol that solves the ϵ^4-Nash equilibrium problem in games of size $N \times N$ for $N = 2^{\Theta(n)}$ induces a communication protocol for the problem Simulation End-of-the-Line: Alice constructs her utility in the above presented game using her private information of the αs, Bob constructs his utility using the βs. They implement the communication protocol to find an ϵ^4-Nash equilibrium, and then both of them know $\mathsf{E}_{\mathbf{x} \sim \mathcal{A}}[\mathbf{x}]$ which is a δ-approximate fixed point of f (by Corollary 3.2.12). Using D_v they decode the vertex v^* and they know the first coordinate of v^*.

Using Corollary 3.2.4 we deduce that the communication complexity of ϵ^4-Nash equilibrium in games of size $2^{\Theta(n)} \times 2^{\Theta(n)}$ is at least $2^{\Omega(n)}$. □

3.2.5 n-player game

Theorem (Theorem 3.0.2, restated)**.** *There exists a constant $\epsilon > 0$ such that the communication complexity of (ϵ, ϵ)-weak approximate Nash equilibrium in n-player binary-action games is at least $2^{\epsilon n}$.*

The proof follows similar lines to those in the proof of Theorem 3.0.1. Rather than two players whose actions correspond to $\Theta(n)$-long vectors, we have a player for each bit of (an encoding of) those vectors. We construct a game with $8m'$-players for $m' = \Theta(n)$ such that Alice holds the utility function of (the first) $3m'$ players, Bob holds the utilities of (the last) $5m'$ players, Alice's players utilities depend only on the αs, Bob's utilities depend only on the βs, and every $(\epsilon^5/82, \epsilon^5/82)$-weak approximate Nash equilibrium corresponds to a δ-fixed point of the function f from Proposition 3.2.5.

Players and Actions

In section 3.2.4 we have used error correcting code to encode vertices that are deduced from $\mathbf{x}$ and $\mathbf{y}$. Here, since we consider *weak* approximate equilibria, we should add additional encodings for $I_v^A, I_v^A, v^B, w^B, \beta_v^B$ and β_w^B. Since we want to use the same number of players for each of the above objects, it will be convenient to encode them in the same space $\{0,1\}^{m'}$. We let the following be encoding functions of binary error correcting codes with constant (relative) distance:

- $E_I : \{0,1\}^3 \to \{0,1\}^{m'}$.
- $E_u : \{0,1\}^{2n+\log(n+1)} \to \{0,1\}^{m'}$.
- $E_\beta : \{0,1\}^{3\log M} \to \{0,1\}^{m'}$ (note that $3\log M = \Theta(n)$).

Let E and m denote encoding function and block length of the error correcting code from Section 3.2.3, i.e.:

- $E : \{0,1\}^{2n+\log(n+1)} \to \{0,1\}^{m}$.

For vectors $\mathbf{x}, \mathbf{y} \in [-1,2]^{4m}$, we use $(\frac{3}{\epsilon} - 1)$ bits to encode each continuous coordinate (up to precision ϵ) in unary encoding. We choose m' such that $m' = 4(\frac{3}{\epsilon} - 1)m$, so the encoding of each of $\mathbf{x}, \mathbf{y}$ also takes m' bits. (For E_β, we must also have $m' > 3\log M$.) Here and henceforth, ϵ is a sufficiently small constant, satisfying $\epsilon = \Theta(\delta)$.

Instead of having a single player, Alice, with actions $(\mathbf{x}, I_v^A, I_w^A) \in \{-1, -1+\epsilon, \ldots, 2-\epsilon, 2\}^m \times \{0,1\}^3 \times \{0,1\}^3$ we replace her by $3m'$ players with binary actions. We have three types of Alice players:

- $\mathbf{x}$-type players. Player $\mathbf{x}_j^i$ chooses one of the actions $a_j^i \in \{j, j+\epsilon\}$ for every $i \in [4m]$ and $j \in \{-1, -1+\epsilon, \ldots, 2-2\epsilon, 2-\epsilon\}$. Note that the total number of $\mathbf{x}$-type players is $4m(\frac{3}{\epsilon} - 1) = m'$.

- I^v-type and I^w-type players. Player I^v_i chooses a bit 0 or 1 for every $i \in [m']$. Similarly for I^w-type players.

In the communication problem, we assume that Alice knows the utilities of all the above players.

Instead of having a single player, Bob, with actions $(\mathbf{y}, v^B, w^B, \beta^B_v, \beta^B_w) \in \{-1, -1+\epsilon, \ldots, 2-\epsilon, 2\}^m \times \{0,1\}^{2n+\log(n+1)} \times \{0,1\}^{2n+\log(n+1)} \times [M]^3 \times [M]^3$ we replace him by $5m'$ players with binary actions. We have five types of players:

- $\mathbf{y}$-type players. Player $\mathbf{y}^i_j$ chooses one of the actions $b^i_j \in \{j, j+\epsilon\}$ for every $i \in [4m]$ and $j \in \{-1, -1+\epsilon, \ldots, 2-2\epsilon, 2-\epsilon\}$.

- v-type players. Player v_i chooses a bit 0 or 1 for every $i \in [m']$. Similarly for w-type players.

- β^v-type players. Player β^v_i chooses a bit 0 or 1 for every $i \in [m']$. Similarly for β^w-type players.

In the communication problem, we assume that Bob knows the utilities of all the above players.

Utilities

Before getting to the description of the utilities we define the notions of *realized number* and *realized point* by a set of players. For every $i \in [m]$, for simplicity of notations we add a dummy player $\mathbf{x}^i_2$ who has a single action $a^i_2 = 2$. Given an action profile $a^i = (a^i_{-1}, a^i_{-1+\epsilon}, \ldots, a^i_2)$ of the players $\{\mathbf{x}^i_j\}_j$, the *realized number* $r(a^i) \in [-1, 2]$ is defined to be the minimal j such that $a^i_j = j$. Note that $r(a_i)$ is well defined because the last player $\mathbf{x}^i_2$ plays 2. Given an action profile $a = (a^i_j)_{i,j}$ of all $\mathbf{x}$-type players we denote by $r(a) = (r(a_i))_i \in [-1, 2]^m$ the realized point. Similarly we define the realized point of $\mathbf{y}$-type players.

The utilities are defined similarly to the two-player case with the following differences:

1. $\mathbf{x}$-type/ $\mathbf{y}$-type players' utilities are defined with respect to the *realized* points of the opponents. In addition, player that is responsible to the i-th coordinate of the point pays the distance from the i-th coordinate of the opponent's point/the ith coordinate of the f operation of the opponent's point.

2. For all other types, the i-th player chooses the value of the i-th bit in the (alleged) codeword in $\{0,1\}^{m'}$.

Formally the payoffs are defined as follows:

- For $\mathbf{x}$-type players, $U^{\mathbf{x}^i_j}\left(a^i_j; b^i\right) \triangleq -|a^i_j - r(b^i)|^2$, where we recall that player $\mathbf{x}^i_j$ is allowed to choose only $a_{i,j} = j$ or $a_{i,j} = j + \epsilon$, and b^i is the profile of action played by players $\{\mathbf{y}^i_j\}_j$.
- For a v-type player v_i, we define $U^{v_i}(v_i; a) = 1$ iff he announces the bit $[E_u(D_v(r(a)))]_i$ (where the decoding function D_v is defined in Section 3.2.4.1). Otherwise, $U^{v_i}(v_i; a) = 0$. Namely, the i-th player tries to guess the i-th coordinate of the encoded vector $E_u(v) \in \{0,1\}^{m'}$, were v is computed using the decoding operation D_v on the realized point $r(a) \in [-1,2]^{4m}$. Similarly we define the utility of a w-type player.
- For a β^v-type player β^v_i, we define $U^{\beta^v_i}(\beta^v_i; a) = 1$ iff he announces the bit $[E_\beta(\beta^S_{D_v(r(a))})]_i$. Namely, the i-th player tries to guess the i-th coordinate of the encoded vector $E_\beta(\beta^S_v)$, were v, as in the previous bullet, is computed using decoding. Similarly we define the utilities of β^w-type players.
- For a I^v-type player I^v_i, we define $U^{I^v_i}(I^v_i, \beta^v) = 1$ iff she announces the bit $\left[E_u\left(\alpha^T_{\overline{v}}([\overline{\beta}]_1), \alpha^S_{\overline{v}}([\overline{\beta}]_2), \alpha^P_{\overline{v}}([\overline{\beta}]_3)\right)\right]_i$ where $\overline{v}$ is the decoded vertex announced by v-type players and $\overline{\beta}$ is the decoded vector of indexes announced by β^v-type players. Similarly we define the utilities of I^w-type players.
- For $\mathbf{y}$-type players, $U_{\mathbf{y}^i_j} = -|b^i_j - f_{\overline{I^v},\overline{I^w}}(r(a))|^2$, where $\overline{I^v}$ and $\overline{I^w}$ are the decoding of the vertices announced by I^v-type and I^w-type players. We recall that the function f_{I^v,I^w} is defined in Proposition 3.2.5.

3.2.5.1 Analysis of game

We analyse $(\overline{\epsilon}, \overline{\epsilon})$-weak approximate equilibria for $\overline{\epsilon} = \epsilon^5/82$. The analysis of the game follows the same sequence of Lemmas as the analysis in the two-player case (Section 3.2.4.2). The analogue of Lemma 3.2.6 is the following.

Lemma 3.2.13. *In every $(\overline{\epsilon}, \overline{\epsilon})$-weak approximate equilibrium $(\mathcal{A}, \mathcal{B})$, the realized point by the $\mathbf{x}$-type players $r(a)$ satisfies*

$$\|r(a) - \mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2 \leq \epsilon^2 \tag{3.11}$$

with high probability[6] (the probability is over the mixed strategy of the $\mathbf{x}$-type players).

[6] Here and throughout this section, we use "with high probability" to mean with probability approaching 1 as n grows (in fact, with an exponential dependence); in particular, the probability is approaching 1 faster than any polynomial in ϵ.

Proof. We say that player $\mathbf{x}_j^i$'s action j is *wrong* if $\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)] \geq j+\epsilon$; similarly, we say that action $j+\epsilon$ is *wrong* $\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)] \leq j$. Note that if for some coordinate i, *no* player $\mathbf{x}_j^i$ plays a wrong action, then the realized number $r_i(a_i)$ is ϵ-close to $\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]$. We show that indeed in an $(\bar{\epsilon},\bar{\epsilon})$-weak approximate equilibrium we will have many such coordinates i.

Recall that player $\mathbf{x}_j^i$'s utility when she plays j is given by

$$\begin{aligned} u(j) &\triangleq \mathsf{E}_{b^i\sim\mathcal{B}}\left[U^{\mathbf{x}_j^i}\left(j;b^i\right)\right] \\ &= \mathsf{E}_{b^i\sim\mathcal{B}}[-|j-r(b^i)|^2] \\ &= -|j-\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]|^2 - \mathrm{Var}_{b^i\sim\mathcal{B}}[r(b^i)]. \end{aligned}$$

Similarly, when she plays $j+\epsilon$ her utility is given by

$$\begin{aligned} u(j+\epsilon) &\triangleq \mathsf{E}_{b^i\sim\mathcal{B}}\left[U^{\mathbf{x}_j^i}\left(j+\epsilon;b^i\right)\right] \\ &= \mathsf{E}_{b^i\sim\mathcal{B}}[-|j+\epsilon-r(b^i)|^2] \\ &= -|j+\epsilon-\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]|^2 - \mathrm{Var}_{b^i\sim\mathcal{B}}[r(b^i)]. \end{aligned}$$

When j is wrong (i.e. $\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)] \geq j+\epsilon$) the difference in the utilities $u(j+\epsilon)-u(j)$ is given by

$$\begin{aligned} u(j+\epsilon)-u(j) &= -(\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]-j-\epsilon)^2 + (\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]-j)^2 \\ &= (2\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]-2j-\epsilon)\epsilon \geq \epsilon^2 \end{aligned}$$

For $j+\epsilon$ is wrong ($\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)] \leq j$) the difference in the utilities $u(j)-u(j+\epsilon)$ is given by

$$\begin{aligned} u(j)-u(j+\epsilon) &= -(j-\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)])^2 + (j+\epsilon-\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)])^2 \\ &= (2j-2\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]+\epsilon)\epsilon \geq \epsilon^2 \end{aligned}$$

Therefore, player $\mathbf{x}_j^i$ can always increase her payoff by at least ϵ^2 by deviating from a wrong action. Note that if player $\mathbf{x}_j^i$ is $\bar{\epsilon}$-best replying, she assigns a probability of at most $\bar{\epsilon}/\epsilon^2$ to a wrong action. In addition, the fraction of $\mathbf{x}$-type players that are not $\bar{\epsilon}$-best replying is at most $8\bar{\epsilon}$ (because we have 8 types of players of equal cardinality). Therefore, in the expected fraction of $\mathbf{x}$-type players playing a wrong is at most $8\bar{\epsilon}+2\bar{\epsilon}/\epsilon^2 < 2.5\bar{\epsilon}/\epsilon^2$. Therefore, with high probability over $\mathbf{x}$-type players mixed strategies, at most a $3\bar{\epsilon}/\epsilon^2$-fraction play a wrong action (e.g. by Chernoff bound). Therefore the fraction of coordinates $i \in [4m]$ where at least one player $\mathbf{x}_j^i$

plays a wrong action is at most $9\overline{\epsilon}/\epsilon^3$ (because we have $3/\epsilon$ players in each coordinate). So in $(1-9\overline{\epsilon}/\epsilon^3)$ fraction of coordinates we have $|r_i(a_i)-\mathsf{E}_{b^i\sim\mathcal{B}}[r(b^i)]| \le \epsilon$, which implies

$$\begin{aligned}\|r(a)-\mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2 &= \frac{1}{4m}\sum_i |r(a_i)-\mathsf{E}_{b\sim\mathcal{B}}[r(b^i)]|^2 \\ &\le (1-\frac{9\overline{\epsilon}}{\epsilon^3})\epsilon^2 + \frac{9\overline{\epsilon}}{\epsilon^3}3^2 < \frac{82\overline{\epsilon}}{\epsilon^3} = \epsilon^2\end{aligned}$$

□

The analogue of Lemma 3.2.7 is the following.

Lemma 3.2.14. *In every $(\overline{\epsilon},\overline{\epsilon})$-weak approximate equilibrium $(\mathcal{A},\mathcal{B})$, if the first m-tuple of coordinates of $\mathsf{E}_{b\sim\mathcal{B}}[r(b)]$ is $6h$-close to the binary encoding $E(v)$ of a vertex v, then*

1. *The decoding of the action profile of the v-type players is v with probability $1-o(\epsilon)$.*
2. *The decoding of the action profile of the β^v-type players is $(\beta_v^T,\beta_v^S,\beta_v^P)$ with probability $1-o(\epsilon)$.*

Proof. Whenever (3.11) holds, $D_v(r(a)) = v$. In particular, for each $i \in [m']$, $[E_u(D_v(r(a)))]_i = [E_u(v)]_i$ with high probability. Therefore, by playing the action $[E_u(v)]_i$ player v_i has expected utility of $1-o(1)$ whereas by playing the action $1-[E_u(v)]_i$ his expected utility is $o(1)$.

Every player that is $\overline{\epsilon}$-best replying, assigns probability of at least $1-O(\overline{\epsilon})$ to the correct bit. In addition, we have at most $8\overline{\epsilon}$ fraction of v-type players who are not $\overline{\epsilon}$-best replying (because we have 8 types of players of equal cardinality). Therefore the expected fraction of v-type players who play the wrong bit is $O(\overline{\epsilon})$. By Chernoff bound, it also holds that with high probability at most an $O(\overline{\epsilon})$-fraction of v-type players play the wrong bit. Whenever this is the case, v is indeed decoded correctly.

Similarly we prove the second claim in the lemma for β^v-type players. □

In a similar way we can show that analogues of Lemmas 3.2.8, 3.2.9, 3.2.10, and 3.2.11 hold for the n-player game. In particular,

Lemma 3.2.15. *In every $(\overline{\epsilon},\overline{\epsilon})$-weak approximate equilibrium $(\mathcal{A},\mathcal{B})$, $f_{I_v^A,I_w^A}(\mathbf{x}) = f(\mathbf{x})$ with high probability.*

Now we get to the analogue of the last Corollary 3.2.12.

Corollary 3.2.16. *In every $(\overline{\epsilon},\overline{\epsilon})$-weak approximate equilibrium $(\mathcal{A},\mathcal{B})$, the expectation of the realized point $\mathsf{E}_{a\sim\mathcal{A}}[r(a)]$ is a δ-approximate equilibrium of f; i.e.,*

$$\|\mathsf{E}_{a\sim\mathcal{A}}[r(a)] - f(\mathsf{E}_{a\sim\mathcal{A}}[r(a)])\|_2 \le \delta.$$

Proof. The proof is similar to the proof of Corollary 3.2.12. We recall that in Lemma 3.2.13 we have proved that

$$\|r(a) - \mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2 \le \epsilon^2 \tag{3.12}$$

with high probability. This, in particular, implies that $r(a)$ is, with high probability, close to its expectation:

$$\begin{aligned}
&\|r(a) - \mathsf{E}_{a'\sim\mathcal{A}}[r(a')]\|_2^2 && (3.13)\\
&\le 2\|r(a) - \mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2 + 2\|\mathsf{E}_{a\sim\mathcal{A}}[r(a)] - \mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2\\
&\le 2\|r(a) - \mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2 + 2\mathsf{E}_{a'\sim\mathcal{A}}\left[\|r(a') - \mathsf{E}_{b\sim\mathcal{B}}[r(b)]\|_2^2\right]\\
&= O(\epsilon^2), && (3.14)
\end{aligned}$$

with high probability. Where the first inequality follows from Triangle inequality, second follows from convexity, and the last is Lemma 3.2.13.

Using the $O(1)$-Lipschitzness of f we deduce that

$$\|f(r(a)) - f(\mathsf{E}_{a'\sim\mathcal{A}}[r(a')])\|_2^2 = O(\epsilon^2) \tag{3.15}$$

with high probability.

Using similar arguments to those of Lemma 3.2.13 we can show that

$$\left\|r(b) - \mathsf{E}_{a'\sim\mathcal{A}}[f_{\overline{I^v},\overline{I^w}}(r(a'))]\right\|_2^2 = O(\epsilon^2) \tag{3.16}$$

with high probability, where we recall that $\overline{I^v},\overline{I^w}$ denote the decoded line information of the action profile played by the I^v, I^w-types players. By an analogous argument to (3.13),

$$\|r(b) - \mathsf{E}_{b'\sim\mathcal{B}}[r(b')]\|_2^2 = O(\epsilon^2) \tag{3.17}$$

with high probability.

By Lemma 3.2.15,

$$\left\|\mathsf{E}_{a'\sim\mathcal{A}}[f_{\overline{I^v},\overline{I^w}}(r(a'))] - \mathsf{E}_{a'\sim\mathcal{A}}[f(r(a'))]\right\|_2^2 = O(\epsilon^2). \tag{3.18}$$

By Equations (3.13),(3.12),(3.17),(3.16),(3.18),(3.15) (applied exactly in this order) and the triangle inequality we get

$$\|\mathsf{E}_{a\sim\mathcal{A}}[r(a')] - f(\mathsf{E}_{a\sim\mathcal{A}}[r(a')])\|_2^2 = O(\epsilon^2) < \delta^2. \tag{3.19}$$

□

Proof of Theorem 3.0.2. Any communication protocol that solves the $(\epsilon^5/82, \epsilon^5/82)$-weak approximate Nash equilibrium problem in $\Theta(n)$-player games with binary actions induces a communication protocol for the problem Simulation End-of-the-Line: Alice constructs the utilities of her players using her private information of the αs, Bob constructs his utility using the βs. They implement the communication protocol to find an $(\epsilon^5/82, \epsilon^5/82)$-weak approximate Nash equilibrium, and then both of them know $\mathsf{E}_{a\sim\mathcal{A}}[r(a)]$ which is a δ-approximate fixed point of f (by Corollary 3.2.16). Finally, they round and decode the approximate fixed point to reocever the end of the line.

Using Corollary 3.2.4 we deduce that the communication complexity of $(\epsilon^5/82, \epsilon^5/82)$-weak approximate Nash equilibrium problem in $\Theta(n)$-player games with binary actions is at least $2^{\Omega(n)}$. □

3.3 An open problem: correlated equilibria in 2-player games

As mentioned in Section 3.0.3, it is known that for n-player, $O(1)$-action games, even *exact* correlated equilibrium can be found with $\mathsf{poly}(n)$ deterministic communication complexity (see [HM10; PR08; JLB15]).

In two-player $N \times N$ games, for approximate correlated equilibrium with constant value of approximation, to the best of our knowledge, no non-trivial results are known (neither positive nor negative). Does a $\mathsf{polylog}(N)$ communication protocol for approximate correlated equilibrium exist? Is there a $\mathsf{poly}(N)$ communication lower bound? For small values of approximation, recently [GS17] have shown that $1/N$-correlated equilibrium requires $\mathsf{poly}(N)$ communication.

Chapter 4

Brouwer's fixed point

Brouwer's fixed point theorem guarantees that any continuous functions $f : [0,1]^n \to [0,1]^n$ has a fixed point, i.e. a point $\mathbf{x} \in [0,1]^n$ such that $f(\mathbf{x}) = \mathbf{x}$. In this section we construct continuous functions $f : [0,1]^n \to [0,1]^n$ where even finding an approximate fixed point (i.e. $\mathbf{x}$ such that $\mathbf{x} \approx f(\mathbf{x})$) is hard.

In particular, we reduce the End-of-a-Line problem to the problem of finding an approximate fixed point. Our reductions require only local, computationally efficient, black-box access to the End-of-a-Line instance. Thus they immediately apply both to the computational setting (defined in Section 2.2) and the query-complexity model (which is the focus of Chapter 3).

We present two reductions. We begin with an easier construction, where we think of the error $\mathbf{x} - f(\mathbf{x})$ in terms of ℓ_∞-norm; it is a good starting point for introducing the main ideas, and the special structure of the construction is also useful for the results in Chapter 6. We then add error correcting codes to obtain a stronger construction for ℓ_2-norm.

4.1 Brouwer with ℓ_∞

Below we state and prove the hardness of finding an approximate fixed point in ℓ_∞-norm. For the results in Chapter 6 we require a stronger characterization of the construction, which we omit from the theorem statement for simplicity (see Fact 4.1.3 for details).

Theorem 4.1.1 (ℓ_∞ Brouwer)**.** *There exists a constant $\epsilon > 0$ such that the following holds. Given (local, black-box access to) an instance of* End-of-a-Line*, we can construct a computationally-efficient, continuous function $f : [0,1]^{O(n)} \to [0,1]^{O(n)}$, such that:*

- *f is $O(1)$-Lipschitz in ℓ_∞-norm; and*
- *given $\mathbf{x}$ for which $\|f(\mathbf{x}) - \mathbf{x}\|_\infty < \epsilon$, we can efficiently reconstruct a solution to the* End-of-a-Line *instance.*

Proof. In the first step (Subsection 4.1.2), we embed the End-of-a-Line problem (over $\{0,1\}^n$) as a collection H of vertex-disjoint paths over the $(2n+1)$-dimensional hypercube graph. Given H, our second step (Subsection 4.1.3) is to construct a continuous mapping $f:[0,1]^{2n+2} \to [0,1]^{2n+2}$ whose fixed points correspond to ends of paths in H. This step generalizes a construction of Hirsch et al [HPV89] for embedding a single path. □

4.1.1 Preliminaries: ℓ_∞-norm Geometry

Throughout this section, we work with the ℓ_∞-norm. This has some implications that may contradict our geometric intuition. For example: in a ℓ_∞-norm world, a circle is a square.

ℓ_∞-norm interpolation Given coordinates $x, y \geq 0$, we define the *ℓ_∞-norm angle*[1] that point (x,y) forms with the X-axis (in the XY-plane) as

$$\theta_\infty(x,y) = \frac{y}{x+y}$$

The ℓ_∞-norm angle is useful for interpolation. Given the values of $f:[0,1]^n \to [0,1]^n$ on two neighboring facets of the hypercube, we can extend f to all points of the hypercube by *angular interpolation*: interpolate according to the ℓ_∞-norm angle $\theta_\infty(x_i, x_j)$ where x_i and x_j are the respective distances from the two facets. When f is defined on two opposite facets, we can simply use *Cartesian interpolation*, which again means to interpolate according to the distance from each facet.

ℓ_∞-norm local polar coordinates Given a point $\mathbf{z} \in \mathbb{R}^n$ we define a new *local ℓ_∞-norm polar coordinate* system around $\mathbf{z}$. Every $\mathbf{x} \in \mathbb{R}^n$ is transformed into $\langle r, \mathbf{p} \rangle_{\mathbf{z}} \in \mathbb{R} \times \mathbb{R}^n$ where $r = \|\mathbf{x} - \mathbf{z}\|$ is the ℓ_∞-norm radius, and $\mathbf{p} = (\mathbf{x} - \mathbf{z})/r$ is the ℓ_∞-norm unit vector that points from $\mathbf{z}$ in the direction of $\mathbf{x}$.

[1] Our ℓ_∞-norm angle was called *unit* in [HPV89].

4.1.2 Embedding the END-OF-A-LINE graph as paths in $\{0,1\}^{2n+1}$

Our first step in the reduction is to embed an END-OF-A-LINE graph $G_{S,P}$ as vertex-disjoint paths on the $(2n+1)$-dimensional hypercube graph. We construct a collection H of vertex-disjoint paths and cycles over the $(2n+1)$-dimensional hypercube graph, such that there is a 1-to-1 correspondence between starting and end points of paths in H and starting and end points of lines in $G_{S,P}$.

In order to construct our embedding we divide the $2n+1$ coordinates as follows: the first n coordinates store the current vertex $\mathbf{u}$, the next n coordinates for the next vertex in the line, $\mathbf{v}$, and finally, the last coordinate b stores a compute-next vs copy bit. When $b=0$, the path proceeds to update $\mathbf{v} \leftarrow S(\mathbf{u})$, bit-by-bit. When this update is complete, the value of b is changed to 1. Whenever $b=1$, the path proceeds by copying $\mathbf{u} \leftarrow \mathbf{v}$ bit-by-bit, and then changes that value of b again. Finally, when $\mathbf{u}=\mathbf{v}=S(\mathbf{u})$ and $b=0$, the path reaches an end point. For example, the edge $\mathbf{u} \to \mathbf{v}$ maps into the path:

$$(\mathbf{u},\mathbf{u},0) \to \cdots \to (\mathbf{u},\mathbf{v},0) \to (\mathbf{u},\mathbf{v},1) \to \cdots \to (\mathbf{v},\mathbf{v},1) \to (\mathbf{v},\mathbf{v},0)\,.$$

Notice that the paths in H do not intersect. Furthermore, given a vector in $\mathbf{p} \in \{0,1\}^{2n+1}$, we can efficiently output whether $\mathbf{p}$ belongs to a path in H, and if so which are the previous and consecutive vectors in the path. Finding a starting or end point of any path in H (other than $\mathbf{0}_{2n+1}$) is therefore equivalent to finding an odd-degree vertex (other than $\mathbf{0}_n$) in $G_{S,P}$.

4.1.3 Continuous mapping on $[0,1]^{2n+2}$

In order to construct a hard instance of Brouwer function, we use techniques introduced by Hirsch, Papadimitriou, and Vavasis [HPV89]. The continuous Brouwer function is denoted by $f\colon [0,1]^{2n+2} \to [0,1]^{2n+2}$, while the associated displacement function is denoted by $g(\mathbf{x}) \triangleq f(\mathbf{x}) - \mathbf{x}$. The following lemma (proven below) completes the proof of Theorem 4.1.1.

Lemma 4.1.2. *The displacement g satisfies:*

1. *g is $O(1)$-Lipschitz (thus, f is also $O(1)$-Lipschitz)*
2. *$\|g(\mathbf{x})\|_\infty = \Omega(1)$ for every $\mathbf{x}$ that does not correspond to a starting or end point in H.*
3. *The value of g at each point $\mathbf{x}$ can be computed (efficiently) with local, black-box access to S, P.*

4.1.3.1 Overview of the construction

The domain of f is the $2n+2$-dimensional (solid) hypercube. The hypercube is divided into subcubes, of side length h (we fix $h = 1/4$). We define f separately on each subcube such that it agrees on the intersections (no interpolation is needed in this sense).

The last ($(2n+2)$-th) dimension is special; we use "up" (resp. "down") to refer to the positive (negative) $(2n+2)$-th direction. All the action takes place in the second-from-bottom $(2n+1)$-dimensional layer of subcubes; this layer is called the *slice*. Within the slice, we also ignore the subcubes that are near the boundary of the hypercube (those compose the *frame*); we are left with the subcubes in the center of the slice, which we call the *picture*. We identify between the vertices of the $(2n+1)$-dimensional hypercube graph (over which H was defined) and the 2^{2n+1} subcubes of the picture.

The subset of subcubes into which we embed each path or cycle from H is called a *tube*. The *home subcube*, the subcube that corresponds to the $\mathbf{0}_{2n+1}$-vertex, is special: all the flow from all subcubes that do not belong to any tube leads to this subcube.

Below we define the displacement in the following regions, and argue that it satisfies the desiderata of Lemma 4.1.2:

- Inside the picture, but not in any tube;
- inside a tube; and
- outside the picture.

4.1.3.2 Default displacement

Most of the slice has the same *default displacement*: directly upward, i.e. $g(\mathbf{x}) = \delta\boldsymbol{\xi}_{2n+2}$, where $\boldsymbol{\xi}_{2n+2}$ is the $(2n+2)$-unit vector, and $\delta > 0$ is a small constant. Formally,

Fact 4.1.3. $g(\mathbf{x}) = \delta\boldsymbol{\xi}_{2n+2}$, for every $\mathbf{x}$ such that at least one of the following holds:

1. $\mathbf{x}$ lies on a corner, i.e. the intersection of two or more facets of a subcube;
2. $\mathbf{x}$ lies on an outer facet of a tube subcube, i.e. a facet other than the two facets that continue the path; or
3. $\mathbf{x}$ lies in a subcube that does not belong to any tube.

Intuitively, Property 2 implies that *all subcubes -whether they belong to the tube or not- look the same from the outside* (except for the two facets that continue the path). In particular, the displacement on both sides of each facet is the same; so if

the displacement is $O(1)$-Lipschitz on each subcube, it is also $O(1)$-Lipschitz on the entire hypercube.

Property 1, stating that *all corners look the same*, is key to the sampling gadgets in Chapter 6, because it liberates us from having to disambiguate the position of a point near the corners (that is, deciding exactly to which subcube it belongs).

4.1.3.3 Displacement at a tube

The mapping is defined so that in the center of the tube, the flow goes along the direction of the path; slightly outside the center, the flow points towards the center of the tube; further away from the center, the flow goes *against* the direction of the path; at the outer boundary of the tube, as we previously described, the flows goes upwards.

We first define g on facets. Let $\langle r, \mathbf{p} \rangle_{\mathbf{z}}$ be a point on the facet centered at $\mathbf{z}$, and suppose that the tube enters the subcube through $\mathbf{z}$, advancing in the positive i-th coordinate. We define

$$g\left(\langle r, \mathbf{p} \rangle_{\mathbf{z}}\right) \triangleq \begin{cases} \delta \boldsymbol{\xi}_i & r = 0 \\ -\delta \mathbf{p} & r = h/8 \\ -\delta \boldsymbol{\xi}_i & r = h/4 \\ \delta \boldsymbol{\xi}_{2n+2} & r = h/2 \end{cases} \tag{4.1}$$

(Recall that h is the subcube side length, and δ is some small constant.) Notice that at each r, the displacement g is $O(1)$-Lipschitz and has magnitude $\|g(\mathbf{x})\|_\infty = \Omega(1)$ (thus satisfying the first two desiderata of Lemma 4.1.2).

For $r \in (0, h/8)$, interpolate between $\delta \boldsymbol{\xi}_i$ and $-\delta \mathbf{p}$ ([HPV89] call this *radial interpolation*), and similarly for $r \in (h/8, h/4)$ and $r \in (h/4, h/2)$. See also illustration in Figure 4.1. It is easy to see that the $O(1)$-Lipschitz property is preserved. Notice also that $\boldsymbol{\xi}_i$ is orthogonal to $\mathbf{p}$ and $\boldsymbol{\xi}_{2n+2}$; this guarantees that the interpolation does not lead to cancellation, i.e. we still have $\|g(\mathbf{x})\|_\infty = \Omega(1)$.

In the last couple of paragraphs we defined g on two facets for each subcube that belongs to the tubes; for all other points in the tubes we interpolate (angular interpolation) between those two facets: Consider a point $\mathbf{x}$ in the tube, and assume (w.l.o.g.) that $x_i, x_j > 1/2$, and suppose that the value of $f(\cdot)$ on the $y_i = 1/2$ and $y_j = 1/2$ facets of the subcube containing $\mathbf{x}$ is determined by (4.1). Let

$$\begin{aligned} \mathbf{x}^i &= \left(\mathbf{x}_{-i,j}, \frac{1}{2}, \max\{x_i, x_j\}\right) \\ \mathbf{x}^j &= \left(\mathbf{x}_{-i,j}, \max\{x_i, x_j\}, \frac{1}{2}\right) \end{aligned}$$

Figure 4.1: A facet of the Hirsch et al construction

An illustration of the displacement on a facet between two subcubes in a tube; the direction of the path is into the paper. In the center, the displacement points into the paper; slightly further outside, the displacement points towards the center; further outside, the displacement points out of the paper; finally in the outer layer, the displacements points in the special $2n+2$ dimension.

denote the corresponding "ℓ_∞-norm projections" to the respective $y_i = 1/2$ and $y_j = 1/2$ facets. We set

$$g(\mathbf{x}) \;=\; \theta_\infty\left(x_i - \frac{1}{2}, x_j - \frac{1}{2}\right) g\left(\mathbf{x}^i\right) + \left(1 - \theta_\infty\left(x_i - \frac{1}{2}, x_j - \frac{1}{2}\right)\right) g\left(\mathbf{x}^j\right).$$

Notice that $\mathbf{x}^i$ and $\mathbf{x}^j$ are at the same distance from the respective facet centers, i.e. they have they correspond to the same r. For each case of (4.1), the (i,j)-components of the displacements at $\mathbf{x}^i$ and $\mathbf{x}^j$ are orthogonal, and for the rest of the components they are aligned. Therefore, when we interpolate between $g(\mathbf{x}^i)$ and $g(\mathbf{x}^j)$ there is again no cancellation, i.e. $\|g(\mathbf{x})\|_\infty = \Omega\left(\|g(\mathbf{x}^i)\|_\infty\right) = \Omega(1)$. Finally, recall that the displacement on each facet is $O(1)$-Lipschitz, and the displacements agree on the intersection of the facets. Therefore the interpolated displacement is $O(1)$-Lipschitz over the entire subcube by a triangle-inequality argument.

The home subcube is defined using (4.1) as if the tube enters from above, i.e. coming down the $(2n+2)$-dimension, and exits through another facet (in one of the first $(2n+1)$ dimensions) in the direction of the path (here again we have $\|g(\mathbf{x})\|_\infty = \Omega(1)$). For all other starting and end points, we define $g(\mathbf{x}) = \delta\boldsymbol{\xi}_{2n+2}$ on the facet opposite the one that continues the tube, and interpolate between the opposite facets using Cartesian interpolation. Notice that this gives a fixed point when the interpolation cancels the default displacement at the opposite facet, with the displacement $-\delta\boldsymbol{\xi}_{2n+2}$ at the point on the tube facet which is at distance $h/8$ above the path.

4.1.3.4 Outside the picture

For all points in the frame and below the slice, the displacement points directly upward, i.e. $g(\mathbf{x}) = \delta\boldsymbol{\xi}_{2n+2}$. Moving above the slice, let $\mathbf{z}[\text{top}]$ be the point on the top facet of the hypercube which is directly above the center of the home subcube. For all points $\langle r, \mathbf{p}\rangle_{\mathbf{z}[\text{top}]}$ on the top facet of the hypercube, define the displacement

Figure 4.2: Outside the picture

An illustration of the displacement outside the picture.

as follows:

$$g\left(\langle r, \mathbf{p}\rangle_{\mathbf{z}}[\text{top}]\right) = \begin{cases} -\delta\boldsymbol{\xi}_{2n+2} & r = 0 \\ -\delta\mathbf{p} & r \geq h/8 \end{cases}$$

and interpolate for $r \in (0, h/8)$. Notice that this displacement is $O(1)$-Lipschitz and has $\Omega(1)$ magnitude for each r, and this is preserved after interpolation.

Notice that the definition of g on the slice from the previous subsection, implies that all the points in the top facet of the slice, except for the top of the home subcube, point directly upwards. Let $\mathbf{z}[\text{home}]$ denote the center of the top facet of the home subcube. We therefore have that for any $\langle r, \mathbf{p}\rangle_{\mathbf{z}}[\text{home}]$ in the top facet of the slice,

$$g\left(\langle r, \mathbf{p}\rangle_{\mathbf{z}}[\text{home}]\right) = \begin{cases} -\delta\boldsymbol{\xi}_{2n+2} & r = 0 \\ -\delta\mathbf{p} & r = h/8 \\ \delta\boldsymbol{\xi}_{2n+2} & r \geq h/4 \end{cases}$$

where we again interpolate radially for r in $(0, h/8)$ and $(h/8, h/4)$.

Finally, to complete the construction above the slice, simply interpolate (using Cartesian interpolation) between the top facets of the slice and the hypercube. See also illustration in Figure 4.2.

4.2 Euclidean Brouwer

Theorem 4.2.1 (Euclidean Brouwer)**.** *There exist constants $\delta, h > 0$ such that the following holds. Given (local, black-box access to) an instance $I = (T, S, P)$ of* Membership End-of-a-Line*, we can construct a computationally-efficient, continuous function $f : [0,1]^{O(n)} \to [0,1]^{O(n)}$, such that*

- *f is $O(1)$-Lipschitz in ℓ_2-norm; and*
- *given $\mathbf{x}$ for which $\|f(\mathbf{x}) - \mathbf{x}\|_2 < \epsilon$, we can efficiently reconstruct a solution to the* Membership End-of-a-Line *instance.*

4.2.1 Discrete embedding of a graph in the Euclidean space

Each vertex v is embedded to the point $(E(v), E(v), \mathbf{0}_m, \mathbf{0}_m) \in [-1,2]^{4m}$, which is called *the embedded vertex*.

For every edge (v,w)we define five vertices:

$$\begin{aligned}
\mathbf{x}^1(v,w) &\triangleq (E(v), E(v), \mathbf{0}_m, \mathbf{0}_m) \\
\mathbf{x}^2(v,w) &\triangleq (E(v), E(v), \mathbf{1}_m, \mathbf{0}_m) \\
\mathbf{x}^3(v,w) &\triangleq (E(v), E(w), \mathbf{1}_m, \mathbf{0}_m) \\
\mathbf{x}^4(v,w) &\triangleq (E(v), E(w), \mathbf{0}_m, \mathbf{0}_m) \\
\mathbf{x}^5(v,w) &\triangleq (E(w), E(w), \mathbf{0}_m, \mathbf{0}_m)\,.
\end{aligned}$$

The vertices $\mathbf{x}^i(v,w)$ are called *Brouwer vertices*. Note that $\mathbf{x}^1(v,w)$ is the embedded vertex v, $\mathbf{x}^5(v,w)$ is the embedded vertex w. The line that connects the points $\mathbf{x}^i(v,w)$ and $\mathbf{x}^{i+1}(v,w)$ is called a *Brouwer line segment*. The union of these four Brouwer line segments is called the *embedded edge* (v,w).

4.2.2 The function f

We set h to be a sufficiently small constant such that the $\sqrt{h}$ neighborhood of any two Brouwer vertices will not intersect and such that the $3h$ neighborhood of any two Brouwer line segments will not intersect- unless they share the same common Brouwer vertex. We take δ to be a constant arbitrarily smaller than h ($\delta = h^3$ suffices). We define a displacement function $g : [-1,2]^{4m} \to [-\delta,\delta]^{4m}$ and $f(\mathbf{x}) \triangleq \mathbf{x} + g(\mathbf{x})$. In order that Properties (1)-(3) of Proposition 3.2.5 will be satisfied, we should define g such that:

1. $\|g(\mathbf{x})\|_2 = \Omega(\delta)$ for every x that is not $2\sqrt{h}$-close to the Brouwer line segments of any non-trivial end or starting of a line.
2. g is $O(1)$-Lipschitz.
3. g is defined "locally", which will allow as to generate the class of functions $\{f_{I_1,I_2}\}$.

We think of the $4m$ coordinates as partitioned into four parts: the first m-tuple of coordinates represent the current vertex in the line; the second m-tuple represent the next vertex in the line; we think of the third m-tuple as all being equal to a single bit that monitors helps altering between computing the next vertex, and copying

from the second to first m-tuple. Finally, the last m coordinates represent a special default direction in which the displacement points when far from all Brouwer line segments (similarly to the single special coordinate in [HPV89]).

We consider a path starting at $(\mathbf{0}_{3m}, 2\cdot\mathbf{1}_m)$, i.e. the concatenation of 0 on the first $3m$ coordinates, and 2 on the last m coordinates. The path first goes to $(\mathbf{0}_{4m})$ (in a straight line), and thereafter the last m coordinates remain constantly 0 (note that every Brouwer vertex has $\mathbf{0}_m$ in its last m-tuple). The first $3m$ coordinates follow the line according to the embedding in Section 3.2.3. This path corresponds to the line starting at $\mathbf{0}_{2n+1}$; for any additional line starting at vertex u, we have another path starting at $(E(u), E(u), \mathbf{0}_{2m})$.

We say that a point $\mathbf{x}$ is in the *picture* if $\frac{1}{m}\sum_{i=3m+1}^{4m} x_i < 1/2$. We construct g separately inside and outside the picture (and make sure that the construction agrees on the hyperplane $\frac{1}{m}\sum_{i=3m+1}^{4m} x_i = 1/2$).

Truncation In order for $g(\cdot)$ to be a displacement function, we must ensure that it never sends any points outside the hypercube, i.e. $\forall \mathbf{x} \in [-1,2]^{4m}$, we require that also $\mathbf{x} + g(\mathbf{x}) \in [-1,2]^{4m}$. Below, it is convenient to first define an *untruncated* displacement function $\hat{g} : [-1,2]^{4m} \to [-\delta,\delta]^{4m}$ which is not restricted by the above condition. We then truncate each coordinate to fit in $[-1,2]$: $[g(\mathbf{x})]_i = \max\{-1, \min\{2, x_i + [\hat{g}(\mathbf{x})]_i\}\} - x_i$. It is clear that if $\hat{g}(\cdot)$ is $(M-1)$-Lipschitz, then $g(\cdot)$ is M-Lipschitz. It is, however, important to make sure that the magnitude of the displacement is not compromised. Typically, some of the coordinates may need to be truncated, but we design the displacement so that most coordinates, say 99%, are not truncated. If $\hat{g}(\mathbf{x})$ has a non-negligible component in at least 5% of the coordinates, then in total $g(\mathbf{x})$ maintains a non-negligible magnitude.

4.2.2.1 Inside the picture

The line $\mathbf{0} = v_0, v_1, \ldots, v^*$ is embedded to a path in $[-1,2]^{4m}$ that goes in straight lines through the following sequence of Brouwer vertices:

$$(\mathbf{0}_{3m}, 2\cdot\mathbf{1}_m), (\mathbf{0}_{4m}) = \mathbf{x}^1(v_0,v_1), \mathbf{x}^2(v_0,v_1), \ldots, \mathbf{x}^5(v_0,v_1) = \mathbf{x}^1(v_1,v_2), \mathbf{x}^2(v_1,v_2), \ldots, \mathbf{x}^5(P(v^*),v^*).$$

Similarly, if I contains another line $u, \ldots, w$, it is embedded as a path through:

$$(E(u), E(u), \mathbf{0}_{2m}) = \mathbf{x}^1(u, S(u)), \ldots, \mathbf{x}^5(P(w), w) = (E(w), E(w), \mathbf{0}_{2m}),$$

and analogously for cycles.

Now we *cut the corners* of this path as follows: For two consecutive Brouwer vertices $\mathbf{s}, \mathbf{y}$ in the embedded path we let $\mathbf{z}^1_{(\mathbf{s}\to\mathbf{y})}$ be the point in the Brouwer line

segment $[\mathbf{s},\mathbf{y}]$ that is exactly $\sqrt{h}$-far from $\mathbf{s}$. Similarly, $\mathbf{z}^2_{(\mathbf{s}\to\mathbf{y})}$ is the point in $[\mathbf{s},\mathbf{y}]$ that is exactly $\sqrt{h}$-far from $\mathbf{y}$. For three consecutive Brouwer vertices $\mathbf{s}\to\mathbf{y}\to\mathbf{t}$, the path after "cutting the corners" goes in straight lines through

$$\ldots,\mathbf{z}^1_{(\mathbf{s}\to\mathbf{y})},\mathbf{z}^2_{(\mathbf{s}\to\mathbf{y})},\mathbf{z}^1_{(\mathbf{y}\to\mathbf{t})},\mathbf{z}^2_{(\mathbf{y}\to\mathbf{t})},\ldots$$

instead of going through $\mathbf{s}\to\mathbf{y}\to\mathbf{t}$.

First, for all points inside the picture that are $3h$-far from the embedded path after cutting the corners we use the same *default displacement* which points in the positive special direction: $\hat{g}(\mathbf{x}) = (\mathbf{0}_{3m},\delta\cdot\mathbf{1}_m)$. Because $\mathbf{x}$ is inside the picture, the truncated displacement $g(\mathbf{x})$ is close to $\hat{g}(\mathbf{x})$, and therefore satisfies $\|g(\mathbf{x})\|_2 = \Omega(\delta)$.

Now we define the displacement $3h$-close to the embedded path in two regions:

1. For points that are $3h$-close to a segment of the form $[\mathbf{z}^1_{(\mathbf{s}\to\mathbf{y})},\mathbf{z}^2_{(\mathbf{s}\to\mathbf{y})}]$ but (approximately[2]) $\sqrt{h}$-far from both Brouwer vertices $\mathbf{s},\mathbf{y}$.
2. For the remaining points, those that are $3h$-close to a segment of the form $[\mathbf{z}^2_{(\mathbf{s}\to\mathbf{y})},\mathbf{z}^1_{(\mathbf{y}\to\mathbf{t})}]$ and (approximately[2]) $\sqrt{h}$-close to the Brouwer vertex $\mathbf{y}$.

We make sure that the definitions agree on the interface between the two regions, as well as on the interface with the points that receive the default displacement.

4.2.2.2 Close to the path but far from a Brouwer vertex

On the Brouwer line segment, the displacement points in the direction of the path; at distance h from the Brouwer line segment, the displacement points in towards the Brouwer line segment; at distance $2h$ from the Brouwer line segment, the displacement points against the direction of the path; at distance $3h$, the displacement points in the default direction.

Formally, let $\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x})$ denote the magnitude of the component of $\mathbf{x}-\mathbf{s}$ in the direction of line $(\mathbf{s}\to\mathbf{t})$,

$$\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x}) \triangleq \frac{(\mathbf{t}-\mathbf{s})}{\|\mathbf{s}-\mathbf{t}\|_2}\cdot(\mathbf{x}-\mathbf{s}),$$

where $\cdot$ denotes the (in-expectation) dot product. Let $\mathbf{z} = \mathbf{z}(\mathbf{x})$ be the point nearest to $\mathbf{x}$ on the Brouwer line segment; notice that $\mathbf{z}$ satisfies

$$\mathbf{z} = \sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x})\,\mathbf{t} + \big(1-\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x})\big)\,\mathbf{s}.$$

[2] It will be more convenient to set the threshold of points $\mathbf{x}$ that are "far"/"close" from/to a Brouwer vertex using the expression $\sigma_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x})$ that is defined below. $\sigma_{(\mathbf{s}\to\mathbf{y})}(x)$ is closely related to the distance of $\mathbf{x}$ from the points $\mathbf{s},y$ but is not precisely the distance.

For points near the Brouwer line segment ($\|\mathbf{x}-\mathbf{z}\|_2 \le 3h$), but far from its endpoints ($\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x}) \in \left[\sqrt{h}, 1-\sqrt{h}\right]$), we define the displacement:

$$\hat{g}(\mathbf{x}) \triangleq \begin{cases} \delta\frac{(\mathbf{t}-\mathbf{s})}{\|\mathbf{t}-\mathbf{s}\|_2} & \|\mathbf{x}-\mathbf{z}\|_2 = 0 \\ \delta\frac{(\mathbf{z}-\mathbf{x})}{h} & \|\mathbf{x}-\mathbf{z}\|_2 = h \\ \delta\frac{(\mathbf{s}-\mathbf{t})}{\|\mathbf{t}-\mathbf{s}\|_2} & \|\mathbf{x}-\mathbf{z}\|_2 = 2h \\ \delta(\mathbf{0}_{3m}, \mathbf{1}_m) & \|\mathbf{x}-\mathbf{z}\|_2 = 3h \end{cases} \tag{4.2}$$

At intermediate distances from the Brouwer line segment, we interpolate: at distance $\|\mathbf{x}-\mathbf{z}\|_2 = \frac{1}{3}h$, for example, we have $\hat{g}(\mathbf{x}) = \frac{2}{3}\delta\frac{(\mathbf{t}-\mathbf{s})}{\|\mathbf{t}-\mathbf{s}\|_2} + \frac{1}{3}\delta\frac{(\mathbf{z}-\mathbf{x})}{h}$. Notice that every two of $(\mathbf{t}-\mathbf{s})$, $(\mathbf{z}-\mathbf{x})$, and $(\mathbf{0}_{3m}, \mathbf{1}_m)$ are orthogonal, so the interpolation does not lead to cancellation. Also, every point $\mathbf{z}$ on the Brouwer line segment is $\Omega(1)$-far in every coordinate from $\{-1, 2\}$, so the truncated displacement $g(\mathbf{x})$ still satisfies $\|g(\mathbf{x})\|_2 = \Omega(\delta)$. For each case in (4.2), $\hat{g}(\cdot)$ is either constant, or (in the case of $\|\mathbf{x}-\mathbf{z}\|_2 = h$) $O(\delta/h)$-Lipschitz ($\frac{(\mathbf{z}-\mathbf{x})}{h}$ is $O(1/h)$-Lipschitz because two "antipodal" points at distance $2h$ have opposite directions, both pointing parallel to the Brouwer line segment); by choice of $\delta \ll h$, it follows that $\hat{g}(\cdot)$ is in particular $O(1)$-Lipschitz. Furthermore, notice that $\|\mathbf{x}-\mathbf{z}\|_2$ is 1-Lipschitz, so after interpolating for intermediate distances, $\hat{g}(\cdot)$ continues to be $O(1)$-Lipschitz. Notice also that at distance $3h$ the displacement defined in (4.2) agrees with the displacements for points far from every Brouwer line segment, so Lipschitz continuity is preserved.

4.2.2.3 Close to the path and a Brouwer vertex

Let $L_{\mathbf{y}}$ be the line that connects the points $\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}$ and $\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}$. Given $\mathbf{x}$, we let $\mathbf{z}$ be the closest point to $\mathbf{x}$ on $L_{\mathbf{y}}$.

Our goal is to interpolate between the line displacement for $(\mathbf{s}\to\mathbf{y})$ (which is defined up to $\sigma_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}) = 1-\sqrt{h}$), and the line displacement for $(\mathbf{y}\to\mathbf{t})$ (which begins at $\sigma_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) = \sqrt{h}$). Let $\Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}) \triangleq \sigma_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}) - \left(1-\sqrt{h}\right)$, and $\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) \triangleq \sqrt{h} - \sigma_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x})$. We set our interpolation parameter $\tau = \tau(\mathbf{x}) \triangleq \frac{\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x})}{\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x})+\Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x})}$, and set

$$\mathbf{z} \triangleq \tau\mathbf{z}_{(\mathbf{s}\to\mathbf{y})} + (1-\tau)\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}. \tag{4.3}$$

For points $\mathbf{x}$ near $\mathbf{y}$ such that $\Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}), \Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) \ge 0$, we can now define the

displacement analogously to (4.2):

$$\hat{g}(\mathbf{x}) \triangleq \begin{cases} \delta \cdot \left[\tau \frac{(\mathbf{y}-\mathbf{s})}{\|\mathbf{y}-\mathbf{s}\|_2} + (1-\tau) \frac{(\mathbf{t}-\mathbf{y})}{\|\mathbf{t}-\mathbf{y}\|_2} \right] & \|\mathbf{x}-\mathbf{z}\|_2 = 0 \\ \delta \frac{(\mathbf{z}-\mathbf{x})}{h} & \|\mathbf{x}-\mathbf{z}\|_2 = h \\ \delta \cdot \left[\tau \frac{(\mathbf{s}-\mathbf{y})}{\|\mathbf{y}-\mathbf{s}\|_2} + (1-\tau) \frac{(\mathbf{y}-\mathbf{t})}{\|\mathbf{t}-\mathbf{y}\|_2} \right] & \|\mathbf{x}-\mathbf{z}\|_2 = 2h \\ \delta(\mathbf{0}_{3m}, \mathbf{1}_m) & \|\mathbf{x}-\mathbf{z}\|_2 \geq 3h \end{cases}. \tag{4.4}$$

At intermediate distances, interpolate according to $\|\mathbf{x}-\mathbf{z}\|_2$. Notice that for each fixed choice of $\tau \in [0,1]$ (and $\mathbf{z}$), $\hat{g}$ is $O(\delta/h) = O(1)$-Lipschitz. Furthermore, $\Delta_{(\mathbf{s}\to\mathbf{y})}$ and $\Delta_{(\mathbf{y}\to\mathbf{t})}$ are 1-Lipschitz in $\mathbf{x}$. For any $\mathbf{z} \in L_{\mathbf{y}}$, $\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{z}) + \Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{z}) = \sqrt{h}$. For general $\mathbf{x}$, we have

$$\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) + \Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}) \geq \Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{z}) + \Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{z}) - 2\|\mathbf{x}-\mathbf{z}\|_2 = \sqrt{h} - 2\|\mathbf{x}-\mathbf{z}\|_2; \tag{4.5}$$

so τ is $O\left(1/\sqrt{h}\right)$-Lipschitz whenever $\|\mathbf{x}-\mathbf{z}\|_2 < 3h$, and otherwise has no effect on $\hat{g}(\mathbf{x})$. We conclude that $\hat{g}$ is $O(1)$-Lipschitz when interpolating across different values of τ. At the interface with (4.2) τ is 1 (0 near $\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}$), so (4.2) and (4.4) are equal. Therefore $\hat{g}$ is $O(1)$-Lipschitz on all of $[-1,2]^{4m}$.

To lower bound the magnitude of the displacement, we argue that $(\mathbf{z}-\mathbf{x})$ is orthogonal to $\left[\tau \frac{(\mathbf{y}-\mathbf{s})}{\|\mathbf{y}-\mathbf{s}\|_2} + (1-\tau) \frac{(\mathbf{t}-\mathbf{y})}{\|\mathbf{t}-\mathbf{y}\|_2}\right]$. First, observe that we can restrict our attention to the component of $(\mathbf{z}-\mathbf{x})$ that belongs to the plane defined by $\mathbf{s},\mathbf{y},\mathbf{t}$ (in which $\mathbf{z}$ also lies). Let $P_{\mathbf{s},\mathbf{y},\mathbf{t}}(\mathbf{x})$ denote the projection of $\mathbf{x}$ to this plain. We can write points in this plane in terms of their $\Delta(\cdot) \triangleq \left(\Delta_{(\mathbf{s}\to\mathbf{y})}(\cdot), \Delta_{(\mathbf{y}\to\mathbf{t})}(\cdot)\right)$ values. (Recall that $(\mathbf{s}\to\mathbf{y})$ and $(\mathbf{y}\to\mathbf{t})$ are orthogonal.)

First, observe that $\Delta\left(\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}\right) = \left(0, \sqrt{h}\right)$, $\Delta\left(\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}\right) = \left(\sqrt{h}, 0\right)$ and $\Delta(\mathbf{y}) = \left(\sqrt{h}, \sqrt{h}\right)$. Notice also that

$$\left[\tau \frac{(\mathbf{y}-\mathbf{s})}{\|\mathbf{y}-\mathbf{s}\|_2} + (1-\tau) \frac{(\mathbf{t}-\mathbf{y})}{\|\mathbf{t}-\mathbf{y}\|_2}\right] = \left[\tau \frac{\left(\mathbf{y}-\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}\right)}{\sqrt{h}} + (1-\tau) \frac{\left(\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}-\mathbf{y}\right)}{\sqrt{h}}\right].$$

Putting those together, we have that

$$\Delta\left(\left[\tau \frac{\mathbf{y}}{\|\mathbf{y}-\mathbf{s}\|_2} + (1-\tau) \frac{\mathbf{t}}{\|\mathbf{t}-\mathbf{y}\|_2}\right]\right) - \Delta\left(\left[\tau \frac{\mathbf{s}}{\|\mathbf{y}-\mathbf{s}\|_2} + (1-\tau) \frac{\mathbf{y}}{\|\mathbf{t}-\mathbf{y}\|_2}\right]\right) = (\tau, 1-\tau). \tag{4.6}$$

For $\mathbf{z}$, we have

$$\Delta(\mathbf{z}) = \tau\Delta\left(\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}\right) + (1-\tau)\Delta\left(\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}\right) = \sqrt{h}(1-\tau, \tau).$$

Finally, for $P_{\mathbf{s},\mathbf{y},\mathbf{t}}(\mathbf{x})$, we can write

$$\begin{aligned}\Delta\left(P_{\mathbf{s},\mathbf{y},\mathbf{t}}(\mathbf{x})\right) &= \left(\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}), \Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x})\right) \\ &= \frac{1}{\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) + \Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x})}(1-\tau,\tau).\end{aligned}$$

Therefore $\Delta(\mathbf{z}) - \Delta\left(P_{\mathbf{s},\mathbf{y},\mathbf{t}}(\mathbf{x})\right)$ is orthogonal to (4.6).

4.2.2.4 Close to an end-of-any-line

Close to the non-trivial end or start of any line, we don't have to be as careful with defining the displacement: any Lipschitz extension of the displacement we defined everywhere else would do, since here we are allowed (in fact, expect) to have fixed points.

For concreteness, let $(\mathbf{s}\to\mathbf{t})$ be the last Brouwer line segment in a path. In (4.2), we defined the displacement for points $\mathbf{x}$ such that $\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x}) \le 1-\sqrt{h}$. For points such that $\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x}) = 1$ (i.e. at the hyperplane through $\mathbf{t}$ and perpendicular to $(\mathbf{s}\to\mathbf{t})$), we simply set the default displacement $\hat{g}(\mathbf{x}) \triangleq \delta(\mathbf{0}_{3m}, \mathbf{1}_m)$. For intermediate values of $\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x}) \in [1-\sqrt{h}, 1]$, we simply interpolate according to $\sigma_{(\mathbf{s}\to\mathbf{t})}(\mathbf{x})$. Notice that this induces a fixed point for some intermediate point, since for $\mathbf{x}$ directly "above" the Brouwer line segment, $\delta\frac{\mathbf{z}-\mathbf{x}}{h}$ perfectly cancels $\delta(\mathbf{0}_{3m}, \mathbf{1}_m)$. Define the displacement analogously at the (non-trivial) start of a path.

4.2.2.5 Outside the picture

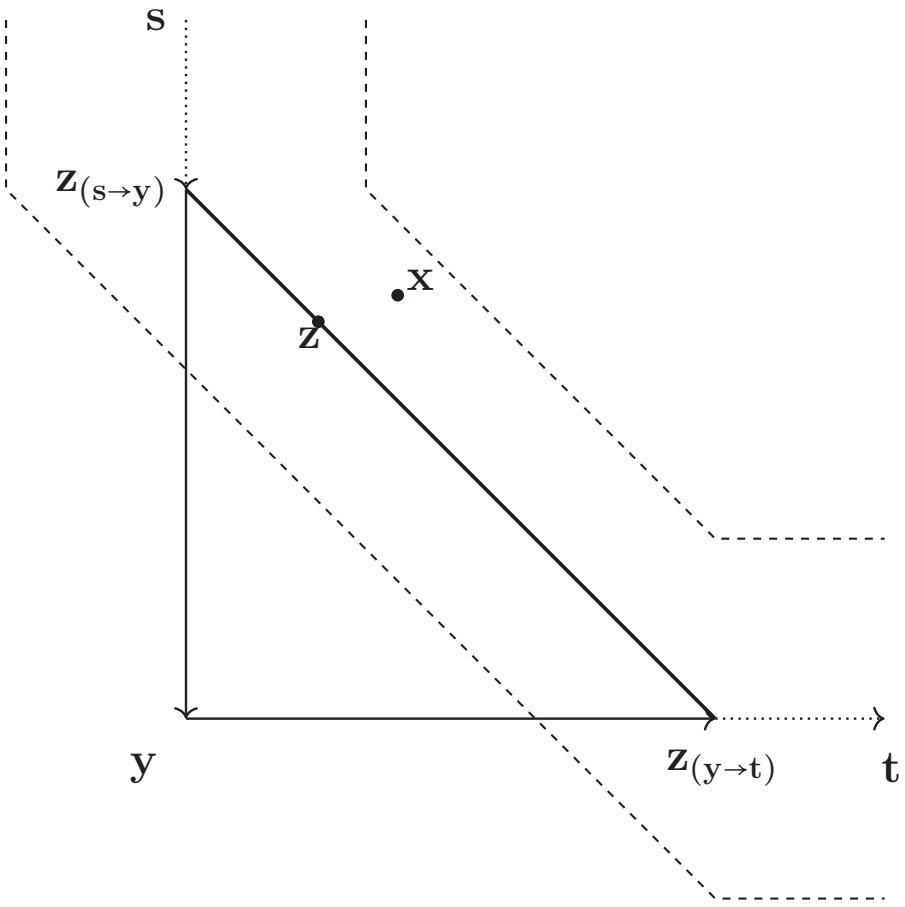
The displacement outside the picture is constructed by interpolating the displacement at $\frac{1}{m}\sum_{i=3m+1}^{4m} x_i = 1/2$, and the displacement at points in the "top" of the hypercube, where $x_i = 2$ for every i in the last m coordinates. The former displacement, where $\mathsf{E}_{i\in\{3m+1,\ldots,4m\}} x_i = 1/2$ is defined to match the displacement inside the picture. Namely, it is the default displacement everywhere except near the first Brouwer line segment which goes "down" from $\mathbf{s} = (\mathbf{0}_{3m}, 2\cdot\mathbf{1}_m)$ to $\mathbf{t} = (\mathbf{0}_{4m})$. Near this line, it is defined according to (4.2). (Notice that $\|\mathbf{t}-\mathbf{s}\|_2 = 1$.)

Formally, let $\mathbf{z}_{1/2} = \left(\mathbf{0}_{3m}, \frac{1}{2}\cdot\mathbf{1}_m\right)$; for $\mathbf{x}$ on the boundary of the picture, we have:

$$\hat{g}(\mathbf{x}) \triangleq \begin{cases} \delta(\mathbf{0}_{3m}, -\mathbf{1}_m) & \left\|\mathbf{x}-\mathbf{z}_{1/2}\right\|_2 = 0 \\ \delta\frac{(\mathbf{z}_{1/2}-\mathbf{x})}{h} & \left\|\mathbf{x}-\mathbf{z}_{1/2}\right\|_2 = h \\ \delta(\mathbf{0}_{3m}, \mathbf{1}_m) & \left\|\mathbf{x}-\mathbf{z}_{1/2}\right\|_2 \ge 2h \end{cases} \tag{4.7}$$

For points $\mathbf{x}$ such that $\sum_{i=3m+1}^{4m} x_i$ is very close to 2, the displacement $\delta(\mathbf{0}_{3m}, \mathbf{1}_m)$ is not helpful because it points outside the hypercube, i.e. it would get completely

Figure 4.3: Geometry near a Brouwer vertex



The figure (not drawn to scale) shows some of the important points near a Brouwer vertex $\mathbf{y}$: There is an incoming Brouwer line segment from $\mathbf{s}$ through $\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}$, and an outgoing Brouwer line segment to $\mathbf{t}$ through $\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}$. For each point $\mathbf{x}$ between the dashed lines, we assign a point $\mathbf{z}$ on the line $L_{\mathbf{y}}$ as in (4.3), and define the displacement according to (4.4). Outside the dashed lines (including at $\mathbf{y}$ itself), we use the default displacement $\delta\left(\mathbf{0}_{3m}, \mathbf{1}_m\right)$.

erased by the truncation. Instead, we define the displacement as follows:

$$\hat{g}(\mathbf{x}) \triangleq \begin{cases} \delta(\mathbf{0}_{3m}, -\mathbf{1}_m) & \|\mathbf{x} - \mathbf{z}_2\|_2 = 0 \\ \delta\frac{(\mathbf{z}_1 - \mathbf{x})}{h} & \|\mathbf{x} - \mathbf{z}_2\|_2 \geq h, \end{cases} \tag{4.8}$$

where $\mathbf{z}_2 = (\mathbf{0}_{3m}, 2 \cdot \mathbf{1}_m)$. When $\theta \triangleq \sum_{i=3m+1}^{4m} x_i \in (1/2, 2)$, we interpolate between (4.7) and (4.8) according to $\frac{\theta - 1/2}{3/2}$.

4.2.3 Locally computing the Brouwer function

The function f defined above is *local* in two different ways: First, in order to compute $f(\mathbf{x})$ at a point $\mathbf{x}$ which is close to the embedding of one or a few vertices in the Membership End-of-a-Line instance I, we only need to understand I at those vertices. The second type of locality observes that in order to compute just a single coordinate $f_i(\mathbf{x})$, we only need partial information about $\mathbf{x}$.

Locality in the Membership End-of-a-Line instance

Fact 4.2.2. The function f defined below is *local* in the sense that there exists a class of functions $\{f_{I_1,I_2} : [-1,2]^{4m} \to [-1,2]^{4m}\}$ which do not depend on I, and f can be defined as an interpolation between these functions such that:

1. If the first m-tuple of coordinates of $\mathbf{x}$ is $12\sqrt{h}$-close to the encoded vertex $E(v)$, but the second m-tuple of coordinates of $\mathbf{x}$ is $12\sqrt{h}$-far from any encoded vertex $E(w)$ then $f_{I(v),I_2}(\mathbf{x}) = f(\mathbf{x})$ for every I_2.

2. If the second m-tuple of coordinates of $\mathbf{x}$ is $12\sqrt{h}$-close to the encoded vertex $E(w)$, but the first m-tuple of coordinates of $\mathbf{x}$ is $12\sqrt{h}$-far from any encoded vertex $E(v)$ then $f_{I_1,I(w)}(\mathbf{x}) = f(\mathbf{x})$ for every I_1.

3. If the first m-tuple of coordinates of $\mathbf{x}$ is $12\sqrt{h}$-close to the encoded vertex $E(v)$, and the second m-tuple of coordinates of $\mathbf{x}$ is $12\sqrt{h}$-close to the encoded vertex $E(w)$ then $f_{I(v),I(w)}(\mathbf{x}) = f(\mathbf{x})$.

4. If none of the above conditions are satisfied, then $f_{I_1,I_2}(\mathbf{x}) = f(\mathbf{x})$ for every I_1, I_2.

Locality in a single coordinates

In order to compute all of $f(\mathbf{x})$ exactly, we essentially need to know $\mathbf{x}$ in every coordinate. However, in order to compute $f_i(\mathbf{x})$ (the i-th coordinate of $f(\mathbf{x})$), it suffices to know x_i and that $\mathbf{x}$ is:

- inside the picture, but far from every Brouwer line segment;
- close to some point $\mathbf{z}$ on Brouwer line segment $(\mathbf{s} \to \mathbf{t})$ (but far from $\mathbf{s}$ and $\mathbf{t}$),
  - also need to know s_i, t_i, z_i, $\|\mathbf{x}-\mathbf{z}\|_2$ and $\|\mathbf{t}-\mathbf{s}\|_2$;
- close to some point $\mathbf{z}$ on line $L_{\mathbf{y}}$ for Brouwer vertex $\mathbf{y}$ on the intersection of Brouwer lines $(\mathbf{s} \to \mathbf{y})$ and $(\mathbf{y} \to \mathbf{t})$,
  - also need to know s_i, y_i, t_i, z_i, $\|\mathbf{x}-\mathbf{z}\|_2, \|\mathbf{y}-\mathbf{s}\|_2, \|\mathbf{t}-\mathbf{y}\|_2$, and α; or
- outside the picture,
  - also need to know $\mathsf{E}_{i\in\{3m+1,\ldots,4m\}} x_i$ and $\|\mathbf{x}-\mathbf{z}\|_2$, where $\mathbf{z}$ is the $\left(\mathsf{E}_{i\in\{3m+1,\ldots,4m\}} x_i\right)$-weighted average of $\mathbf{z}_{1/2}$ and $\mathbf{z}_2$.

By Lipschitz continuity, if we only want to compute $f_i(\mathbf{x})$ to within $\pm O(\epsilon)$, it suffices to know all the quantities above to within $\pm\epsilon$. Furthermore, at distance $\pm\epsilon$ near interfaces between the different cases (inside/outside the picture, close to 0/1/2 lines), we can use the wrong displacement, and still be within $\pm O(\epsilon)$ of $f_i(\mathbf{x})$.

Part III

PPAD

Chapter 5

PPAD-hardness of approximation

In this part of the dissertation we introduce and prove our PPAD-hardness of approximation results, in particular for finding for Nash equilibria. As we discussed in the introduction, for a constant number of players, we are unlikely to prove PPAD-hardness since a quasi-polynomial time approximation algorithm exists [LMM03]. Our main focus in this part is games with a large number of players. For such games, there is a question of representation: the normal form representation is exponential in the number of the players. Instead, we consider three natural and well-studied classes of many-player games that have succinct representations:

Definition 5.0.1. Polymatrix games In a polymatrix game [Yan68], each pair of players simultaneously plays a separate two-player game. Every player has to play the same strategy in every two-player subgame, and her utility is the sum of her subgame utilities. The game is given in the form of the payoff matrix for each two-player game.

Graphical games In a graphical game [Kea07], the utility of each player depends only on the action chosen by a few other players. This game now naturally induces a directed graph: we say that $(i, j) \in E$ if the utility of player j depends on the strategy chosen by player i. When the maximal incoming degree is bounded, the game has a representation polynomial in the number of players and strategies.

Succinct games Most generally, in a succinct game [SV12][1] the normal form representation is succinctly described by some small circuit.

[1] Related notions for two-player games have been also considered by [FKS95] and [For+08].

Any of the above restrictions suffices to guarantee that the game has a succinct representation. Our main result in this part is that even for games that satisfy all restrictions simultaneously, finding an ϵ-approximate Nash equilibrium is PPAD-complete.

Theorem 5.0.2. *There exists a constant $\epsilon > 0$, such that given a degree 3, bipartite, polymatrix game where each player has two actions, finding an ϵ-approximate Nash equilibrium is PPAD-complete.*

The notion of ϵ-approximate Nash equilibrium used in Theorem 5.0.2 requires that *every* player plays ϵ-optimally. The most interesting open question left open in this part of the dissertation is whether the equilibrium computation problem remains PPAD-hard even if we only require that *most* of the players play ϵ-optimally. This is the "PCP Conjecture for PPAD":

Conjecture 5.0.3 (PCP for PPAD)**.** *There exist constants $\epsilon, \delta > 0$ such that given a degree 3, bipartite, polymatrix game where each player has two actions, finding an (ϵ, δ)-WeakNash is PPAD-complete.*

While, proving (or disproving) the "PCP Conjecture for PPAD" remains open (see additional discussion in Section 16.1.1), we do prove in this section that for the more general class of succinct games, finding an (ϵ, δ)-WeakNash is indeed PPAD-hard.

Theorem 5.0.4. *There exist constants $\epsilon, \delta > 0$, such that finding an (ϵ, δ)-WeakNash is PPAD-hard for succinct multiplayer games where each player has a constant number of actions.*

Besides Theorem 5.0.4, all our results rely in this part of the dissertation rely on the hardness of approximation for the generalized circuit problem. Generalized circuits are similar to standard algebraic circuits, the main difference being that generalized circuits contain cycles, which allow them to verify fixed points of continuous functions. A generalized circuit induces a constraint satisfaction problem, ϵ-GCIRCUIT [CDT09]: find an assignment for the values on the lines of the circuit, that simultaneously ϵ-approximately satisfies all the constraints imposed by the gates (see Chapter 6 for a formal definition). ϵ-GCIRCUIT was implicitly proven PPAD-complete for exponentially small ϵ by Daskalakis et al [DGP09], and explicitly for polynomially small ϵ by Chen et al [CDT09]. Here we prove that it continues to be PPAD-complete for some constant ϵ.

Theorem 5.0.5 (Generalized circuit)**.** *There exists a constant $\epsilon > 0$ such that ϵ-GCIRCUIT with fan-out 2 is PPAD-complete.*

This result, in turn builds on the hardness of finding an ℓ_{infty}-approximate fixed point, which we proved in Theorem 4.1.1.

We note that except for the hardness of the course allocation problem (Chapter 10), all the problems we consider in this part of the dissertation were previously known to be PPAD-hard for polynomial approximation factors.

Chapter 6

The generalized circuit problem

Generalized circuits are similar to the standard algebraic circuits, the main difference being that generalized circuits contain cycles, which allow them to verify fixed points of continuous functions. We restrict the class of generalized circuits to include only a particular list of gates described below. Formally,

Definition 6.0.1 (Generalized circuits, [CDT09])**.** A *generalized circuit* $\mathcal{S}$ is a pair $(V, \mathcal{T})$, where V is a set of nodes and $\mathcal{T}$ is a collection of gates. Every gate $T \in \mathcal{T}$ is a 5-tuple $T = G(\zeta \mid v_1, v_2 \mid v)$, in which $G \in \{G_\zeta, G_{\times\zeta}, G_=, G_+, G_-, G_<, G_\vee, G_\wedge, G_\neg\}$ is the type of the gate; $\zeta \in \mathbb{R} \cup \{nil\}$ is a real parameter; $v_1, v_2 \in V \cup \{nil\}$ are the first and second input nodes of the gate; and $v \in V$ is the output node.

The collection $\mathcal{T}$ of gates must satisfy the following important property: For every two gates $T = G(\zeta \mid v_1, v_2 \mid v)$ and $T' = G'(\zeta' \mid v'_1, v'_2 \mid v')$ in $\mathcal{T}$, $v \neq v'$.

Alternatively, we can think of each gate as a constraint on the values on the incoming and outgoing wires. We are interested in the following constraint satisfaction problem: given a generalized circuit, find an assignment to all the wires that simultaneously satisfies all the gates. When every gate computes a continuous function of the incoming wires (with inputs and output in $[0, 1]$), a solution must exist by Brouwer's fixed point theorem.

In particular, we are interested in the approximate version of this CSP, where we must approximately satisfy every constraint.

Definition 6.0.2. Given a generalized circuit $\mathcal{S} = (V, \mathcal{T})$, we say that an assignment $\mathbf{x}: V \to [0, 1]$ *ϵ-approximately satisfies* $\mathcal{S}$ if for each of the following gates, $\mathbf{x}$ satisfies the corresponding constraints:

Gate	Constraint
$G_\zeta(\alpha \parallel a)$	$\mathbf{x}[a] = \alpha \pm \epsilon$
$G_{\times\zeta}(\alpha \mid a \mid b)$	$\mathbf{x}[b] = \alpha \cdot \mathbf{x}[a] \pm \epsilon$
$G_=(\mid a \mid b)$	$\mathbf{x}[b] = \mathbf{x}[a] \pm \epsilon$
$G_+(\mid a, b \mid c)$	$\mathbf{x}[c] = \min(\mathbf{x}[a] + \mathbf{x}[b], 1) \pm \epsilon$
$G_-(\mid a, b \mid c)$	$\mathbf{x}[c] = \max(\mathbf{x}[a] - \mathbf{x}[b], 0) \pm \epsilon$
$G_<(\mid a, b \mid c)$	$\mathbf{x}[c] = \begin{cases} 1 \pm \epsilon & \mathbf{x}[a] < \mathbf{x}[b] - \epsilon \\ 0 \pm \epsilon & \mathbf{x}[a] > \mathbf{x}[b] + \epsilon \end{cases}$
$G_\vee(\mid a, b \mid c)$	$\mathbf{x}[c] = \begin{cases} 1 \pm \epsilon & \mathbf{x}[a] = 1 \pm \epsilon \text{ or } \mathbf{x}[b] = 1 \pm \epsilon \\ 0 \pm \epsilon & \mathbf{x}[a] = 0 \pm \epsilon \text{ and } \mathbf{x}[b] = 0 \pm \epsilon \end{cases}$
$G_\wedge(\mid a, b \mid c)$	$\mathbf{x}[c] = \begin{cases} 1 \pm \epsilon & \mathbf{x}[a] = 1 \pm \epsilon \text{ and } \mathbf{x}[b] = 1 \pm \epsilon \\ 0 \pm \epsilon & \mathbf{x}[a] = 0 \pm \epsilon \text{ or } \mathbf{x}[b] = 0 \pm \epsilon \end{cases}$
$G_\neg(\mid a \mid b)$	$\mathbf{x}[b] = \begin{cases} 1 \pm \epsilon & \mathbf{x}[a] = 0 \pm \epsilon \\ 0 \pm \epsilon & \mathbf{x}[a] = 1 \pm \epsilon \end{cases}$

(Where G_ζ and $G_{\times\zeta}$ also take a parameter $\alpha \in [0,1]$.)

Given a generalized circuit $\mathcal{S} = (V, \mathcal{T})$, ϵ-GCIRCUIT is the problem of finding an assignment that ϵ-approximately satisfies it.

Brittle comparators Intuitively, in order for (approximate) solutions to the circuit problem to correspond to (approximate) equilibria, all our gates should implement continuous (Lipschitz) functions. The gate $G_<(\mid a, b \mid c)$, for example, approximates that the function $c(a,b) = \begin{cases} 1 & a < b \\ 0 & a \geq b \end{cases}$, which is not continuous. To overcome this problem, Daskalakis et al [DGP09] defined the *brittle comparator*: when a is (ϵ-) larger than b, it outputs 0; when b is (ϵ-) larger than a, it outputs 1. However, when a and b are (ϵ-approximately) equal, its behavior is undefined.

Brittleness introduces difficulties in the transition from continuous to discrete solutions. This challenge is overcome by an averaging gadget, which is described in detail in Section 6.2.

Our results

ϵ-GCIRCUIT was implicitly proven PPAD-complete for exponentially small ϵ by Daskalakis et al [DGP09], and explicitly for polynomially small ϵ by Chen et al [CDT09]. Here

we prove that it continues to be PPAD-complete for some constant ϵ.

Theorem (Generalized circuit; Theorem 5.0.5 restated)**.** *There exists a constant* $\epsilon > 0$ *such that* ϵ-GCIRCUIT *with fan-out* 2 *is* PPAD*-complete.*

6.1 Proof overview

The key idea that enables us to improve over previous hardness of approximation for ϵ–GCIRCUIT (and Nash equilibrium) [DGP09; CDT09] is our particular choice of hard fixed point instance in Section 4.1. The first advantage is that our instance is simply harder to approximate: finding an ϵ-approximate fixed point (i.e. $\mathbf{x}$ such that $\|f(\mathbf{x}) - \mathbf{x}\|_\infty \leq \epsilon$) is PPAD-hard for $\epsilon = \Omega(1)$ (as opposed to $\epsilon = 1/\exp(n)$ for [DGP09] and $\epsilon = 1/\mathsf{poly}(n)$ in [CDT09]).

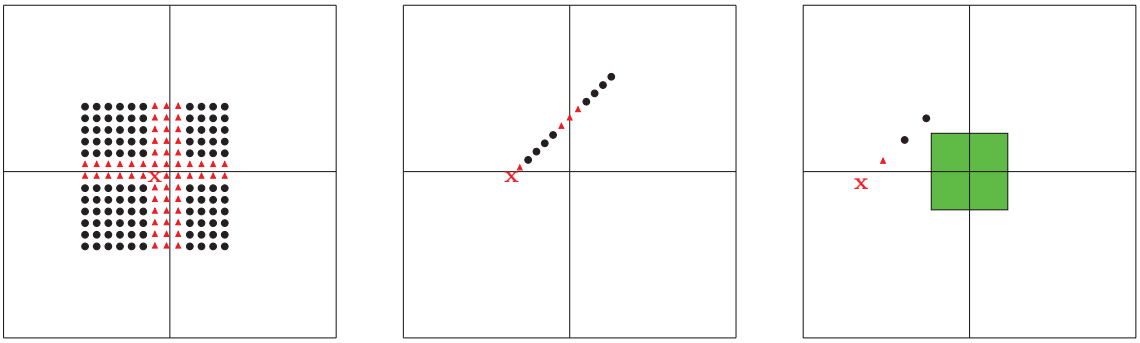
A simpler averaging gadget Our construction of hard instances of Brouwer functions, as do the ones form previous works, partitions the (continuous) hypercube into subcubes, and define the function separately on each subcube. When we construct a circuit that approximately simulates such a Brouwer function, we have a problem near the facets of the subcubes: using *approximate gates* and *brittle comparators* (both defined formally in Chapter 6), one cannot determine to which subcube the input belongs. This is the most challenging part of our reduction, as was also the case in [DGP09; CDT09].

Originally, Daskalakis et al [DGP09] tackled this obstacle by approximating $f(\mathbf{x})$ as the average over a ball around $\mathbf{x}$. The key observation is that even if $\mathbf{x}$ is close to a facet between subcubes, most of the points in its neighborhoods will be sufficiently far. Yet if f is Lipschitz they are mapped approximately to the same point as $\mathbf{x}$. This works fine in $O(1)$ dimensions, but then the inapproximability parameter is inherently exponentially small (in constant dimensions, it is easy to construct a $1/\mathsf{poly}(n)$-net over the unit hypercube). For $\mathsf{poly}(n)$ dimensions, the (discretization of the) ball around $\mathbf{x}$ contains exponentially many points.

Chen et al [CDT09] overcome this problem using *equiangle sampling*: consider many translations of the input vector by adding small multiples of the all-ones vector; compute the displacement for each translation, and average. Since each translation may be close to a facet in a different dimension, Chen et al consider a polynomial number of translations. Thus, all translations must be polynomially close to each other - otherwise they will be too far to approximate the true input.

We avoid this problem by observing another nice property of the [HPV89]'s construction: when the input vector lies near two or more facets, the displacement is

Figure 6.1: Comparison of averaging gadgets



Daskalakis et al [DGP09] Chen et al [CDT09] This paper

A comparison of the averaging gadgets of [DGP09], [CDT09], and this paper. $\mathbf{x}$ is the point whose displacement we would like to estimate using imprecise gates and brittle comparators. Points that are too close to a facet between subcubes are denoted by triangles, while points that are sufficiently far are denoted by circles. Finally, in this paper we have a "safe" zone (shaded) around the corner where we don't need to parse the subcube; thus we only need to avoid one facet.

(approximately) the same, regardless of the subcube. Once we rule out such points, it suffices to sample only a constant number of points (as at most one of them may be too close to a facet). See also illustration in Figure 6.1.

Completing the proof Given a point $\mathbf{x}' \approx \mathbf{x}$ which is safely in the interior of one subcube, we can parse the corresponding binary vector, use logical operator gates to simulate the End-of-a-Line circuit, and then approximately compute $f(\mathbf{x}')$. This is tedious, but mostly straightforward.

One particular challenge that nevertheless arises is preventing the error from accumulating when concatenating approximate gates. Of course this is more difficult in our setting where each gate may err by a constant $\epsilon > 0$. Fortunately, the definition of ϵ-Gcircuit provides logical operator gates that round the output to $\{0, 1\}$ before introducing new error. As long as the inputs are unambiguous bits, approximate logical operator gates can be concatenated without accumulating errors.

In order to carry out the reduction to Nash equilibrium (Section 7.1), we must first ensure that every gate in our generalized circuit has a constant fan-out (Section 6.3). We can replace each logical operator gate with a binary tree of fan-out 2,

alternating negation gates (that do not accumulate error). Given an arithmetic gate with large fan-out, we convert its output to unary representation[1] using a constant number of (fan-out 2) gates. Then we copy the unary representation using a binary tree of negation gates. Finally, we convert each copy back to a real number using a constant number of gates.

6.2 From Brouwer to ϵ-GCIRCUIT

In this section we prove a slightly easier version of Theorem 5.0.5, for a generalized circuit with unbounded fan-out. We reduce to constant fan-out in the next section.

Proposition 6.2.1. *There exists a constant $\epsilon > 0$ such that ϵ-GCIRCUIT is* PPAD*-complete.*

Proof. We continue to denote the hard Brouwer function by $f:[0,1]^{2n+2} \to [0,1]^{2n+2}$, and its associated displacement by $g(\mathbf{y}) = f(\mathbf{y}) - \mathbf{y}$. We design a generalized circuit $\mathcal{S}$ that computes f, and verifies that the output is equal to the input. We show that every ϵ-approximate solution to $\mathcal{S}$ corresponds to an $O(\epsilon^{1/4})$-approximate fixed point of f.

Recall that the construction from Section (4.1) divides the hypercube into equal-sized subcubes (of length 1/4). Furthermore, all the paths in H are embedded in the 2^{2n+1} subcubes that belong to the picture. For ease of exposition, we present a construction that only works for points in the picture, i.e. $\mathbf{y} \in [1/4, 3/4]^{2n+1} \times [1/4, 1/2]$. It is straightforward how to use the same ideas to extend the circuit to deal with all $\mathbf{y} \in [0,1]^{2n+2}$.

The most challenging part of the construction is the extraction of the information about the local subcube: is it part of a tube? if so, which are the entrance and exit facets? This is done by extracting the binary representation of the current subcube, and feeding it to the (Boolean) circuit that computes H (recall that H is our collection of paths and cycles from Section (4.1.2)). Notice that whenever we have valid logic bits, i.e. $\mathbf{x}[b] < \epsilon$ or $\mathbf{x}[b] > 1 - \epsilon$, we can perform logic operations on them without increasing the error.

Once we know the behavior of the path on the current subcube, we simply have to locally implement the mapping from the previous section, for which we have a closed form description, using the available gates in the definition of generalized circuits.

[1] Unary representation of numbers with constant precision is prevalent throughout our implementation of the generalized circuit. We prefer unary representation over binary, because in the former at most one bit can be ambiguous due to the use of brittle comparators.

Since this definition does not include multiplication and division, we implement multiplication and division in Algorithms 2 and 3 in Subsection 6.2.1.

Our construction has four parts: (1) equiangle sampling segment, (2) computing the displacement, (3) summing the displacement vectors, and (4) closing the loop. The first part contains a new idea introduced in this paper: using a constant size sample. The second part is a more technical but straightforward description of the implementation of the closed-form mapping by approximate gates. The third and fourth parts are essentially identical to [CDT09]. □

6.2.1 Subroutines

In this subsection we show how to implement a few useful subroutines using the gates in the definition of ϵ-GCIRCUIT.

6.2.1.1 If-Else

We begin by describing how to implement a simple if-else. Similar ideas can be used to implement more involved cases such as (4.1).

Claim 6.2.2. In any ϵ-approximate solution to IF-ELSE$(\mid a,b,c \mid d)$,

$$\mathbf{x}[d] = \begin{cases} \mathbf{x}[c] \pm O(\epsilon) & \text{if } \mathbf{x}[a] < \sqrt{\epsilon} \\ \mathbf{x}[b] \pm O(\epsilon) & \text{if } \mathbf{x}[a] > 1 - \sqrt{\epsilon} \end{cases}.$$

Proof. By definition of $G_\neg$, we have that

$$\mathbf{x}[\overline{a}] = \begin{cases} 1 \pm \epsilon & \text{if } \mathbf{x}[a] < \sqrt{\epsilon} \\ 0 \pm \epsilon & \text{if } \mathbf{x}[a] > 1 - \sqrt{\epsilon} \end{cases}.$$

Algorithm 1 IF-ELSE$(\mid a,b,c \mid d)$

1. $G_\neg(\mid a \mid \overline{a})$# $\overline{a}$ is the negation of a
2. $G_-(\mid b,\overline{a} \mid b')$# b' is (approximately) equal to b iff $a = 1$
3. $G_\neg\left(\mid \overline{a} \mid \overline{\overline{a}}\right)$# $\overline{\overline{a}}$ is the roudning of a to $\{0,1\}$
4. $G_-\left(\mid c,\overline{\overline{a}} \mid c'\right)$# c' is (approximately) equal to c iff $a = 0$
5. $G_-(\mid b',c' \mid d)$

Therefore by definition of G_-,

$$\mathbf{x}[b'] = \begin{cases} 0 \pm O(\epsilon) & \text{if } \mathbf{x}[a] < \sqrt{\epsilon} \\ \mathbf{x}[b] \pm O(\epsilon) & \text{if } \mathbf{x}[a] > 1 - \sqrt{\epsilon} \end{cases}.$$

Similarly,

$$\mathbf{x}[c'] = \begin{cases} \mathbf{x}[c] \pm O(\epsilon) & \text{if } \mathbf{x}[a] < \sqrt{\epsilon} \\ 0 \pm O(\epsilon) & \text{if } \mathbf{x}[a] > 1 - \sqrt{\epsilon} \end{cases}.$$

Finally, the claim follows by definition of G_+. □

6.2.1.2 Multiply

Claim 6.2.3. In any ϵ-approximate solution to MULTIPLY($|\, a, b \,|\, c$),

$$\mathbf{x}[c] = \mathbf{x}[a] \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right).$$

Proof. For any k, the first gate implies that

$$\mathbf{x}[\zeta_k] = k\sqrt{\epsilon} \pm \epsilon.$$

The second gate thus gives

$$\mathbf{x}[\overline{a_k}] = \begin{cases} 0 \pm \epsilon & \text{if } \mathbf{x}[a] > k\sqrt{\epsilon} + O(\epsilon) \\ 1 \pm \epsilon & \text{if } \mathbf{x}[a] < k\sqrt{\epsilon} - O(\epsilon). \end{cases} \tag{6.1}$$

Notice that the above equation is ambiguous for at most one value of k. In particular,

$$\sum_k \left(1 - \mathbf{x}[\overline{a_k}]\right)\sqrt{\epsilon} = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right). \tag{6.2}$$

We also have

$$\mathbf{x}[d_k] = \mathbf{x}[b] \cdot \sqrt{\epsilon} \pm \epsilon.$$

The subtraction gate zeros $\mathbf{x}[d_k]$ for all k such that $\mathbf{x}[a] < k\sqrt{\epsilon} - O(\epsilon)$, and has negligible effect for k such that $\mathbf{x}[a] > k\sqrt{\epsilon} + O(\epsilon)$:

$$\mathbf{x}[e_k] = \begin{cases} \mathbf{x}[b] \cdot \sqrt{\epsilon} \pm 2\epsilon & \text{if } \mathbf{x}[a] > k\sqrt{\epsilon} + O(\epsilon) \\ 0 \pm 2\epsilon & \text{if } \mathbf{x}[a] < k\sqrt{\epsilon} - O(\epsilon). \end{cases}$$

The sum of the $\mathbf{x}[e_k]$'s satisfies:

$$\sum_k \mathbf{x}[e_k] = \mathbf{x}[a] \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right),$$

Algorithm 2 MULTIPLY($| a, b | c$)

1. $G_\zeta(0 \,\|\, h_0)$
2. **for** each $k \in \left[1/\sqrt{\epsilon}\right]$:

   a) $G_\zeta\left(k\sqrt{\epsilon} \,\|\, \zeta_k\right)$, $G_<(| a, \zeta_k | \overline{a_k})$
   # The vector $(\overline{a_k})$ is the unary representation of a

   $$\sum_{k:\mathrm{X}[\overline{a_k}]<\epsilon} \sqrt{\epsilon} = \max_{k:\mathrm{X}[\overline{a_k}]<\epsilon} k\sqrt{\epsilon} = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right)$$

   b) $G_{\times\zeta}\left(\sqrt{\epsilon} \mid b \mid d_k\right)$
   # The vector (d_k) is simply equal to $b \cdot \sqrt{\epsilon}$ everywhere.

   $$\sum_{k:\mathrm{X}[\overline{a_k}]<\epsilon} \mathbf{x}[d_k] = \left(\sum_{k:\mathrm{X}[\overline{a_k}]<1-\epsilon} \sqrt{\epsilon}\right) \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right) = \mathbf{x}[a] \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right)$$

   c) $G_-(| d_k, \overline{a_k} | e_k)$
   # The vector (e_k) is $b \cdot \sqrt{\epsilon}$ only when $(\overline{a_k}) < \epsilon$.

   $$\sum_{k:\mathrm{X}[\overline{a_k}]<\epsilon} \mathbf{x}[e_k] = \mathbf{x}[a] \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right)$$

   d) $G_+(| h_{k-1}, e_k | h_k)$
   # Finally, we sum the e_k's to get $a \cdot b$

   $$\mathbf{x}\left[h_{1/\sqrt{\epsilon}}\right] = \mathbf{x}[a] \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right)$$

3. $G_=\left(| h_{1/\sqrt{\epsilon}} | c\right)$

where we have an error of $\pm O\left(\sqrt{\epsilon}\right)$ arising from aggregating $\pm 2\epsilon$ for $1/\sqrt{\epsilon}$ distinct k's, and another $\pm O\left(\sqrt{\epsilon}\right)$ from (6.2).

By induction, each h_k is approximately equal to the sum of the first $\mathbf{x}[e_j]$'s:

$$\mathbf{x}[h_k] = \sum_{j=1}^{k} \mathbf{x}[e_j] \pm k\epsilon.$$

In particular, we have

$$\begin{aligned} \mathbf{x}\left[h_{1/\sqrt{\epsilon}}\right] &= \sum_k \mathbf{x}[e_k] \pm \sqrt{\epsilon} \\ &= \mathbf{x}[a] \cdot \mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right). \end{aligned}$$

□

6.2.1.3 Divide

Claim 6.2.4. In any ϵ-approximate solution to DIVIDE$(|\, a, b \,|\, c)$,

$$\mathbf{x}[c] \cdot \mathbf{x}[b] = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right).$$

Notice that for Algorithm 3, in any ϵ-approximate solution, $\mathbf{x}[c] = \mathbf{x}[a]/\mathbf{x}[b] \pm O\left(\sqrt{\epsilon}\right)/\mathbf{x}[b]$; when $\mathbf{x}[b]$ and ϵ are bounded away from 0, this is only a constant factor increase in the error.

Proof. For each k, we have

$$\mathbf{x}[b_k] = k\sqrt{\epsilon} \cdot \mathbf{x}[b] \pm \epsilon.$$

Thus also

$$\mathbf{x}[d_k] = \begin{cases} 1 \pm \epsilon & \text{if } \mathbf{x}[a] > k\sqrt{\epsilon} \cdot \mathbf{x}[b] + O(\epsilon) \\ 0 \pm \epsilon & \text{if } \mathbf{x}[a] < k\sqrt{\epsilon} \cdot \mathbf{x}[b] - O(\epsilon). \end{cases}$$

Notice that $\mathbf{x}[d_k]$ is ambiguous for at most k. Furthermore, aggregating the $\pm\epsilon$ error over $1/\sqrt{\epsilon}$ distinct k's, we have:

$$\sum \mathbf{x}[d_k] \cdot \sqrt{\epsilon}\mathbf{x}[b] = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right).$$

$\mathbf{x}[e_k]$'s are a step closer to what we need:

$$\mathbf{x}[e_k] = \mathbf{x}[d_k]\sqrt{\epsilon} \pm \epsilon,$$

and therefore also

$$\sum \mathbf{x}[e_k] \cdot \mathbf{x}[b] = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right). \tag{6.3}$$

Finally, by induction

$$\mathbf{x}\left[h_{1/\sqrt{\epsilon}}\right] = \sum \mathbf{x}[e_k] \pm \sqrt{\epsilon},$$

and the claim follows by plugging into (6.3). □

Algorithm 3 DIVIDE($| a, b | c$)

1. $G_{\zeta}(0 \,\|\, h_0)$
2. **for** each $k \in \left[1/\sqrt{\epsilon}\right]$:

   a) $G_{\times\zeta}\left(k\sqrt{\epsilon} \mid b \mid b_k\right)$, $G_{<}(| b_k, a | d_k)$
   # The vector (d_k) is the unary representation of a/b

   $$\left(\sum_{k:\mathrm{X}[d_k]>\epsilon} \sqrt{\epsilon}\right) \cdot \mathbf{x}[b] = \left(\max_{k:\mathrm{X}[d_k]>\epsilon} k\sqrt{\epsilon}\right) \cdot \mathbf{x}[b] = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right)$$

   b) $G_{\times\zeta}\left(\sqrt{\epsilon} \mid d_k \mid e_k\right)$
   # The vector (e_k) is a $\left(\sqrt{\epsilon}\right)$-scaled version of (d_k)

   $$\left(\sum \mathbf{x}[e_k]\right) \cdot \mathbf{x}[b] = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right)$$

   c) $G_{+}(| h_{k-1}, e_k | h_k)$
   # Finally, we sum the e_k's

   $$\mathbf{x}\left[h_{1/\sqrt{\epsilon}}\right] \cdot \mathbf{x}[b] = \mathbf{x}[a] \pm O\left(\sqrt{\epsilon}\right)$$

3. $G_{=}\left(| h_{1/\sqrt{\epsilon}} | c\right)$

6.2.1.4 Max

Claim 6.2.5. In any ϵ-approximate solution to $\textsc{Max}(|\, a_1, \ldots a_n \,|\, b)$,

$$\mathbf{x}[b] = \max \mathbf{x}[a_i] \pm O\left(\sqrt{\epsilon}\right).$$

Proof. Similarly to (6.1), we have that for each i, k

$$\mathbf{x}[c_{k,i}] = \begin{cases} 0 \pm \epsilon & \text{if } \mathbf{x}[a_i] < k\sqrt{\epsilon} - O(\epsilon) \\ 1 \pm \epsilon & \text{if } \mathbf{x}[a_i] > k\sqrt{\epsilon} + O(\epsilon). \end{cases}$$

For each k, taking OR of all the $c_{k,i}$'s gives (approximately) 1 iff any of the $\mathbf{x}[a_i]$'s is sufficiently large; in particular if the maximum is:

$$\mathbf{x}[d_{k,n}] = \begin{cases} 0 \pm \epsilon & \text{if } \max_i \mathbf{x}[a_i] < k\sqrt{\epsilon} - O(\epsilon) \\ 1 \pm \epsilon & \text{if } \max_i \mathbf{x}[a_i] > k\sqrt{\epsilon} + O(\epsilon). \end{cases}$$

Therefore also (similarly to (6.2)),

$$\sum_k \mathbf{x}[d_{k,n}]\sqrt{\epsilon} = \max_i \mathbf{x}[a_i] \pm O\left(\sqrt{\epsilon}\right).$$

The $\mathbf{x}[e_k]$ take care of scaling by $\sqrt{\epsilon}$:

$$\sum_k \mathbf{x}[e_k] = \max_i \mathbf{x}[a_i] \pm O\left(\sqrt{\epsilon}\right).$$

Finally, by induction,

$$\mathbf{x}\left[h_{1/\sqrt{\epsilon}}\right] = \max \mathbf{x}[a_i] \pm O\left(\sqrt{\epsilon}\right).$$

□

6.2.1.5 Interpolate

Algorithm 4 MAX($| a_1, \ldots a_n \mid b$)

1. $G_\zeta \left(0 \parallel h_0\right)$
2. **for** each $k \in \left[1/\sqrt{\epsilon}\right]$:
   a) $G_\zeta \left(k\sqrt{\epsilon} \parallel \zeta_k\right)$
   b) $G_\zeta \left(0 \parallel d_{k,0}\right)$
   c) **for** each $i \in [n]$:
      i. $G_< \left(| \zeta_k, a_i \mid c_{k,i}\right)$
      # The vector $\left(c_{k,i}\right)_k$ is the unary representation of a_i:

$$\forall i \ \left(\max_{k:\mathbf{x}[c_{k,i}]>\epsilon} k\sqrt{\epsilon}\right) = \mathbf{x}\left[a_i\right] \pm O\left(\sqrt{\epsilon}\right)$$

      ii. $G_\vee \left(| d_{k,i-1}, c_{k,i} \mid d_{k,i}\right)$

   # The vector $(d_{k,n})$ is the unary representation of $\max a_i$:

$$\left(\max_{k:\mathbf{x}[d_{k,n}]>\epsilon} k\sqrt{\epsilon}\right) = \max \mathbf{x}\left[a_i\right] \pm O\left(\sqrt{\epsilon}\right)$$

   d) $G_{\times\zeta} \left(\sqrt{\epsilon} \mid d_{k,n} \mid e_k\right)$
   # The vector (e_k) is a $\left(\sqrt{\epsilon}\right)$-scaled version of (d_k)

$$\left(\sum \mathbf{x}\left[e_k\right]\right) = \max \mathbf{x}\left[a_i\right] \pm O\left(\sqrt{\epsilon}\right)$$

   e) $G_+ \left(| h_{k-1}, e_k \mid h_k\right)$
   # Finally, we sum the e_k's

$$\mathbf{x}\left[h_{1/\sqrt{\epsilon}}\right] = \max \mathbf{x}\left[a_i\right] \pm O\left(\sqrt{\epsilon}\right)$$

3. $G_= \left(| h_{1/\sqrt{\epsilon}} \mid b\right)$

Claim 6.2.6. In any ϵ-approximate solution to $\textsc{Interpolate}(a, w_a, b, w_b \mid c)$,

$$\mathbf{x}[c](\mathbf{x}[w_a] + \mathbf{x}[w_b]) = (\mathbf{x}[w_a] \cdot \mathbf{x}[a] + \mathbf{x}[w_b] \cdot \mathbf{x}[b]) \pm O\left(\sqrt{\epsilon}\right).$$

Proof. By Claim 6.2.4, we have

$$\begin{aligned} \mathbf{x}[\overline{w_a}] \cdot (\mathbf{x}[w_a] + \mathbf{x}[w_b]) &= \mathbf{x}[w_a] \pm O\left(\sqrt{\epsilon}\right) \\ \mathbf{x}[\overline{w_b}] \cdot (\mathbf{x}[w_a] + \mathbf{x}[w_b]) &= \mathbf{x}[w_b] \pm O\left(\sqrt{\epsilon}\right). \end{aligned}$$

Therefore, by Claim 6.2.3,

$$\begin{aligned} \mathbf{x}[c_a] \cdot (\mathbf{x}[w_a] + \mathbf{x}[w_b]) &= \mathbf{x}[w_a] \cdot \mathbf{x}[c] \pm O\left(\sqrt{\epsilon}\right) \\ \mathbf{x}[c_b] \cdot (\mathbf{x}[w_a] + \mathbf{x}[w_b]) &= \mathbf{x}[w_b] \cdot \mathbf{x}[c] \pm O\left(\sqrt{\epsilon}\right). \end{aligned}$$

The claim follows by definition of G_+. □

Algorithm 5 INTERPOLATE$(a, w_a, b, w_b \mid c)$

1. $G_{\times\zeta}\left(1/2 \mid w_a \mid w_{a/2}\right)$ and $G_{\times\zeta}\left(1/2 \mid w_b \mid w_{b/2}\right)$
   # We divide by 2 before adding in order to stay in $[0,1]$
2. $G_{+}\left(\mid w_{a/2}, w_{b/2} \mid w_{a/2+b/2}\right)$ Add the weights
3. DIVIDE$\left(\mid w_{a/2}, w_{a/2+b/2} \mid \overline{w_a}\right)$ and DIVIDE$\left(\mid w_{b/2}, w_{a/2+b/2} \mid \overline{w_b}\right)$,
   # $\overline{w_a}$ and $\overline{w_b}$ are the normalized weights:

$$\mathbf{x}\left[\overline{w_a}\right] \cdot \left(\mathbf{x}\left[w_a\right] + \mathbf{x}\left[w_b\right]\right) = \mathbf{x}\left[w_a\right] \pm O\left(\sqrt{\epsilon}\right)$$

4. MULTIPLY$(\mid \overline{w_a}, a \mid c_a)$ and MULTIPLY$(\mid \overline{w_b}, b \mid c_b)$
   # c_a and c_b are the a and b components, respectively, of c:

$$\mathbf{x}\left[c_a\right] = \mathbf{x}\left[w_a\right] \cdot \mathbf{x}\left[a\right] / \left(\mathbf{x}\left[w_a\right] + \mathbf{x}\left[w_b\right]\right) \pm O\left(\sqrt{\epsilon}\right) / \left(\mathbf{x}\left[w_a\right] + \mathbf{x}\left[w_b\right]\right)$$

5. $G_{+}\left(\mid c_a, c_b \mid c\right)$

Finally, c is the interpolation of a and b:

$$\begin{aligned} \mathbf{x}\left[c\right] \;=\; & \left(\mathbf{x}\left[w_a\right] \cdot \mathbf{x}\left[a\right] + \mathbf{x}\left[w_b\right] \cdot \mathbf{x}\left[b\right]\right) / \left(\mathbf{x}\left[w_a\right] + \mathbf{x}\left[w_b\right]\right) \\ & \pm O\left(\sqrt{\epsilon}\right) / \left(\mathbf{x}\left[w_a\right] + \mathbf{x}\left[w_b\right]\right) \end{aligned}$$

6.2.2 Equiangle sampling segment

The first information we require in order to compute the Hirsch et al mapping $f(\mathbf{y})$ is about the subcube to which $\mathbf{y}$ belongs: is it part of the tube? if so, which are the entrance and exit facets? In order to answer those questions, we extract the binary representation of the cube. Recall that our circuit uses brittle comparators; thus when $\mathbf{y}$ is close to a facet between subcubes, the behavior of the brittle comparators may be unpredictable. We start with the easy case, where $\mathbf{y}$ is actually far from every facet:

Definition 6.2.7. We say that $\mathbf{y}$ is an *interior point* if for every i, $|y_i - 1/2| > \epsilon$; otherwise, we say that $\mathbf{y}$ is a *boundary point*.

A very nice property of the Hirsch et al construction is that whenever $\mathbf{y}$ is at the intersection of two or more facets, the displacement is the same: $g(\mathbf{y}) = \delta\boldsymbol{\xi}_{2n+2}$. Thus, by the Lipschitz property of g, whenever $\mathbf{y}$ is close to the intersection of two or more facets, the displacement is approximately $\delta\boldsymbol{\xi}_{2n+2}$. For such $\mathbf{y}$'s, we don't care to which subcube they belong.

Definition 6.2.8. We say that $\mathbf{y}$ is a *corner point* if there exist distinct $i, j \in [2n+2]$ such that $|y_i - 1/2| < \epsilon^{1/4}$ and $|y_j - 1/2| < \epsilon^{1/4}$.

(Notice that $\mathbf{y}$ may be an interior point and a corner point at the same time.)

We still have a hard time handling $\mathbf{y}$'s which are neither an interior point nor a corner point. To mitigate the effect of such $\mathbf{y}$'s we use an equiangle averaging scheme. Namely we consider the set:

$$E^{\epsilon}(\mathbf{y}) = \left\{\mathbf{y}^l = \mathbf{y} + (6l \cdot \epsilon)\,\mathbf{1} : 0 \le l < 1/\sqrt{\epsilon}\right\}$$

where $\mathbf{1}$ denotes the all-ones vector. Notice that since g is λ-Lipschitz for constant λ, $g(\mathbf{y}^l)$ will be approximately the same for all $\mathbf{y}^l \in E^{\epsilon}(\mathbf{y})$.

Fact 6.2.9. If any $\mathbf{y}^l \in E^{\epsilon}(\mathbf{y})$ is not a corner point, then at most one $\mathbf{y}^{l'} \in E^{\epsilon}(\mathbf{y})$ is a boundary point.

Proof. For each dimension, at most one element in $E^{\epsilon}(\mathbf{y})$ can be ϵ-close to the $(1/2)$ facet. Thus if two elements in $E^{\epsilon}(\mathbf{y})$ are boundary points, it must be because of distinct dimensions - and therefore every $\mathbf{y}^l$ is a corner point. □

Given input $\mathbf{y}$, we compute the displacement $g(\cdot)$ separately and in parallel for each $\mathbf{y}^l \in E^{\epsilon}$, and average at the end. Since at most one $\mathbf{y}^l$ is a boundary point, this will incur an error of at most $\sqrt{\epsilon}$.

In the generalized circuit we can construct E^ϵ using $\left(1/\sqrt{\epsilon}\right)$ auxiliary nodes and G_ζ and G_+ gates:

$$\mathbf{x}\left[y_i^l\right] = \min\left\{\mathbf{x}\left[y_i^0\right] + \left(6l\cdot\epsilon\right), 1\right\} \pm 2\epsilon$$

6.2.3 Computing the displacement

For each $\mathbf{y}^l \in E^\epsilon$, we construct a disjoint circuit that approximates the displacement $g\left(\mathbf{y}^l\right)$. In the description of the circuit below we omit the index l.

Lemma 6.2.10. *The circuit below $O\left(\sqrt{\epsilon}\right)$-approximately simulates the computation of the Hirsch et al displacement:*

1. *Whenever $\left(\mathbf{x}\left[y_i\right]\right)_{i\in[2n+2]}$ is an interior point,*

$$\mathbf{x}\left[g_i^+\right] - \mathbf{x}\left[g_i^-\right] = g_i\left(\mathbf{x}\left[\mathbf{y}\right]\right) \pm O\left(\epsilon^{1/4}\right)$$

2. *Furthermore, whenever $\left(\mathbf{x}\left[y_i\right]\right)_{i\in[2n+2]}$ is a corner point,*

$$\mathbf{x}\left[g_{2n+2}^+\right] - \mathbf{x}\left[g_{2n+2}^-\right] = \delta \pm O\left(\sqrt{\epsilon}\right)$$

and $\forall i < 2n+2$:

$$\mathbf{x}\left[g_i^+\right] - \mathbf{x}\left[g_i^-\right] = 0 \pm O\left(\sqrt{\epsilon}\right)$$

Proof. We construct the circuit in five stages: (1) given $\mathbf{y}$, we extract $\mathbf{b}$, that is the binary representation of the corresponding subcube in $\{0,1\}^{2n+2}$; (2) we then compute whether $\mathbf{b}$ belongs to a path in H, and if so which are the previous and next vertices; (3) we compute the centers of the coordinate systems corresponding to the entrance and exit facets, and label them $\mathbf{z}^{in}$ and $\mathbf{z}^{out}$; (4) we project $\mathbf{y}$ to each facet, and transform this projection to the local polar coordinate systems - $\left(r^{in}, \mathbf{p}^{in}\right)$; and (5) finally, we use all the information above to compute the displacement $\mathbf{g} = g\left(\mathbf{y}\right)$.

The correctness of Lemma 6.2.10 follows from Claims 6.2.11-6.2.17.

Extract $\mathbf{b} \in \{0,1\}^{2n+2}$

Our first step is to extract the binary vector $\mathbf{b}$ which represents the subcube to which $\mathbf{y}$ belongs. In other words we want b_i to be the indicator of $y_i < 1/2$. We do that by adding the following gadgets: $G_\zeta\left(1/2 \parallel c_{1/2}\right)$ and, for each i, $G_<\left(\mid y_i, c_{1/2} \mid b_i\right)$. Observe that now

$$\mathbf{x}\left[b_i\right] \quad = \quad \begin{cases} 0 \pm \epsilon & \mathbf{x}\left[y_i\right] < \mathbf{x}\left[c_{1/2}\right] - \epsilon \\ 1 \pm \epsilon & \mathbf{x}\left[y_i\right] > \mathbf{x}\left[c_{1/2}\right] + \epsilon \end{cases}$$

Claim 6.2.11. If $\mathbf{x}\left[\mathbf{y}\right]$ is an interior point, $\mathbf{x}\left[\mathbf{b}\right]$ is the correct representation (up to ϵ error) of the corresponding bits in $\{0,1\}^{2n+2}$.

Neighbors in H

Given $\mathbf{x}[\mathbf{b}]$ we can construct, using $G_\wedge$'s and $G_\neg$'s and a polynomial number of unused nodes, the circuits S^H and P^H that give the next and previous vertex visited by our collection of paths, H. The output of each circuit is represented by $2n+2$ unused nodes $\{P_i^H(\mathbf{b})\}$ and $\{S_i^H(\mathbf{b})\}$.

Recall that H is defined in $\{0,1\}^{2n+1}$, so the last input bit is simply ignored (inside the picture it is always 0); the last output bit is used as follows. Our convention is that starting points and end points correspond to $P^H(\mathbf{b}) = \mathbf{b}$ and $S^H(\mathbf{b}) = \mathbf{b}$, respectively, and likewise for points that do not belong to any path. An exception to this is the $\mathbf{0}$ starting point, which will correspond to $P^H(\mathbf{0}) = (0^{2n+1}; 1)$: This is in accordance with the Hirsch et al construction, where the home subcube is constructed as if it continues a path from the subcube above it.

Claim 6.2.12. If $\mathbf{x}[\mathbf{b}]$ is an ϵ-approximate binary vector, i.e. $\mathbf{x}[\mathbf{b}] \in ([0,\epsilon] \cup [1-\epsilon, 1])^{2n+2}$, then $\mathbf{x}[P^H(\mathbf{b})]$ and $\mathbf{x}[S^H(\mathbf{b})]$ correctly represent (up to ϵ error) the previous vertex and next vertex in H.

Entrance and exit facets

Let $b_i^{+in} = b_i \wedge \neg P_i^H(\mathbf{b})$, i.e. b_i^{+in} is 1 if the path enters the current subcube via the positive i-th direction; define b_i^{-in} analogously. Let b_i^{in} denote the OR of b_i^{+in} and b_i^{-in}.

The center of the entrance facet is constructed via G_ζ, $G_{\times\zeta}$, G_+, and G_- according to the formula:

$$z_i^{in} = \begin{cases} 1/2 - h/2 & b_i^{in} = 0 \text{ AND } b_i = 0 \\ 1/2 + h/2 & b_i^{in} = 0 \text{ AND } b_i = 1 \\ 1/2 & b_i^{in} = 1 \end{cases}$$

Construct $\mathbf{z}^{out}$ analogously.

Notice that if we know on which coordinate the path enters, in $\{0,1\}^{2n+2}$ it has only one possible direction; in the Hirsch et al hypercube this corresponds to always entering from the center (i.e. from the $y_i = 1/2$ facet). Also, if $\mathbf{b}$ corresponds to a non-trivial starting point $\mathbf{b}^{in} = \mathbf{0}$ and $\mathbf{z}^{in}$ is simply the center of the subcube (and similarly for $\mathbf{b}^{out}, \mathbf{z}^{out}$ when $\mathbf{b}$ is an end point).

Claim 6.2.13. If $\mathbf{x}[\mathbf{b}]$, $\mathbf{x}[P^H(\mathbf{b})]$, and $\mathbf{x}[S^H(\mathbf{b})]$ are ϵ-approximate binary vectors, then $\mathbf{x}[\mathbf{z}^{in}]$ and $\mathbf{x}[\mathbf{z}^{out}]$ are $O(\epsilon)$-approximations to the centers of the entrance facet and exit facets, respectively.

Max-norm polar coordinates

We are now ready to compute the local max-norm polar coordinates of the projections of $\mathbf{y}$ on the entrance and exit facets.

The max-norm radius is given by

$$r^{in} = \max_{i:b_i^{in}=0} \left| z_i^{in} - y_i \right|$$

Finding the maximum of a (length $2n+1$) vector requires some care when on each gate we can incur a constant error, the details are described in Algorithm 4.

The direction (max-norm) unit-vector, $\mathbf{p}$, is given by

$$p_i^{in} = \left(z_i^{in} - y_i \right) / r^{in}$$

Division is computed using DIVIDE, introducing an error of $O\left(\sqrt{\epsilon}/r^{in}\right)$; this approximation suffices because for $r^{in} < h/8$, we multiply p_i^{in} by r^{in} when we interpolate. Also, we will use two nodes for each p_i^{in} to represent the positive and negative values. We do the same for $(r^{out}, \mathbf{p}^{out})$.

Claim 6.2.14. If $\mathbf{x}[\mathbf{y}]$ is an interior point, then $\mathbf{x}[r^{in}]$ and $\mathbf{x}[r^{out}]$ are $O\left(\sqrt{\epsilon}\right)$-approximations to the distances of $\mathbf{x}[\mathbf{y}]$ from $\mathbf{x}[\mathbf{z}^{in}]$ and $\mathbf{x}[\mathbf{z}^{out}]$, respectively. Furthermore, $\mathbf{x}[\mathbf{p}^{in}]$ and $\mathbf{x}[\mathbf{p}^{out}]$ are $O\left(\sqrt{\epsilon}/\mathbf{x}[r^{in}]\right)$- and $O\left(\sqrt{\epsilon}/\mathbf{x}[r^{out}]\right)$- to the unit-length vectors that point from $\mathbf{x}[\mathbf{y}]$ in the directions of $\mathbf{x}[\mathbf{z}^{in}]$ and $\mathbf{x}[\mathbf{z}^{out}]$, respectively.

The final displacement

Given $\mathbf{p}^{in}$ and $\mathbf{b}^{in}$, we can compute $\mathbf{g}^{in}$ for the special values of r^{in}. Recall that

$$\mathbf{g}^{in}\left(\left\langle r^{in}, \mathbf{p}^{in} \right\rangle\right) = \begin{cases} \delta\left(\mathbf{b}^{+in} - \mathbf{b}^{-in}\right) & r^{in} = 0 \\ -\delta \mathbf{p}^{in} & r^{in} = h/8 \\ \delta\left(\mathbf{b}^{-in} - \mathbf{b}^{+in}\right) & r^{in} = h/4 \\ \delta \boldsymbol{\xi}_{2n+2} & r^{in} = h/2 \end{cases}$$

We use Algorithm 5 to interpolate for intermediate values of r^{in}. We also need to interpolate between $\mathbf{g}^{in}$ and $\mathbf{g}^{out}$. The ratio at which we interpolate is exactly the ratio between the distance of $\mathbf{y}$ from the entrance and exit facets. We label the positive and negative components of this last interpolation $\mathbf{g}^{(interior)+}$ and $\mathbf{g}^{(interior)-}$, respectively.

When $\mathbf{y}$ is close to both facets, the interpolation may be inaccurate; however, in this case it is a corner point. (We remark that this seems to be the only part

of the proof which requires us to set the threshold for a corner point and the final error at $\Theta(\epsilon^{1/4})$ rather than $\Theta(\sqrt{\epsilon})$; this issue may be avoidable by a more careful construction of Algorithms 3 and 5.)

Claim 6.2.15. If $\mathbf{x}[\mathbf{y}]$ is an interior point of an intermediate subcube in the tube, and it is not a corner point, then $(\mathbf{x}[\mathbf{g}^{(interior)+}] - \mathbf{x}[\mathbf{g}^{(interior)-}])$ is a $O(\epsilon^{1/4})$-approximation of the Hirsch et al displacement $g(\mathbf{x}[\mathbf{y}])$.

Corner points We must ensure that if $\mathbf{y}$ is a corner point, we set $\mathbf{g}^{(corner)+} = \delta\boldsymbol{\xi}_{2n+2}$ and $\mathbf{g}^{(corner)-} = \mathbf{0}$: check over all pairs of coordinates whether $|\mathbf{x}[y_i] - 1/2| < 2\epsilon^{1/4}$ and $|\mathbf{x}[y_j] - 1/2| < 2\epsilon^{1/4}$. Let z be the variable representing the OR of those $\binom{2n+2}{2}$ indicators. Interpolate (e.g. using Algorithm 5) between the $(\mathbf{g}^{(interior)+}, \mathbf{g}^{(interior)-})$ we computed earlier and $\delta\boldsymbol{\xi}_{2n+2}$ with weights z and $\neg z$. Label the result of the interpolation $(\mathbf{g}^{(tube)+}, \mathbf{g}^{(tube)-})$.

We remark that whenever z is ambiguous, i.e. the second smallest $|\mathbf{x}[y_i] - 1/2|$ is very close to $2\epsilon^{1/4}$, then we cannot predict the value of $\mathbf{x}[z]$; it can take any value in $[0,1]$. Nevertheless, in this case $\mathbf{x}[\mathbf{y}]$ is not a corner point, thus for most $\mathbf{y}^l \in E^\epsilon$, $\mathbf{x}[\mathbf{y}^l]$ is an interior point. This means that by Claim 6.2.15, we would compute the (approximately) correct interior displacement $(\mathbf{x}[\mathbf{g}^{(interior)+}] - \mathbf{x}[\mathbf{g}^{(interior)-}])$. Since $\mathbf{x}[\mathbf{y}]$ is close to a corner point, this value is very close to $\delta\boldsymbol{\xi}_{2n+2} = (\mathbf{x}[\mathbf{g}^{(corner)+}] - \mathbf{x}[\mathbf{g}^{(corner)-}])$. Therefore, although we don't know the value of $\mathbf{x}[z]$, we use it to interpolate between two (approximately) equal vectors - so the result is guaranteed to be (approximately) correct regardless of the value of $\mathbf{x}[z]$.

Claim 6.2.16. If $\mathbf{x}[\mathbf{y}]$ is a corner point, then $(\mathbf{x}[\mathbf{g}^{(tube)+}] - \mathbf{x}[\mathbf{g}^{(tube)-}])$ is a $O(\sqrt{\epsilon})$-approximation of $\delta\boldsymbol{\xi}_{2n+2}$, and thus also a $O(\epsilon^{1/4})$-approximation of the Hirsch et al displacement $g(\mathbf{x}[\mathbf{y}])$. Furthermore, Claim 6.2.15 continues to hold for $(\mathbf{g}^{(tube)+}, \mathbf{g}^{(tube)-})$.

Start/end subcubes and subcubes outside the tube For start/end subcubes (except the home subcube) we use a slightly different (Cartesian) interpolation that yields a fixed point in the center of the subcube, and a displacement of $\delta\boldsymbol{\xi}_{2n+2}$ on all facets but the exit/entrance facet, respectively. For subcubes in the picture but outside the tube, we again set $\mathbf{g} = \delta\boldsymbol{\xi}_{2n+2}$.

Notice that we can infer the type of subcube from the following two-bits vector:

$$T = \left(\vee_i b_i^{in}, \vee_i b_i^{out}\right)$$

If $T = (0,0)$, the subcube is outside the tube; when $T = (0,1)$, we are at a start subcube, while $T = (1,0)$ corresponds to an end subcube; $T = (1,1)$ is an intermediate subcube in the tube. Finally, interpolate between the displacement for each type of subcube using T and $\neg T$; label the result of the interpolation $(\mathbf{g}^+, \mathbf{g}^-)$.

Claim 6.2.17. If $\mathbf{x}[\mathbf{y}]$ is either an interior point or a corner point, of any subcube in the slice, then $(\mathbf{x}[\mathbf{g}^+] - \mathbf{x}[\mathbf{g}^-])$ is an $O(\epsilon^{1/4})$-approximation of the Hirsch et al displacement $g(\mathbf{x}[\mathbf{y}])$.

□

6.2.4 Summing the displacement vectors

We are now ready to average over the displacement vectors we computed for each $\mathbf{y}^l$. Using $G_{\times\zeta}$ and G_+ we have that

$$\mathbf{x}[g_i^+] = \sum_{l=1}^{1/\sqrt{\epsilon}} \left(\sqrt{\epsilon}\mathbf{x}\left[g_i^{l+}\right]\right) \pm O\left(\sqrt{\epsilon}\right) \quad \text{and} \quad \mathbf{x}[g_i^-] = \sum_{l=1}^{1/\sqrt{\epsilon}} \left(\sqrt{\epsilon}\mathbf{x}\left[g_i^{l-}\right]\right) \pm O\left(\sqrt{\epsilon}\right)$$

Lemma 6.2.18. *For every input* $\mathbf{x}[\mathbf{y}]$ *and every* $i \in [2n+2]$*,*

$$\mathbf{x}[g_i^+] - \mathbf{x}[g_i^-] = g_i(\mathbf{x}[\mathbf{y}]) \pm O\left(\epsilon^{1/4}\right)$$

Proof. By Fact 6.2.9, every $\mathbf{y}^l \in E^\epsilon$, except at most one, is either an interior point or a corner point. By Lemma 6.2.10, for all those $\mathbf{y}^l$, $(\mathbf{x}[g_i^{+l}] - \mathbf{x}[g_i^{-l}])$ is at most $O(\epsilon^{1/4})$-far from the right displacement. The single point which is neither an interior point nor a corner point increases the error by at most $O\left(\sqrt{\epsilon}\right)$, as does the summation above. Finally, because g is λ-Lipschitz for constant λ, the error added due to the distance between the $\mathbf{y}^l$'s is again at most $O\left(\sqrt{\epsilon}\right)$. □

6.2.5 Closing the loop

Finally, for each $i \in [2n+2]$: $G_+(|\, y_i^1, g_i^+ \mid y_i')$, $G_+(|\, y_i', g_i^- \mid y_i'')$ and $G_=(|\, y_i'' \mid y_i^1)$.

6.3 GCRCUIT with Fan-out 2

In the previous section, we proved that ϵ-GCIRCUIT is PPAD-complete for some constant $\epsilon > 0$. Each generalized circuit gate has fan-in at most 2, which would eventually correspond to a bound on the *incoming* degree of each player in the graphical game. In order to bound the total degree (as well as for the application to A-CEEI), we need to also bound fan-out of each gate.

Proof of Theorem 5.0.5. We present a black-box reduction that converts a general ϵ'-GCIRCUIT instance to an instance of ϵ-GCIRCUIT with fan-out 2, for $\epsilon' = \Theta\left(\sqrt{\epsilon}\right)$. Daskalakis et al [DGP09] bound the fan-out of the generalized circuit by introducing

Algorithm 6 REAL2UNARY$\left(\mid a \mid b_1, \ldots b_{4/\epsilon'}\right)$

1. $G_{=}\left(\mid a \mid c_0\right)$
2. **for** each $k \in [4/\epsilon']$:
   a) $G_{=}\left(\mid c_{k-1} \mid c_k\right)$
   # The c_k's are simply copies of a, ensuring that each gate has fan-out at most 2:
   $$\forall k \;\; \mathbf{x}[c_k] = \mathbf{x}[a] \pm (k+1)\,\epsilon$$
   b) $G_{\zeta}\left(k\epsilon'/4 \parallel \zeta_k\right)$
   c) $G_{<}\left(\mid \zeta_k, c_k \mid b_k\right)$
   # The vector (b_k) is the unary representation of a:
   $$\forall i \;\; \left(\max_{k:\mathrm{X}[b_k]>\epsilon} k\epsilon'/4\right) = \mathbf{x}[a] \pm \epsilon'/2$$

a binary tree of $G_{=}$ gates. Unfortunately, this blows up the error: each $G_{=}$ gate introduces an additive error of ϵ, so the increase is proportional to the depth of the tree, i.e. $\Theta(\epsilon \cdot \log n)$. While this was acceptable for [DGP09] who used an exponentially small ϵ, it clearly fails in our setting.

We overcome this obstacle by resorting to logical gates ($G_{<}$, $G_{\vee}$, $G_{\wedge}$, and in particular $G_{\neg}$). Recall that the logical gates are at most ϵ'-far from the correct Boolean value. Therefore, concatenating multiple logical gates does not amplify the error. In particular, if any logical gate has a large fan-out, we can distribute its output using a binary tree of $G_{\neg}$ gates (we use trees of even depth). When the gate is arithmetic (G_{ζ}, $G_{\times\zeta}$, $G_{=}$, G_{+}, or G_{-}), we convert its output to unary representation over $\Theta(\epsilon')$ logical gates (Algorithm 6). Then, we copy the unary representation using trees of $G_{\neg}$ gates. Finally, we use $G_{\times\zeta}$ and G_{+} gates to convert each copy of the unary representation back to a real value in $[0,1]$ (Algorithm 7).

It is interesting to note that for constant ϵ and ϵ', the unary representation has constant size, so the number of new gates is proportional to the original fan-out (i.e. the number of leaves of the binary tree that copies the unary representation). In particular this reduction increases the size of the circuit by a factor of $\Theta(1/\epsilon')$.

□

Algorithm 7 UNARY2REAL$\big(\mid b_1, \ldots b_{4/\epsilon'} \mid a'\big)$

1. $G_{\zeta}\,(0 \parallel d_0)$
2. **for** each $k \in [4/\epsilon']$:
   a) $G_{\times\zeta}\,(\epsilon'/4 \mid b_k \mid c_k)$
   # The sum of the c_k's is approximately a:
   $$\sum \mathbf{x}\,[c_k] = \mathbf{x}\,[a] \pm (\epsilon'/2 + \epsilon \cdot 4/\epsilon')$$
   b) $G_{+}\,(\mid d_{k-1}, c_k \mid d_k)$
3. $G_{=}\,(\mid d_k \mid a')$
   # a' approximately recovers the original a:
   $$\mathbf{x}\,[a'] = \mathbf{x}\,[a] \pm \epsilon'$$

Chapter 7

Many-player games

In this chapter we prove our hardness results for games with a large number of players. We begin in Section 7.1 with our result for the more restricted class of graphical and polymatrix games, and then (Section 7.2) prove a stronger hardness of approximation for the more general class of succinct games. The two results follow relatively easily from Theorems 5.0.5 and 4.1.1, respectively.

7.0.1 Related works: tractable special cases

For games with many players and a constant number of strategies, PTAS were given for the special cases of anonymous games by Daskalakis and Papadimitriou [DP15] and polymatrix games on a tree by Barman et al. [BLP15]. Finally, let us return to the more general class of succinct n-player games, and mention an approximation algorithm due to Goldberg and Roth [GR16]; their algorithm runs in exponential time, but uses only a polynomial number of oracle queries.

7.1 Graphical, polymatrix games

Theorem (Theorem 5.0.2 restated)**.** *There exists a constant $\epsilon > 0$, such that given a degree 3, bipartite, polymatrix game where each player has two actions, finding an ϵ-approximate Nash equilibrium is* PPAD*-complete.*

The proof proceeds in two steps. First, we reduce to the problem of finding an ϵ-Well Supported Nash Equilibrium in a polymatrix, (degree 3, bipartite) graphical game. This reduction is implicit in Daskalakis et al [DGP09][1]. In the second step we

[1] In [DGP09], polymatrix games are called *games with additive utility functions.*

use the fact that our graphical game have a constant degree to reduce to the more lenient notion ϵ-approximate Nash equilibrium.

7.1.1 Hardness of Well Supported Nash Equilibrium

Proposition (Essentially [DGP09])**.** 2ϵ-GCIRCUIT *with fan-out* 2 *is polynomial-time reducible to finding an* ϵ*-Well Supported Nash Equilibrium in a graphical (bipartite, degree* 3*), polymatrix binary action game.*

The reduction closely follows the lines of [DGP09]. We replace each gate in the generalized circuit with a *subgame* over a constant number of players. Each subgame has one or more designated *input players*, a single *output player*, and potentially some *auxiliary players*. The utility of the input players does not depend on the subgame at all. The guarantee is that in any ϵ-WSNE of the subgame, the mixed strategy profile $O(\epsilon)$-approximately implements the gate, i.e. the probability that the output player assigns to strategy 1 is (approximately) equal to the gate function applied to the corresponding probabilities for the input players.

Finally, we concatenate all the subgames together, i.e. we identify between the output player of one gadget and an input player of (at most 2) other gadgets. Any ϵ-WSNE in the resulting large game $O(\epsilon)$-approximately implements all the gates in the generalized circuit - so we can extract valid assignments to all the generalized circuit lines from the probability that each output player assigns to strategy 1.

In the gadget constructions below, every input and output player has degree 1 (i.e. it only interacts with one other player, in the sense of Definition 5.0.1). Each player participates in at most one gadget as an output player, and at most 2 gadgets as an input player (since the circuit has maximum fan-out 2). Therefore, the total degree of each player is ≤ 3.

Below we construct the gadgets for each of the gates. By the above argument, Lemmata 7.1.1-7.1.4 together imply Proposition 7.1.1.

$G_{\times\zeta}$, G_{ζ}, G_{+}, $G_{=}$ gadgets

Lemma 7.1.1 ($G_{\times\zeta}$, G_{ζ}, G_{+}, $G_{=}$ gadgets)**.** *Let* $\alpha, \beta > 0$. *Let* $v_{\text{IN1}}, v_{\text{IN2}}, v_{\text{OUT}}, w$ *be players in a graphical game, and suppose that the payoffs of* v_{OUT} *and* w *are as follows.*

Payoff for v_{out}:

	w *plays* 0	w *plays* 1
v_2 *plays* 0	0	1
v_2 *plays* 1	1	0

Payoffs for w:

game with v_{in1}:

	v_{IN1} *plays* 0	v_{IN1} *plays* 1
w *plays* 0	0	α
w *plays* 1	0	0

game with v_{in2}:

	v_{IN2} *plays* 0	v_{IN2} *plays* 1
w *plays* 0	0	β
w *plays* 1	0	0

game with v_{out}:

	v_{OUT} *plays* 0	v_{OUT} *plays* 1
w *plays* 0	0	0
w *plays* 1	0	1

Then, in every ϵ-WSNE

$$\mathbf{p}\left[v_{\text{OUT}}\right] = \min\left(\alpha\mathbf{p}\left[v_{\text{IN1}}\right] + \beta\mathbf{p}\left[v_{\text{IN2}}\right], 1\right) \pm \epsilon, \tag{7.1}$$

where $\mathbf{p}[u]$ denotes the probability that u assigns to strategy 1.

Notice that each of $G_{\times\zeta}$, G_{ζ}, G_{+}, $G_{=}$ can be implemented using (7.1): for $G_{\times\zeta}$ set $\alpha = \zeta, \beta = 0$; for G_{ζ} set the same α, β, and modify w's game with v_{IN1} as if the latter always plays strategy 1; etc.

Proof of Lemma 7.1.1. Assume by contradiction that (7.1) is violated:

- If $\mathbf{p}\left[v_{\text{OUT}}\right] > \alpha\mathbf{p}\left[v_{\text{IN1}}\right] + \beta\mathbf{p}\left[v_{\text{IN2}}\right] + \epsilon$, then in every ϵ-WSNE $\mathbf{p}[w] = 1$. But then v_{OUT} is incentivized to play strategy 0, which contradicts $\mathbf{p}\left[v_{\text{OUT}}\right] > \epsilon$.
- Similarly, if $\mathbf{p}\left[v_{\text{OUT}}\right] < \alpha\mathbf{p}\left[v_{\text{IN1}}\right] + \beta\mathbf{p}\left[v_{\text{IN2}}\right] - \epsilon$, then in every ϵ-WSNE $\mathbf{p}[w] = 0$. Therefore, $\mathbf{p}\left[v_{\text{OUT}}\right] \geq 1 - \epsilon$.

□

$G_{\neg}$, G_{-} gadgets

Lemma 7.1.2 ($G_{\neg}$, G_{-} gadgets)**.** *Let $v_{\text{IN1}}, v_{\text{IN2}}, v_{\text{OUT}}, w$ be players in a graphical game, and suppose that the payoffs of v_{OUT} and w are as follows.*

Payoff for v_{out}:

	w *plays* 0	w *plays* 1
v_2 *plays* 0	0	1
v_2 *plays* 1	1	0

Payoffs for w:

game with v_{in1}:

	v_{IN1} *plays* 0	v_{IN1} *plays* 1
w *plays* 0	0	1
w *plays* 1	0	0

game with v_{in2}:

	v_{IN2} *plays* 0	v_{IN2} *plays* 1
w *plays* 0	1	0
w *plays* 1	0	0

game with v_{out}:

	v_{OUT} *plays* 0	v_{OUT} *plays* 1
w *plays* 0	0	0
w *plays* 1	0	1

Then, in every ϵ-WSNE

$$\mathbf{p}[v_{\text{OUT}}] = \max(\mathbf{p}[v_{\text{IN1}}] - \mathbf{p}[v_{\text{IN2}}], 0) \pm \epsilon, \tag{7.2}$$

where $\mathbf{p}[u]$ denotes the probability that u assigns to strategy 1.

Notice that each of $G_{\neg}$, G_{-} can be implemented using (7.2).

Proof of Lemma 7.1.2. Assume by contradiction that (7.2) is violated:

- If $\mathbf{p}[v_{\text{OUT}}] > \mathbf{p}[v_{\text{IN1}}] - \mathbf{p}[v_{\text{IN2}}] + \epsilon$, then in every ϵ-WSNE $\mathbf{p}[w] = 1$. But then v_{OUT} is incentivized to play strategy 0, which contradicts $\mathbf{p}[v_{\text{OUT}}] > \epsilon$.
- Similarly, if $\mathbf{p}[v_{\text{OUT}}] < \mathbf{p}[v_{\text{IN1}}] - \mathbf{p}[v_{\text{IN2}}] - \epsilon$, then in every ϵ-WSNE $\mathbf{p}[w] = 0$. Therefore, $\mathbf{p}[v_{\text{OUT}}] \geq 1 - \epsilon$.

□

$G_{<}$ gadget

Lemma 7.1.3 ($G_{<}$ gadget)**.** *Let $v_{\text{IN1}}, v_{\text{IN2}}, v_{\text{OUT}}, w$ be players in a graphical game, and suppose that the payoffs of v_{OUT} and w are as follows.*

Payoff for v_{out}:

	w *plays* 0	w *plays* 1
v_2 *plays* 0	0	1
v_2 *plays* 1	1	0

Payoffs for w:

game with v_{in1}:

	v_{IN1} *plays* 0	v_{IN1} *plays* 1
w *plays* 1	0	0
w *plays* 0	1	1

game with v_{in2}:

	v_{IN2} *plays* 0	v_{IN2} *plays* 1
w *plays* 0	1	0
w *plays* 1	0	0

Then, in every ϵ-WSNE

$$\mathbf{p}[v_{\text{OUT}}] = \begin{cases} \geq 1-\epsilon & \mathbf{p}[v_{\text{IN1}}] < \mathbf{p}[v_{\text{IN2}}] - \epsilon \\ \leq \epsilon & \mathbf{p}[v_{\text{IN1}}] > \mathbf{p}[v_{\text{IN2}}] + \epsilon \end{cases}, \tag{7.3}$$

where $\mathbf{p}[u]$ denotes the probability that u assigns to strategy 1.

Proof. The payoff for player w for action 0 is $1+\mathbf{p}[v_{\text{IN1}}]-\mathbf{p}[v_{\text{IN2}}]$, whereas for action 1 it is $1-\mathbf{p}[v_{\text{IN1}}]+\mathbf{p}[v_{\text{IN2}}]$. Therefore, in every ϵ-WSNE

$$\mathbf{p}[w] = \begin{cases} \leq \epsilon & \mathbf{p}[v_{\text{IN1}}] < \mathbf{p}[v_{\text{IN2}}] - \epsilon \\ \geq 1-\epsilon & \mathbf{p}[v_{\text{IN1}}] > \mathbf{p}[v_{\text{IN2}}] + \epsilon \end{cases}$$

(7.3) follows immediately because player v_{OUT} want to play the opposite of w. □

$G_\vee$, $G_\wedge$ gadgets

Lemma 7.1.4 ($G_\vee$, $G_\wedge$ gadgets)**.** *Let $v_{\text{IN1}}, v_{\text{IN2}}, v_{\text{OUT}}, w$ be players in a graphical game, and suppose that the payoffs of v_{OUT} and w are as follows.*

Payoff for v_{out}:

	w *plays* 0	w *plays* 1
v_2 *plays* 0	0	1
v_2 *plays* 1	1	0

Payoffs for w:

game with v_{in1}:

	v_{IN1} *plays* 0	v_{IN1} *plays* 1
w *plays* 0	1	0
w *plays* 1/2	0	1

game with v_{in2}:

	v_{IN2} *plays* 0	v_{IN2} *plays* 1
w *plays* 0	1	0
w *plays* 1/2	0	1

Then, in every ϵ-WSNE

$$\mathbf{p}[v_{\text{OUT}}] = \begin{cases} \geq 1-\epsilon & \mathbf{p}[v_{\text{IN1}}] + \mathbf{p}[v_{\text{IN2}}] < 2/3 - \epsilon \\ \leq \epsilon & \mathbf{p}[v_{\text{IN1}}] + \mathbf{p}[v_{\text{IN2}}] > 2/3 + \epsilon \end{cases}, \tag{7.4}$$

where $\mathbf{p}[u]$ denotes the probability that u assigns to strategy 1.

Notice that (7.4) implements an $G_\vee$ (OR) gadget when the input values are approximately Boolean. A $G_\wedge$ (AND) gadget can be implemented analogously. by interchanging the 1 and 1/2 values in w's payoff matrices.

Proof of Lemma 7.1.4. The payoff for player w for action 0 is $\mathbf{p}[v_{\text{IN1}}] + \mathbf{p}[v_{\text{IN2}}]$, whereas for action 1 it is $(1-\mathbf{p}[v_{\text{IN1}}])/2+(1-\mathbf{p}[v_{\text{IN2}}])/2 = 1-(\mathbf{p}[v_{\text{IN1}}]+\mathbf{p}[v_{\text{IN2}}])/2$. In particular, the latter payoff from action 1 is larger whenever $\mathbf{p}[v_{\text{IN1}}] + \mathbf{p}[v_{\text{IN2}}] < 2/3$.

Therefore, in every ϵ-WSNE

$$\mathbf{p}[w] = \begin{cases} \leq \epsilon & \mathbf{p}[v_{\text{IN1}}] + \mathbf{p}[v_{\text{IN2}}] < 2/3 - \epsilon \\ \geq 1-\epsilon & \mathbf{p}[v_{\text{IN1}}] + \mathbf{p}[v_{\text{IN2}}] > 2/3 + \epsilon \end{cases}$$

(7.4) follows immediately because player v_{OUT} want to play the opposite of w. □

7.1.2 From $\sqrt{\epsilon}$-WSNE to $\Theta(\epsilon)$-ANE

The reduction above shows hardness for the slightly stronger notion (therefore weaker hardness) of ϵ-WSNE. Daskalakis et al [DGP09] show a reduction from $\sqrt{\epsilon}$-WSNE to $\Theta(\epsilon)$-ANE for games with a constant number of players. It is easy to see that the same reduction holds for graphical games with constant degree. We sketch the proof below.

Lemma 7.1.5. *(Essentially [DGP09]) Given an ϵ-ANE of a graphical game with payoffs in $[0,1]$ and incoming degree d_{in}, we can construct in polynomial time a $\sqrt{\epsilon}\cdot\left(\sqrt{\epsilon}+1+4d_{in}\right)$-WSNE.*

Proof. Let V be the set of players, where each $v \in V$ has utility U^v and action set A^v. Let $\mathbf{x} = (x_a^v) \in \Delta(\times_v A^v)$ be an ϵ-ANE. Let $U_a^v(\mathbf{x}^{-v})$ denote the expected payoff for playing a when all other players play according to $\mathbf{x}$. Note that $U_a^v(\mathbf{x}^{-v}) = U_a^v(\mathbf{x}^{\mathcal{N}_{in}(v)})$ depends only on the distributions of the players in the incoming neighborhood of v, which we denote $\mathcal{N}_{in}(v)$. Finally, let $U_{\max}^v(\mathbf{x}^{-v}) = \max_{a\in A^v} U_a^v(\mathbf{x}^{-v})$.

Let $k = k(\epsilon) > 0$ be some large number do be specified later. We construct our new approximate equilibrium by taking, for each player only the strategies that are within ϵk of the optimum:

$$\hat{x}_a^v = \begin{cases} \frac{x_a^v}{1-z^v} & U_a^v(\mathbf{x}^{-v}) \geq U_{\max}^v(\mathbf{x}^{-v}) - \epsilon k \\ 0 & \text{otherwise} \end{cases}$$

where z^v is the total probability that player v assigns to strategies that are more than ϵk away from the optimum.

The above division is well-defined because for $k > 1$, z^v is bounded away from 1. Moreover, the following claim from [DGP09] formalizes the intuition that when k is sufficiently large, the total weight on actions removed is small, so $\hat{\mathbf{x}}^v$ is close to $\mathbf{x}^v$:

Claim 7.1.6. (Claim 6 of [DGP09])

$$\forall v \in V \quad \sum_{a \in A^v} |\hat{x}_a^v - x_a^v| \leq \frac{2}{k-1}$$

Now, the total change to the expected payoff to player v for each action a, is bounded by the total change in mixed strategies of its *incoming neighbors*:

$$\begin{aligned} |U_a^v(\mathbf{x}^{-v}) - U_a^v(\hat{\mathbf{x}}^{-v})| &= \left|U_a^v\left(\mathbf{x}^{\mathcal{N}_{in}(v)}\right) - U_a^v\left(\hat{\mathbf{x}}^{\mathcal{N}_{in}(v)}\right)\right| \\ &\leq \sum_{\mathbf{a} \in A^{\mathcal{N}(v)}} \left|\hat{x}_{\mathbf{a}}^{\mathcal{N}_{in}(v)} - x_{\mathbf{a}}^{\mathcal{N}_{in}(v)}\right| \leq \sum_{w \in \mathcal{N}(v)} \sum_{a \in A^w} |\hat{x}_a^w - x_a^w| \leq \frac{2d_{in}}{k-1} \end{aligned}$$

It follows that $\hat{x}_a^v$ is a $\left(k\epsilon + \frac{4d_{in}}{k-1}\right)$-WSNE:

$$U_a^v(\hat{\mathbf{x}}^{-v}) \geq U_a^v(\mathbf{x}^{-v}) - \frac{2d_{in}}{k-1} \geq U_{\max}^v(\mathbf{x}^{-v}) - \epsilon k - \frac{2d_{in}}{k-1} \geq U_{\max}^v(\hat{\mathbf{x}}^{-v}) - \epsilon k - \frac{4d_{in}}{k-1}$$

Finally, take $k = 1 + 1/\sqrt{\epsilon}$ to get that

$$k\epsilon + \frac{4d_{in}}{k-1} \leq \sqrt{\epsilon} \cdot \left(\sqrt{\epsilon} + 1 + 4d_{in}\right)$$

□

7.2 Succinct games

Theorem (Theorem 5.0.4 restated)**.** *There exist constants $\epsilon, \delta > 0$, such that finding an (ϵ, δ)-WeakNash is* PPAD*-hard for succinct multiplayer games where each player has a constant number of actions.*

Proof. Let f be the hard function guaranteed by Theorem 4.1.1. We construct a game with two groups of n players each. The action set of each player corresponds to $\{0, 1/k, 2/k \ldots, 1\}$ for a sufficiently large constant $k > 0$. We denote the choice of strategies for the first group $\mathbf{a} \triangleq (a_1 \ldots a_n)$, and $\mathbf{b} \triangleq (b_1, \ldots b_n)$ for the second group.

Each player (A, i) in the first group attempts to imitate the behavior of the corresponding player in the second group. Her utility is given by

$$u_i(a_i, b_i) \triangleq -|a_i - b_i|^2.$$

The second group players attempt to imitate the value of f, when applied to the vector of actions taken by all the players in the first group. The utility of the j-th player, (B,j), is

$$v_j(b_j,\mathbf{a}) \triangleq -\left|f_j(\mathbf{a})-b_j\right|^2,$$

where f_j denotes the j-th output of f.

Observe that the expected utility for (A,i) is given by:

$$\mathsf{E}\left[u_i(a_i,b_i)\right] = -\left|a_i-\mathsf{E}(b_i)\right|^2 - \mathrm{Var}(b_i).$$

For any value of $\mathsf{E}(b_i)$, player (A,i) has an action $\alpha_i \in [\mathsf{E}(b_i)-1/2k, \mathsf{E}(b_i)+1/2k]$. Her utility when playing α_i is lower bounded by:

$$\mathsf{E}\left[u_i(\alpha_i,b_i)\right] \geq -\frac{1}{4k^2} - \mathrm{Var}(b_i).$$

On the other hand, for any $\widehat{a_i} \notin \{\alpha_i, \alpha_i+1/k\}$,

$$\mathsf{E}\left[u_i(\widehat{a_i},b_i)\right] \leq -\frac{1}{k^2} - \mathrm{Var}(b_i).$$

Therefore in every $(\delta,\delta/k^2)$-WeakNash, it holds for all but a 2δ-fraction of i's, that player (A,i) assigns probability at least $(1-O(\delta))$ to strategies $\{\alpha_i,\alpha_i+1/k\}$. In particular, with high probability over the randomness of the players, it holds that all but an $O(\delta)$-fraction of the players play one of those strategies. Therefore with high probability

$$\left\|\mathbf{a}-\mathsf{E}(\mathbf{b})\right\|_2 = O\left(\sqrt{\delta}+1/k\right). \tag{7.5}$$

Similarly, player (B,j) (the j-th player in the second group) has an action $\beta_j \in [\mathsf{E}(f_j(\mathbf{a}))-1/2k, \mathsf{E}(f_j(\mathbf{a}))+1/2k]$. Therefore in every $(\delta,\delta/k^2)$-WeakNash, it holds for all but a 2δ-fraction of j's, that player (B,j) assigns probability at least $(1-O(\delta))$ to strategies $\{\beta_j,\beta_j+1/k\}$. In particular, with high probability over the randomness of the players, it holds that all but an $O(\delta)$-fraction of the players play one of those strategies. Therefore with high probability

$$\left\|\mathbf{b}-\mathsf{E}(f(\mathbf{a}))\right\|_2 = O\left(\sqrt{\delta}+1/k\right).$$

Now, by the Lipschitz condition on f, whenever (7.5) holds,

$$\left\|f(\mathbf{a})-f(\mathsf{E}(\mathbf{b}))\right\|_2 = O\left(\sqrt{\delta}+1/k\right).$$

Therefore $\mathsf{E}(\mathbf{a})$ and $\mathsf{E}(\mathbf{b})$ are both solutions to the $\Theta\left(\sqrt{\delta}+1/k\right)$-approximate Euclidean BROUWER instance. □

7.2.1 Binary actions

Corollary 7.2.1. *There exist constants $\epsilon, \delta > 0$, such that finding an (ϵ, δ)-WeakNash is* PPAD*-hard for succinct multiplayer games where each player has two actions.*

Proof. We replace each player (A, i) (respectively, (B, j)) in the proof of Theorem 5.0.4 with $k+1$ players, denoted: $(A, i, 0), (A, i, 1/k), \ldots, (A, i, 1)$. Each new player has two actions: $\{+, -\}$. Given a $(k+1)$-tuple of actions for players $\{(A, i, \cdot)\}$, we define the *realized value* $r_i \in [0, 1]$ to be

$$r_i \triangleq \max\{x \text{ s.t. } (A, i, x) \text{ plays action } +\}.$$

Let the realized value q_j for players $(B, j, 0), \ldots, (B, j, 1)$ be defined analogously.

We let player (A, i, x)'s utility be:

$$\begin{aligned} u_{i,x}(+, \mathbf{b}) &\triangleq -\left|\left(x + \frac{1}{k}\right) - q_i\right|^2 \\ u_{i,x}(-, \mathbf{b}) &\triangleq -\left|\left(x - \frac{1}{k}\right) - q_i\right|^2. \end{aligned}$$

Similarly, for player (B, j, y), we have

$$u_{j,y}(\pm, \mathbf{a}) \triangleq -\left|\left(y \pm \frac{1}{k}\right) - f_i(\mathbf{r})\right|^2,$$

where $\mathbf{r} \triangleq (r_1, \ldots, r_n)$ is the vector of realized values on the first group.

For any (A, i, x), we have

$$\mathsf{E}[(u_{i,x}(\pm, \mathbf{b}))] = -\left|\left(x \pm \frac{1}{k}\right) - \mathsf{E}[q_i]\right|^2 - \mathrm{Var}[q_i].$$

Subtracting her two possible payoffs, we have

$$\begin{aligned} \mathsf{E}[(u_{i,x}(+, \mathbf{b})) - (u_{i,x}(-, \mathbf{b}))] &= \left|(x - \mathsf{E}[q_i]) - \frac{1}{k}\right|^2 - \left|(x - \mathsf{E}[q_i]) + \frac{1}{k}\right|^2 \\ &= -\frac{4}{k}(x - \mathsf{E}[q_i]). \end{aligned}$$

In particular, if $\mathsf{E}[q_i] > x + k\sqrt{\epsilon}$, player (A, i, x) must assign probability at most $\sqrt{\epsilon}$ to action $\{-\}$ in order to play ϵ-optimally. For any i such that all players $(A, i, \cdot)$ use ϵ-optimal mixed strategies, we have that

$$|\mathsf{E}[q_i] - \mathsf{E}[r_i]| = O\left(k\sqrt{\epsilon} + 1/k\right). \tag{7.6}$$

In any (ϵ, δ)-WeakNash, for all but a $2\delta k$-fraction of i's it holds that all players $(A, i, \cdot)$ use ϵ-optimal mixed strategies; thus (7.6) holds for all but a $2\delta k$-fraction of i's. Therefore,

$$\|\mathsf{E}[\mathbf{q}] - \mathsf{E}[\mathbf{r}]\|_2 = O\left(\sqrt{\delta k} + k\sqrt{\epsilon} + 1/k\right).$$

By the Lipschitz condition on f, the latter also implies,

$$\|f(\mathsf{E}[\mathbf{q}]) - f(\mathsf{E}[\mathbf{r}])\|_2 = O\left(\sqrt{\delta k} + k\sqrt{\epsilon} + 1/k\right).$$

Similarly, for every j such that all players $(B, j, \cdot)$ use ϵ-optimal mixed strategies, we have that

$$|\mathsf{E}[q_j] - f_j(\mathsf{E}[\mathbf{r}])| = O\left(k\sqrt{\epsilon} + 1/k\right).$$

Thus in any (ϵ, δ)-WeakNash,

$$\|\mathsf{E}[\mathbf{q}] - f(\mathsf{E}[\mathbf{r}])\|_2 = O\left(\sqrt{\delta k} + k\sqrt{\epsilon} + 1/k\right).$$

Therefore $\mathsf{E}(\mathbf{q})$ and $\mathsf{E}(\mathbf{r})$ are both solutions to the $\Theta\left(\sqrt{\delta} + 1/k\right)$-approximate Euclidean BROUWER instance. □

Chapter 8

Bayesian Nash equilibrium

In this short chapter we supplement our hardness of approximation of Nash equilibrium in many-player games and many-actions game, by showing a different reason of complexity: players' uncertainty. Specifically, we prove that finding an approximate Bayesian Nash equilibrium in two-player games with incomplete information is PPAD-complete, even when the players have only a constant number of actions.

In a game with incomplete information, each player i has a type t_i known only to her, and the players' types $\mathbf{t} = (t_1, t_2)$ are drawn from a joint distribution which is known to everyone. The payoff for player i is a function $u_i(\mathbf{a}, t_i)$ of her own type and all the players' actions.

Definition 8.0.1. (e.g. [SSW04])

In a *Bayesian Nash equilibrium*, for every player i and every type t_i, the mixed strategy $x_i(t_i)$ must be a best response in expectation over the other players' types and actions:

$$\mathbb{E}_{\mathbf{t}|t_i}\left[\mathbb{E}_{\mathbf{a}\sim(x_i(t_i),x_{-i}(t_{-i}))}\left[u_i(\mathbf{a};t_i)\right]\right] \geq \max_{x_i'(t_i)\in\Delta A_i} \mathbb{E}_{\mathbf{t}|t_i}\left[\mathbb{E}_{\mathbf{a}\sim(x_i'(t_i),x_{-i}(t_{-i}))}\left[u_i(\mathbf{a};t_i)\right]\right].$$

Similarly, in an *ϵ-approximate Bayesian Nash equilibrium*, for every player i and every type t_i, the mixed strategy $x_i(t_i)$ must be an ϵ-best response in expectation over the other players' types and actions:

$$\mathbb{E}_{\mathbf{t}|t_i}\left[\mathbb{E}_{\mathbf{a}\sim(x_i(t_i),x_{-i}(t_{-i}))}\left[u_i(\mathbf{a};t_i)\right]\right] \geq \max_{x_i'(t_i)\in\Delta A_i} \mathbb{E}_{\mathbf{t}|t_i}\left[\mathbb{E}_{\mathbf{a}\sim(x_i'(t_i),x_{-i}(t_{-i}))}\left[u_i(\mathbf{a};t_i)\right]\right] - \epsilon.$$

Before we prove our main corollary for incomplete information games, it is helpful to prove the following slightly weaker statement, for two players with many strategies.

Lemma 8.0.2. *There exists a constant $\epsilon > 0$, such that given a two-player game with incomplete information where each player has n actions, finding an ϵ-approximate Bayesian Nash equilibrium is* PPAD*-complete.*

Proof. We reduce from a bipartite polymatrix game, and let the typeset for each of the two players in the incomplete information game correspond to one side of the bipartite graph. The utility of player i on edge (i, j) of the polymatrix game depends on her identity (i), as well as the identity (j) of the vertex on the other side of that edge. We use the types of the incomplete information game, to encode i. We encode j using the strategies of the second player in the incomplete information game.

In more detail, consider a bipartite polymatrix game for which it is PPAD-hard to compute a 4ϵ-approximate Nash equilibrium. Use an affine transformation to change all the payoffs from $[0, 1]$ to $[1/2, 1]$. It is PPAD-hard to find a 2ϵ-approximate Nash equilibrium in the transformed game.

We now construct the two-player incomplete information game: As we hinted before, we let the typeset of each player correspond to the vertices on one side of the bipartite graphical game. Player i has $|T_i|$ types and $2|T_i|$ strategies, where each strategy corresponds to a pair of vertex and strategy for that vertex. If a player plays a strategy whose vertex does not match her type, her payoff is 0. Therefore in every ϵ-approximate Bayesian Nash equilibrium, every player, on every type, plays the two strategies that correspond to her type with probability at least $1 - 2\epsilon$.

Let the joint distribution over types be as follows: pick a random edge in the bipartite graph, and assign the types corresponding to its vertices. Whenever both players play strategies that match their respective types, their payoffs are the payoffs in the (transformed) bimatrix game on that edge. In every ϵ-approximate Bayesian Nash equilibrium, every player, on every type, plays a mixed strategy which is ϵ-best response. Since the other player plays strategies that correspond to the correct vertex with probability at least $1 - 2\epsilon$, the same mixture must be a 2ϵ-best response for the vertex player in the bipartite game. □

In order to prove the main corollary, we need to reduce the number of actions in the above construction. Observe that we don't need each player to choose an action that uniquely identifies her type. Rather, it suffices to specify which neighbor of the other player's vertex is chosen. This can be done concisely via a coloring of the vertices such that every pair of vertices of distance exactly two have different colors; i.e. a coloring of the square of the polymatrix game's graph. The squared graph has degree $3 \cdot (3 - 1) = 6$, and therefore we can efficiently find a 7-coloring. It suffices for each player to specify one of 7 colors, together with one of 2 strategies for the vertex with that color. Therefore we can encode this choice using only 14 strategies.

Corollary 8.0.3. *There exists a constant $\epsilon > 0$, such that given a two-player game with incomplete information where each player has 14 actions, finding an ϵ-approximate Bayesian Nash equilibrium is* PPAD*-complete.*

Chapter 9

Market Equilibrium

In this chapter we formally introduce our result for non-monotone markets, discuss it, and compare it to the original result of Chen, Paparas, and Yannakakis [CPY13]. A sketch of the proof appears in the next section.

Intuitively, a market is monotone if increasing the price of some good, while fixing the rest of the prices, never increases the excess demand for that good. Formally, we have the following definition by Chen et al:

Definition 9.0.1. ([CPY13]) Let M be a market with $k \geq 2$ goods. We say that M is *non-monotone* at price vector $\boldsymbol{\pi}$ if there exist $c > 0$, and a good g_1 such that:

- the excess demand $Z_1(y_1, \ldots y_k)$ is a continuous function (rather than correspondence) over $\mathbf{y} \in B(\boldsymbol{\pi}, c)$;
- $Z_1(\boldsymbol{\pi}) > 0$;
- the partial derivative of $\partial Z_1/\partial y_1$ exists and is continuous over $B(\boldsymbol{\pi}, c)$;
- and $\partial Z_1/\partial y_1(\boldsymbol{\pi}) > 0$.

We say that a market M is *non-monotone* if there exists such a rational price vector $\boldsymbol{\pi} \geq \mathbf{0}$, and $Z_1(\boldsymbol{\pi})$ is *moderately computable*; i.e. for any $\gamma > 0$, $Z_1(\boldsymbol{\pi})$ can be approximated to within γ in time polynomial in $1/\gamma$.

In general we want to talk about non-monotone families of utility functions, i.e. ones that support non-monotone markets. Formally,

Definition 9.0.2. ([CPY13]) We say that a family $\mathcal{U}$ of utility functions is *non-monotone* if:

- $\mathcal{U}$ is countable;
- if $u\colon [0,\infty)^k \to \mathbb{R}$ is in $\mathcal{U}$, then so is $u'(x_1, \ldots x_m) = a \cdot u(x_{l_1}/b_1, \ldots x_{l_k}/b_k)$ for any indices $l_1, \ldots, l_k \in [m]$ and positive (rational) $a, b_1, \ldots, b_k$;
- $u(\boldsymbol{x}) = g^*(x_i)$ is in $\mathcal{U}$ for some strictly increasing $g\colon [0,\infty) \to \mathbb{R}$; and
- there exists a non-monotone market $M_{\mathcal{U}}$ with utilities from $\mathcal{U}$.

We need to include one more definition: that of ϵ-tight market equilibrium.

Definition 9.0.3. A price vector $\boldsymbol{\pi}$ is an ϵ-tight approximate market equilibrium of M if there exists a $\mathbf{z} \in Z(\boldsymbol{\pi})$ (the excess demand at $\boldsymbol{\pi}$) such that for every good j, $|z_j| \leq \epsilon W_j$, where W_j is the sum of the endowments of good j.

Our main result for non-monotone markets equilibria is now formally defined:

Theorem 9.0.4 (Non-monotone markets)**.** *Let $\mathcal{U}$ be any non-monotone family of utility functions. There exists a constant $\epsilon_{\mathcal{U}} > 0$ such that given a market M where the utility of each trader is either linear or taken from $\mathcal{U}$, finding an $\epsilon_{\mathcal{U}}$-tight approximate market equilibrium is* PPAD*-hard.*

9.0.1 Why are non-monotone markets hard?

Before delving into the details of the construction, we attempt to reach some intuition: why should we expect equilibrium computation to be hard in non-monotone markets? Probably the most intuitive algorithm for finding market equilibrium is via tatonnement: raise the prices of over-demanded goods, and decrease the prices of under-demanded goods. For many markets, the tatonnement process is also computationally efficient [CMV05]. One obvious problem is that when the market is non-monotone, the tatonnement step actually takes us further away from equilibrium. However, the non-monotonicity is only local: if we set the (relative) price of the non-monotone good high enough, even the most enthusiastic traders can only afford a small amount.

The "real" reason that tatonnement fails to converge efficiently for non-monotone markets is a little more subtle. What happens when the demand for the non-monotone good g increases by a factor of $(1+\delta)$ for some small δ? The tatonnement increases the price of g, which further increases the demand. Eventually, the price is high enough, and the demand is reduced; but due to the non-monotonicity we may have to increase the price by larger factor, i.e. $(1+\delta')$ for $\delta' > \delta$. Now, another trader with a positive endowment of g has increased her spending budget by $(1+\delta')$,

further increasing the demand for yet another good (by a larger factor). Thus a small change in the demand for one good may cause a much larger change in the demand for another good. Exploiting this "butterfly effect" lies at the heart of Chen et al's construction.

9.0.2 High-level structure of the proof

Our reduction from polymatrix games to non-monotone markets closely follows the footsteps of [CPY13]. To gain some intuition, consider two goods g_{2i-1} and g_{2i} for each player i, corresponding to her two available strategies (soon each of those goods will become a subset of goods). Let $\pi(g_{2i-1})$ and $\pi(g_{2i})$ denote their corresponding prices; those prices correspond to the probabilities that player i assigns to her respective strategies. For every $i, j \in [n]$, we add a trader who is interested in selling g_{2i-1} and buying g_{2j-1} (and similarly for $(2i, 2j-1)$, $(2i-1, 2j)$, and $(2i, 2j)$). This trader has an endowment of g_{2i-1} that is proportional to $P_{2i-1,2j-1}$, the utility of player j in the bimatrix game with player i, when they both play the first strategy. Qualitatively, if the price $\pi(g_{2i-1})$ is high (player i assigns a high probability to her first strategy), and $P_{2i-1,2j-1}$ is high (player j receives a high utility from playing her first strategy to i's first strategy), then the demand for good g_{2j-1} is high - implying a high price in every approximate market equilibrium (i.e. player j indeed assigns a high probability to her first strategy).

In order to turn this qualitative intuition into a proof we use a more complex construction. The main difficulty comes from the need to amplify the effect of a higher income for one trader on the incomes of other traders. To this end we consider, for each $i \in [n]$, two sequences of goods: $g_{2i-1} = g_{2i-1,0}, g_{2i-1,1}, \ldots, g_{2i-1,4t} = h_{2i-1}$ and $g_{2i} = g_{2i,0}, g_{2i,1}, \ldots, g_{2i,4t} = h_{2i}$. The trader mentioned in the previous paragraph actually owns $P_{2i-1,2j-1}$ units of good h_{2i-1}; she is still interested in good g_{2j-1}. Now we construct (Lemma 9.1.7) a chain of gadgets that use copies of the non-monotone markets in $\mathcal{U}$ to amplify the small gap guaranteed between $\pi(g_{2j-1})$ and $\pi(g_{2j})$ to a larger gap between $\pi(h_{2j-1})$ and $\pi(h_{2j})$.

Additionally, we want to bound the range that these prices may take. In Lemma 9.1.4 we use a price regulating gadget [Che+09; VY11] to control the relative prices of $\pi(g_{2i-1,j})$ and $\pi(g_{2i,j})$. In Lemma 9.1.6 we show that the sums $\pi_{i,j} = \pi(g_{2i-1,j}) + \pi(g_{2i,j})$ are approximately equal. Finally, in Section 9.1.5 we combine these lemmata to formalize a quantitative version of the qualitative intuition described above.

9.0.3 Adaptations for constant factor inapproximability

As mentioned in the introduction, Theorem 9.0.4 has a weakness in comparison to the results of Chen et al [CPY13]: it only applies to *tight* approximate market equilibrium.

Maintaining the constant hardness of approximation through most of [CPY13]'s proof is rather straightforward, but there are a few hurdles along the way. To understand the first obstacle, we must consider a subtle issue of normalization. Chen et al normalize the bimatrix game between every pair of players to have an average value of 1/2. While this does not change the absolute utility gained from any deviation, the relative utility from deviation is now divided by a factor of $\Theta(n)$. In contrast, in Theorem 5.0.2 we prove hardness for a constant ϵ when normalizing with respect to a constant degree (3), i.e. each player participates in only a constant number of bimatrix games. We overcome this difficulty by using a different normalization: only edges (i.e. bimatrix games) that belong to the game graph will have an average utility of 1/2, while the utilities on other edges remains 0. Since we proved hardness for a degree 3 graphical game, the normalization only costs us a constant factor.

More serious complications arise when trying to prove [CPY13]'s Lemma 3[1] for a constant ϵ. This lemma says that certain prices (in fact, these are sums of prices), denoted $\pi_{i,0}$ for $i \in [n]$, are approximately equal. A key step in the proof of [CPY13] is to show, roughly, that in every $\epsilon(n)$-approximate market equilibrium,

$$\pi_{i,0} \geq \frac{1}{n} \sum_{j \in [n]} \pi_{j,0} - O(\epsilon(n))$$

When $\epsilon(n)$ is polynomially small, this immediately implies that $\min_{i \in [n]} \pi_{i,0}$ is within $O(\epsilon(n))$ of the average, and therefore it must also be that $\max_{i \in [n]} \pi_{i,0}$ is within $O(n \cdot \epsilon(n))$ of the average. When taking a larger $\epsilon(n)$, this reasoning breaks. The first modification we make to overcome this obstacle, is to require $\epsilon(n)$-*tight* approximate market equilibrium. This gives a two-sided bound:

$$\left| \pi_{i,0} - \frac{1}{n} \sum_{j \in [n]} \pi_{j,0} \right| = O(\epsilon(n)) \tag{9.1}$$

A second issue that arises in the same inequality, is that with our new normalization, which depends on the game graph G, we can only prove that $\pi_{i,0}$ approximates the values of its neighbors, denoted $\mathcal{N}_G(i)$. In other words, (9.1) becomes

[1] Compare with our Lemma 9.1.6. The reader may also want to refer to Lemma 6 in the full version of [CPY13].

$$\left|\pi_{i,0} - \frac{1}{|\mathcal{N}_G(i)|}\sum_{j\in\mathcal{N}_G(i)}\pi_{j,0}\right| = O(\epsilon(n)) \tag{9.2}$$

In order to relate the value of $\pi_{i,0}$ to the corresponding values of the neighboring vertices, $\pi_{j,0}$'s, we consider T consecutive applications of (9.2): $\pi_{i,0}$ is $O(T\cdot\epsilon)$-close to the expectation over $\pi_{j,0}$ where j is taken from the distribution of a T-steps random walk on G starting from i. For example, if G is a constant degree expander, the random walk converges in $O(\log n)$ steps, yielding a $(1/\log n)$-inapproximability result.

Achieving constant hardness

Finally, in order to achieve hardness for a constant ϵ, we want a graph with constant mixing time - and this clearly cannot be done with a constant degree[2]. Instead, in Section 9.1.2 we construct a normalized game whose graph has a constant mixing time, each vertex has degree $O(\sqrt{n})$, and yet approximating Nash equilibrium is hard for a constant ϵ. In short, we take n copies of the original n-player game (our new game has n^2 players). For any pair of players that play a (non-trivial) bimatrix game in the original game, we have a copy of the same bimatrix game between all $\binom{n}{2}$ pairs of their respective copies. We also add a trivial bimatrix game between every pair of players that belong to the same copy of the original game. In Section 9.1.2 we argue that these newly added trivial edges are only a constant fraction of all edges in the new game graph, yet this graph has a constant mixing time.

9.1 Non-monotone markets: proof of inapproximability

In this section we prove our main inapproximability result for non-monotone markets (Theorem 9.0.4)

9.1.1 Normalized polymatrix games

We identify n-player, 2-strategy polymatrix graphical games with $2n\times 2n$ matrices by letting the (i,j)-th 2×2 block correspond to the payoff matrix of player j in the bimatrix game with player i.

[2]In fact, it seems that a graph where the random walks starting from any pair of neighbors converge in constant time would suffice. We do not know whether such graphs can be constructed with constant degree.

Given a game $\mathcal{G}$, let $\mathbf{P}'$ be the $2n \times 2n$ induced payoff matrix. We normalize $\mathbf{P}'$ as follows:

$$P_{2i,2j-1} = \begin{cases} 1/(2\Delta) + \left(P'_{2i,2j-1} - P'_{2i,2j}\right)/(2\Delta) & (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{9.3}$$

where E is the edge set for the graphical game[3] and Δ is the maximum degree. We define $P_{2i,2j}$, $P_{2i-1,2j-1}$, $P_{2i-1,2j}$ analogously. Notice that $\mathbf{P}$ and $\mathbf{P}'$ have the same ϵ-WSNE, up to the normalization by the degree Δ. In particular finding an (ϵ/Δ)-WSNE in $\mathbf{P}$ continues to be PPAD-complete.

Observe that in this formulation, finding an ϵ-WSNE is equivalent to finding a vector $\mathbf{x} \in [0,1]^{2n}$ s.t. $x_{2i-1} + x_{2i} = 1$ and

$$\begin{aligned} \mathbf{x}^\top \cdot \mathbf{P}_{2i-1} > \mathbf{x}^\top \cdot \mathbf{P}_{2i} + \epsilon \quad &\Longrightarrow \quad x_{2i} = 0 \\ \mathbf{x}^\top \cdot \mathbf{P}_{2i-1} < \mathbf{x}^\top \cdot \mathbf{P}_{2i} - \epsilon \quad &\Longrightarrow \quad x_{2i-1} = 0 \end{aligned}$$

9.1.2 Games on graphs with a constant mixing time

Given the correspondence defined above between n-player games and $2n \times 2n$ matrices, we see that the structure of the game graph plays a non-trivial role in the construction. In particular, adding trivial edges between vertices, i.e. adding zero-utility bimatrix games between players, has no affect on the utility of the players, but changes the corresponding normalized matrix. For reasons that will become clear much later in the proof, we would like our game graph to have a constant mixing time.

Indeed, a trivial candidate with very fast mixing is the complete graph. However, such a blowup in the degree would dilute our inapproximability factor in the *normalized game*. Instead, we consider n copies $(v^1, \ldots, v^n)$ of each player $v \in V$ in the original game. If players u and v play a bimatrix game $\mathcal{G}_{u,v}$ in the original game $\mathcal{G}$, then for every $i, j \in [n]$, we construct the same bimatrix game $\mathcal{G}_{u,v}$ between u^i and v^j. Our game graph now consists of n^2 vertices, each with degree[4] $3n$. Finally, within each copy V^i, we add trivial edges between all the vertices not otherwise connected (including self-loops). Normalize this game using (9.3). We use $\overline{\mathcal{G}}$ to denote the new game and $\overline{G}$ for the new game graph; we henceforth let $\Delta = 4n - 3$ denote the degree. In the next two lemmata we show that this game satisfies the two properties we need: finding an ϵ-WSNE of $\overline{\mathcal{G}}$ is PPAD-complete, and the mixing time of $\overline{G}$ is constant.

[3]Notice that this definition allows self-loops in the game graph.

[4]Theorem 5.0.2 promises a graphical game of degree at most 3. It is not hard to extend to a 3-regular graph game with only a constant loss in the approximation factor.

Lemma 9.1.1. *Given an ϵ-WSNE in $\overline{\mathcal{G}}$, we can (efficiently) construct a $(4\epsilon/3)$-WSNE for $\mathcal{G}$.*

Proof. For each player v, we take the average of the mixed strategies of $v^1, \ldots, v^n$. The utility of v is the same as the average of utilities of $v^1, \ldots, v^n$, and if v has a $(4\epsilon/3)$-improving deviation, then at least one of the copies v^i has an ϵ-improving deviation. (The $(4/3)$ factor comes from the change in the degree.) □

Lemma 9.1.2. *Let $\pi_{v^i,T}$ be the distribution of a random walk on $\overline{G}$ after T steps, starting from v^i, and let π^* be the uniform distribution on the vertices of $\overline{G}$. Then*

$$\left\|\pi_{v^i,T} - \pi^*\right\|_1 \leq \left(\frac{1}{4}\right)^{T/2} + \left(\frac{3}{4}\right)^{T/2} = 2^{-O(T)}$$

Proof. At each step of the random walk, there is a constant probability (greater than 3/4) of walking on a non-trivial edge, which takes us to another (uniformly random) copy of the original game; thereafter the copy of the game remains uniformly random. Similarly, at each step there is a constant probability (greater than 1/4) of moving to a vertex within the same copy (again, uniformly random). Thus conditioned on having walked on a non-trivial edge, and then on an edge within the same copy, the distribution is uniform. Since all vertices have the same degree, this is also the stationary distribution, and we never leave it. □

For simplicity, in the following we redefine n to be the size of $\overline{G}$ (and hence $\Delta \approx 4\sqrt{n}$).

9.1.3 Construction

Let N be a sufficiently large constant, and let $t = \log N$. Note that N depends on the parameters of the non-monotone market in $\mathcal{U}$, but not on the size n of our construction. We use the notation $O_N(\cdot)$ to denote the asymptotic behavior when N goes to infinity (but arbitrarily slower than n). We prove that it is PPAD-hard to find an η-tight approximate market clearing equilibrium for $\eta = N^{-8}\epsilon$, where ϵ is the inapproximability factor from Lemma 9.1.1.

For each vertex $i \in [n]$ we construct a series of $4t+1$ gadgets $\mathcal{R}_{i,j}$, for $j \in [0:4t]$. Each gadget is composed of:

Main goods $g_{2i-1,j}$ and $g_{2i,j}$ are the main goods in the reduction. They are used to encode the weights assigned to strategies x_{2i-1} and x_{2i}, respectively.

Non-monotone gadget For each $j \in [4t]$, we include additional goods $s_{i,j,3}, \ldots, s_{i,j,k}$ and a non-monotone gadget

$$\mathbf{nm}\left(\mu, \gamma, g_{2i-1,j}, g_{2i,j}, s_{i,j,3}, \ldots, s_{i,j,k}\right)$$

This means that we scale the non-monotone market guaranteed to exist in $\mathcal{U}$ according to parameters γ and μ such that when all the prices are approximately the same, the excess demand of $g_{2i-1,j}$ increases linearly with its price. Formally, we have the following lemma by Chen et al.

Lemma 9.1.3. *(Lemma 3.1 of [CPY13]) There exist two (not necessarily rational) positive constants c and d with the following property. Given any $\gamma > 0$, one can build a market M_γ with utilities from $\mathcal{U}$ and goods $g_{2i-1,j}, g_{2i,j}, s_{i,j,3}, \ldots, s_{i,j,k}$ in polynomial time in $1/\gamma$ such that:*

Let $f_{\gamma,\mu}(x)$ denote the excess demand function of $g_{2i-1,j}$ when the price of g_{2i-1} is $1+x$, and the prices of all other $k-1$ goods are $1-x$. Then $f_{\gamma,\mu}$ is well defined over $[-c,c]$ with $|f_{\gamma,\mu}(0)| \le \mu\gamma$ and its derivative $f'_{\gamma,\mu}(0) = d > 0$. For any $x \in [-c,c]$, $f_{\gamma,\mu}(x)$ also satisfies

$$|f_{\gamma,\mu}(x) - f_{\gamma,\mu}(0) - \mu d x| \le |\mu x / D|, \text{ where } D = \max\{20, 20/d\}.$$

Finally, we would like to set $\mu = \Delta/d$; in particular, this would imply that $f_{\gamma,\mu}(x) \approx \Delta x$. However, as mentioned above, d may be irrational. Instead, let d^* be a positive rational constant that satisfies

$$1 - 1/D \le d^* \cdot d \le 1$$

We set the parameters $\mu = d^*\Delta$ and $\gamma = 1/N^6$.

Price regulating gadget For $j \in [4t]$, we include a price regulating gadget

$$\mathbf{pr}\left(\tau, \alpha_j, g_{2i-1,j}, g_{2i,j}, s_{i,j,3}, \ldots s_{i,j,k}\right),$$

whereas for $j = 0$, we don't have goods $s_{i,0,3}, \ldots s_{i,0,k}$, and simply include the gadget

$$\mathbf{pr}\left(\tau, \alpha_0, g_{2i-1,0}, g_{2i,0}\right).$$

The parameters are set to $\tau = N\Delta$ and $\alpha_i = 2^i/N^5$. Notice that $\alpha_0 = N^{-5}$ and $\alpha_{4t} = N^{-1} = \beta$.

This gadget ensures that in any approximate equilibrium, the price ratio $\pi(g_{2i-1,j})/\pi(g_{2i,j})$ is always in the range $\left[\frac{1-\alpha_j}{1+\alpha_j}, \frac{1+\alpha_j}{1-\alpha_j}\right]$. Furthermore, within each gadget $\mathcal{R}_{i,j}$, the prices of all the goods besides $g_{2i-1,j}$ are exactly equal:

$$\pi(g_{2i}, j) = \pi(s_{i,j,3}) = \cdots = \pi(s_{i,j,k}).$$

More specifically, we have two traders T_1 and T_2 with endowments $(k-1)\tau$ of $g_{2i-1,j}$, for T_1 and τ of each of the other goods for T_2. The utilities are defined as

$$\begin{aligned} u_1 &= (1+\alpha_j)\, x(g_{2i-1,j}) + (1-\alpha_j)\left(x(g_{2i,j}) + \sum_{l=3}^{k} x(s_{i,j,l})\right) \\ u_2 &= (1-\alpha_j)\, x(g_{2i-1,j}) + (1+\alpha_j)\left(x(g_{2i,j}) + \sum_{l=3}^{k} x(s_{i,j,l})\right). \end{aligned}$$

In particular, T_1 and T_2 do not trade whenever $\pi(g_{2i-1,j})/\pi(g_{2i,j}) \in \left(\frac{1-\alpha_j}{1+\alpha_j}, \frac{1+\alpha_j}{1-\alpha_j}\right)$.

Auxiliary goods For $j = 0$, we also include an auxiliary good AUX_i. Its eventual purpose is to disentangle the price of $g_{2i-1,0}$ and $g_{2i,0}$ from the utility that the actions of player i causes to other players.

Single-minded traders graph

We connect the groups of goods ($\mathcal{R}_{i,j}$'s) using the following single-minded traders. We use $(w, g_1 : g_2)$ to denote a trader with endowment w of good g_1 who only wants good g_2. Similarly, we use $(w, g_1, g_2 : g_3)$ to denote a trader who has an endowment w of each of g_1 and g_2, and only wants g_3.

1. For each $i \in [n]$ and $j \in [0 : 4t-1]$, we add two traders from $\mathcal{R}_{i,j}$ to $\mathcal{R}_{i,j+1}$: $(\Delta, g_{2i-1,j} : g_{2i-1,j+1})$ and $(\Delta, g_{2i,j} : g_{2i,j+1})$. These traders help propagate price discrepancies from $g_{i,0}$ to $g_{i,4t}$.

2. Recall that we use g_i as short for $g_{i,0}$ and h_i for $g_{i,4t}$. For each pair $(i,j) \in E$ we add the following four traders: $(\Delta P_{2i-1,2j-1}, h_{2i-1} : g_{2j-1})$, $(\Delta P_{2i,2j-1}, h_{2i} : g_{2j-1})$, $(\Delta P_{2i-1,2j}, h_{2i-1} : g_{2j})$, $(\Delta P_{2i,2j}, h_{2i} : g_{2j})$. Since $\mathbf{P}$ is normalized, we have

$$\Delta P_{2i-1,2j-1} + \Delta P_{2i-1,2j} = \Delta P_{2i,2j-1} + \Delta P_{2i,2j} = 1$$

These traders will enforce the approximate Nash equilibrium.

3. Connect the auxiliary goods: We let

$$r_{2j-1} = 2\Delta - \Delta \sum_{i \in \mathcal{N}(j)} \left(P_{2i-1,2j-1} + P_{2i,2j-1}\right) > 0$$

$$r_{2j} = 2\Delta - \Delta \sum_{i \in \mathcal{N}(j)} \left(P_{2i-1,2j} + P_{2i,2j}\right) > 0$$

note that $r_{2j-1} + r_{2j} = 2\Delta$.

We add the following traders: $\left((1-\beta)\, r_{2j-1}, \text{AUX}_j : g_{2j-1}\right)$, $\left((1-\beta)\, r_{2j}, \text{AUX}_j : g_{2j}\right)$, and $\left((1-\beta)\, \Delta, g_{2j-1}, g_{2j} : \text{AUX}_j\right)$.

Notice that the economy graph is strongly connected (because G is strongly connected); therefore an equilibrium always exists [Max97]. The supplies and demands for each good are summarized in Table 9.1.

9.1.4 Structure of a market equilibrium

We now prove some properties that every η-tight approximate equilibrium $\boldsymbol{\pi}$ must satisfy. Recall that $\eta = N^{-8}\epsilon$, where ϵ is the inapproximability factor for the polymatrix game.

We begin with the application of the price regulating markets:

Lemma 9.1.4. *For every $i \in [n]$ and $j \in [0:4t]$,*

$$\frac{1-\alpha_j}{1+\alpha_j} \leq \frac{\pi\left(g_{2i-1}, j\right)}{\pi\left(g_{2i}, j\right)} \leq \frac{1+\alpha_j}{1-\alpha_j}$$

and

$$\pi\left(g_{2i}, j\right) = \pi\left(s_{i,j,3}\right) = \cdots = \pi\left(s_{i,j,k}\right)$$

Proof. Follows from the construction of the price regulating markets. For more details see the proof[5] of Lemma 6 in the full version of [CPY13], or previous works that use similar gadgets [Che+09; VY11]. □

We henceforth use $\pi_{i,j}$ to denote the sum of the (i,j)-th main goods: $\pi_{i,j} = \pi\left(g_{2i-1,j}\right) + \pi\left(g_{2i,j}\right)$.

Lemma 9.1.5.

[5] In their statement, Chen et al require an ϵ-additively approximate equilibrium, and for a much smaller ϵ. However their proof continues to hold with our parameters.

Table 9.1: Goods and traders

good [total supply]	supplied by	demanded by
g_{2i-1}, g_{2i} $\left[N\Delta\left(1+o_N\left(1\right)\right)\right]$	**pr** trader; $(\Delta, g_{2i-1} : g_{2i-1,1})$, $(\Delta, g_{2i} : g_{2i,1})$; $((1-\beta)\,\Delta, g_{2i-1}, g_{2i} : \mathrm{AUX}_i)$	**pr** traders; $(\Delta P_{2j-1,2i-1}, h_{2j-1} : g_{2i-1})$, ⋮ $(\Delta P_{2j,2i}, h_{2j} : g_{2i})$; $((1-\beta)\,r_{2i-1}, \mathrm{AUX}_i : g_{2i-1})$, $((1-\beta)\,r_{2i}, \mathrm{AUX}_i : g_{2i})$
h_{2i-1} $\left[(k-1)\,N\Delta\left(1+o_N\left(1\right)\right)\right]$	**pr** trader; **nm** traders; $(\Delta P_{2i-1,2j-1}, h_{2i-1} : g_{2j-1})$, $(\Delta P_{2i-1,2j}, h_{2i-1} : g_{2j})$	**pr** traders; **nm** traders; $(\Delta, g_{2i-1,4t-1} : h_{2i-1})$
h_{2i} $\left[N\Delta\left(1+o_N\left(1\right)\right)\right]$	**pr** trader; **nm** traders; $(\Delta P_{2i,2j-1}, h_{2i} : g_{2j-1})$, $(\Delta P_{2i,2j}, h_{2i} : g_{2j})$	**pr** traders; **nm** traders; $(\Delta, g_{2i,4t-1} : h_{2i})$
$g_{2i-1,j}$ $\left[(k-1)\,N\Delta\left(1+o_N\left(1\right)\right)\right]$	**pr** trader; **nm** traders; $(\Delta, g_{2i-1,j} : g_{2i-1,j+1})$	**pr** traders; **nm** traders; $(\Delta, g_{2i-1,j-1} : g_{2i-1,j})$
$g_{2i,j}$ $\left[N\Delta\left(1+o_N\left(1\right)\right)\right]$	**pr** trader; **nm** traders; $(\Delta, g_{2i,j} : g_{2i,j+1})$	**pr** traders; **nm** traders; $(\Delta, g_{2i,j-1} : g_{2i,j})$
$s_{i,j,l}$ $\left[N\Delta\left(1+o_N\left(1\right)\right)\right]$	**pr** trader; **nm** traders;	**pr** traders; **nm** traders;
AUX_i $\left[N\Delta\left(1+o_N\left(1\right)\right)\right]$	$((1-\beta)\,r_{2i-1}, \mathrm{AUX}_i : g_{2i-1})$, $((1-\beta)\,r_{2i}, \mathrm{AUX}_i : g_{2i})$	$((1-\beta)\,\Delta, g_{2i-1}, g_{2i} : \mathrm{AUX}_i)$

$$(1 - O_N(\eta))\,\pi_{i,0}/2 \le \pi(\text{AUX}_i) \le (1 + O_N(\eta))\,\pi_{i,0}/2$$

Proof. The total supply of AUX_i is $2(1-\beta)\Delta$, yet the demand from the single-minded trader $((1-\beta)\Delta, g_{2i-1}, g_{2i} : \text{AUX}_i)$ is $(1-\beta)\Delta\frac{\pi_{i,0}}{\pi(\text{AUX}_i)}$. (For the upper bound we use the fact that π is a *tight* approximate market equilibrium.) □

We are now ready to prove that the cost of every pair of main goods is approximately the same. Let $\delta = N^2\eta$.

Lemma 9.1.6. *Let* $\pi_{\max} = \max_{i,j}\pi_{i,j}$ *and* $\pi_{\min} = \min_{i,j}\pi_{i,j}$, *then*

$$\pi_{\max}/\pi_{\min} \le 1 + O_N(\delta).$$

Proof. The proof of this lemma is the main obstacle which requires the tightness assumption of the market equilibrium, as well as our bound on the mixing time from Lemma 9.1.2.

Recall that by Lemma 9.1.4, the prices of all the goods in each gadget $\mathcal{R}_{i,j}$ are approximately equal. Thus, using our bound on the clearing error, we have that for each $(i,j) \in [n] \times [0:4t]$,

$$|\text{total spent on } \mathcal{R}_{i,j} - \text{total worth of } \mathcal{R}_{i,j}| \le O_N(\eta \cdot kN\Delta)\,\pi_{i,j} = O_N(\eta \cdot N\Delta)\,\pi_{i,j} \tag{9.4}$$

By Walras' Law, the traders within each $\mathcal{R}_{i,j}$ (i.e. the price regulating and non-monotone gadgets) contribute the same to both quantities in (9.4). Similarly, by Lemma 9.1.5, the auxiliary traders contribute $O_N(\eta\Delta\pi_{i,0})$-approximately the same (for $j = 0$). Therefore the money spent on $\mathcal{R}_{i,j}$ by the single minded traders is approximately the same as the total worth of endowments in $\mathcal{R}_{i,j}$ of single minded traders:

- For each $(i,j) \in [n] \times [4t]$, the restriction of (9.4) to the single-minded traders gives
$$|\Delta\pi_{i,j-1} - \Delta\pi_{i,j}| = O_N(\eta \cdot N\Delta)\,\pi_{i,j} \tag{9.5}$$
- Similarly, for each group $\mathcal{R}_{i,0}$, we have
$$\left|\sum_{l \in \mathcal{N}(i)} \pi_{l,4t} - \Delta\pi_{i,0}\right| = O_N(\eta \cdot N\Delta)\,\pi_{i,0} \tag{9.6}$$

Applying (9.5) inductively, we have that for any $i \in [n]$ and for any $j, l \in [0{:}4t]$,

$$|\pi_{i,j} - \pi_{i,l}| = O_N(\eta \cdot Nt)\,\pi_{i,l}$$

Combining, with (9.6) we have,

$$\left|\pi_{i,0} - \frac{1}{\Delta}\sum_{l \in \mathcal{N}_{G^{\mathcal{G}}}(i)} \pi_{l,0}\right| = O(\eta \cdot Nt)\,\pi_{i,0}$$

Thus for each i, $\pi_{i,0}$ is $O(\eta \cdot Nt)$-approximately equal to the average of its neighbors in $\overline{G}$. Repeating this argument T times, we have that $\pi_{i,0}$ is $O_N(T\eta \cdot Nt)$-approximately equal to the expectation over a T-step random walk in $\overline{G}$ starting from i. By Lemma 9.1.2, after $T = O(\log \delta)$ steps the random walk δ-approximately converges to the uniform distribution, and we have

$$\begin{aligned}\left|\pi_{i,0} - \frac{1}{n}\sum_{l \in [n]} \pi_{l,0}\right| &= O_N(T\eta \cdot Nt)\,\frac{1}{n}\sum_{l \in [n]} \pi_{l,0} + \delta \max_{l \in [n]} \pi_{l,0} \\ &= O_N(\delta) \max_{l \in [n]} \pi_{l,0}\end{aligned}$$

□

Finally, we have the following lemma which describes the action of the non-monotone gadgets.

Lemma 9.1.7. *(Lemma 6 of [CPY13])*

$$\frac{1+\alpha_{j-1}}{\pi(g_{2i-1,j-1})} = \frac{1-\alpha_{j-1}}{\pi(g_{2i,j-1})} \implies \frac{1+\alpha_j}{\pi(g_{2i-1,j})} = \frac{1-\alpha_j}{\pi(g_{2i,j})} \text{ and}$$
$$\frac{1-\alpha_{j-1}}{\pi(g_{2i-1,j-1})} = \frac{1+\alpha_{j-1}}{\pi(g_{2i,j-1})} \implies \frac{1-\alpha_j}{\pi(g_{2i-1,j})} = \frac{1+\alpha_j}{\pi(g_{2i,j})}$$

Proof. The demand for $g_{2i-1,j}$ and $g_{2i,j}$ comes from three sources: the single-minded traders, $(\Delta, g_{2i-1,j-1} : g_{2i-1,j})$ and $(\Delta, g_{2i,j-1} : g_{2i,j})$; the non-monotone gadget; and the price regulating gadget. Assume without loss of generality that the first premise holds, i.e. $\frac{1+\alpha_{j-1}}{\pi(g_{2i-1,j-1})} = \frac{1-\alpha_{j-1}}{\pi(g_{2i,j-1})}$. When the prices of $g_{2i-1,j}$ and $g_{2i,j}$ are equal, the demand from $(\Delta, g_{2i-1,j-1} : g_{2i-1,j})$ is larger since she has more income from $g_{2i-1,j-1}$. In order to account for this difference, $\pi(g_{2i-1,j})$ must be higher - but then the demand from the traders in the non-monotone market increases. Thus we further have to increase $\pi(g_{2i-1,j})$, until we reach the threshold of the price regulating traders: $(1+\alpha_j)/(1-\alpha_j)$.

Formally, normalize $\boldsymbol{\pi}$ such that $\pi_{i,j} = \pi\left(g_{2i-1,j}\right) + \pi\left(g_{2i,j}\right) = 2$. Thus by Lemma 9.1.6, $\pi_{i,j-1}$ is also $O_N(\delta)$-close to 2. Let $f(x)$ denote the excess demand from the traders in the non-monotone gadget when $\pi\left(g_{2i-1,j}\right) = 1+x$ and $\pi\left(g_{2i-1,j}\right) = \pi\left(s_{i,j,3}\right) = \cdots = \pi\left(s_{i,j,k}\right) = 1-x$ (recall from Lemma 9.1.4 that the latter prices are always equal to each other). By Lemma 9.1.3, we have that $|f(0)| \le \mu\gamma$, and for all $x \in [-c, c]$:

$$|f(x) - f(0) - \mu dx| \le |\mu x/D|.$$

Now, let $\pi\left(g_{2i-1,j-1}\right) = 1 + y$; notice that by Lemma 9.1.6, $y = \alpha_{j-1} \pm O_N(\delta)$. Let $h(x, y)$ excess demand from all traders besides the two that belong to the price regulating gadget. Then,

$$h(x,y) = f(x) + \frac{\Delta(1+y)}{1+x} - \Delta = f(x) - \frac{\Delta x}{1+x} + \frac{\Delta y}{1+x}$$

For small x, we show that $f(x) \approx \Delta x/(1+x)$. More precisely,

$$\begin{aligned} |f(x) - \Delta x/(1+x)| &\le |f(x) - \mu dx| + |\mu dx - \Delta x| + \Delta\,|x - x/(1+x)| \\ &\le |f(0)| + 2\,|\mu x/D| + 2\Delta x^2 \\ &\le \Delta \cdot \left(d^* \cdot \gamma + 2x/20 + 2x^2\right) \\ &\le \frac{\Delta y}{3} \end{aligned}$$

where the first inequality follows from the triangle inequality; the second follows by application of Lemma 9.1.3 for the first difference and the definitions of μ and d^* for the second; the third inequality applies the Lemma 9.1.3 again; finally the last inequality holds because for sufficiently large constant N, the parameters γ and x are sufficiently small.

Therefore, the excess demand must be balanced by the demand from the price regulating traders, implying that indeed $\frac{1+\alpha_j}{\pi(g_{2i-1,j})} = \frac{1-\alpha_j}{\pi(g_{2i,j})}$. □

9.1.5 From market equilibrium to Nash equilibrium

To complete the proof of Theorem 9.0.4, we must construct an ϵ-WSNE from any η-tight approximate market equilibrium.

For each $i \in [n]$, let $\theta_i = \left(\pi\left(h_{2i-1}\right) + \pi\left(h_{2i}\right)\right)/2$. We define

$$x_{2i-1} = \frac{\pi\left(h_{2i-1}\right)/\theta_i - (1-\beta)}{2\beta} \text{ and } x_{2i} = \frac{\pi\left(h_{2i}\right)/\theta_i - (1-\beta)}{2\beta} \tag{9.7}$$

Observe that $x_{2i-1} + x_{2i} = 1$.

Suppose that

$$\mathbf{x}^\top \cdot \mathbf{P}_1 \geq \mathbf{x}^\top \cdot \mathbf{P}_2 + \epsilon$$

We show that this forces $x_1 = 1$ and $x_2 = 0$; by the discussion in Section 9.1.1 this implies that $\mathbf{x}$ is indeed an ϵ-WSNE.

The following traders spend money on g_1:

1. For each $i \in \mathcal{N}_{\overline{G}}(1)$, there is a $(\Delta P_{i,1}, h_i; g_1)$ trader. The total money these traders spend on g_1 is

$$\sum \Delta P_{i,1} \cdot \pi(h_1) = \sum \Delta P_{i,1} (1 - \beta + 2\beta \cdot x_i) \theta_{\lceil i/2 \rceil}$$

2. For each $i \in \mathcal{N}_{\overline{G}}(2)$, there is a $(\Delta P_{i,2}, h_i; g_2)$ trader. The total money these traders spend on g_2 is

$$\sum \Delta P_{i,2} \cdot \pi(h_2) = \sum \Delta P_{i,2} (1 - \beta + 2\beta \cdot x_i) \theta_{\lceil i/2 \rceil}$$

3. $((1-\beta) r_1, \text{AUX}_1 : g_1)$ and $((1-\beta) r_2, \text{AUX}_1 : g_2)$ traders

Let M_1 be the total amount that these traders spend on g_1. Then

$$M_1 = \sum_{i \in \mathcal{N}_{\overline{G}}(1)} \Delta P_{i,1} (1 - \beta + 2\beta \cdot x_i) \theta_{\lceil i/2 \rceil} + (1-\beta) r_1 \pi(\text{AUX}_1)$$

Normalizing the prices such that $\frac{1}{n} \sum_{i=1}^{n} \theta_i = 1/\Delta$, this means that

$$M_1 \geq 2(1-\beta) + 2\beta \mathbf{x}^\top \cdot \mathbf{P}_1 - O_N(\delta)$$

Similarly,

$$M_2 \leq 2(1-\beta) + 2\beta \mathbf{x}^\top \cdot \mathbf{P}_2 + O_N(\delta)$$

Therefore,

$$M_1 \geq M_2 + 2\beta\epsilon - O_N(\delta) = M_2 + \Theta_N(\beta\epsilon)$$

so the difference between the demands for g_1 and g_2 from these traders is

$$\frac{M_1}{\pi(g_1)} - \frac{M_2}{\pi(g_2)} \geq \frac{M_2 + \Theta_N(\beta\epsilon)}{\pi(g_1)} - \frac{M_2(1+\alpha_0)}{\pi(g_1)(1-\alpha_0)} = \Theta_N(\beta\epsilon)$$

Thus the price regulating traders T_1 and T_2 must have different demands for g_1 and g_2 - but this can only happen when

$$\frac{1+\alpha_0}{\pi(g_1)} = \frac{1-\alpha_0}{\pi(g_2)}$$

Therefore, by consecutive applications of Lemma 9.1.7,

$$\frac{1+\beta}{\pi(h_1)} = \frac{1-\beta}{\pi(h_2)}$$

Finally, by (9.7) this implies that $x_1 = 1$ and $x_2 = 0$.

□

Chapter 10

CourseMatch

University courses have limited capacity, and some are more popular than others. This creates an interesting allocation problem. Imagine that each student has ordered all the possible schedules—bundles of courses—from most desirable to least desirable, and the capacities of the classes are known. What is the best way to allocate seats in courses to students? There are several desiderata for a course allocation mechanism:

Fairness In what sense is the mechanism "fair"?

Efficiency Are seats in courses allocated to the students who want them the most?

Feasibility Are any courses oversubscribed?

Truthfulness Are students motivated to honestly report their preferences to the mechanism?

Computational efficiency Can the allocation be computed from the data in polynomial time?

Competitive Equilibrium from Equal Incomes (CEEI) [Fol67; Var74; TV85] is a venerable mechanism with many attractive properties: In CEEI all agents are allocated the same amount of "funny money", next they declare their preferences, and then a price equilibrium is found that clears the market. The market clearing guarantees Pareto efficiency and feasibility. The mechanism has a strong, albeit technical, *ex post* fairness guarantee that emerges from the notion that agents who miss out on a valuable, competitive item will have extra funny money to spend on other items at equilibrium. Truthfulness is problematic—as usual with market mechanisms—but potential incentives for any individual agent to deviate are mitigated by the large number of agents. However, CEEI only works when the resources to be allocated

are divisible and the utilities are relatively benign. This restriction has both benefits and drawbacks. It ensures computational feasibility, because CEEI can be computed in polynomial time with a linear or convex program, depending on the utilities involved [Var74; Dev+08; Gho+11]; on the other hand, it is easy to construct examples in which a CEEI does not exist when preferences are complex or the resources being allocated are not divisible. Indeed, both issues arise in practice in a variety of allocation problems, including shifts to workers, landing slots to airplanes, and the setting that we focus on here, courses to students [Var74; Bud11].

It was recently shown in [Bud11] that an approximation to a CEEI solution, called A-CEEI, exists even when the resources are indivisible and agent preferences are arbitrarily complex, as required by the course allocation problems one sees in practice. The approximate solution guaranteed to exist is approximately fair (in that the students are given almost the same budget), and approximately Pareto efficient and feasible (in that all courses are filled close to capacity, with the possible exception of courses with more capacity than popularity). This result seems to be wonderful news for the course allocation problem. However, there is a catch: Budish's proof is non-constructive as it relies on Kakutani's fixed-point theorem.

A heuristic search algorithm for solving A-CEEI was introduced in [OSB10]. The algorithm resembles a traditional tâtonnement process, in which the prices of courses that are oversubscribed are increased and the prices of courses that are undersubscribed are decreased. A modified version of this algorithm that guarantees courses are not oversubscribed is currently used by the Wharton School (University of Pennsylvania) to assign their MBA students to courses [Bud+14]. While it has been documented that the heuristic algorithm often produces much tighter approximations than the theoretical bound, on some instances it fails to find even the guaranteed approximation [Bud11, Section 9].

Thus A-CEEI is a problem where practical interest motivates theoretical inquiry. We have a theorem that guarantees the existence of an approximate equilibrium—the issue is finding it. Can the heuristic algorithms currently used to assign Wharton MBAs to their courses be replaced by a fast and rigorous algorithm for finding an approximate CEEI? Or are there complexity obstacles to approximating CEEI?

In this chapter, we show that finding the guaranteed approximation to CEEI is an intractable problem:

Theorem (Theorem 10.1.5, informal statement)**.** *The problem of finding an A-CEEI as guaranteed by [Bud11] is* PPAD*-complete.*

10.1 The Course Allocation Problem

Even though the A-CEEI and the existence theorem in [Bud11] are applicable to a broad range of allocation problems, we shall describe our results in the language of the course allocation problem.

We are given a set of M courses with integer capacities (the supply) $(q_j)_{j=1}^M$, and a set of N students, where each student i has a set $\Psi_i \subseteq 2^M$ of permissible course bundles, with each bundle containing at most $k \leq M$ courses. The set Ψ_i encodes both scheduling constraints (e.g., courses that meet at the same time) and any constraints specific to student i (e.g., prerequisites).

Each student i has a strict ordering over her permissible schedules, denoted by $\preccurlyeq_i$. We allow arbitrarily complex preferences—in particular, students may regard courses as substitutes or complements. More formally:

Definition 10.1.1. Course Allocation Problem The input to a course allocation problem consists of:

- For each student i a set of course bundles $(\Psi_i)_{i=1}^N$.
- The students' reported preferences, $(\preccurlyeq_i)_{i=1}^N$,
- The course capacities, $(q_j)_{j=1}^M$, and

The output to a course allocation problem consists of:

- Prices for each course $(p_j^*)_{j=1}^M$,
- Allocations for each student$(x_i^*)_{i=1}^N$, and
- Budgets for each student $(b_i^*)_{i=1}^N$.

How is an allocation evaluated? The *clearing error* of a solution to the allocation problem, is the $\mathcal{L}_2$ norm of the length-M vector of seats oversubscribed in any course, or undersubscribed seats in courses with positive price.

Definition 10.1.2. The *clearing error* α of an allocation is

$$\alpha \equiv \sqrt{\sum_j z_j^2}$$

Where z_j is given by

$$z_j = \begin{cases} \sum_i x_{ij}^* - q_j & \text{if } p_j^* > 0; \\ \max\left[\left(\sum_i x_{ij}^* - q_j\right), 0\right] & \text{if } p_j^* = 0. \end{cases}$$

We can now define the notion of *approximate* CEEI. The quality of approximation is characterized by two parameters: α, the clearing error (how far is our solution from a true competitive equilibrium?) and β, the bound on the difference in budgets (how far from equal are the budgets?). Informally, α can be thought of as the approximation loss on efficiency, and β can be thought of as the approximation loss on fairness.

Definition 10.1.3. An allocation is a (α, β)-*CEEI* if:

1. Each student is allocated their most preferred affordable bundle. Formally

$$\forall i : x_i^* = \arg\max_{\preccurlyeq_i} \left[x_i \in \Psi_i : \sum_j x_{ij} p_j^* \leq b_i^* \right]$$

2. Total clearing error is at most α.

3. Every budget $b_i^* \in [1, 1+\beta]$.

In [Bud11] it is proved that an (α, β)-approximate CEEI always exists, for some quite favorable (and as we shall see, essentially optimal) values of α and β:

Theorem 10.1.4 ([Bud11])**.** *For any input preferences, there exists an* (α, β)-*CEEI with* $\alpha = \sqrt{kM/2}$ *and any* $\beta > 0$.

Recall that k is the maximum bundle size.

The bound of $\alpha = \sqrt{kM/2}$ means that, for large number of students and course capacities, the market-clearing error converges to zero quite fast as a fraction of the endowment. It is also shown in [Bud11] that the mechanism which allocates courses according to such an A-CEEI satisfies attractive criteria of approximate fairness, approximate truthfulness, and approximate Pareto efficiency. The reader may consult [Bud11] for the precise definitions of the economic properties of the A-CEEI mechanism.

10.1.1 Our results

We our now ready to formally state our result for the complexity of A-CEEI:

Theorem 10.1.5. *Computing a* $\left(\sqrt{\frac{kM}{2}}, \beta\right)$-*CEEI is* PPAD-*complete, for some small constant* $\beta > 0$.

In Section 10.2, we prove that computing a $\left(\sqrt{\frac{kM}{2}}, \beta\right)$-CEEI is PPAD-hard; this is accomplished by a reduction from ϵ-GCIRCUIT. Then, in Section 10.3, we prove that the same problem also belongs to the class PPAD; this proof mostly follows along the lines of Budish's existence proof [Bud11], but certain probabilistic arguments must be constructively derandomized.

10.2 A-CEEI is PPAD-hard

]

Informally, in this section we provide a construction demonstrating that it is possible to define a set of courses, students, and preferences such that the price of the courses in an A-CEEI simulates the various "basic circuit functions" (e.g., an OR gate) that, when combined and wired together, are the necessary building blocks sufficient to emulate any continuous function. Therefore, any algorithm capable of solving A-CEEI in polynomial time would also suffice to solve ϵ-GCIRCUIT, and hence any problem in PPAD, in polynomial time as well.

Overview of the Reduction

We shall reduce ϵ-GCIRCUIT with fan-out 2 to the problem of finding an (α, β)-CEEI, with approximation parameters $\alpha = \Theta(N/M)$ and $\epsilon = \epsilon(\beta)$ (Note that, by increasing N, we can make α arbitrarily large as a function of M; in particular, $\alpha > \sqrt{kM/2}$.)

We will construct gadgets (that is, small sets of courses, students, capacities and preferences) for the various types of gates in the generalized circuit problem. Each gadget that we construct has one or more dedicated "input course", a single "output course", and possibly some "interior courses". An output course of one gadget can (and will) be an input to another. The construction will guarantee that in any A-CEEI the price of the output course will be approximately equal to the gate applied to the prices of the input courses.

10.2.1 The NOT gate ($G_\neg$) gadget

To illustrate what needs to be done, we proceed to construct a gadget for the $G_\neg$ gate; in particular, this implements a logical NOT.

Lemma 10.2.1. *(NOT gate ($G_\neg$) gadget)*
Let $n_x > 4\alpha$ and suppose that the economy contains the following courses:

- *c_x (the "input course") ;*

- *c_{1-x} with capacity $q_{1-x} = n_x/2$ (the "output course");*

and the following set of students:

- *n_x students interested only in the schedule $\{c_x, c_{1-x}\}$;*

and suppose further that at most $n_{1-x} = n_x/4$ other students are interested in course c_{1-x}.

Then in any (α, β)-CEEI

$$p^*_{1-x} \in [1 - p^*_x, 1 - p^*_x + \beta]$$

Proof. Observe that:

- If $p^*_{1-x} > 1 - p^*_x + \beta$, then none of the n_x students will be able to afford the bundle $\{c_x, c_{1-x}\}$, and therefore there will be at most $n_{1-x} = n_x/4$ students enrolled in the c_{1-x} - much less than the capacity $n_x/2$. Therefore $z_{1-x} \geq n_x/4$.
- On the other hand, if $p^*_{1-x} < 1 - p^*_x$, then all n_x students can afford the bundle $\{c_x, c_{1-x}\}$ - therefore the course will be overbooked by $n_x/2$; thus, $z_{1-x} \geq n_x/2$.

Therefore if $p^*_{1-x} \notin [1 - p^*_x, 1 - p^*_x + \beta]$, then $\|z\|_2 \geq n_x/4 > \alpha$ - a contradiction to (α, β)-CEEI. □

Similarly, in Subsection 10.2.3, we construct gadgets that simulate all the gates of the generalized circuit problem.

10.2.2 Course-size amplification

In the next subsection, we will construct gadgets that compute all the gates necessary for the circuit in the reduction from ϵ-GCIRCUIT. What happens when we try to concatenate them to form a circuit? Recall the penultimate sentence in the statement of Lemma 10.2.1: It says that the output course's price continues to behave like the output of the simulated gate, as long as there are not too many additional students that try to take the output course. (If there are more students, they may raise the price of the course beyond what we expect.) In particular, *the number of additional students that may want the output course is smaller than the number of students that want the input course.*

If we concatenated the gadgets without change, we would need to have larger course sizes as we increase the depth of the simulated circuit. This increase in course size is exponential in the depth of the circuit. Things get even worse—since we reduce

from *generalized* circuits, our gates form cycles. If the course size must increase at every gate it would have to be infinite!

To overcome this problem we construct a special $G_=$ gadget that (approximately) preserves the price from the input course, but is robust to twice as many additional students:

Lemma 10.2.2. *(Course-size amplification gadget)*
Let $n_x \geq 100\alpha$ and suppose that the economy contains the following courses:

- *c_x (the "input course")*
- *for $i = 1, \ldots 10$, c_i with capacities $q_i = 0.5 \cdot n_x$ ("interior courses");*
- *$c_{x'}$ with capacity $q_{x'}$, s.t. $q_x \leq q_{x'} \leq 4n_x$ ("output course");*

and the following sets of students:

- *n_x students interested in schedules $(\{c_x, c_i\})_{i=1}^{10}$ (in this order);*
- *$n_i = 0.49 \cdot n_x$ students $(\forall i)$ interested in schedules $(\{c_{x'}, c_i\}, \{c_i\}, \{c_{i+1}\}, \ldots, \{c_{10}\})$ (in this order);*

and suppose further that at most $n_{x'} = 2n_x$ other students are interested in course $c_{x'}$.

Then in any (α, β)-CEEI

$$p^*_{x'} \in [p^*_x - \beta, p^*_x + \beta]$$

In particular, notice that the price of $c_{x'}$ is guaranteed to approximate the price of c_x, even in the presence of additional $n_{x'} = 2n_x$ students - twice as many students as we added to c_x.

Proof. We start by proving that all the c_i's simulate NOT gadgets simultaneously, i.e. for every i and every (α, β)-CEEI, $p^*_i \in [1 - p^*_x, 1 - p^*_x + \beta]$.

- If $p^*_i > 1 - p^*_x + \beta$, assume wlog that it is the first such i, i.e. $p^*_j \leq 1 - p^*_x + \beta < p^*_i$ for every $j < i$.

  None of the n_x students can afford buying both c_x and c_i. Furthermore, for every $j < i$, none of the n_j students will prefer c_i over c_j. Therefore at most n_i students will take this course: $z^*_i \geq 0.01n_x$.

- If, on the other hand, $p_i^* < 1 - p_x^*$, then all n_x students will buy course c_i or some previous course c_j (for $j \le i$); additionally for every $j \le i$, each of the n_j corresponding students will buy some course c_k for $j \le k \le i$. Therefore the total overbooking of courses $1, \ldots, i$ will be at least $\sum_{j \le i} z_j^* \ge n_x \cdot (1 - 0.01i)$ - a contradiction to (α, β)-CEEI.

Now that we established that $p_i^* \in [1 - p_x^*, 1 - p_x^* + \beta]$, we shall prove the main claim, i.e. that $p_{x'}^* \in [p_x^* - \beta, p_x^* + \beta]$.

- If $p_{x'}^* > p_x^* + \beta$, then none of the n_i students, for any n_i, can afford buying both $c_{x'}$ and c_i. Therefore, even in the presence of additional $n_{x'} = 2n_x$ students who want to take $c_{x'}$, the course will be undersubscribed by $z_{x'}^* \ge q_{x'} - n_{x'} = 2n_x$
- If $p_{x'}^* < x + \beta$, then all n_i students, for each i, can afford to buy their top schedule - both $\{c_i, c_{x'}\}$. Therefore $c_{x'}$ will be oversubscribed by at least $z_{x'}^* \ge 0.9 \cdot n_x$ - a contradiction to (α, β)-CEEI.

□

Finally, given an instance of ϵ-GCIRCUIT with fan-out 2, we can use the gadgets we constructed in Lemmata 10.2.1-10.2.2 to construct an instance of (α, β)-CEEI that simulates the generalized circuit. We concatenate gadgets by identifying the output course of one gadget with the input course of the next two gadgets. In particular, after each gate gadget, we insert a series of course-size amplifying gadgets. Each amplifying gadget doubles the number of additional students that the gadget can tolerate, so a constant number of amplifying gadgets suffice; thus the blowup in error is also constant. As for the size of the reduction, each gadget introduces a constant number of new courses, and $\Theta(\alpha)$ new students; thus $M = \Theta(|V|)$ and $N = \Theta(\alpha \cdot |V|)$, where $|V|$ is the number of gates in the generalized circuit.

10.2.3 Additional gate gadgets

In this section we construct the rest of the gate gadgets, completing the proof of the PPAD-hardness.

In the lemma below we construct gadgets for a slightly modified set of gates. In particular, instead of implementing gates G_ζ and $G_{\times\zeta}$ from Definition 6.0.2, we only consider the special cases corresponding to $\zeta = \frac{1}{2}$ (denoted $G_{\frac{1}{2}}$ and $G_{/2}$, respectively).

Lemma 10.2.3. *Let $n_x \ge 2^8 \cdot \alpha$ and suppose that the economy has courses c_x and c_y. Then for any of the functions f listed below, we can add: a course c_z, and*

at most n_x students interested in each of c_x and c_y, such that in any (α, β)-CEEI $p_z^* \in \left[f\left(p_x^*, p_y^*\right) - 2\beta, f\left(p_x^*, p_y^*\right) + 2\beta\right]$

1. *HALF:* $f_{G_{/2}}(x) = x/2$
2. *VALUE:* $f_{G_{\frac{1}{2}}} \equiv \frac{1}{2}$
3. *SUM:* $f_{G_+}(x, y) = \min(x + y, 1)$
4. *DIFF:* $f_{G_-}(x, y) = \max(x - y, 0)$
5. *LESS:* $f_{G_<}(x, y) = \begin{cases} 1 & x > y + \beta \\ 0 & y > x + \beta \end{cases}$
6. *AND:* $f_{G_\wedge}(x, y) = \begin{cases} 1 & \left(x > \frac{1}{2} + \beta\right) \wedge \left(y > \frac{1}{2} + \beta\right) \\ 0 & \left(x < \frac{1}{2} - \beta\right) \vee \left(y < \frac{1}{2} - \beta\right) \end{cases}$
7. *OR:* $f_{G_\vee}(x, y) = \begin{cases} 1 & \left(x > \frac{1}{2} + \beta\right) \vee \left(y > \frac{1}{2} + \beta\right) \\ 0 & \left(x < \frac{1}{2} - \beta\right) \wedge \left(y < \frac{1}{2} - \beta\right) \end{cases}$

In particular, $p_z^* \in \left[f\left(p_x^*, p_y^*\right) - 2\beta, f\left(p_x^*, p_y^*\right) + 2\beta\right]$ in every (α, β)-CEEI even if up to $n_z \leq n_x/2^8$ additional students (beyond the ones specified in the proofs below) are interested in course c_z.

Proof.

HALF $G_{/2}$

Let c_z have capacity $q_z = n_x/8$, let $n_z = q_z/2$, and consider three auxiliary courses c_1, c_2, and $c_{\overline{x}}$ of capacities $q_1 = q_2 = q_z$ and $q_{\overline{x}} = n_x/2$. Using lemma 10.2.1 add n_x students that will guarantee $p_{\overline{x}} \in [1 - p_x^*, 1 - p_x^* + \beta]$. Additionally, consider $n_{\overline{x}} = n_x/4$ students with preference list: $(\{c_z, c_1, c_{\overline{x}}\}, \{c_z, c_2, c_{\overline{x}}\}, \{c_1, c_2, c_{\overline{x}}\})$ (in this order), then:

- If the total price $p_i^* + p_j^*$ of any pair $i, j \in \{1, 2, z\}$ is less than $p_x^* - \beta$, then all $n_{\overline{x}}$ students will be able to afford some subset in their preference list, leaving a total overbooking of at least $z_z^* + z_1^* + z_2^* \geq 2n_{\overline{x}} - 3q_z = n_x/8$, which violates the (α, β)-CEEI conditions
- If the total price of any of the pairs above (wlog, $p_1^* + p_2^*$) is greater than $p_x^* + \beta$, then none of the $n_{\overline{x}}$ students will be able to afford the subset $\{c_1, c_2, c_{\overline{x}}\}$. Therefore the number of students taking c_z will be at least the sum of students

taking c_1 or c_2. Therefore, even after taking into account n_z additional students, we have that $z_z^* + z_1^* + z_2^* \geq q_z - n_z = n_x/16$.

VALUE $G_{\frac{1}{2}}$

Similarly to the HALF gadget, consider two auxiliary courses c_1 and c_2, and let n_x students have preferences: $(\{c_z, c_1\}, \{c_z, c_2\}, \{c_1, c_2\})$. Then, following the argument for the HALF gadget, it is easy to see that $p_z^* \in \left[\frac{1}{2}, \frac{1}{2} + \beta\right]$ in any (α, β)-CEEI, with $n_z = n_x/8$.

DIFF G_-

Let $c_{\overline{x}}$ be a course with price $p_{\overline{x}}^* \in [1 - p_x^*, 1 - p_x^* + \beta]$, $q_{\overline{x}} = n_x/2$, and consider $n_{\overline{x}} = n_x/4$ students willing to take $\{c_{\overline{x}}, c_y, c_z\}$. Then it is easy to see that

$$\begin{aligned} p_z^* &\in \left[1 - p_{\overline{x}}^* - p_y^*, 1 - p_{\overline{x}}^* - p_y^* + \beta\right] \\ &\subseteq \left[p_x^* - p_y^* - \beta, p_x^* - p_y^* + \beta\right] \end{aligned}$$

with $n_z = n_x/16$

SUM G_+

Concatenating NOT and DIFF gadgets, we have:

$$\begin{aligned} p_{\overline{x}}^* &\in [1 - p_x^*, 1 - p_x^* + \beta] \\ p_{\overline{z}}^* &\in \left[p_{\overline{x}}^* - p_y^* - \beta, p_{\overline{x}}^* - p_y^* + \beta\right] \\ p_z^* &\in \left[1 - \left(p_{\overline{x}}^* - p_y^* + \beta\right), 1 - \left(p_{\overline{x}}^* - p_y^* - \beta\right) + \beta\right] \\ &\subseteq \left[p_x^* + p_y^* - 2\beta, p_x^* + p_y^* + 2\beta\right] \end{aligned}$$

for $n_z = n_x/2^8$

LESS $G_<$

Let $c_{\overline{x}}$ be a course with price $p_{\overline{x}}^* \in [1 - p_x^*, 1 - p_x^* + \beta]$, $q_{\overline{x}} = n_x/2$; let $q_z = n_x/8$ and $n_z = n_x/16$. Consider $n_x/4$ students wishing to take $(\{c_{\overline{x}}, c_y\}\{c_z\})$, in this order:

- If $p_y^* > p_x^* + \beta$, then $p_{\overline{x}}^* + p_y^* > 1 + \beta$, and therefore none of the $n_x/4$ students will be able to afford the first pair; they will all try to sign up to c_z which will be overbooked unless $p_z^* > 1$
- If $p_x^* > p_y^* + \beta$, then all $n_x/4$ students will sign up for the first pair, forcing $p_z^* = 0$ in any (α, β)-CEEI.

AND $G_\wedge$

Let $c_{\frac{1}{2}}$ be a course with price $p^*_{\frac{1}{2}} \in \left[\frac{1}{2}, \frac{1}{2}+\beta\right]$ and $n_{\frac{1}{2}} = n_x/8$, as guaranteed by gadget VALUE; let $q_z = n_x/32$ and $n_z = n_x/64$. Consider $n_x/16$ students wishing to take $\left(\left\{c_x, c_{\frac{1}{2}}\right\}, \left\{c_y, c_{\frac{1}{2}}\right\}, \{c_z\}\right)$, in this order.

- If $\left(p^*_x > \frac{1}{2}+\beta\right) \wedge \left(p^*_y > \frac{1}{2}+\beta\right)$, then the $n_x/16$ students can afford neither pair. They will all try to sign up for c_z, forcing $p^*_z > 1$, in any (α, β)-CEEI.
- If $\left(x < \frac{1}{2}-\beta\right) \vee \left(y < \frac{1}{2}-\beta\right)$, then the $n_x/16$ students can afford at least one of the pairs and will register for those courses. Thus $p^*_z = 0$.

OR $G_\vee$

Similar to the AND gadget; students will want $\left(\left\{c_x, c_y, c_{\frac{1}{2}}\right\}, \{c_z\}\right)$, in this order.

□

10.3 A-CEEI ∈ PPAD

that the problem belongs to the class PPAD; this proof is much harder than usual.

In this section we establish that computing a $\left(\frac{\sqrt{\sigma M}}{2}, \beta\right)$-CEEI is in PPAD, for $\sigma = \min\{2k, M\}$. We follow the steps of the existence proof in [Bud11], and show that each one can be carried out either in polynomial time, or through a fixed point. One difficulty is that certain steps of Budish's proof are randomized and must be constructively derandomized in polynomial time.

Remark 10.3.1. We assume that the student preferences $(\succsim_i)$ are given in the form of an ordered list of all the bundles in Ψ_i (i.e., all the bundles that student i prefers over the empty bundle). In particular, we assume that the total number of permissible bundles is polynomial.

Remark 10.3.2. In fact, we prove that the following, slightly more general problem, is in PPAD: Given any $\beta, \epsilon > 0$ and initial approximate-budgets vector $\mathbf{b} \in [1, 1+\beta]^N$, find a $\left(\frac{\sqrt{\sigma M}}{2}, \beta\right)$-CEEI with budgets $\mathbf{b}^*$ such that $|b_i - b^*_i| < \epsilon$ for every i.

Our proof will follow the steps of the existence proof by [Bud11]. We will use the power of PPAD to solve the Kakutani problem, and derandomize the other nonconstructive ingredients.

10.3.1 Preliminaries

Our algorithm receives as input an economy $\left((q_j)_{j=1}^M, (\Psi_i)_{i=1}^N, (\succsim_i)_{i=1}^N\right)$, parameters $\beta, \epsilon > 0$, and an initial approximate-budgets vector $\mathbf{b} \in [1, 1+\beta]^N$. We denote $\bar{\beta} = \min\{\beta, \epsilon\}/2$.

We will consider M-dimensional price vectors in $\mathcal{P} = [0, 1+\beta+\epsilon]^M$. In order to define a price adjustment function, we consider an enlargement $\tilde{\mathcal{P}} = [-, +\beta+\epsilon]^{\mathcal{M}}$, as well as a truncation function $t : \tilde{\mathcal{P}} \to \mathcal{P}$ (whose j-th coordinate is given by $t_j(\tilde{p}_j) = \min\{\max\{\tilde{p}_j, 0\}, 1+\beta+\epsilon\}$).

For each student i, we denote her demand at prices $\tilde{\mathbf{p}}$ with budget b_i by

$$d_i(\tilde{\mathbf{p}}, b_i) = \max_{(\succsim_i)} \{x' \in \Psi_i : \tilde{\mathbf{p}} \cdot x' \leq b_i\}$$

Given the total demand of all the students, we can define the excess demand to be:

$$\mathbf{z}(\tilde{\mathbf{p}}, \mathbf{b}) = \sum_{i=1}^{N} d_i(\tilde{\mathbf{p}}, b_i) - \mathbf{q}$$

A key ingredient to the analysis is the budget-constraint hyperplanes. These are the hyperplanes in price space along which a student can exactly afford a specific bundle. For each student i and bundle x, the corresponding *budget-constraint hyperplane* is defined as $H(i, x) = \{\tilde{\mathbf{p}} \in \mathcal{P} : \tilde{\mathbf{p}} \cdot \ = \ \}$.

10.3.2 Deterministically finding a "general position" perturbation (step 1)

It is convenient to assume that the budget-constraint hyperplanes are in "general position", i.e. there is no point $\tilde{\mathbf{p}} \in \mathcal{P}$ at which any subset of linearly dependent budget-constraint hyperplanes intersect (in particular, no more than M hyperplanes intersect at any point). In the existence proof, this is achieved by assigning a small random reverse tax $\tau_{i,x} \in (-\epsilon, \epsilon)$, for each student i and bundle x; i's modified cost for bundle x at prices $\tilde{\mathbf{p}}$ becomes $\tilde{\mathbf{p}} \cdot x - \tau_{i,x}$. Given taxes $\tau = (\tau_{i,x})_{i \in \mathcal{S}, x \in \Psi_i}$, we redefine $d_i(\tilde{\mathbf{p}}, b_i, \tau_i)$, $\mathbf{z}(\tilde{\mathbf{p}}, \mathbf{b}, \tau)$, and $H(i, x, \tau_{i,x})$ analogously.

In this section, we show how to deterministically choose these taxes.

Lemma 10.3.3. *There exists a polynomial-time algorithm that finds a vector of taxes $\tau = (\tau_{i,x})_{i \in \mathcal{S}, x \in \Psi_i}$ such that:*

1. *$-\epsilon < \tau_{i,x} < \epsilon$ (taxes are small)*
2. *$\tau_{i,x} > \tau_{i,x'}$ if $x \succ_i x'$ (taxes prefer more-preferred bundles)*

3. $1 \le \min_{i,x}\{b_i + \tau_{i,x}\} \le \max_{i,x}\{b_i + \tau_{i,x}\} \le 1 + \beta$ *(inequality bound is preserved)*
4. $b_i + \tau_{i,x} \ne b_{i'} + \tau_{i',x'}$ *for* $(i,x) \ne (i',x')$ *(no two perturbed prices are equal)*
5. *there is no price* $\tilde{\mathbf{p}} \in \mathcal{P}$ *at which any subset of linearly dependent budget-constraint hyperplanes intersect*[1]

Proof. Assume wlog that $\mathbf{b}$ is rounded to the nearest integer multiple of $\bar{\beta}M^{-M}$: otherwise we can include this rounding in the taxes.

We proceed by induction on the pairs (i,x) of students and bundles: at each step let $\tau_{i,x}$ be much smaller in absolute value than all the taxes introduced so far. (For each i, we consider the (i,x)'s either in the order $\succsim_i$ or in the reverse order, maintaining Property 2 depending on the sign of $\tau_{i,x}$.)

More precisely, if (i,x) is the ν^{th} pair to be considered, then we set

$$\tau_{i,x} \in \pm\bar{\beta}M^{-2\nu M},$$

where the sign is chosen such that condition 3 in the statement of the lemma is preserved.

Now, assume by contradiction that there exists a k-tuple $H(i_1, x_1, \tau_{i_1,x_1}), \ldots, H(i_k, x_k, \tau_{i_k,x_k})$ of hyperplanes that intersect at price vector $\tilde{\mathbf{p}}$, and such that the x_i's are linearly dependent. (Note that the latter holds, in particular, for every $(M+1)$-tuple.)

Assume further, wlog, that this is the first such k-tuple, with respect to the order of the induction. In particular, this means that $\{x_1, \ldots, x_{k-1}\}$ are linearly independent. Now consider the system

$$\begin{pmatrix} x_1^T & \ldots & x_{k-1}^T \end{pmatrix}(\alpha) = (x_k)$$

Notice that it has rank $k-1$. We can now take $k-1$ linearly independent rows $j_1, \ldots j_{k-1}$ such that the following system has the same unique solution α:

$$\begin{pmatrix} x_{1,j_1} & \ldots & x_{k-1,j_1} \\ \vdots & \ddots & \\ x_{1,j_{k-1}} & & x_{k-1,j_{k-1}} \end{pmatrix}(\alpha) = \begin{pmatrix} x_{k,j_1} \\ \vdots \\ x_{k,j_{k-1}} \end{pmatrix}$$

Denote

$$X = \begin{pmatrix} x_{1,j_1} & \ldots & x_{k-1,j_1} \\ \vdots & \ddots & \\ x_{1,j_{k-1}} & & x_{k-1,j_{k-1}} \end{pmatrix}$$

[1] The original existence proof of [Bud11] requires only that no more than M hyperplanes intersect at any point; this causes problems in the conditional expectation argument [Bud11, Step 5].

Since X is a square matrix of full rank it is invertible, so we have that

$$\alpha = X^{-1}\begin{pmatrix} x_{k,j_1} \\ \vdots \\ x_{k,j_{k-1}} \end{pmatrix}$$

Now, recall that

$$X^{-1} = \frac{1}{\det X}\begin{pmatrix} X_{1,1} & \cdots & X_{k-1,1} \\ \vdots & \ddots & \\ X_{1,k-1} & & X_{k-1,k-1} \end{pmatrix}$$

where $X_{i,j}$ is the (i,j)-cofactor of X. Finally, since X is a Boolean matrix, its determinant and all of its cofactors are integers of magnitude less than $(k-1)^{k-1} \leq M^M$. The entries of α are therefore rational fractions with numerators and denominators of magnitude less than M^M.

Now, by our assumption by contradiction, k hyperplanes intersect at $\tilde{\mathbf{p}}$:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix}(\tilde{\mathbf{p}}) = \begin{pmatrix} b_{i_1} + \tau_{i_1,x_1} \\ \vdots \\ b_{i_k} + \tau_{i_k,x_k} \end{pmatrix}$$

Therefore,

$$b_{i_k} + \tau_{i_k,x_k} = x_k \cdot \tilde{\mathbf{p}} = \sum_{l=1}^{k-1} \alpha_l \left(x_l \cdot \tilde{\mathbf{p}}\right) = \sum_{l=1}^{k-1} \alpha_l \left(b_{i_l} + \tau_{i_l,x_l}\right) \tag{10.1}$$

However, if (i_k, x_k) is the ν^{th} pair added by the induction, then the following is an integer:

$$\sum_{l=1}^{k-1} \left(\det(X) \cdot \alpha_l\right) \cdot \frac{M^{2(\nu-1)M}}{\bar{\beta}} \left(b_{i_l} + \tau_{i_l,x_l}\right)$$

By our assumption that all the budgets are rounded, $\frac{M^M}{\bar{\beta}} \cdot b_{i_k}$ is also an integer. Yet $\left|\det(X) \cdot \frac{M^{2(\nu-1)M}}{\bar{\beta}} \cdot \tau_{i_k,x_k}\right| \leq \left|\frac{M^{(2\nu-1)M}}{\bar{\beta}} \cdot \tau_{i_k,x_k}\right| \leq M^{-M}$ is not an integer. This yields a contradiction to Equation (10.1). □

10.3.3 Finding a fixed point (steps 2-4)

This subsection describes the price adjustment correspondence of [Bud11], and is brought here mostly for completeness.

We first define the price adjustment function:

$$f\left(\tilde{\mathbf{p}}\right) = t\left(\tilde{\mathbf{p}}\right) + \frac{1}{2N}\mathbf{z}\left(t\left(\tilde{\mathbf{p}}\right); \mathbf{b}, \tau\right)$$

Observe that if $\tilde{\mathbf{p}}^*$ is a fixed point $\tilde{\mathbf{p}}^* = f(\tilde{\mathbf{p}}^*)$ of f, then its truncation $t(\tilde{\mathbf{p}}^*) = \mathbf{p}^*$ defines an exact competitive equilibrium[2]. Yet, we know that the economy may not have an exact equilibrium - and indeed f is discontinuous at the budget constraint hyperplanes, and so it is not guaranteed to have a fixed point.

Instead, we define an upper hemicontinuous, set-valued "convexification" of f:

$$F(\mathbf{p}) = co\{\mathbf{y}: \exists \text{ a sequence } \mathbf{p^w} \to \mathbf{p},\ \mathbf{p} \neq \mathbf{p^w} \in \mathcal{P} \text{ such that } f(\mathbf{p^w}) \to \mathbf{y}\}$$

The correspondence F is upper hemicontinuous, non-empty, and convex; therefore, by Kakutani's fixed point theorem it has a fixed point (i.e. a price vector that satisfies $\tilde{\mathbf{p}}^* \in F(\tilde{\mathbf{p}}^*)$).

By [Pap94], finding a Kakutani fixed point of F is in PPAD.

Working with finite precision

To be rigorous, we need to complete a few subtle numerical details about finding a fixed point of F. We round all price vectors to a the nearest integer multiple of $\delta := \left(\bar{\beta} M^{\frac{1}{2}-2(\nu_{\max}+1)M}\right)$ (this precision suffices to implement the algorithm in Lemma 10.3.3).

At any point on the δ-grid, the price of any bundle is an integer multiple of δ, so, any budget-constraint hyperplane which does not contain $\mathbf{p}$, must be at (L_1) distance at least δ. In particular, this means that every $\delta/2$-approximate fixed point of F is also an exact fixed point. Finally, we can use the PPAD algorithm of [Pap94] to find a $\delta/2$-approximate fixed point.

There is also an issue of computing the correspondence F. From the proof of [Pap94] it follows that it suffices to compute just a single point in $F(\mathbf{p})$ for every $\mathbf{p}$. This is important because the number of points in $F(\mathbf{p})$ on the δ-grid may be exponential. As we mentioned earlier, every budget-constraint hyperplane which does not contain $t(\mathbf{p})$, must be at least δ-far. Therefore, we can take any point $\mathbf{p'}$ whose truncation $t(\mathbf{p'})$ is at distance $\delta/2$ from $t(\mathbf{p})$, and does not lie on any hyperplanes. ($\mathbf{p'}$ will not be on the δ-grid.) Because no budget-constraint hyperplanes lie between $t(\mathbf{p'})$ and $t(\mathbf{p})$, it follows that $t(\mathbf{p}) + \frac{1}{2N}\mathbf{z}(t(\mathbf{p'}); \mathbf{b}, \tau) \in F(\mathbf{p})$.

10.3.4 From a fixed point to an approximate CEEI (steps 5-9)

Lemma 10.3.4. *Given a fixed point $\mathbf{p}^*$ of F, we can find in polynomial time a vector of prices $\mathbf{p}^{\phi'}$ such that $\left\|\mathbf{z}\left(\mathbf{p}^{\phi'}, \mathbf{b}, \tau\right)\right\|_2 \leq \frac{\sqrt{\sigma M}}{2}$*

[2]See [Bud11, Appendix A, Step 2] for more details.

Proof. We use the method of conditional expectation to derandomize Step 8 of [Bud11].

Recall from the previous subsection that there exists a neighborhood around $\mathbf{p}^*$ which does not intersect any budget-constraint hyperplanes (beyond those that contain $\mathbf{p}^*$). Let $1, \ldots, L'$ be the indices of students whose budget-constraint hyperplanes intersect at $\mathbf{p}^*$. For student $i \in [L']$, let w_i be the number of corresponding hyperplanes $H\left(i, x_i^1, \tau_{i,x_i^1}\right), \ldots H\left(i, x_i^{w_i}, \tau_{i,x_i^{w_i}}\right)$ intersecting at $\mathbf{p}^*$, and assume wlog that the superindices of $x_i^1, \ldots x_i^{w_i}$ are ordered according to $\succsim_i$.

Let d_i^0 be agent i's demand when prices are slightly perturbed from $\mathbf{p}^*$ such that all x_i^j's are affordable. Such a perturbation exists and is easily computable because the hyperplanes are linearly independent[3]. Similarly, let d_i^1 denote agent i's demand when $x_i^2, \ldots x_i^{w_i}$ are affordable, but x_i^1 is not, and so on. Finally, let $z_{S \setminus [L']}\left(\mathbf{p}^*, \mathbf{b}, \tau\right) = d_{S \setminus [L']}\left(\mathbf{p}^*, \mathbf{b}, \tau\right) - \mathbf{q}$ be the market clearing error when considering the rest of the students. (The demands of $S \setminus [L']$ is constant in the small neighborhood $\mathbf{p}^*$ which does not intersect any additional hyperplanes.)

By Lemma 3 of [Bud11], there exist distributions a_i^f over d_i^f:

$$\begin{aligned} a_i^f &\in [0,1] \quad \forall \quad i \in [L'], \forall f \in \{0\} \cup [w_i] \\ \sum_{f=0}^{w_i} a_i^f &= 1 \quad \forall \quad i \in [L'] \end{aligned}$$

such that the clearing error of the *expected demand* is 0:

$$z_{S \setminus [L']}\left(\mathbf{p}^*, \mathbf{b}, \tau\right) + \sum_{i=1}^{L'} \sum_{f=0}^{w_i} a_i^f d_i^f = 0$$

We first find such a_i^f in polynomial time using linear programming.

The existence proof then considers, for each i, a random vector $\Theta_i = (\Theta_i^1, \ldots, \Theta_i^{w_i})$: the vectors are independent and in any realization θ_i satisfy $\sum_{f=0}^{w_i} \theta_i^f = 1$, while the variables each have support $\operatorname{supp}\left(\Theta_i^f\right) = \{0,1\}$, and expectation $\mathrm{E}\left[\Theta_i^f\right] = a_i^f$.

By Lemma 4 of [Bud11], the *expected clearing error* is bounded by:

$$\mathrm{E}_{\Theta_! \ldots \Theta_{L'}} \left\| \sum_{i=1}^{L'} \sum_{f=0}^{w_i} \left(a_i^f - \theta_i^f\right) d_i^f \right\|_2^2 = \sum_{i=1}^{L'} \mathrm{E}_{\Theta_i} \left\| \sum_{f=0}^{w_i} \left(a_i^f - \theta_i^f\right) d_i^f \right\|_2^2 \leq \frac{\sigma M}{4}$$

[3]This perturbation eventually guarantees the existence of $\mathbf{p}^{\phi'}$. As we mentioned in Footnote 1, Budish does not require that the hyperplanes are linearly independent, so $\mathbf{p}^{\phi'}$ may not exist. However, it seems that $\mathbf{p}^{\phi'}$ is not actually crucial to the overall existence proof. In particular, as Budish points out, even if it exists it may be infeasible (i.e. require negative prices), so the final solution uses $\mathbf{p}^*$ instead.

We now proceed by induction on the students. For each i, if the conditional expectation on $\left(\hat{\theta}_j\right)_{j<i}$ satisfies

$$\mathrm{E}_{\Theta_i \ldots \Theta_{L'}}\left[\left\|\sum_{i=1}^{L'}\sum_{f=0}^{w_i}\left(a_i^f-\theta_i^f\right)d_i^f\right\|_2^2 \mid \hat{\theta}_1,\ldots,\hat{\theta}_{i-1}\right] \leq \frac{\sigma M}{4}$$

then at least one $\hat{\theta}_i$ must also satisfy the above bound. We can find such $\hat{\theta}_i$ in polynomial time by computing the conditional expectation for every feasible $\hat{\theta}_i'$:

$$\begin{aligned}\mathrm{E}_{\Theta_{i+1}\ldots\Theta_{L'}}\left[\left\|\sum_{j=1}^{L'}\sum_{f=0}^{w_j}\left(a_j^f-\theta_j^f\right)d_j^f\right\|_2^2 \mid \hat{\theta}_1,\ldots,\hat{\theta}_i\right] &= \sum_{j=1}^{i}\left\|\sum_{f=0}^{w_j}\left(a_j^f-\hat{\theta}_j^f\right)d_j^f\right\|_2^2 \\ &+ \sum_{j=i+1}^{L'}\mathrm{E}_{\Theta_j}\left\|\sum_{f=0}^{w_j}\left(a_j^f-\theta_j^f\right)d_j^f\right\|_2^2 \\ &+ \sum_{j\neq h\leq i}\sum_{f=0}^{w_j}\sum_{g=0}^{w_h}\left(a_j^f-\hat{\theta}_j^f\right)\left(a_h^g-\hat{\theta}_h^g\right)\end{aligned}$$

Finally, the choice of $\left(\hat{\theta}_i\right)_{i=1}^{L'}$ induces the promised price vector $\mathbf{p}^{\phi'}$.

□

The chosen $\left(\hat{\theta}_i\right)_{i=1}^{L'}$ define an allocation $\mathbf{x}^*$ with bounded clearing error. We now follow step 9 of [Bud11] in order to define budgets $\mathbf{b}^*$ such that $\mathbf{x}^*$ is the preferred consumption by all the students at price $\mathbf{p}^*$.

We define, for every i, $b_i^* = b_i + \tau_{i,x_i^*}$. For $i > L'$ we have $x_i^* = d_i\left(\mathbf{p}^*, b_i, \tau_i\right)$. By requirement 2 of lemma 10.3.3, every bundle that student i prefers over x_i^* had a greater tax and was still unaffordable at $\mathbf{p}^*$; it now costs more than $b_i + \tau_{i,x_i^*}$.

For $i \leq L'$ notice that every bundle $x_i^\perp$ that i prefers over x_i^* and was exactly affordable at $\mathbf{p}^*$ with taxes τ and budget $\mathbf{b}$, $x^\perp$ must cost strictly more than i's new budget b_i^*. Therefore, $(\mathbf{x}^*, \mathbf{b}^*, \mathbf{p}^*)$ is a $\left(\frac{\sqrt{\sigma M}}{2}, \beta\right)$-CEEI

□

Part IV

Quasi-polynomial Time

Chapter 11

Birthday repetition

This part of the thesis deals with several fundamental problems that admit quasi-polynomial ($n^{\log n}$) time algorithms. What can we learn from such an algorithm? On one hand, assuming the Exponential Time Hypothesis (ETH, see Hypothesis 1), it means that they are not NP-hard. On the other hand, it does not meet our gold standard of efficiency, polynomial time (P). Furthermore, the logarithmic factor in the exponent is still prohibitive in applications in practice.

The approach we take in this this thesis is inspired by the *birthday repetition* meta-reduction due to Aaronson, Impagliazzo, and Moshkovitz [AIM14]. The birthday repetition is best explained as a game[1] between two provers (Alice and Bob) and a verifier. The omniscient but untrusted provers want to convince the verifier that a certain 3-SAT formula is (approximately) satisfiable. The provers agree in advance on an assignment, and are then placed in separate rooms. The verifier asks Alice for the assignments of 3 variables on one randomly chosen clause, and asks Bob for the assignment of one of the variables in the same clause. It is not hard to see that if at most 90% of the clauses are satisfiable, the provers have probability at most 90% to succeed in sending assignments that both satisfy Alice's clause and agree on the Bob's variable. Otherwise, the verifier can detect that Alice and Bob are lying and the 3-SAT formula is not satisfiable. But this requires coordinating the challenge-messages sent to Alice and Bob.

In the birthday-repetition version of the same two-prover one-verifier game, the verifier sends Alice a random selection of $\sqrt{n}$ clauses, and sends Bob an independently random subset of $\sqrt{n}$ variables. By the birthday paradox, we expect that one of Bob's variables appears in one of Alice's clauses. Thus their probability of tricking the verifier into believing that a far-from-satisfiable formula is satisfiable is not much

[1] The sense in which we use the word "game" here is unrelated to the game theoretic "game".

higher than in the original (no-repetition) two-prover one-verifier game. The main advantage in this approach is that the challenges sent to Alice and Bob are independent; this is sometimes called a *free game*. This is particularly helpful when reducing to bilinear optimization problems such as Densest k-Subgraph or Nash equilibrium (recall (1.1) in the introduction).

The other interesting feature of the birthday repetition game is that it blows up the number of possible questions to Alice and Bob, and (more importantly) the length of their answers. In particular, given a list of $\sqrt{n}$ variables, Bob has $N \triangleq 2^{\sqrt{n}}$ distinct choices of truth assignments to those variables. Therefore, if Alice and Bob can devise an (approximately) optimal strategy much faster than $N^{\log N} = 2^n$, they would violate ETH. Hence with this approach we can obtain quasi-polynomial lower bounds on the running time that almost exactly match the running time of the best known algorithms.

There is one more property that is common to all our quasi-polynomial lower bounds: while they are all based on the birthday repetition approach, each problem introduces new obstacle and requires new ideas.

11.1 Warm-up: best ϵ-Nash

In order to introduce the "birthday repetition" framework, we begin with a particularly simple application. As we have already discussed, proving hardness of total problem is notoriously difficult both conceptually and technically. Often, however, there are related decision problems that admit much easier proofs of intractability. For example, Gilboa and Zemel [GZ89] proved that deciding whether there exists a Nash equilibrium with certain welfare guarantees is NP-hard.

By looking at the analogous question for approximate Nash equilibrium, we circumvent the difficulty of totality. But the quasi-polynomial algorithm of [LMM03] can approximately[2] solve this question as well. Hence, assuming ETH, this problem is still not NP-hard.

Braverman et al. [BKW15] showed a nearly matching quasi-polynomial lower bound on the running time for approximating the best ϵ-Nash equilibrium. The theorem below obtains slightly better parameters. More importantly, our proof is significantly simpler[3] .

[2]This is a bi-criteria approximation: in quasi-polynomial time the algorithm can distinguish between a game that has a high-payoff exact (or $\epsilon/2$)-approximate Nash equilibrium, and a game where every ϵ-approximate Nash equilibrium has low payoff. Notice however that this does not imply that the algorithm can approximate (to within an additive ϵ) the value of the best ϵ-approximate Nash equilibrium.

[3]Briefly, in our proof we focus on a single partition of the variables, where it is guaranteed that

Theorem 11.1.1. *There exists a constant $\epsilon > 0$ such that, given a two-player, N-strategy game with utilities in $[0,1]$ distinguishing between the following requires $N^{\tilde{\Omega}(\log N)}$ time (assuming the Exponential Time Hypothesis):*

Completeness *The game has an exact Nash equilibrium with expected payoff > 0.99 for both players.*

Soundness *In every ϵ-approximate Nash equilibrium, both players have expected payoff < 0.01.*

The rest of this section is devoted to the proof of Theorem 11.1.1. We first prove a small ($O(\epsilon)$) additive gap between completeness and soundness. This is the main step that exhibits the birthday repetition. We then achieve a large gap by a simple amplification step, where we introduce a new action for each player, and a spurious low-payoff pure equilibrium: in the completeness case, the players don't want to deviate from the high-payoff mixed equilibrium; but for the soundness, they are lured by unilateral deviations into the low-payoff equilibrium.

11.1.1 Constructing the first gap

Lemma 11.1.2. *There exist absolute constants $\epsilon, \delta > 0$ such that the following holds. Given a two-player, N-strategy game with utilities in $[0,1]$ distinguishing between the following requires $N^{\tilde{\Omega}(\log N)}$ time (assuming the Exponential Time Hypothesis):*

Completeness *The game has an exact Nash equilibrium with expected payoff c for both players.*

Soundness *In every ϵ-Nash equilibrium, at least one player has expected payoff $s \leq c - \delta$.*

Proof. We reduce from a bipartite, constant alphabet 2CSP. By the PCP Theorem 2.4.1, distinguishing between a completely satisfiable instance and one where only $(1-\eta)$-fraction (for some constant $\eta > 0$) of the clauses are satisfiable requires time $2^{\tilde{\Omega}(n)}$, assuming ETH. We construct a game where, at every approximate Nash equilibrium, Alice's and Bob's respective mixed strategies encode an assignment (or distribution over assignments) to the 2CSP[4]. Alice's and Bob's expected payoff will depend on the number of satisfied clauses.

every pair of subsets induces approximately the right number of constraints. In contrast, [BKW15] enumerate over all the subsets of size $\approx \sqrt{n}$. Then it is only true that *most* pairs of subsets induce the right number of constraints, which somewhat complicates the proof.

[4]Note that so far *every* approximate equilibrium encodes *some* assignment (or distribution over assingmnets) - even in the soundness case.

Construction

At a high level, our construction is composed of three subgames. The *main subgame* will test the 2CSP assignment; in this subgame we perform the "birthday repetition". The remaining two *auxiliary subgames* enforce the structure of (approximate) equilibria; in particular, they force that the players to assign (approximately) uniform probabilities in the main subgame to (almost) all 2CSP variables; this step uses a construction due to Althofer [Alt94].

The players' final payoff is a weighted average of the payoffs in the three subgame. Since we want to make sure that the structural constraints on the equilibria are enforced, we place only ϵ weight on the main subgame, and weight $\frac{1-\epsilon}{2}$ on each of the auxiliary subgames.

Main subgame We partition Alice's and Bob's respective sets of variables into $\sqrt{n}$ subsets $\{S_i\}$ and $\{T_j\}$, respecitively. By Lemma 2.7.6, we can guarantee that each subset has at most $2\sqrt{n}$ variables, and that the number of constraints between any pair S_i, T_j is at most 8 times the expectation.

In the main subgame, each player, chooses an index $i, j \in \left[\sqrt{n}\right]$, and a partial assignment for the corresponding respective subsets $\sigma \in {\Sigma_A}^{S_i}, \tau \in {\Sigma_B}^{T_j}$. If the partial assignments jointly satisfy all the induced constraints, both players receive payoff 1; otherwise they receive payoff 0.

Auxiliary subgames In addition to the choices of i, σ in the main subgame, Alice chooses a subset $C_A \subset \left[\sqrt{n}\right]$ of cardinality $|C_A| = \sqrt{n}/2$. In the first auxiliary subgame, Alice wants to use the set C_A to "catch" Bob's index j: Alice has payoff 1 if $j \in C_A$, and 0 otherwise. Bob, on the other hand, tries to "escape": his payoff is 0 if $j \in C_A$, and 1 otherwise.

The second auxiliary subgame is defined analogously (Bob chooses a set C_B of cardinality $|C_B| = \sqrt{n}/2$ and tries to catch Alice's index i.)

Analysis

Game size Each player has an action set of size $N \leq \sqrt{n} \cdot \Sigma^{2\sqrt{n}} \cdot \binom{\sqrt{n}}{\sqrt{n}/2} = 2^{O(\sqrt{n})}$. Therefore, assuming ETH, finding an (approximately) satisfying assignment to the Label Cover instance requires time $2^{\Omega(n)} = N^{\tilde{\Omega}(\log N)}$.

Completeness If the 2CSP has a satisfying assignment, the players can play the following high-welfare equilibrium: in the main subgame, Alice and Bob choose

i, j uniformly at random, and σ, τ as the respective restrictions of the satisfying assignments. In the auxiliary subgames, Alice and Bob choose C_A, C_B uniformly at random.

Because i, j are chosen uniformly at random, they have no incentives to deviate from their choice of C_A, C_B; similarly, since C_A, C_B are chosen uniformly at random (and the payoff in the main subgame is always 1), neither player has an incentive to deviate from the choice of i, j. Therefore this is indeed a Nash equilibrium.

Finally, the expected payoff for each player in each auxiliary subgame is $1/2$ (they have probability $1/2$ of winning), and 1 in the main subgame; in total it is $c \triangleq 1/2 + \epsilon/2$.

Soundness

First, we claim that in every ϵ-approximate Nash equilibrium (ϵ-ANE), the players' respective distributions over i, j are $O(\epsilon)$-close to uniform. Assume by contradiction that Alice's distribution over i is 36ϵ-far from uniform. By Lemma 2.7.8, Bob can pick a set $C_B \in \binom{\lceil\sqrt{n}\rceil}{\sqrt{n}/2}$ such that $i \in C_B$ with probability $1/2 + 9\epsilon$. Therefore Bob can guarantee a payoff of at least $1/2 + 9\epsilon$ from the second auxiliary subgame (without affecting his payoffs from the other subgames). Therefore at any ϵ-ANE, he guarantees himself more than $1/2 + 6\epsilon$ (otherwise, having a 3ϵ-improving deviation on the auxiliary subgame implies a $> \epsilon$-deviation in total). Therefore Alice's payoff on this subgame is less than $1/2 - 6\epsilon$, whereas she can guarantee herself a payoff of $1/2$ by picking i uniformly at random. Thus she can improve her total payoff by more than 2ϵ, which is a violating deviation even after we subtract ϵ for potential loss of payoff due to changing her strategy in the main subgame.

Any mixed strategy profile induces a distribution over assignments to the original 2CSP instance: for each set S_i, we take the distribution over ${\Sigma_A}^{S_i}$ induced by Alice's mixed strategy restricted to actions where she picks index i (if she never picks i, choose an arbitrary assignment for S_i). Alice's expected payoff in the main subgame is equal to the probability that an assignment drawn from the induced distribution satisfies the constraints on a random pair (S_i, T_j), where i and j are drawn according to Alice and Bob's respective mixed strategies. Since those strategies are $O(\epsilon)$-close to uniform, Alice's expected payoff in the main subgame is within $O(\epsilon)$ of the probability that the constraints between a uniformly random (S_i, T_j) are satisfied. If the value of the 2CSP is $1 - \eta$, then the probability that any consistent assignment (in particular, one drawn from the induced distribution) satisfies a uniformly random (S_i, T_j) is at most $1 - \eta/8$; this is true because, by our construction of the partition, the η-fraction of unsatisfied constraints cannot concentrate on less than $\eta/8$-fraction of pairs. Therefore Alice's (respectively Bob's) expected payoff in the main subgame

is at most $1-\eta/8+O(\epsilon)$. Since the auxiliary subgames are 1-sum games, at least one of Alice or Bob must have total expected payoff at most $1/2+\epsilon/2\,(1-\eta/8+O(\epsilon)) < c-\eta\epsilon/16+O(\epsilon^2)$. Choosing ϵ sufficiently small compared to η (yet still constant) completes the proof. □

11.1.2 Gap amplification

Proof of Theorem 11.1.1. Consider the game from Lemma 11.1.2. First, we scale all the payoffs by a 10^{-4}-factor and shift them so that they are between $1-10^{-4}$ and 1. Now, by Lemma 11.1.2, it is hard to distinguish between the game having an exact Nash equilibrium with expected payoff c' and every ϵ'-Nash equilibrium having at least one player with expected payoff $< c'-\delta'$. We choose ϵ sufficiently small; in particular, $\epsilon < \epsilon'/4, \delta'/4 < 10^{-4}$.

We add one new action to each player. When both players play old actions, their payoffs are as in the game from Lemma 11.1.2 (with modified payoffs). When Alice plays the new action and Bob plays an old action, Alice's payoff is $c'-\delta'/2$, and Bob's payoff is 0. Similarly, when Alice plays an old action and Bob plays his new action, Alice's payoff is 0 and Bob's payoff is $c'-\delta'/2$. Finally, when they both play their new actions their payoff is 10^{-3}.

Notice that playing the new action is an (exact) pure Nash equilibrium. Notice further that when the 2CSP is satisfied, the original equilibrium from the completeness of Lemma 11.1.2 is still a Nash equilibrium since the average payoff to both players from that equilibrium is higher than what they would get by deviating to the new strategy.

To complete the proof we need to argue for soundness. In particular, we claim that if the value of the 2CSP is only $1-\eta$, then every ϵ-approximate Nash equilibrium is 10^{-3}-close (in total variation distance) to the new strategy pure equilibrium. Assume by contradiction that this is not the case, i.e. there is an ϵ-approximate equilibrium where (wlog) Alice assigns probability more than 10^{-3} to old strategies. First, notice that Bob must assign at least 1/2 probability to old strategies: whenever Bob plays the new strategy, Alice has a large incentive to deviate to her new strategy. Therefore, by a symmetric argument, Alice must also assign at least 1/2 probability to old actions.

Finally, the mixed strategy profile restricted to the old strategies must be a 4ϵ-approximate Nash equilibrium of the scaled old game. By Lemma 11.1.2, the expected payoff to one of the players is at most $c'-\delta'$. But then that player has at least $\delta'/4 > \epsilon$ incentive to deviate to her new strategy. □

Chapter 12

Densest k-Subgraph

k-Clique is one of the most fundamental problems in computer science: given a graph, decide whether it has a fully connected induced subgraph on k vertices. Since it was proven NP-complete by Karp [Kar72], extensive research has investigated the complexity of relaxed versions of this problem.

This work focuses on two natural relaxations of k-Clique which have received significant attention from both algorithmic and complexity communities: The first one is to relax "k", i.e. looking for a smaller subgraph:

Problem 12.0.1 (Approximate Max Clique, Informal)**.** Given an n-vertex graph G, decide whether G contains a clique of size k, or all induced cliques of G are of size at most δk for some $1 > \delta(n) > 0$.

The second natural relaxation is to relax the "Clique" requirement, replacing it with the more modest goal of finding a subgraph that is almost a clique:

Problem 12.0.2 (Densest k-Subgraph with perfect completeness, Informal)**.** Given an n-vertex graph G containing a clique of size k, find an induced subgraphs of G of size k with (edge) density at least $(1-\varepsilon)$, for some $1 > \varepsilon > 0$. (More modestly, given an n-vertex graph G, decide whether G contains a clique of size k, or all induced k-subgraphs of G have density at most $(1-\varepsilon)$).

Today, after a long line of research [Fei+96; AS98; Aro+98; Hås99; Kho01; Zuc07] we have a solid understanding of the inapproximability of Problem 12.0.1. In particular, we know that it is NP-hard to distinguish between a graph that has a clique of size k, and a graph whose largest induced clique is of size at most $k' = \delta k$ for $\delta = 1/n^{1-\varepsilon}$ [Zuc07]. The computational complexity of the second relaxation (Problem 12.0.2) remained largely open. There are a couple of (very different) quasi-polynomial algorithms that guarantee finding a $(1-\varepsilon)$-dense k subgraph in every graph containing

a k-clique [FS97; Bar15], suggesting that this problem is not NP-hard. Yet we know neither polynomial-time algorithms, nor general impossibility results for this problem.

In this work we provide a strong evidence that the aforementioned quasi-polynomial time algorithms for Problem 12.0.2 [FS97; Bar15] are essentially tight, assuming the (deterministic) *Exponential Time Hypothesis* (ETH) (Hypothesis 1). In fact, we show that under ETH, both parameters of the above relaxations are simultaneously hard to approximate:

Theorem 12.0.3 (Main Result)**.** *There exists a universal constant $\varepsilon > 0$ such that, assuming the (deterministic) Exponential Time Hypothesis, distinguishing between the following requires time $n^{\tilde{\Omega}(\log n)}$, where n is the number of vertices of G.*

Completeness *G has an induced k-clique; and*

Soundness *Every induced subgraph of G size $k' = k \cdot 2^{-\Omega(\frac{\log n}{\log\log n})}$ has density at most $1-\varepsilon$,*

Our result has implications for two major open problems whose computational complexity remained elusive for more than two decades: The (general) Densest k-Subgraph problem, and the Planted Clique problem.

Densest k-Subgraph

The Densest k-Subgraph problem, $\mathsf{DkS}(\eta, \varepsilon)$, is the same as (the decision version of) Problem 12.0.2, except that in the "completeness" case, G has a k-subgraph with density η, and in the "soundness" case, every k-subgraph is of density at most ε, where $\eta \gg \varepsilon$. Since Problem 12.0.2 is a special case of this problem, our main theorem can also be viewed as a new inapproximability result for $\mathsf{DkS}(1, 1-\varepsilon)$. We remark that the aforementioned quasi-polynomial algorithms for the "perfect completeness" regime completely break in the sparse regime, and indeed it is believed that $\mathsf{DkS}(n^{-\alpha}, n^{-\beta})$ (for $k = n^{\varepsilon}$) in fact requires much more than quasi-polynomial time [Bha+12]. The best to-date approximation algorithm for Densest k-Subgraph due to Bhaskara et. al, is guaranteed to find a k-subgraph whose density is within an $\sim n^{1/4}$-multiplicative factor of the densest subgraph of size k [Bha+10], and thus $\mathsf{DkS}(\eta, \varepsilon)$ can be solved efficiently whenever $\eta \gg n^{1/4} \cdot \varepsilon$ (this improved upon a previous $n^{1/3-\delta}$-approximation of Feige et. al [FKP01]). Making further progress on either the lower or upper bound frontier of the problem is a major open problem.

Several inapproximability results for Densest k-Subgraph were known against specific classes of algorithms [Bha+12] or under incomparable assumptions of Unique

Games with expansion [RS10] and hardness of random k-CNF [Fei02; Alo+11]. The most closely related result is by Khot [Kho06], who shows that the DENSEST k-SUBGRAPH problem has no PTAS unless SAT can be solved in time 2^{n^ε}, as opposed to $2^{n^{1/2+\varepsilon}}$ in our paper. While Khot's work uses a slightly weaker assumption, an important advantage of our work is simplicity: our construction is very simple, especially in contrast to Khot's reduction.

We stress that the result of [Kho06], as well as other aforementioned works, focus on the sub-constant density regime, i.e. they show hardness for distinguishing between a graph where every k-subgraph is sparse, and one where every k-subgraph is even sparser. In contrast, our result has perfect completeness and provides the first *additive* inapproximability for DENSEST k-SUBGRAPH — the best one can hope for as per the upper bound of [Bar15].

Planted Clique

The PLANTED CLIQUE problem is a special case of our problem, where the inputs come from a specific distribution ($G(n,p)$ versus $G(n,p)+$ "a planted clique of size k", where p is some constant[1]). The *Planted Clique Conjecture* ([Alo+07; AKS98; Jer92; Kuc95; FK00; DGGP10]) asserts that distinguishing between the aforementioned cases for $p = 1/2, k = o(\sqrt{n})$ cannot be done in polynomial time, and has served as the underlying hardness assumption in a variety of recent applications including machine-learning and cryptography (e.g. [Alo+07; BR13]) that inherently use the average-case nature of the problem, as well as in reductions to worst-case problems (e.g. [HK11; Alo+11; KZ11; Bal+13; Che+15a; Bad+16]).

The main drawback of average-case hardness assumptions is that many average-case instances (even those of worst-case-hard problems) are in fact tractable. In recent years, the centrality of the planted clique conjecture inspired several works that obtain lower bounds in restricted models of computation [Fel+13; MPW15; DM15; Hop+16; Bar+16]. Nevertheless, a general lower bound for the average-case planted clique problem appears out of reach for existing techniques. Therefore, an important potential application of our result is replacing average-case assumptions such as the planted-clique conjecture, in applications that do not inherently rely on the distributional nature of the inputs (e.g., when the ultimate goal is to prove a worst-case hardness result). In such applications, there is a good chance that planted clique hardness assumptions can be replaced with a more "conventional" hardness assumption, such as the ETH, even when the problem has a quasi-polynomial algo-

[1] PLANTED CLIQUE typically refers to $p = 1/2$, while our hardness result is analogous to $p = 1-\delta$, for a small constant $\delta > 0$. Nevertheless, in almost all applications of PLANTED CLIQUE, hardness for any constant p suffices.

rithm. Recently, such a replacement of the planted clique conjecture with ETH was obtained for the problem of finding an approximate Nash equilibrium with approximately optimal social welfare [BKW15].

We also remark that, while showing hardness for Planted Clique from worst-case assumptions seems beyond the reach of current techniques, our result can also be seen as circumstantial evidence that this problem may indeed be hard. In particular, any polynomial time algorithm (if exists) would have to inherently use the (rich and well-understood) structure of $G(n,p)$.

Followup work by Manurangsi

The soundness in our result was greatly improved by Manurangsi [Man17], who showed that even $\mathsf{DkS}(1, n^{-1/\mathsf{poly}\log\log n})$ may be intractable; i.e. in the NO case the maximal density is almost inverse-polynomial.

Techniques

Our simple construction is inspired by the "birthday repetition" technique: given a 2CSP (e.g. 3COL), we have a vertex for each $\tilde{\Omega}\left(\sqrt{n}\right)$-tuple of variables and assignments (respectively, 3COL vertices and colorings). We connect two vertices by an edge whenever their assignments are consistent and satisfy all 2CSP constraints induced on these tuples. In the completeness case, a clique consists of choosing all the vertices that correspond to a fixed satisfying assignment. In the soundness case (where the value of the 2CSP is low), the "birthday paradox" guarantees that most pairs of vertices (i.e. two $\tilde{\Omega}\left(\sqrt{n}\right)$-tuples of variables) will have a significant intersection (nonempty CSP constraints), thus resulting in lower densities whenever the 2CSP does not have a satisfying assignment. In the language of two-prover games, the intuition here is that the verifier has a "constant chance in catching the players in a lie if they are trying to cheat" in the game while not satisfying the CSP.

While our construction is simple, analyzing it is intricate. The main challenge is to rule out a "cheating" dense subgraph that consists of different assignments to the same variables (inconsistent colorings of the same vertices in 3COL). Intuitively, this is similar in spirit to *proving a parallel repetition theorem where the provers can answer some questions multiple times, and completely ignore other questions.* Continuing with the parallel repetition metaphor, notice that the challenge is doubled: in addition to a cheating prover correlating her answers (the standard obstacle to parallel repetition), each prover can now also correlate which questions she chooses to answer. Our argument follows by showing that a sufficiently large subgraph must accumulate many non-edges (violations of either 2CSP or consistency constraints).

To this end we introduce an information theoretic argument that carefully counts the entropy of choosing a random vertex in the dense subgraph.

We note that our entropy-based argument is completely different from all other known applications of "birthday repetition" to other problems. The main reason is that enforcing consistency is much more difficult in the case of DENSEST k-SUBGRAPH than in other applications because the problem formulation is so simple. In fact, even the follow-up work on the same problem by [Man17] used a completely different (and quite elegant) argument that is based on counting small bi-cliques in any given subgraph.

12.1 Construction (and completeness)

12.1.1 Construction

Let ψ be the **2CSP** instance produced by the reduction in Theorem 2.4.4, i.e. a constraint graph over n variables with alphabet A of constant size. We construct the following graph $G_\psi = (V, E)$:

- Let $\rho := \sqrt{n} \log \log n$ and $k := \binom{n}{\rho}$.
- Vertices of G_ψ correspond to all possible assignments (colorings) to all ρ-tuples of variables in ψ, i.e $V = [n]^\rho \times A^\rho$. Each vertex is of the form $v = (y_{x_1}, y_{x_2}, \ldots, y_{x_\rho})$ where $\{x_1, \ldots, x_\rho\}$ are the chosen variables of v, and y_{x_i} is the corresponding assignment to variable x_i.
- If $v \in V$ *violates any* **2CSP** *constraints*, i.e. if there is a constraint on (x_i, x_j) in ψ which is not satisfied by $\left(y_{x_i}, y_{x_j}\right)$, then v is an isolated vertex in G_ψ.
- Let $u = (y_{x_1}, y_{x_2}, \ldots, y_{x_\rho})$ and $v = (y'_{x'_1}, y'_{x'_2}, \ldots, y'_{x'_\rho})$. $(u, v) \in E$ iff:
  - (u, v) does not *violate any consistency constraints*: for every shared variable x_i, the corresponding assignments agree, $y_{x_i} = y'_{x_i}$;
  - and (u, v) also does not *violate any* **2CSP** *constraints*: for every **2CSP** constraint on $\left(x_i, x'_j\right)$ (if exists), the assignment $\left(y_{x_i}, y'_{x'_j}\right)$ satisfy the constraint.

Notice that the size of our reduction (number of vertices of G_ψ) is $N = \binom{n}{\rho} \cdot |A|^\rho = 2^{\tilde{O}(\sqrt{n})}$.

Completeness If $\mathrm{OPT}(\psi) = 1$, then G_ψ has a k-clique: Fix a satisfying assignment for ψ, and let S be the set of all vertices that are consistent with this assignment. Notice that $|S| = \binom{n}{\rho} = k$. Furthermore its vertices do not violate any consistency constraints (since they agree with a single assignment), or **2CSP** constraints (since we started from a satisfying assignment).

12.2 Soundness

Suppose that $\mathrm{OPT}(\psi) < 1-\eta$, and let $\epsilon_0 > 0$ be some constant to be determined later. We shall show that for any subset S of size $k' \geq k \cdot |V|^{-\epsilon_0/\log\log|V|}$, $\mathrm{den}(S) < 1 - \delta$, where δ is some constant depending on η. The remainder of this section is devoted to proving the following theorem:

Theorem 12.2.1. *If* $\mathrm{OPT}(\psi) < 1 - \eta$, *then* $\forall S \subset V$ *of size* $k' \geq k \cdot |V|^{-\epsilon_0/\log\log|V|}$, $\mathrm{den}(S) < 1 - \delta$ *for some constant* δ.

12.2.1 Setting up the entropy argument

Fix some subset S of size k', and let $v \in_R S$ be a uniformly chosen vertex in S (recall that v is a vector of ρ coordinates, corresponding to labels for a subset of ρ chosen variables). For $i \in [n]$, let X_i denote the indicator variable associated with v such that $X_i = 1$ if the i'th variable appears in v and 0 otherwise. We let Y_i represent the coloring assignment (label) for the i'th variable whenever $X_i = 1$, which is of the form $l \in A$. Throughout the proof, let

$$W_{i-1} = X_{<i}, Y_{<i}$$

denote the i'th prefix corresponding to v. We can write :

$$\begin{aligned} H(Y_i|W_{i-1}, X_i) &= \Pr[X_i = 0] \cdot H(Y_i|W_{i-1}, X_i = 0) \\ &\quad + \Pr[X_i = 1] \cdot H(Y_i|W_{i-1}, X_i = 1) \\ &= \Pr[X_i = 1] \cdot H(Y_i|W_{i-1}, X_i = 1) \end{aligned}$$

since $H(Y_i|W_{i-1}, X_i = 0) = 0$. Notice that since (XY) and v determine each other, and v was uniform on a set of size $|S| = k'$, we have

Observation 12.2.2. $H(XY) = \log k'$.

Thus, in total, the choice of challenge and the choice of assignments should contribute $\log k'$ to the entropy of v. If much of the entropy comes from the assignment

distribution (conditioned on the fixed challenge variables), we will show that S must have many consistency violations, implying that S is sparse. If, on the other hand, almost all the entropy comes from the challenge distribution, we will show that this implies many CSP constraint violations (implied by the soundness assumption). From now on, we denote

$$\alpha_i := H(X_i|X_{<i}, Y_{<i}) \quad \text{and} \quad \beta_i := H(Y_i|X_{\leq i}, Y_{<i}).$$

When conditioning on the i'th prefix, we shall write $\alpha_i(w_{i-1}) := H(X_i|X_{<i}, Y_{<i} = w_{i-1})$, and similarly for $\beta_i(\cdot)$. Also for brevity, we denote

$$q_i := \Pr[X_i = 1] \quad \text{and} \quad q_i(w_{i-1}) := \Pr[X_i = 1|w_{i-1}].$$

Prefix graphs

The consistency constraints induce, for each i, a graph over the prefixes: the vertices are the prefixes, and two prefixes are connected by an edge if their labels are consistent. (We can ignore the **2CSP** constraints for now — the prefix graph will be used only in the analysis of the consistency constraints.) Formally,

Definition 12.2.3 (Prefix graph)**.** For $i \in [n+1]$ let the *i-th prefix graph*, $G_i = (V_i, E_i)$ be defined over the prefixes of length $i-1$ as follows. We say that w_{i-1} is a neighbor of σ_{i-1} if they do not violate any consistency constraints. Namely, for all $j < i$, if $X_j = 1$ for both w_{i-1} and σ_{i-1}, then w_i and σ_i assign the same label Y_j.

In particular, we will heavily use the following notation: let $\mathcal{N}(w_{i-1})$ be the *prefix neighborhood* of w_{i-1}; i.e. it is the set of all prefixes (of length $i-1$) that are consistent with w_{i-1}. For technical issues of normalization, we let $w_{i-1} \in \mathcal{N}(w_{i-1})$, i.e. all the prefixes have self-loops.

Notice that G_{n+1} is defined over the vertices of S (the original subgraph). The set of edges on S is contained in the set of edges of G_{n+1}, since in the latter we only remove pairs that violated consistency constraints (recall that we ignore the **2CSP** constraints).

Unless stated otherwise, we always think of prefixes as weighted by their probabilities. Naturally, we also define the weighted degree and weighted edge density of the prefix graph.

Definition 12.2.4 (Prefix degree and density)**.** The *prefix degree* of w_{i-1} is given by:

$$\deg(w_{i-1}) = \sum_{\sigma_{i-1} \in \mathcal{N}(w_{i-1})} \Pr[\sigma_{i-1}].$$

Similarly, we define the *prefix density* of G_i as:

$$\mathrm{den}(G_i) = \sum_{w_{i-1}} \sum_{\sigma_{i-1} \in \mathcal{N}(w_{i-1})} \Pr[w_{i-1}] \cdot \Pr[\sigma_{i-1}].$$

When it is clear from the context, we henceforth drop the *prefix* qualification, and simply refer to the *neighborhood* or *degree*, etc., of w_{i-1}.

Notice that in G_{n+1}, the probabilities are uniformly distributed. In particular, $\mathrm{den}(G_{n+1}) \geq \mathrm{den}(S)$, since, as we mentioned earlier, the set of edges in S is contained in that of G_{n+1}. Finally, observe also that because we accumulate violations, the density of the prefix graphs is monotonically non-increasing with i.

Observation 12.2.5.

$$\mathrm{den}(G_1) \geq \cdots \geq \mathrm{den}(G_{n+1}) \geq \mathrm{den}(S).$$

Useful approximations

We use the following bounds on α_i and β_i many times throughout the proof:

Fact 12.2.6.

$$\alpha_i = \mathsf{E}\left[H(q_i(w_{i-1}))\right] \leq H(\mathsf{E}\left[q_i(w_{i-1})\right]) = H(q_i)$$

Fact 12.2.7.

$$\beta_i = \mathsf{E}\left[\beta_i(w_{i-1}))\right] \leq \mathsf{E}\left[q_i(w_{i-1}) \cdot \log|A|\right] = q_i \log|A|$$

Proof. The bound on α_i follows from concavity of entropy (Fact 2.6.3). For the second bound, observe that β_i is maximized by spreading q_i mass uniformly over alphabet A. □

We also recall some elementary approximations to logarithms and entropies that will be useful in the analysis. The proofs are deferred to the appendix.

Fact 12.2.8. For $k = \binom{n}{\rho}$ then,

$$\log k = nH\left(\frac{\rho}{n}\right) \pm O\left(\log n\right) = \left(\frac{1}{2} - o\left(1\right)\right)\rho \log n$$

More useful to us will be the following bounds on $\log k'$:

Fact 12.2.9. Let $\epsilon_1 \geq 5\epsilon_0$, and k, k', V, n, ρ as specified in the construction. Then,

$$\log k' \geq \max\left\{\log k, nH\left(\frac{\rho}{n}\right)\right\} - \underbrace{\epsilon_1 \log k / \log n}_{\approx \frac{\epsilon_1}{2} \cdot \rho}.$$

This means that most indices i should contribute roughly $H\left(\frac{\rho}{n}\right)$ entropy to the choice of v.

We will also need the following bound which relates the entropies of a very biased coin and a slightly less biased one:

Fact 12.2.10. Let $1/n \ll |\upsilon| \ll 1$. Then

$$H\left(\frac{1+\upsilon}{n}\right) = H\left(\frac{1}{n}\right) - \frac{\upsilon}{n}\log\frac{1}{n} - (\log e)\,\frac{\upsilon^2}{2n} + O\left(n^{-2}\right) + O\left(\frac{\upsilon^3}{n}\right)$$

A useful lemma: bias implies less entropy

In Fact 12.2.6 we saw that always $\alpha_i \le H(q_i)$. Equality happens only if the q_i-mass is evenly distributed across all prefixes. We argue that if q_i is far from evenly distributed, then the inequality is also far from tight. In particular:

Claim 12.2.11. Let $B \subset V_i$ be a subset of prefixes such that for some $0 < a < b < 1$,

1. $\sum_{w_{i-1}\in B} \Pr[w_{i-1}] \le b$; but also
2. $\sum_{w_{i-1}\in B} \Pr[w_{i-1}] q_i(w_{i-1}) > a$.

Then $\alpha_i \le H(q_i) - q_i \mathsf{D}_{\mathsf{KL}}\left(a \middle\| b\right)$.

Proof. Abusing notation, let $B(\cdot)$ be the indicator variable for $W_{i-1} \in B$. By the data-processing inequality (Fact 2.6.8),

$$\begin{aligned} \alpha_i &= H(X_i \mid W_{i-1}) \\ &\le H(X_i \mid B(W_{i-1})) \\ &= H(X_i) - I(X_i; B(W_{i-1})) \end{aligned} \tag{12.1}$$

Since we can write mutual information as expected KL-divergence (Fact 2.6.10), and KL-divergence is non-negative, we get

$$\begin{aligned} I(X_i; B(W_{i-1})) &= \mathsf{E}_{x_i}\left[\mathsf{D}_{\mathsf{KL}}\left(B(W_{i-1})|x_i \middle\| B(W_{i-1})\right)\right] \\ &\ge q_i \mathsf{D}_{\mathsf{KL}}\left(\Pr[B(W_{i-1}) = 1 \mid x_i = 1] \middle\| B(W_{i-1}) = 1\right) \\ &\ge q_i \mathsf{D}_{\mathsf{KL}}\left(a \middle\| b\right), \end{aligned}$$

where the second inequality follows from the premise assumptions that $\Pr[B(W_{i-1})] \le b$ and $\Pr[B(W_{i-1} = 0 \mid x_i = 1] \ge a$

Plugging into (12.1) we have:

$$\alpha_i \leq H(q_i) - q_i \mathsf{D}_{\mathsf{KL}}\left(a \middle\| b\right). \tag{12.2}$$

□

12.2.2 Consistency violations

In this section, we show that if the total entropy contribution of the assignments $(\sum_i \beta_i))$ is large, there are many consistency violations between vertices, which lead to constant density loss. First, we show that if $\sum_i \beta_i > 5\varepsilon_1 \log k / \log n$, then at least a constant fraction of such entropy is concentrated on *good variables* that contribute to both "types" of entropy.

Definition 12.2.12 (Good Variables)**.** We say that an index i is *good* if

- $\alpha_i \geq H(q_i) - 2q_i \log |A|$
- $\beta_i \geq \frac{1}{2}\varepsilon_1 q_i$

where ε_1 is a constant to be determined later in the proof.

Claim 12.2.13. For any constant ε_1, if $\sum_i \beta_i > 5\varepsilon_1 \log k / \log n$,

$$\sum_{\text{good } i\text{'s}} q_i^2 \geq \left(\frac{1}{5}\varepsilon_1 \rho\right)^2 / (n \log^2 |A|) = \Omega(\rho^2/n).$$

Proof. We want to show that many of the indices i have both a large α_i and a large β_i simultaneously. Let $\iota \subseteq [n]$ denote the set of i such that $\alpha_i + \beta_i < H(q_i) - q_i \log |A|$. We can write

$$\sum_{i \in [n]} (\alpha_i + \beta_i) = \sum_{i \in \iota} (\alpha_i + \beta_i) + \sum_{i \notin \iota} (\alpha_i + \beta_i)$$

Using Facts 12.2.6 and 12.2.7, we have

$$\sum_{i \in [n]} (\alpha_i + \beta_i) \leq \sum_{i \in \iota} (H(q_i) - \beta_i) + \sum_{i \notin \iota} (H(q_i) + \beta_i).$$

Because the subgraph is of size k', from the expansion of $\log k'$ (Fact 12.2.9),

$$\begin{aligned} \sum_{i \in [n]} (\alpha_i + \beta_i) &\geq nH\left(\frac{\rho}{n}\right) - \epsilon_1 \log k / \log n \\ &\geq \sum H(q_i) - \epsilon_1 \log k / \log n, \end{aligned}$$

where the second inequality follows from the concavity of entropy. Plugging into (12.2.2), we have

$$\begin{aligned}\sum_{i\notin\iota}\beta_i &\geq \sum_{i\in\iota}\beta_i - \epsilon_1 \log k/\log n \\ &= \left(\sum_i \beta_i - \sum_{i\notin\iota}\beta_i\right) - \epsilon_1 \log k/\log n\end{aligned}$$

Rearranging, we get

$$\sum_{i\notin\iota}\beta_i \geq \frac{1}{2}\sum_i \beta_i - \epsilon_1 \log k/\log n \tag{12.3}$$

For all the i's in the LHS summation, $\alpha_i \geq H(q_i) - 2q_i \log|A|$ by Fact 12.2.7. From now on, we will consider only i's that satisfy this condition. Now, using the premise on $\sum_i \beta_i$ and (12.3) we have:

$$\begin{aligned}\sum_{i:\alpha_i \geq H(q_i)-2q_i\log|A|} \beta_i &\geq (5/2-1)\epsilon_1 \log k/\log n \\ &\geq 0.7\varepsilon_1\rho,\end{aligned}$$

where the second inequality follows from our approximation for $\log k$ (Fact 12.2.8).

We want to further restrict our attention to i's for which β_i is at least $\frac{1}{2}\varepsilon_1 q_i$ (aka good i's). Note that the above inequality can be decomposed to

$$\sum_{\text{good } i\text{'s}} \beta_i + \sum_{\substack{i:\alpha_i \geq H(q_i)-2q_i\log|A| \\ \beta_i < \frac{1}{2}\varepsilon_1 q_i}} \beta_i \geq 0.7\varepsilon_1\rho$$

Now via a simple sum bound,

$$\sum_{\substack{i:\alpha_i \geq H(q_i)-2q_i\log|A| \\ \beta_i < \frac{1}{2}\varepsilon_1 q_i}} \beta_i \leq \frac{1}{2}\varepsilon_1 \sum_i q_i = \frac{1}{2}\varepsilon_1\rho$$

Rearranging, we get,

$$\sum_{\text{good } i\text{'s}} \beta_i \geq \frac{1}{5}\varepsilon_1\rho$$

By Cauchy-Schwartz we have:

$$\sum_{\text{good } i\text{'s}} \beta_i^2 \geq \left(\frac{1}{5}\varepsilon_1\rho\right)^2/n$$

Finally, since $\beta_i \le q_i \log|A|$,

$$\sum_{\text{good } i\text{'s}} q_i^2 \ge \left(\frac{1}{5}\varepsilon_1\rho\right)^2 / (n\log^2|A|).$$

□

In the same spirit, we now define a notion of a "good" prefix. Intuitively, conditioning on a good prefix leaves a significant amount of entropy on the i'th index. We also require that a good prefix has a high prefix degree; that is, it has many neighbors it could potentially lose when revealing the i-th label.

Definition 12.2.14 (Good Prefixes)**.** We say w_{i-1} is a *good* prefix if:

- i is good;
- $\sum_{\sigma_{i-1}\in\mathcal{N}(w_{i-1})} q_i(\sigma_{i-1})\Pr[\sigma_{i-1}] \ge (1-\varepsilon_2)q_i$;
- $\beta_i(w_{i-1}) \ge \varepsilon_3 q_i(w_{i-1})$,

for $\varepsilon_3 = (\varepsilon_4+\kappa)\log|A|$, with ε_4 an arbitrarily small constant that denotes the fraction of assignments that disagree with the majority of the assignments, $\kappa = \Theta(1/\log|A|)$, and ε_2 a constant that satisfies $\delta = \left(\frac{\varepsilon_2}{|A|^{2/\varepsilon_2}}\right)^4$, with $\mathrm{den}(S) = 1-\delta$.

In the following claim, we show that these prefixes contribute some constant fraction of entropy, assuming that our subset is dense.

Claim 12.2.15. If $\mathrm{den}(S) > 1-\delta$, where $\delta = \left(\frac{\varepsilon_2}{|A|^{2/\varepsilon_2}}\right)^4$ and $\varepsilon_1 \ge 4\varepsilon_2\log|A| + 8\varepsilon_3$, then for every good index i, it holds that

$$\sum_{\text{good } w_{i-1}\text{'s}} \Pr[w_{i-1}]\beta_i(w_{i-1}) \ge \beta_i/4$$

Proof. We begin by proving that most prefixes satisfy the degree condition of Definition 12.2.14. Let w_{i-1} be *popular* if i is a good variable and its degree in the prefix graph G_i is at least $deg(w_{i-1}) := \sum_{\sigma_{i-1}\in\mathcal{N}(w_{i-1})}\Pr[\sigma_{i-1}] \ge 1-\sqrt{\delta}$. Recall that $\mathrm{den}(G_i) \ge \mathrm{den}(S) \ge (1-\delta)$ (by Observation 12.2.5). Thus by Markov inequality, at most $\sqrt{\delta}$-fraction of the prefixes are unpopular.

We now argue that:

$$\sum_{\text{unpopular } w_{i-1}\text{'s}} \Pr[w_{i-1}]q_i(w_{i-1}) \le \varepsilon_2 q_i. \tag{12.4}$$

Otherwise, by Claim 12.2.11, $\alpha_i \geq H(q_i) - q_i \mathsf{D}_{\mathsf{KL}}\left(\varepsilon_2 \middle\| \sqrt{\delta}\right)$. On the other hand, recall that since i is good, $\alpha_i \geq H(q_i) - 2q_i \log|A|$. Recall also that $\delta = \left(\frac{\varepsilon_2}{|A|^{2/\varepsilon_2}}\right)^4$, and therefore $\mathsf{D}_{\mathsf{KL}}\left(\varepsilon_2 \middle\| \sqrt{\delta}\right) \geq 2\log|A|$. Thus, we get a contradiction.

Ineq. (12.4) implies that even if the assignment is uniform over the alphabet, the contribution to $\sum \beta_i$ from unpopular prefixes is small:

$$\begin{aligned}\sum_{\text{unp.}} \Pr[w_{i-1}]\beta_i(w_{i-1}) &\leq \sum_{\text{unp.}} \Pr[w_{i-1}]q_i(w_{i-1})\log|A| \\ &\leq \varepsilon_2 q_i \log|A| \\ &\leq \frac{1}{4}\varepsilon_1 q_i \leq \frac{1}{2}\beta_i\end{aligned}$$

where first inequality follows from Fact 12.2.7, second from (12.4), third from our setting of $\varepsilon_1 \geq 4\varepsilon_2 \log|A|$, and fourth from $\beta_i \geq \frac{1}{2}\varepsilon_1 q_i$ since i is good. Therefore,

$$\begin{aligned}\sum_{\text{pop.}} \Pr[w_{i-1}]\beta_i(w_{i-1}) &= \beta_i - \sum_{\text{unp.}} \Pr[w_{i-1}]\beta_i(w_{i-1}) \\ &\geq \beta_i/2\end{aligned}$$

Using a similar argument, we show that for any popular w_{i-1}, most of the q_i mass is concentrated on its neighbors. Consider any popular w_{i-1}, and let $\mathcal{N}^C(w_{i-1})$ denote the complement of $\mathcal{N}(w_{i-1})$. Then we can rewrite α_i as:

$$\begin{aligned}\alpha_i = &\sum_{\sigma_{i-1}\in\mathcal{N}(w_{i-1})} \Pr[\sigma_{i-1}]\alpha_i(\sigma_{i-1}) \\ &+ \sum_{\sigma_{i-1}\in\mathcal{N}^C(w_{i-1})} \Pr[\sigma_{i-1}]\alpha_i(\sigma_{i-1})\end{aligned}$$

Notice that since w_{i-1} is popular, $\mathcal{N}^C(w_{i-1})$ has measure at most $\sqrt{\delta}$. Thus, if an ε_2-fraction of the q_i mass is concentrated on $\mathcal{N}^C(w_{i-1})$, Claim 12.2.11 implies:

$$\alpha_i \leq H(q_i) - q_i \mathsf{D}_{\mathsf{KL}}\left(\varepsilon_2 \middle\| \sqrt{\delta}\right),$$

which (as in (12.4)) would yield a contradiction to i being a good variable. Therefore every popular prefix also satisfies the q_i-weighted condition on the degree:

$$\sum_{\sigma_{i-1}\in\mathcal{N}(w_{i-1})} \Pr[\sigma_{i-1}]q_i(\sigma_{i-1}) \geq (1-\varepsilon_2)\, q_i \tag{12.5}$$

Recall that a prefix w_{i-1} is good if it also satisfies $\beta_i(w_{i-1}) \geq \varepsilon_3 \cdot q_i(w_{i-1})$. Fortunately, prefixes that violate this condition (i.e. those with small $\beta_i(w_{i-1})$), cannot account for much of the weight on β_i:

$$\sum_{\beta_i(w_{i-1}) < \varepsilon_3 q_i(w_{i-1})} \Pr[w_{i-1}]\beta_i(w_{i-1}) \leq \varepsilon_3 q_i.$$

Since i is good and $\varepsilon_1 \geq 8\varepsilon_3$, this implies:

$$\sum_{\text{good } w_{i-1}\text{'s}} \Pr[w_{i-1}]\beta_i(w_{i-1}) \geq \beta_i/2 - \varepsilon_3 q_i \geq \beta_i/4$$

since

$$\varepsilon_3 q_i \leq \frac{1}{8}\varepsilon_1 q_i \leq \frac{1}{4}\beta_i$$

where last inequality follows from i being good. □

Corollary 12.2.16. *For every good index i,*

$$\sum_{good\ w_{i-1}\text{'s}} \Pr[w_{i-1}]q_i(w_{i-1}) \geq \frac{\varepsilon_1}{8\log|A|} q_i.$$

Proof.

$$\begin{aligned}\sum_{\text{good}} \Pr[w_{i-1}]q_i(w_{i-1}) &\geq \sum_{\text{good}} \Pr[w_{i-1}]\beta_i/\log|A| \\ &\geq \beta_i/(4\log|A|) \\ &\geq \frac{\varepsilon_1}{8\log|A|} q_i.\end{aligned}$$

Where the first inequality follows by Fact 12.2.7, the second by Claim 12.2.15, and the last by definition of good i's. □

With Claim 12.2.13 and Corollary 12.2.16, we are ready to prove the main lemma of this section:

Lemma 12.2.17 (Labeling Entropy Bound)**.** *If $\sum_i H(Y_i|X_{\leq i}, Y_{<i}) > \frac{5\varepsilon_1 \log k}{\log n}$, then $\text{den}(S) < 1 - \delta$.*

Proof. Assume for a contradiction that $\text{den}(S) \geq 1 - \delta$. For prefix w_{i-1}, let $\mathcal{D}_{w_{i-1}}$ denote the induced distribution on labels to the i-th variable, conditioned on w_{i-1} and $x_i = 1$. (If $q_i(w_{i-1}) = 0$, take an arbitrary distribution.) After revealing each variable i, the loss in prefix density is given by the probability of "fresh violations":

the sum over all prefix edges (w_{i-1}, σ_{i-1}) of the probability that they assign different labels to the i-th variable:

$$\begin{aligned} \mathrm{den}(G_i) - \mathrm{den}(G_{i+1}) = \sum_{w_{i-1}} \sum_{\sigma_{i-1} \in \mathcal{N}(w_{i-1})} \dots \\ \Big(\Pr[w_{i-1}] \Pr[\sigma_{i-1}] q_i(w_{i-1}) q_i(\sigma_{i-1}) \Big) \Pr_{\substack{Y_i \sim \mathcal{D}_{w_{i-1}} \\ Y_i' \sim \mathcal{D}_{\sigma_{i-1}}}} [Y_i \neq Y_i'] \end{aligned} \tag{12.6}$$

We now lower-bound $\Pr_{\mathcal{D}_{w_{i-1}} \times \mathcal{D}_{\sigma_{i-1}}}[Y_i \neq Y_i']$ for good w_{i-1} (notice that we assume nothing about σ_{i-1}). A simple calculation shows that for $\kappa < 1/2$, if

$$\begin{aligned} &\beta_i(w_{i-1}) \geq \\ &\quad (\kappa \log |A| - \kappa \log \kappa - (1-\kappa) \log(1-\kappa))\, q_i(w_{i-1}), \end{aligned}$$

then the probability mass (under $\mathcal{D}(w_{i-1})$) on the most common label is at most $1 - \kappa$. Observe that this probability is an upper bound on $\Pr_{\mathcal{D}_{w_{i-1}} \times \mathcal{D}_{\sigma_{i-1}}}[Y_i = Y_i']$. For good w_{i-1}, we indeed have

$$\begin{aligned} &\beta_i(w_{i-1}) \geq \varepsilon_3 q_i(w_{i-1}) \geq \\ &(\varepsilon_4 \log |A| - \varepsilon_4 \log \varepsilon_4 - (1-\varepsilon_4) \log(1-\varepsilon_4))\, q_i(w_{i-1}), \end{aligned}$$

where the second inequality follows from choice of ε_4. Therefore $\Pr_{\mathcal{D}_{w_{i-1}} \times \mathcal{D}_{\sigma_{i-1}}}[Y_i \neq Y_i'] \geq \varepsilon_4$.

We now have, for every good index i,

$$\begin{aligned} \mathrm{den}(G_i) - \mathrm{den}(G_{i+1}) &\geq \sum_{\text{good } w_{i-1}\text{'s}} \sum_{\sigma_{i-1} \in \mathcal{N}(w_{i-1})} \varepsilon_4 \cdot \\ &\quad \cdot \Big(\Pr[w_{i-1}] \Pr[\sigma_{i-1}] q_i(w_{i-1}) q_i(\sigma_{i-1}) \Big) \\ &\geq \varepsilon_4 q_i (1 - \varepsilon_2) \sum_{\text{good } w_{i-1}\text{'s}} \Pr[w_{i-1}] q_i(w_{i-1}) \\ &\geq \frac{\varepsilon_1 \varepsilon_4}{10 \log |A|} q_i^2, \end{aligned}$$

where the first inequality follows by Eq. (12.6); the second definition of good prefix; and the last by Corollary 12.2.16 and $\varepsilon_2 < 0.2$.

Finally, summing over all good i's, we get a negative density for S, which is, of course, a contradiction. By Observation 12.2.5 we have:

$$\begin{aligned}
1 - \mathrm{den}(S) &\geq \mathrm{den}(G_1) - \mathrm{den}(G_{n+1}) \\
&= \sum_i \mathrm{den}(G_i) - \mathrm{den}(G_{i+1}) \\
&\geq \sum_{\text{good } i\text{'s}} \mathrm{den}(G_i) - \mathrm{den}(G_{i+1}) \\
&\geq \sum_{\text{good } i\text{'s}} \left(\frac{\varepsilon_1\varepsilon_4}{10\log|A|}\right) q_i^2 \\
&\geq \left(\frac{\varepsilon_1^3\varepsilon_4}{250\log^3|A|}\right)\rho^2/n = \Omega(\rho^2/n),
\end{aligned}$$

where the last inequality follows by Claim 12.2.13. □

12.2.3 2CSP violation

Intuitively, if $\sum_i H(X_i|X_{<i}, Y_{<i})$ is large, then the subgraph approximately corresponds to assignments to all subsets in $\binom{[n]}{\rho}$. More specifically, in this section we show that most of the constraints appear approximately as frequently as we expect. Since in any assignment, a constant fraction of them must be violated, this implies (eventually) that a constant fraction of the edges have a violated constraint.

First, we show that most of the variables appear approximately as frequently as we expect by showing that most of them are "typical."

Definition 12.2.18 (Typical variables)**.** Prefix w_{i-1} is *typical* if

$$(1-\varepsilon_5)\cdot\rho/n < \Pr[X_i = 1|w_{i-1}] < (1+\varepsilon_5)\cdot\rho/n,$$

where ε_5 is some constant such that $\left(\frac{\log e}{8}\right)\varepsilon_5^4 > 14\varepsilon_1$.
Similarly, we say that variable x_i is *typical* if

$$\sum_{\text{typical } w_{i-1}\text{'s}} \Pr[w_{i-1}] \geq 1 - \varepsilon_5$$

Claim 12.2.19. If $\sum_i H(X_i|X_{<i}, Y_{<i}) \geq \left(1 - \frac{6\varepsilon_1}{\log n}\right)\log k = \log k - \Theta(\rho)$, then all but at most $\varepsilon_5 n$ variables are typical.

Proof. Assume by contradiction that there are $\varepsilon_5 n$ atypical variables. That is $\varepsilon_5 n/2$ variables x_i appear with probability at least $(1+\varepsilon_5)\cdot\rho/n$ (or at most $(1-\varepsilon_5)\cdot\rho/n$) for an $(\varepsilon_5/2)$-fraction of the prefixes w_{i-1}. Now, subject only to this constraint and maintaining the correct expected number of variables in each vertex, the entropy is maximized by spreading the $(\varepsilon_5^3/4)$-loss in frequency evenly across all other prefixes and variables. That is on the atypical prefixes, labels are assigned with probability $(1+\varepsilon_5)\,\rho/n$, and with probability $\left(1-\frac{\varepsilon_5^3/4}{1-\varepsilon_5^2/4}\right)\rho/n$ on the rest. Thus,

$$\sum_i H(X_i|X_{<i},Y_{<i}) < \frac{\varepsilon_5^2}{4}n\cdot H\left((1+\varepsilon_5)\,\rho/n\right)+\left(1-\frac{\varepsilon_5^2}{4}\right)nH\left(\left(1-\frac{\varepsilon_5^3/4}{1-\varepsilon_5^2/4}\right)\rho/n\right)$$

Recall from Fact 12.2.10 the expansion of the entropy function:

$$H\left(\frac{1+v}{n}\right)=H\left(\frac{1}{n}\right)-\frac{v}{n}\log\frac{1}{n}-\left(\frac{\log e}{2}\right)\frac{v^2}{n}+O\left(n^{-2}\right)+O\left(\frac{v^3}{n}\right)$$

Therefore,

$$\begin{aligned}\sum_i H(X_i|X_{<i},Y_{<i}) &< \frac{\varepsilon_5^2}{4}n\left[H\left(\frac{\rho}{n}\right)-\varepsilon_5\frac{\rho}{n}\log\frac{\rho}{n}-\left(\frac{\log e}{2}\right)\frac{\rho}{n}\cdot\varepsilon_5^2+O\left(\left(\frac{\rho}{n}\right)^2\right)+O\left(\frac{\rho}{n}\varepsilon_5^3\right)\right]\\ &\quad+\left(1-\frac{\varepsilon_5^2}{4}\right)n\left[H\left(\frac{\rho}{n}\right)+\left(\frac{\varepsilon_5^3/4}{1-\varepsilon_5^2/4}\right)\frac{\rho}{n}\log\frac{\rho}{n}+O\left(\left(\frac{\rho}{n}\right)^2\right)+O\left(\frac{\rho}{n}\varepsilon_5^6\right)\right]\\ &= n\left[H\left(\frac{\rho}{n}\right)-\left(\frac{\log e}{8}\right)\frac{\rho}{n}\cdot\varepsilon_5^4+O\left(\left(\frac{\rho}{n}\right)^2\right)+O\left(\frac{\rho}{n}\varepsilon_5^5\right)\right]\end{aligned}$$

Recall that $-2\log\frac{\rho}{n}<\log n$. Thus for $\left(\frac{\log e}{8}\right)\varepsilon_5^4>14\varepsilon_1$, we have that

$$\left(\frac{\log e}{8}\right)\frac{\rho}{n}\cdot\varepsilon_5^4-O\left(\left(\frac{\rho}{n}\right)^2\right)-O\left(\frac{\rho}{n}\varepsilon_5^5\right)>\frac{\rho}{n}\cdot 12\varepsilon_1>-\frac{\rho}{n}\log\frac{\rho}{n}\cdot 24\varepsilon_1/\log n>(12\varepsilon_1/\log n)\,H\left(\frac{\rho}{n}\right)$$

and therefore,

$$\sum_i H(X_i|X_{<i},Y_{<i})<(1-12\varepsilon_1/\log n)\,nH\left(\frac{\rho}{n}\right)<(1-6\varepsilon_1/\log n)\log k,$$

where the second inequality follows from Fact 12.2.8. Thus we have reached a contradiction. Notice that the $\left(\frac{\log e}{8}\right)\frac{\rho}{n}\cdot\varepsilon_5^4$ term of missing entropy is symmetric (but not the negligible higher order terms); i.e. the same derivation can be used to show a contradiction when many variables appear with probability less than $(1-\varepsilon_5)\,\rho/n$. □

Definition 12.2.20. Let $\mathcal{I}(u,v)$ be defined as the number of (i,j) pairs such that

- In the original **2CSP** instance ψ, there exists an edge (constraint) between typical variables x_i and x_j.
- $X_i = 1$ for u and $X_j = 1$ for v.
- u_{i-1} and v_{j-1} are typical prefixes, where u_{i-1} denotes the prefix represented by u for $X_{<i}, Y_{<i}$, similarly for v_{j-1}.

Intuitively, $\mathcal{I}(u,v)$ is the number of "tests" of **2CSP**-constraints between vertices u, v, when restricting to typical prefixes and variables. We now use the properties of typical prefixes and constraints to show that $\mathcal{I}(u,v)$ behaves "nicely".

Claim 12.2.21. $\mathsf{E}_{u,v}\left[\mathcal{I}\left(u,v\right)\right] \geq \left(1-\varepsilon_7\right)\rho^2/n$ and $\mathsf{E}_{u,v}\left[\mathcal{I}^2\left(u,v\right)\right] \leq \left(1+2\varepsilon_7\right)d^4\left(\mathsf{E}_{u,v}\left[\mathcal{I}\left(u,v\right)\right]\right)^2$, where ε_7 is some constant $\varepsilon_7 \geq 6\varepsilon_5 + \Theta(\varepsilon_5^2)$.

Proof. For any $i, j \in [n]$, we say that $i \in \mathcal{N}^{2CSP}(j)$ if there is a constraint on (x_i, x_j). For the proof of this claim, we also abuse notation and denote $i \in v$ when i is typical, v_{i-1} is a typical prefix, and $X_i = 1$ for v. We also say that $i \in \mathcal{N}(u)$ if i is a typical variable, $i \in \mathcal{N}^{2CSP}(j)$, and $j \in u$ (for some $j \in [n]$). (Do not confuse this notation with prefix neighborhood in the prefix graph.) We can now lower bound the expectation of $\mathcal{I}(u,v)$ as:

$$\mathsf{E}_{u,v}\left[\mathcal{I}\left(u,v\right)\right] \geq \mathsf{E}_u\left[\sum_{i\in\mathcal{N}(u)} \Pr_v\left[i \in v\right]\right]$$

Notice that this bound may not be tight since any $i \in v$ can potentially have d neighbors in u. Thus our upper bound is:

$$\mathsf{E}_{u,v}\left[\mathcal{I}\left(u,v\right)\right] \leq d \cdot \mathsf{E}_u\left[\sum_{i\in\mathcal{N}(u)} \Pr_v\left[i \in v\right]\right]$$

By definition of typical variables, for each typical i, $i \in v$ with probability at least $\left(1-\varepsilon_5\right)^2\rho/n$; thus,

$$\mathsf{E}_{u,v}\left[\mathcal{I}\left(u,v\right)\right] \geq \mathsf{E}_u\left[\sum_{i\in\mathcal{N}(u)} \left(1-\varepsilon_5\right)^2\rho/n\right] = \left(1-\varepsilon_5\right)^2\rho/n \cdot \mathsf{E}_u\left[\left|\mathcal{N}\left(u\right)\right|\right] \tag{12.7}$$

All but $\varepsilon_5 n$ variables are typical, so all but $2\varepsilon_5 n$ variables are typical and have at least one typical neighbor. We restrict our attention to the set of such variables and

fix one typical neighbor for each; this neighbor appears in u with probability at least $(1-\varepsilon_5)^2 \rho/n$. Therefore,

$$\mathsf{E}_u\left[|\mathcal{N}(u)|\right] \geq (1-2\varepsilon_5)n \cdot \left((1-\varepsilon_5)^2\rho/n\right) \geq (1-4\varepsilon_5)\rho \tag{12.8}$$

Combining (12.7) and (12.8), we get the desired bound:

$$\mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right] \geq \left((1-\varepsilon_5)^2 \rho/n\right)(1-4\varepsilon_5)\rho \geq (1-\varepsilon_7)\,\rho^2/n. \tag{12.9}$$

Similarly, for the variance we have

$$\begin{aligned}
\mathsf{E}_{u,v}\left[\mathcal{I}^2(u,v)\right] &\leq d^2 \cdot \mathsf{E}_{u,v}\left(\sum_{i \in v \cap \mathcal{N}(u)} 1\right)^2 \\
&= d^2 \cdot \mathsf{E}_{u,v}\left[\sum_{i \neq j \in v \cap \mathcal{N}(u)} 1 + \sum_{i \in v \cap \mathcal{N}(u)} 1\right] \\
&\leq d^2 \cdot \mathsf{E}_u\left[2 \sum_{i<j \in \mathcal{N}(u)} \Pr_v[i \in v] \Pr_v[j \in v \mid i \in v]\right] + d^2 \cdot \mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right].
\end{aligned}$$

Since for every prefix, each variable receives a typical assignment with probability at most $(1+\varepsilon_5)\cdot\rho/n$, we have that

$$\begin{aligned}
\mathsf{E}_{u,v}\left[\mathcal{I}^2(u,v)\right] &\leq 2d^2 \cdot \mathsf{E}_u\left[\sum_{i<j \in \mathcal{N}(u)} \left((1+\varepsilon_5)\cdot\rho/n\right)^2\right] + d^2 \cdot \mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right] \\
&\leq \left((1+\varepsilon_5)\cdot\rho/n\right)^2 \cdot 2d^2 \cdot \mathsf{E}_u\binom{|\mathcal{N}(u)|}{2} + d^2 \cdot \mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right] \qquad (12.10)
\end{aligned}$$

We would like to bound $\mathsf{E}_u\binom{\mathcal{N}(u)}{2}$.

$$\begin{aligned}
\mathsf{E}_u\binom{\mathcal{N}(u)}{2} &= \sum_{i<j} \sum_{k \in \mathcal{N}^{2CSP}(i)} \sum_{l \in \mathcal{N}^{2CSP}(j)} \Pr_u[k \in u] \Pr_u[l \in u \mid k \in u] \\
&= \sum_{i<j} \sum_{\substack{k \in \mathcal{N}^{2CSP}(i) \\ l \in \mathcal{N}^{2CSP}(j)}} \Pr_u[k \in u] \Pr_u[l \in u \mid k \in u] \qquad (12.11) \\
&\qquad \text{and } k<l \\
&\quad + \sum_{i<j} \sum_{\substack{k \in \mathcal{N}^{2CSP}(i) \\ l \in \mathcal{N}^{2CSP}(j)}} \Pr_u[l \in u] \Pr_u[k \in u \mid l \in u] \qquad (12.12) \\
&\qquad \text{and } k>l \\
&\quad + \sum_{i<j} \sum_{k \in \mathcal{N}^{2CSP}(i) \cap \mathcal{N}^{2CSP}(j)} \Pr_u[k \in u] \qquad (12.13)
\end{aligned}$$

For the first two summands, we can use the condition on the prefixes to conclude that

$$(12.11) + (12.12) \leq \binom{n}{2} d^2 \left((1+\varepsilon_5)\cdot\rho/n\right)^2$$

Whereas to bound the third summand we first change the order of summation:

$$\begin{aligned}(12.13) &= \sum_k \Pr_u[k \in u] \cdot \left|\left\{(i,j) : i \neq j \text{ and } k \in \mathcal{N}^{2CSP}(i) \cap \mathcal{N}^{2CSP}(j)\right\}\right| \\ &\leq ((1+\varepsilon_5)\cdot\rho)\binom{d}{2} = O(\rho)\end{aligned}$$

Summing the last two inequalities, we have

$$2 \cdot \mathsf{E}_u\left(\binom{|\mathcal{N}(u)|}{2}\right) \leq d^2\left((1+\varepsilon_5)\cdot\rho\right)^2 + O(\rho) \leq (1+\varepsilon_5)^3 d^2\rho^2$$

Plugging back into (12.10):

$$\mathsf{E}_{u,v}\left[\mathcal{I}^2(u,v)\right] \leq (1+\varepsilon_5)^5 d^4\rho^4/n^2 + d^2 \cdot \mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right]$$

Using (12.9) and the fact that $\rho = \sqrt{n}\log\log n \gg \sqrt{n}$, this gives

$$\begin{aligned}\mathsf{E}_{u,v}\left[\mathcal{I}^2(u,v)\right] &\leq \frac{d^4(1+\varepsilon_5)^5}{1-\varepsilon_7}\left(\mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right]\right)^2 + d^2 \cdot \mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right] \\ &\leq (1+2\varepsilon_7)\, d^4\left(\mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right]\right)^2\end{aligned}$$

□

It will also be convenient to count the number of tests between a pair of variables.

Definition 12.2.22. For any pair of typical $(i,j) \in \psi$, let $\mathcal{I}^\intercal(i,j)$ be defined as the number of $(u,v) \in (S \times S)$ pairs such that

- $X_i = 1$ for u and $X_j = 1$ for v.
- u_{i-1} and v_{j-1} are typical prefixes, where u_{i-1} denotes the prefix represented by u for $X_{<i}, Y_{<i}$, similarly for v_{j-1}.

We now have two ways to count the total number of tests between typical prefixes to typical variables:

Observation 12.2.23. $\sum_{(u,v)\in(S\times S)} \mathcal{I}(u,v) = \sum_{(i,j)\in\psi} \mathcal{I}^\intercal(i,j)$.

Furthermore, since i and j are typical, the number of tests between also behaves "nicely":

Observation 12.2.24. For every typical $(i,j) \in \psi$, we have $\mathcal{I}^{\top}(i,j) \in |S|^2\rho^2/n^2\Big[\,(1-\varepsilon_5)^4, (1+\varepsilon_5)^2\,\Big]$.

Proof.

$$\begin{aligned}\mathcal{I}^{\top}(i,j) &= \sum_{\text{typical } u_{i-1}\text{'s}} |S|\cdot \Pr[u_{i-1}]\Pr\left[X_i = 1 \mid u_{i-1}\right] \sum_{\text{typical } v_{j-1}\text{'s}} |S|\cdot \Pr\left[v_{j-1}\right]\Pr\left[X_j = 1 \mid v_{j-1}\right] \\ &\in |S|^2\rho^2/n^2\Big[\,(1-\varepsilon_5)^4, (1+\varepsilon_5)^2\,\Big].\end{aligned}$$

□

Armed with these Claims 12.2.19 and 12.2.21 and Observations 12.2.23 and 12.2.24, we are now ready to prove the main lemma of this section. Recall that the soundness of the 2CSP we started with is $1-\eta$ for a small constant η.

Lemma 12.2.25. *If* $\sum_i H(X_i|X_{<i}, Y_{<i}) \geq \left(1 - \frac{6\varepsilon_1}{\log n}\right)\log k$, *then* $\delta(S) < 1-\delta$, *where* $\delta < \frac{\varepsilon_6^2}{d^4(1+2\varepsilon_7)}$ *and* $\varepsilon_6 = (\eta/2-\varepsilon_5)\,(1/|A|^2)\,\frac{(1-\varepsilon_5)^4}{(1+\varepsilon_5)^2}$.

Proof. Let the *mode assignment* be the assignment $\mathcal{A}\colon[n] \to A$ which assigns to each variable x_i its most common typical assignment (i.e. assignment after a typical prefix), breaking ties arbitrarily. In particular, at least $1/|A|$ of the typical assignments for x_i are equal to $\mathcal{A}(i)$. Of course, this assignment cannot satisfy more than a $(1-\eta)$-fraction of the constraints in the original **2CSP**; after removing the $\varepsilon_5 n$ atypical variables, $(\eta/2-\varepsilon_5)\,dn$ constraints out of the $dn/2$ constraints must still be unsatisfied.

Recall that the number of tests for each constraint over typical variables, $\mathcal{I}^{\top}(i,j)$, is approximately the same for every pair of (i,j) — up to a $\frac{(1-\varepsilon_5)^4}{(1+\varepsilon_5)^2}$-multiplicative factor (Observation 12.2.24). Therefore, the total fraction of tests over unsatisfied constraints, out of all tests, is approximately proportional to the fraction of unsatisfied constraints:

$$\sum_{\substack{\text{typical,}\\ \text{unsatisfied}\\ (i,j)\text{'s}}} \mathcal{I}^{\intercal}(i,j) \geq \frac{(1-\varepsilon_5)^4}{(1+\varepsilon_5)^2} \cdot \frac{\left|\left\{\text{typical, unsatisfied } (i,j)\text{'s}\right\}\right|}{\left|\left\{\text{typical } (i,j) \in \psi\right\}\right|} \cdot \sum_{(i,j)\in\psi} \mathcal{I}^{\intercal}(i,j)$$

$$\geq \frac{(1-\varepsilon_5)^4}{(1+\varepsilon_5)^2} \cdot \frac{(\eta/2-\varepsilon_5)\,dn}{dn/2} \cdot \sum_{(i,j)\in\psi} \mathcal{I}^{\intercal}(i,j)$$

$$= \frac{(1-\varepsilon_5)^4}{(1+\varepsilon_5)^2} \cdot (\eta - 2\varepsilon_5) \cdot \sum_{(u,v)\in(S\times S)} \mathcal{I}(u,v) \qquad \text{(Observation 12.2.23)}$$

For each such pair (i,j), on at least a $1/|A|^2$-fraction of the tests both variables receive the mode assignment, so the constraint is violated[2]. Thus the total number of violations is at least $\varepsilon_6 \sum_{(u,v)\in(S\times S)} \mathcal{I}(u,v)$ (where $\varepsilon_6 = (\eta/2-\varepsilon_5)\,(1/|A|^2)\,\frac{(1-\varepsilon_5)^4}{(1+\varepsilon_5)^2}$).

Finally, we show that so many violations cannot concentrate on less than a δ-fraction of the pairs $u, v \in S$; otherwise:

$$\sum_{(u,v)\in(S\times S)\setminus E} \mathcal{I}^2(u,v) \geq \frac{1}{\delta|S|^2}\left(\sum_{(u,v)\in(S\times S)\setminus E} \mathcal{I}(u,v)\right)^2 \qquad \text{(Cauchy-Schwartz)}$$

$$\geq \frac{1}{\delta|S|^2}\left(\varepsilon_6 \sum_{(u,v)\in(S\times S)} \mathcal{I}(u,v)\right)^2$$

$$= \frac{|S|^2\varepsilon_6^2}{\delta}\left(\mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right]\right)^2;$$

yet by Claim 12.2.21,

$$\sum_{(u,v)\in(S\times S)\setminus E} \mathcal{I}^2(u,v) \leq \sum_{(u,v)\in S\times S} \mathcal{I}^2(u,v) \leq (1+2\varepsilon_7)\,d^4|S|^2\left(\mathsf{E}_{u,v}\left[\mathcal{I}(u,v)\right]\right)^2.$$

Thus we have a contradiction since $d^4(1+2\varepsilon_7) < \varepsilon_6^2/\delta$ by our setting of δ. Therefore we have **2CSP**-violations in more than a δ-fraction of the pairs $u, v \in S$. □

With Lemma 12.2.17 and Lemma 12.2.25, we can now complete the proof of Theorem 12.2.1.

[2]We remark that a more careful analysis of the expected number of violations would allow one to save an $|A|^2$-factor in the value of ε_6. Since it does not qualitatively affect the result, we opt for the simpler analysis.

Proof of Theorem 12.2.1. Recall that $\sum_i \alpha_i + \beta_i = \log k' \geq (1 - \frac{\varepsilon_1}{\log n}) \log k$ by Fact 12.2.9. If $\sum_i \beta_i > (\frac{5\varepsilon_1}{\log n}) \log k$, then by Lemma 12.2.17, $\delta(S) < 1 - \delta$. Otherwise, if $\sum_i \alpha_i > (1 - \frac{6\varepsilon_1}{\log n}) \log k$, by Lemma 12.2.25, $\delta(S) < 1 - \delta$. □

Chapter 13

Community detection

Identifying communities is a central graph-theoretic problem with important applications to sociology and marketing (when applied to social networks), biology and bioinformatics (when applied to protein interaction networks), and more (see e.g. Fortunato's classic survey [For10]). Defining what exactly is a *community* remains an interesting problem on its own (see Arora et al [Aro+12] and Borgs et al [Bor+16] for excellent treatment from a theoretical perspective). Ultimately, there is no single "right" definition, and the precise meaning of community should be different for social networks and protein interaction networks.

In this paper we focus on the algorithmic questions arising from one of the simplest and most canonical definitions, which has been considered by several theoretical computer scientists [Mis+08; Aro+12; Bal+13] (see Subsection 13.0.1 for further discussion):

Definition 13.0.1 ((α, β)-Community)**.** Given an undirected graph $G = (V, E)$ an (α, β)-community is a subset $S \subseteq V$ that satisfies:

Strong ties inside the community For every $v \in S$, $|\{v\} \times S| \cap E \geq \alpha \cdot |S|$; and

Weak ties to nodes outside the community For every $u \notin S$, $|\{u\} \times S| \cap E \leq \beta \cdot |S|$.

Arora et al [Aro+12, Theorem 3.1] gave a simple quasi-polynomial ($n^{O(\log n)}$) time for detecting (α, β)-communities whenever $\alpha - \beta$ is at least some positive constant. Similar to the meta-algorithm we described in the introduction, [Aro+12]'s algorithm enumerates over $O(\log n)$-tuples of vertices. For each tuple, consider the set of vertices that are neighbors of an $(\alpha + \beta)/2$-fraction of the tuple; test whether this candidate set is indeed a community.

Here we show that, for *every* constants $\alpha > \beta \in (0,1]$, community detection requires quasi-polynomial time (assuming ETH). For example, when $\alpha = 1$ and $\beta = 0.01$, this means that we can hide a clique C, such that every single vertex not in C is connected to at most 1% of C. Our main result is actually a much stronger inapproximability: even in the presence of a $(1, o(1))$-community, finding any $(\beta + o(1), \beta)$-community is hard.

Theorem 13.0.2. *For every n there exists an $\epsilon = \epsilon(n) = o(1)$ such that, assuming ETH, distinguishing between the following requires time $n^{\tilde{\Omega}(\log n)}$:*

Completeness *G contains an $(1, \epsilon)$-community; and*

Soundness *G does not contain an $(\beta + \epsilon, \beta)$-community for any $\beta \in [0,1]$.*

Unlike many related quasi-polynomial approximation schemes mentioned above, Arora et al's algorithm has the unique property that it can also *exactly count* all the (α, β)-communities. Our second result is that counting even the number of $(1, o(1))$-communities requires quasi-polynomial time. A nice feature of this result is that we can base it on the much weaker #ETH assumption, which asserts that counting the satisfying assignment for a 3SAT instance requires time $2^{\Omega(n)}$. (Note, for example, that #ETH is likely to be true even if $\mathsf{P} = \mathsf{NP}$.)

Theorem 13.0.3. *For every n there exists an $\epsilon = \epsilon(n) = o(1)$ such that, assuming #ETH, counting $(1, \epsilon)$-communities requires time $n^{\log^{1-o(1)} n}$.*

13.0.1 Related works

The most closely related work is a reduction by Balcan, Borgs, Braverman, Chayes, and Teng [Bal+13, Theorem 5.3] from Planted Clique to finding $(1, 1-\gamma)$-communities, for some small (unspecified) constant $\gamma > 0$. Note that our inapproximability in Theorem 13.0.2 is much stronger in all parameters; furthermore, although formally incomparable, our ETH assumption is preferable over the average-case hardness assumption of Planted Clique.

Algorithms for special cases

Mishra, Schreiber, Stanton, and Tarjan [Mis+08] gave a polynomial-time algorithm for finding (α, β)-communities that contain a vertex with very few neighbors outside the community. Balcan et al [Bal+13] give a polynomial-time algorithm for enumerating (α, β)-communities in the special case where the degree of every node is $\Omega(n)$.

Arora, Ge, Sachdeva, and Schoenebeck [Aro+12] consider several semi-random models where the edges inside the community are generated at random, according to the expected degree model. (In fact, their quasi-polynomial time algorithm is also stated in this setting, but only their "Gap Assumption", which is equivalent to $\alpha - \beta = \Omega(1)$, is used in the analysis.)

Stochastic Block Model

Variants of the community detection problem on graphs generated by different stochastic models are extremely popular (see e.g. [BBV16; Ban+16; Che+16; FP16; HWX16; MMV16; MPW16; MX16; Tre+16] for papers in conference proceedings from June 2016). Perhaps the most influential is the *Stochastic Block Model* [HLL83]: The graph is partitioned into two disjoint communities; the edges within each community are present with probability α, independently, whereas edges between communities are present with probability β. Hence this model can also be seen as a special case of the (α, β)-Community Detection problem.

Stochastic models are extremely helpful in physics, for example, because atoms' interactions obey simple mathematical formulas with high precision. Unfortunately, for applications such as social networks, existing models do not describe human behavior with atomic precision, hence casting a shadow over the applicability of algorithms that work on ideal stochastic models. Recent works [MPW16; MMV16] attempted to bridge the gap from ideal model to practice by showing that certain SDP-based algorithms continue to work in a particular semi-random model where a restricted adversary is allowed to modify the random input graph. These success stories beg the question of how strong can one make the adversary? The current paper illuminates some of the computational barriers.

Alternative approaches to modeling communities

As we mentioned above, there are many different definitions of "communities" in networks. For in-depth discussion of different definitions see Arora et al [Aro+12] or Borgs et al [Bor+16]. As pointed out by the latter, for some definitions even verifying that a candidate subset is a community is intractable.

There is also an important literature on axiomatic approaches to the related problem of clustering (e.g. [Kle02; BA08; LM14]); note that while clustering typically aims to partition a set of nodes, our main focus is on detecting just a single community; in particular, different communities may intersect.

13.0.2 Overview of proofs

A good starting point for the technical discussion is the reduction reduction from 3SAT to the related problem of Densest-k-Subgraph we describe in Chapter 12. Let us recall the main two ingredients in that reduction: "birthday repetition" [AIM14] and the "FGLSS graph" [Fei+96].

"Birthday repetition" Starting with an instance of Label Cover (see definition in Section 2.4), the reduction considers a mega-variable for every ρ-tuple of variables, for $\rho \approx \sqrt{n}$. By the birthday paradox, almost every pair of ρ-tuples of variables intersect, inducing a consistency constraint on the two mega-assignments. Similarly, we expect to see some Label Cover edges in the union of the two ρ-tuples, inducing an additional Label Cover constraint between the two mega assignments. Notice that we have $\binom{n}{\rho} \approx 2^{\sqrt{n}}$ mega variables, and the alphabet size is also approximately $N = 2^{\sqrt{n}}$. Therefore, assuming ETH, finding an approximately satisfying assignment for the mega-variables requires time $2^{\Omega(n)} \approx N^{\log N}$.

FGLSS Similarly to the classic reduction by Feige et al. [Fei+96] for the Clique problem, In Chapter 12, we construct a vertex for each mega assignment to each mega variable, and draw an edge between two vertices if the induced assignments do not violate any consistency or Label Cover constraints. Notice that if the Label Cover instance has a satisfying assignment, then the graph contains a clique of size $\binom{n}{\rho}$ where each mega variable receives the mega assignment induced by the globally satisfying assignment. On the other hand, any subgraph that corresponds to a consistent assignment which violates many constraints must be missing most of its edges.

Unfortunately, this simple reduction is still quite far from working for the Community Detection problem. Below we describe some of the obstacles and outline how we overcome them.

Completeness

Surprisingly, the main problem with using the same reduction for Community Detection is the completeness: even if the Label Cover instance has a satisfying assignment, the resulting graph has no (α, β)-communities, for any constants $\alpha > \beta$! Observe, in particular, that the clique that corresponds to the satisfying assignment does not satisfy the weak ties condition. For any vertex v in that clique, consider any vertex v' that corresponds to changing the assignment to just one variable x_i in

v's assignment. If v agrees with the assignments of all other vertices in the clique, v' agrees with almost all of them - except for the negligible fraction that cover x_i or its neighbors in the Label Cover graph.

To overcome this problem of vertices that are "just outside the community", we use error correcting codes. Namely, we encode each assignment as a low-degree bivariate polynomial over finite field $\mathcal{G}$ of size $|\mathcal{G}| \approx \sqrt{n}$. Now vertices correspond to low-degree assignments to rows/columns of the polynomial. This guarantees that the assignments induced by every two vertices are far. If v agrees with all other vertices in the community, then almost all of those vertices disagree with v'.

Soundness

The main challenge for soundness is ruling out communities that do not correspond to a single, globally consistent assignment to the Label Cover instance. The key idea is to introduce auxiliary vertices that punish such communities by violating the weak ties desideratum.

Let us begin with the reduction to the counting variant (Theorem 13.0.3), which is easier, mostly because we are not concerned with approximation (i.e. we only have to show that subsets that are exactly $(1, \epsilon)$-communities correspond to satisfying assignments). Here we further simplify matters by sketching a construction with weighted edges. The full reduction (Section 13.1) uses unweighted edges and is only slightly more involved. Consider, for every $g \in \mathcal{G}$, an auxiliary vertex that is ϵ-connected to all proper vertices that do not correspond to assignments to the g-th row/column. Now if a $(1, \epsilon)$-community C does not contain a vertex with assignment to the g-th row/column, the auxiliary vertex must simultaneously: (i) belong to C so as not to violate the weak ties desideratum; yet (ii) it cannot belong to C because all its edges have weight ϵ (this would violate the strong ties desideratum). Therefore every $(1, \epsilon)$-community assigns values to every row/column in $\mathcal{G}^2$.

The reduction we described above suffices to show that (assuming ETH) deciding whether the graph contains a $(1, \epsilon)$-community also requires quasi-polynomial time. To get the stronger statement of Theorem 13.0.2 we must rule out even $(\beta, \beta + \epsilon)$-communities in case the Label Cover instance is far from satisfiable. In particular, we need to show that subsets that do not correspond to unique, consistent assignments are never $(\beta, \beta + \epsilon)$-communities. Instead of a single column/row, we let each proper vertex correspond to a subset of $t \approx \log n$ columns/rows. Instead of a single $g \in \mathcal{G}$, each auxiliary vertex corresponds to subset $H \subset \mathcal{G}$ of size $|H| = |\mathcal{G}|/2$. We draw an edge between an auxiliary vertex and a proper vertex if the indices of all t columns/rows are contained in H; if they are picked randomly this only happens with polynomially small probability. If, however, a β-fraction of the community

is restricted to a small subset $R \subset \mathcal{G}$, then there are auxiliary vertices for $H \supseteq R$ that connect to all those nodes and violate the weak ties desideratum. Roughly, we show that at least a $(1-\beta)$-fraction of the vertices have assignments that are "well spread" over $\mathcal{G}^2$, and among those assignments there are many violations of the Label Cover constraints.

13.1 Hardness of counting communities

Theorem 13.1.1. *There exists an $\epsilon(n) = o(1)$ such that, assuming #ETH, counting $(1,\epsilon)$-communities requires time $n^{\log^{1-o(1)} n}$.*

Construction

Begin with an instance (A, B, E, π) of Label Cover of size $n = n_A + n_B$ where $n_A \triangleq |A|$ and $n_B \triangleq |B|$. Let $\mathcal{G}$ be a finite field of size $\sqrt{n}/\epsilon^3$, and let $\mathcal{F} \subset \mathcal{G}$ be an arbitrary subset of size $|\mathcal{F}| = \sqrt{n}$. We identify between $A \cup B$ and points in $\mathcal{F}^2$; we also identify between a subset of $\mathcal{G}$ and $\Sigma_A \cup \Sigma_B$. Thus there is a one-to-one correspondence between a subset of assignments to $P_{\mathcal{F}} : \mathcal{F}^2 \to \mathcal{G}$ and assignments to the Label Cover instance. We can extend any such $P_{\mathcal{F}}$ to an individual-degree-$(|\mathcal{F}|-1)$ polynomial $P : \mathcal{G}^2 \to \mathcal{G}$. In the other direction, we think of each low individual degrees polynomial $P : \mathcal{G}^2 \to \mathcal{G}$ as a (possibly invalid) assignment to the Label Cover instance.

For every $g \in \mathcal{G}$, and degree-$(|\mathcal{F}|-1)$ polynomials $p_1, p_2 : \mathcal{G} \to \mathcal{G}$ such that $p_1(g) = p_2(g)$, we construct $1/\epsilon$ vertices $\{v_{g,p_1,p_2,i}\}_{i=1}^{1/\epsilon} \subset V$ in the communities graph. Each vertex naturally induces an assignment (p_1, p_2) on $(\mathcal{G} \times \{g\}) \cup (\{g\} \times \mathcal{G})$. We draw an edge between two vertices in V if they agree on the intersection of their lines, and if their induced assignments satisfy all the Label Cover constraints.

For every $g \in \mathcal{G}$ and $i \in [1/\epsilon]$, we also add two identical auxiliary vertices $u_{g,i}$ which are connected to every $v_{g',p_1,p_2,i}$ for $g' \neq g$ (but not to each other).

Completeness

For each assignment to the Label Cover instance, we construct a $(1,\epsilon)$-community by taking the induced assignment $P_{\mathcal{F}} : \mathcal{F}^2 \to \mathcal{G}$ and extending it to an individual-degree-$(|\mathcal{F}|-1)$ polynomial $P : \mathcal{G}^2 \to \mathcal{G}$. Let C be all the vertices $v_{g,p_1,p_2,i}$ such that p_1, p_2 are the restrictions of P to $(\mathcal{G} \times \{g\}), (\{g\} \times \mathcal{G})$. This correspondence is one-to-one and we need to show that the resulting C is actually a $(1,\epsilon)$-community.

Because all the vertices correspond to a consistent satisfying assignment, C is a clique. Let $v_{g,q_1,q_2,i} \notin C$; wlog q_1 disagrees with the restriction of P to $(\mathcal{G} \times \{g\})$. Since both q_1 and the restriction of P are degree-$(|\mathcal{F}|-1)$ polynomials, they must disagree

on all but at most $(|\mathcal{F}|-1)$ elements of $\mathcal{G}$. For all other $h \in \mathcal{G}$, the vertex $v_{g,q_1,q_2,i}$ does not share edges with any $v_{h,p_1,p_2,j} \in C$. Therefore, $v_{g,q_1,q_2,i}$ has edges to less than an $(|\mathcal{F}|/|\mathcal{G}|)$-fraction of vertices in C. Finally, every auxiliary vertex $u_{g,i}$ has edges to a $\frac{|\mathcal{G}|-1}{|\mathcal{G}|} \cdot \epsilon < \epsilon$-fraction of the vertices in ϵ. Therefore, C is a $(1,\epsilon)$-community.

13.1.1 Soundness

Structure of $(1,\epsilon)$-communities

Claim 13.1.2. Every $(1,\epsilon)$-community C contains exactly $1/\epsilon$ vertices $\{v_{g,p_1,p_2,i}\}_{i=1}^{1/\epsilon}$ for each g.

Proof. First, observe that C cannot contain any auxiliary vertices: if C contains one copy of $u_{g,i}$, it must also contain the other; but they don't have an edge between them, so they cannot both belong to a $(1,\epsilon)$-community.

Now, assume by contradiction that for some $g \in \mathcal{G}$, C does not contain any vertices with assignments for $(\mathcal{G} \times \{g\}) \cup (\{g\} \times \mathcal{G})$. Then every vertex in C is connected to (both copies of) $u_{g,i}$, for some $i \in [1/\epsilon]$. Therefore there is at least one $i \in [1/\epsilon]$ such that $u_{g,i}$ is connected to an ϵ-fraction of the vertices in C. But this is a contradiction since $u_{g,i} \notin C$.

If we ignore the auxiliary vertices (which, as we argued, C does not contain), the different vertices $v_{g,p_1,p_2,i}$ that correspond to the same assignment to the same lines (i.e. if we only change i) are indistinguishable. Therefore if C contains one of them, it must contain all of them (hence, at least $1/\epsilon$ vertices for each g).

Finally, since C is a clique, it cannot contain vertices that disagree on any assignments. (In particular, it cannot contain more than $1/\epsilon$ vertices for each g.) □

Completing the proof

Proof of Soundness. By Claim 13.1.2, every $(1,\epsilon)$-community C contains exactly $1/\epsilon$ vertices $\{v_{g,p_1,p_2,i}\}_{i=1}^{1/\epsilon}$ for each g. Furthermore, since C is a clique, all the induced assignments agree on all the intersections. So every $(1,\epsilon)$-community corresponds to a unique consistent assignment to the Label Cover instance. Finally, appealing again to the fact that C is a clique, this assignment must also satisfy all the Label Cover constraints. □

13.2 Hardness of detecting communities

Theorem 13.2.1. *There exists an $\epsilon(n) = o(1)$ such that, assuming ETH, distinguishing between the following requires time $n^{\tilde{\Omega}(\log n)}$:*

Completeness *G contains an $(1, \epsilon)$-community; and*

Soundness *G does not contain an $(\beta + \epsilon, \beta)$-community for any $\beta \in [0, 1]$.*

The rest of this section is devoted to the proof of Theorem 13.2.1. Our starting point is the Label Cover of Moshkovitz-Raz (Theorem 2.4.4). We compose the birthday repetition technique of [AIM14] with a bi-variate low-degree encoding. We then encode this as a graph a-la FGLSS [Fei+96]. We add auxiliary vertices to ensure that any $(\beta + \epsilon, \beta)$-community corresponds, approximately, to a uniform distribution over the variables.

Construction

Begin with a (d_A, d_B)-bi-regular instance (A, B, E, π) of Label Cover of size $n = n_A + n_B$ where $n_A \triangleq |A|$ and $n_B \triangleq |B|$. Let $\rho \triangleq \sqrt{n} \log n$; let $\mathcal{G}$ be a finite field of size $\rho/\epsilon^3 = \tilde{O}(\rho)$, and let $\mathcal{F} \subset \mathcal{G}$ be an arbitrary subset of size $|\mathcal{F}| = 2\rho$. Let $\mathcal{F}_A, \mathcal{F}_B \subset \mathcal{F}$ be disjoint subsets of size n_A/ρ, n_B/ρ, respectively. By Lemma 2.7.7, we can partition A and B into subsets $X_1, \ldots, X_{|\mathcal{F}_A|}$ and $Y_1, \ldots, Y_{|\mathcal{F}_B|}$ of size at most $|\mathcal{F}|$ such that between every two subsets there are approximately $\frac{d_A \rho^2}{n_B} = \frac{d_B \rho^2}{n_A}$ constraints. For $i \in \mathcal{F}_A$, we think of the points $\{i\} \times \mathcal{F} \subset \mathcal{G}^2$ as representing assignments to variables in X_i; for $j \in \mathcal{F}_B$, we think of $\mathcal{F} \times \{j\} \subset \mathcal{G}^2$ as representing assignments to variables in Y_j. Notice that each point in $\mathcal{F}^2$ may represent an assignment to both a vertex from A and a vertex from B, to one of them, or to neither. In particular, any assignment $P : \mathcal{G}^2 \to \mathcal{G}$ induces an assignment for the Label Cover instance; note that since $|\mathcal{G}| > |\Sigma_A||\Sigma_B|$, one value $P(f_1, f_2) \in \mathcal{G}$ suffices to describe assignments to both $a \in A$ and $b \in B$.

Let $t \triangleq \log n \cdot \left(\frac{|\mathcal{G}|}{|\mathcal{F}_A|} + \frac{|\mathcal{G}|}{|\mathcal{F}_B|} \right) = \mathsf{polylog}(n)$. We say that a subset $S \in \binom{\mathcal{G}}{t}$ is *balanced* if: $|S \cap \mathcal{F}_A| = \frac{|\mathcal{F}_A|}{|\mathcal{G}|} \cdot t$ and $|S \cap \mathcal{F}_B| = \frac{|\mathcal{F}_B|}{|\mathcal{G}|} \cdot t$. For every balanced subset S, consider $2t$ polynomials $q_\ell : \mathcal{G} \to \mathcal{G}$ of degree at most $|\mathcal{F}| - 1$, representing an assignment[1] $Q : (S \times \mathcal{G}) \cup (\mathcal{G} \times S) \to \mathcal{G}$. For balanced S and $2t$-tuple of polynomials (q_ℓ), we construct a corresponding vertex $v_{S,(q_\ell)}$ in the communities graph. Let V denote the set of vertices defined so far. For $g \in \mathcal{G}$ we abuse notation and say that $g \in v_{S,(q_\ell)}$ if

[1] We will only consider polynomials that correspond to a consistent assignment Q; i.e. for each point in $S \times S$ we expect the two corresponding polynomials to agree with each other.

$g \in S$. We construct an edge in the communities graph between two vertices in V if their assignments agree on the variables in their intersection, and their induced assignments to $A \cup B$ satisfy all the Label Cover constraints.

Additionally, for every $H \subset \mathcal{G}$ of size $|H| = |\mathcal{G}|/2$, define $|V|^2$ identical auxiliary vertices u_H in the communities graph. We draw an edge between auxiliary vertex u_H and vertex $v_{S,(q_\ell)}$ if $S \subset H$. Similarly, for every $H_A \subset \mathcal{F}_A$ of size $|H_A| = |\mathcal{F}_A|/2$, we define $|V|^2$ identical auxiliary vertices u_{H_A} with edges to every vertex $v_{S,(q_\ell)}$ such that $(S \cap \mathcal{F}_A) \subset H_A$. For $H_B \subset \mathcal{F}_B$ of size $|H_B| = |\mathcal{F}_B|/2$, we draw edges between u_{H_B} and $v_{S,(q_\ell)}$ such that $(S \cap \mathcal{F}_B) \subset H_B$.

Completeness

Suppose that the Label Cover instance has a satisfying assignment. Let $\mathcal{Z} \subseteq \mathcal{G}^2$ denote the subset of points that correspond to at least one variable in A or B. Let $P_{\mathcal{Z}} : \mathcal{Z} \to \mathcal{G}$ be the induced function on $\mathcal{Z}$ that corresponds to the satisfying assignment, and let $P : \mathcal{G}^2 \to \mathcal{G}$ be the extension of $P_{\mathcal{Z}}$ by setting $P(f_1, f_2) = 0$ for $(f_1, f_2) \in \mathcal{F}^2 \setminus \mathcal{Z}$ (this choice is arbitrary), and then extending to an $(|\mathcal{F}| - 1)$-individual-degree polynomial over all of $\mathcal{G}^2$.

Let C be the set of vertices that correspond to restrictions of P to balanced sets, i.e.

$$C = \left\{ v_{S,(P|_S)} : S \text{ is balanced} \right\},$$

where $P|_S$ denotes the restriction of P to $(S \times \mathcal{G}) \cup (\mathcal{G} \times S)$. Since all those vertices correspond to a consistent satisfying assignment, C is a clique.

For any vertex $v_{S,(q_\ell)} \notin C$, at least one of the polynomials, q_{ℓ^*} disagrees with the restriction of P to the corresponding line. Since both q_{ℓ^*} and the restriction of P to that line are degree-$(|\mathcal{F}| - 1)$ polynomials, they must disagree on at least $\left(1 - \frac{|\mathcal{F}|}{|\mathcal{G}|}\right)$-fraction of the coordinates. The probability that a random balanced set S' is contained in the $O(\epsilon^3)$-fraction of coordinates where they do agree is smaller than ϵ (and in fact polynomially small in n). Therefore $v_{S,(q_\ell)}$ has inconsistency violations with all but (less than) an ϵ-fraction of the vertices in C.

For any auxiliary vertex u_{H_A}, the probability that a random vertex $v_{S,(P|_S)} \in C$ is connected to u_{H_A} is $2^{-|S \cap \mathcal{F}_A|} < 1/n$, and similarly for u_{H_B} and u_H. Therefore, every auxiliary vertex is connected to less than a $(1/n)$-fraction of the vertices in C.

13.2.1 Soundness

Lemma 13.2.2. *If the* Label Cover *instance has value at most ϵ^3, then there are no $(\beta + \epsilon, \beta)$-communities.*

Auxiliary vertices

Claim 13.2.3. Every $(\beta+\epsilon,\beta)$-community does not contain any auxiliary vertices.

Proof. There are $|V|^2$ identical copies of each auxiliary vertex. Since they are identical, any community must either contain all of them, or none of them. If the community contains all $|V|^2$ copies, then it has a vast majority of auxiliary vertices, so none of them can have edges to an ϵ-fraction of the community. □

List decoding

Claim 13.2.4. The vertices in any $(\beta+\epsilon,\beta)$-community C induce at most $4/\epsilon$ different assignments for each variable.

Proof. Suppose by contradiction that this is not the case. Then, wlog, there is a line $\{g_1\}\times\mathcal{G}$ that receives at least $2/\epsilon$ different assignments from vertices in C. Every two assignments agree on at most $|\mathcal{F}|$ points (g_1,g') on the line, so in total there are at most $2|\mathcal{F}|/\epsilon^2$ points where at least two assignments agree. Let $R\subseteq\mathcal{G}$ denote the set of g' such that no two assignments agree on (g_1,g'); we have that $|R|\geq|\mathcal{G}|-2|\mathcal{F}|/\epsilon^2\geq|\mathcal{G}|/2$. Therefore, by the weak ties property, for at most a β-fraction of the vertices $v_{S,(q_\ell)}\in C$, $S\cap R=\varnothing$.

Consider the remaining $(1-\beta)$-fraction of vertices in C. Suppose that v assigns a value to some (g_1,g') for $g'\in R$: this value can only agree with one of the $2/\epsilon$ different assignments to (g_1,g'). Therefore, in expectation, each of the $2/\epsilon$ vertices that assign different values for (g_1,g') is connected to at most a $(\beta+\epsilon/2)$-fraction of the vertices in C. This is a contradiction to C being a $(\beta+\epsilon,\beta)$-community. □

Completing the proof

Proof of Lemma 13.2.2. Suppose that at most a ϵ^3-fraction of the Label Cover constraints can be satisfied by any single assignment, and assume by contradiction that C is a $(\beta+\epsilon,\beta)$-community. By Claim 13.2.4, C induces at most $4/\epsilon$ assignments on each variable, so at most $O(\epsilon)$-fraction of the constraints are satisfied by any pair of assignments.

By Markov's inequality, for at least half of the subsets $X_i\subset A$, only an $O(\epsilon)$-fraction of the constraints that depend on X_i are satisfied. By Claim 13.2.3 at least $(1-\beta)$-fraction of the vertices in C assign values to at least one such X_i. Consider any such vertex $v_{S,(q_\ell)}$ where $S\ni i$. By construction of the partitions (Lemma 2.7.7), each X_i shares approximately the same number of constraints with each Y_j. Therefore, for all but an $O(\epsilon)$-fraction of Y_j's, X_i and Y_j observe a violation - for all the assignments given by vertices in C to the variables in Y_j. In other words, $v_{S,(q_\ell)}$ cannot have edges

to any vertex $v_{T,(r_\ell)}$ such that $T \ni j$, for a $(1 - O(\epsilon))$-fraction of $j \in [n_B/k_B]$. Finally, applying Claim 13.2.3 again, at most a β fraction of vertices in C do not contain any of those j's. This is a contradiction to $v_{S,(q_\ell)}$ having edges to $(\beta + \epsilon)$-fraction of the vertices in C. □

Chapter 14

VC and Littlestone's dimensions

A common and essential assumption in learning theory is that the concepts we want to learn come from a nice, simple concept class, or (in the agnostic case) they can at least be approximated by a concept from a simple class. When the concept class is sufficiently simple, there is hope for good (i.e. sample-efficient and low-error) learning algorithms.

There are many different ways to measure the *simplicity* of a concept class. The most influential measure of simplicity is the VC Dimension, which captures learning in the PAC model. We also consider Littlestone's Dimension [Lit87], which corresponds to minimizing mistakes in online learning (see Section 2.5 for definitions). When either dimension is small, there are algorithms that exploit the simplicity of the class, to obtain good learning guarantees.

Two decades ago, it was shown that (under appropriate computational complexity assumptions) neither dimension can be computed in polynomial time [PY96; FL98]; and these impossibility results hold even in the most optimistic setting where the entire universe and concept class are given as explicit input (a binary matrix whose (x, c)-th entry is 1 iff element x belongs to concept c). The computational intractability of computing the (VC, Littlestone's) dimension of a concept class suggests that even in cases where a simple structure exists, it may be inaccessible to computationally bounded algorithms (see Discussion below).

In this work we extend the results of [PY96; FL98] to show that the VC and Littlestone's Dimensions cannot even be *approximately* computed in polynomial time. We don't quite prove that those problems are NP-hard: both dimensions can be computed (exactly) in quasi-polynomial ($n^{O(\log n)}$) time, hence it is very unlikely that either problem is NP-hard. Nevertheless, assuming the randomized Exponential

Time Hypothesis (ETH)[1], we prove essentially tight quasi-polynomial lower bounds on the running time - that hold even against approximation algorithms.

Theorem 14.0.1 (Hardness of Approximating VC Dimension)**.** *Assuming Randomized ETH, approximating VC Dimension to within a* $(1/2 + o(1))$*-factor requires* $n^{\log^{1-o(1)} n}$ *time.*

Theorem 14.0.2 (Hardness of Approximating Littlestone's Dimension)**.** *There exists an absolute constant* $\varepsilon > 0$ *such that, assuming Randomized ETH, approximating Littlestone's Dimension to within a* $(1-\varepsilon)$*-factor requires* $n^{\log^{1-o(1)} n}$ *time.*

14.0.1 Discussion

As we mentioned before, the computational intractability of computing the (VC, Littlestone's) dimension of a concept class suggests that even in cases where a simple structure exists, it may be inaccessible to computationally bounded algorithms. We note however that it is not at all clear that any particular algorithmic applications are immediately intractable as a consequence of our results.

Consider for example the adversarial online learning zero-sum game corresponding to Littlestone's Dimension: At each iteration, Nature presents the learner with an element from the universe; the learner attempts to classify the element, and loses a point for every wrong classification; at the end of the iteration, the correct (binary) classification is revealed. The Littlestone's Dimension is equal to the worst case loss of the Learner before learning the exact concept. (see Section 2.5 for a more detailed definition.)

What can we learn from the fact that the Littlestone's Dimension is hard to compute? The first observation is that there is no efficient learner that can commit to a concrete mistake bound. But this does not rule out a computationally-efficient learner that plays optimal strategy and makes at most as many mistakes as the unbounded learner. We can, however, conclude that Nature's task is computationally intractable! Otherwise, we could efficiently construct an entire worst-case mistake tree (for a concept class $\mathcal{C}$, any mistake tree has at most $|\mathcal{C}|$ leaves, requiring $|\mathcal{C}| - 1$ oracle calls to Nature).

On a philosophical level, we think it is interesting to understand the implications of an intractable, adversarial Nature. Perhaps this is another evidence that the mistake bound model is too pessimistic?

[1]The randomized ETH (rETH) postulates that there is no $2^{o(n)}$-time Monte Carlo algorithms that solves 3SAT correctly with probability at least 2/3 (i.e. $\mathsf{3SAT} \notin \mathsf{BPTIME}(2^{o(n)})$).

Also, the only algorithm we know for computing the optimal learner's decision requires computing the Littlestone's Dimension. We think that it is an interesting open question whether an approximately optimal computationally-efficient learner exist.

Finally, let us note that in the other direction, computing Littlestone's Dimension exactly implies an exactly optimal learner. However, since the learner has to compute Littlestone's Dimension many times, we have no evidence that an approximation algorithm for Littlestone's Dimension would imply any guarantee for the learner.

Finally, we remark that for either problem (VC or Littlestone's Dimension), we are not aware of any non-trivial approximation algorithms.

14.0.2 Techniques

Our reductions in this chapter are also inspired by "birthday repetition", but surprising challenges arise.

VC Dimension The first challenge we have to overcome in order to adapt this framework to hardness of approximation of VC Dimension is that the number of concepts involved in shattering a subset S is $2^{|S|}$. Therefore any inapproximability factor we prove on the size of the shattered set of elements, "goes in the exponent" of the size of the shattering set of concepts. Even a small constant factor gap in the VC Dimension requires proving a polynomial factor gap in the number of shattering concepts (obtaining polynomial gaps via "birthday repetition" for simpler problems is an interesting open problem [MR16; Man17]). Fortunately, having a large number of concepts is also an advantage: we use each concept to test a different set of 3-Color constraints chosen independently at random; if the original instance is far from satisfied, the probability of passing all $2^{\Theta(|S|)}$ tests should now be doubly-exponentially small ($2^{-2^{\Theta(|S|)}}$)! More concretely, we think of half of the elements in the shattered set as encoding an assignment, and the other half as encoding which tests to run on the assignments.

Littlestone's Dimension Our starting point is the reduction for VC Dimension outlined in the previous paragraph. Without going into the exact definition of Littlestone's Dimension, recall that it corresponds to an online learning model. If the test-selection elements arrive before the assignment-encoding elements, the adversary can adaptively tailor his assignment to pass the specific test selected in the previous steps. To overcome this obstacle, we introduce a special gadget that forces the assignment-encoding elements to arrive first; this makes the reduction to Lit-

tlestone's Dimension somewhat more involved. Note that there is a reduction by [FL98] from VC Dimension to Littlestone's Dimension. Unfortunately, their reduction is not (approximately) gap-preserving, so we cannot use it directly to obtain Theorem 14.0.2 from Theorem 14.0.1.

14.0.3 Related Work

The study of the computational complexity of the VC Dimension was initiated by Linial, Mansour, and Rivest [LMR91], who observed that it can be computed in quasi-polynomial time. [PY96] proved that it is complete for the class LOGNP which they define in the same paper. [FL98] reduced the problem of computing the VC dimension to that of computing Littlestone's Dimension, hence the latter is also LOGNP-hard. (It follows as a corollary of our Theorem 14.0.1 that, assuming ETH, solving any LOGNP-hard problem requires quasi-polynomial time.)

Both problems were also studied in an implicit model, where the concept class is given in the form of a Boolean circuit that takes as input an element x and a concept c and returns 1 iff $x \in c$. Observe that in this model even computing whether either dimension is 0 or not is already NP-hard. Schafer proved that the VC Dimension is Σ_3^{P}-complete [Sch99], while the Littlestone's Dimension is PSPACE-complete [Sch00]. [MU02] proved that VC Dimension is Σ_3^{P}-hard to approximate to within a factor of almost 2; can be approximated to within a factor slightly better than 2 in AM; and is AM-hard to approximate to within $n^{1-\varepsilon}$.

Another line of related work in the implicit model proves computational intractability of PAC learning (which corresponds to the VC Dimension). Such intractability has been proved either from cryptographic assumptions, e.g. [KV94; Kha93; Kha95; Fel+06; Kal+08; KS09; Kli16] or from average case assumptions, e.g. [DS16; Dan16]. [Blu94] showed a "computational" separation between PAC learning and online mistake bound (which correspond to the VC Dimension and Littlestone's Dimension, respectively): if one-way function exist, then there is a concept class that can be learned by a computationally-bounded learner in the PAC model, but not in the mistake-bound model.

Recently, [BFS16] introduced a generalization of VC Dimension which they call Partial VC Dimension, and proved that it is NP-hard to approximate (even when given an explicit description of the universe and concept class).

It is interesting to note that in contrast to other problems discussed in this part of the thesis, VC and Littlestone's dimension do not naturally fit into the bilinear optimization framework, and they can be computed *exactly* in quasi-polynomial time with completely different algorithms: For VC Dimension, it suffices to simply enumerate over all $\log|\mathcal{C}|$-tuples of elements (where $\mathcal{C}$ denotes the concept class

and $\log|\mathcal{C}|$ is the trivial upper bound on the VC dimension) [LMR91]. Littlestone's Dimension can be computed in quasi-polynomial time via a recursive "divide and conquer" algorithm (See our Theorem 14.3.1).

14.1 Inapproximability of VC Dimension

In this section, we present our reduction from Label Cover to VC Dimension, stated more formally below. We note that this reduction, together with Moshkovitz-Raz PCP (Theorem 2.4.4), with parameter $\delta = 1/\log n$ gives a reduction from 3SAT of size n to VC Dimension of size $2^{n^{1/2+o(1)}}$ with gap $1/2 + o(1)$, which immediately implies Theorem 14.0.1.

Theorem 14.1.1. *For every $\delta > 0$, there exists a randomized reduction from a bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e\in E})$ to a ground set $\mathcal{U}$ and a concept class $\mathcal{C}$ such that, if $n \triangleq |A|+|B|$ and $r \triangleq \sqrt{n}/\log n$, then the following conditions hold for every sufficiently large n.*

- *(Size) The reduction runs in time $|\Sigma|^{O(|E|\mathsf{poly}(1/\delta)/r)}$ and $|\mathcal{C}|, |\mathcal{U}| \le |\Sigma|^{O(|E|\mathsf{poly}(1/\delta)/r)}$.*
- *(Completeness) If $\mathcal{L}$ is satisfiable, then* VC-dim$(\mathcal{C},\mathcal{U}) \ge 2r$.
- *(Soundness) If* val$(\mathcal{L}) \le \delta^2/100$, *then* VC-dim$(\mathcal{C},\mathcal{U}) \le (1+\delta)r$ *with high probability.*

14.1.1 A Candidate Reduction (and Why It Fails)

To best understand the intuition behind our reduction, we first describe a simpler candidate reduction and explain why it fails, which will lead us to the eventual construction. In this candidate reduction, we start by evoking Lemma 2.7.7 to partition the vertices $A \cup B$ of the Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e\in E})$ into $U_1 \triangleq (S_1, T_1), \ldots, U_r$ where $r = \sqrt{n}/\log n$. We then create the universe $\mathcal{U}$ and the concept class $\mathcal{C}$ as follows:

- We make each element in $\mathcal{U}$ correspond to a partial assignment to U_i for some $i \in [r]$, i.e., we let $\mathcal{U} = \{x_{i,\sigma_i} \mid i \in [r], \sigma_i \in \Sigma^{U_i}\}$. In the completeness case, we expect to shatter the set of size r that corresponds to a satisfying assignment $\sigma^* \in \Sigma^{A\cup B}$ of the Label Cover instance $\mathcal{L}$, i.e., $\{x_{i,\sigma^*|_{U_i}} \mid i \in [r]\}$. As for the soundness, our hope is that, if a large set $S \subseteq \mathcal{U}$ gets shattered, then we will be able to decode an assignment for $\mathcal{L}$ that satisfies many constraints, which contradicts with our assumption that val$(\mathcal{L})$ is small. Note that the number of

elements used in this candidate reduction is at most $r \cdot |\Sigma|^{O(|E|\mathsf{poly}(1/\delta)r)} = 2^{\tilde{O}(\sqrt{n})}$ as desired.

- As stated above, the intended solution for the completeness case is $\{x_{i,\sigma^*|_{U_i}} \mid i \in [r]\}$, meaning that we must have at least one concept corresponding to each subset $I \subseteq [r]$. We will try to make our concepts "test" the assignment; for each $I \subseteq [r]$, we will choose a set $T_I \subseteq A \cup B$ of $\tilde{O}(\sqrt{n})$ vertices and "test" all the constraints within T_I. Before we specify how T_I is picked, let us elaborate what "test" means: for each T_I-partial assignment ϕ_I that does not violate any constraints within T_I, we create a concept C_{I,ϕ_I}. This concept contains x_{i,σ_i} if and only if $i \in I$ and σ_i agrees with ϕ_I (i.e. $\phi_I|_{T_I \cap U_i} = \sigma_i|_{T_I \cap U_i}$). Recall that, if a set $S \subseteq \mathcal{U}$ is shattered, then each $T \subseteq S$ is an intersection between S and C_{I,ϕ_I} for some I, ϕ_I. We hope that the I's are different for different T so that many different tests have been performed on S.

  Finally, let us specify how we pick T_I. Assume without loss of generality that r is even. We randomly pick a perfect matching between r, i.e., we pick a random permutation $\pi_I : [r] \to [r]$ and let $\Big(\pi_I(1), \pi_I(2)\Big), \ldots, \Big(\pi_I(r-1), \pi_I(r)\Big)$ be the chosen matching. We pick T_I such that all the constraints in the matchings, i.e., constraints between $U_{\pi_I(2i-1)}$ and $U_{\pi_I(2i)}$ for every $i \in [r/2]$, are included. More specifically, for every $i \in [r]$, we include each vertex $v \in U_{\pi_I(2i-1)}$ if at least one of its neighbors lie in $U_{\pi_I(2i)}$ and we include each vertex $u \in U_{\pi_I(2i)}$ if at least one of its neighbors lie in $U_{\pi_I(2i-1)}$. (By Lemma 2.7.7, for every pair in the matching the size of the intersection is at most $\frac{2|E|}{r^2}$, so each concept contains assignments to at most $\frac{2|E|}{r}$ variables; so the total size of the concept class is at most $2^r \cdot |\Sigma|^{\frac{2|E|}{r}}$.)

Even though the above reduction has the desired size and completeness, it unfortunately fails in the soundness. Let us now sketch a counterexample. For simplicity, let us assume that each vertex in $T_{[r]}$ has a unique neighbor in $T_{[r]}$. Note that, since $T_{[r]}$ has quite small size (only $\tilde{O}(\sqrt{n})$), almost all the vertices in $T_{[r]}$ satisfy this property w.h.p., but assuming that all of them satisfy this property makes our life easier.

Pick an assignment $\tilde{\sigma} \in \Sigma^V$ such that none of the constraints in $T_{[r]}$ is violated. From our unique neighbor assumption, there is always such an assignment. Now, we claim that the set $S_{\tilde{\sigma}} \triangleq \{x_{i,\tilde{\sigma}|_{U_i}} \mid i \in [r]\}$ gets shattered. This is because, for every subset $I \subseteq [r]$, we can pick another assignment σ' such that σ' does not violate any constraint in $T_{[r]}$ and $\sigma'|_{U_i} = \tilde{\sigma}|_{U_i}$ if and only if $i \in I$. This implies that $\{x_{i,\tilde{\sigma}_i} \mid i \in I\} = S \cap C_{[r],\sigma'}$ as desired. Note here that such σ' exists because, for

every $i \notin I$, if there is a constraint from a vertex $a \in U_i$ to another vertex $b \in T_{[r]}$, then we can change the assignment to a in such a way that the constraint is not violated; by doing this for every $i \notin I$, we have created the desired σ'. As a result, VC-dim$(\mathcal{C},\mathcal{U})$ can still be as large as r even when the value of $\mathcal{L}$ is small.

14.1.2 The Final Reduction

In this subsection, we will describe the actual reduction. To do so, let us first take a closer look at the issue with the above candidate reduction. In the candidate reduction, we can view each $I \subseteq [r]$ as being a seed used to pick a matching. Our hope was that many seeds participate in shattering some set S, and that this means that S corresponds to an assignment of high value. However, the counterexample showed that in fact only one seed ($I = [r]$) is enough to shatter a set. To circumvent this issue, we will not use the subset I as our seed anymore. Instead, we create r new elements $y_1, \ldots, y_r$, which we will call *test selection elements* to act as seeds; namely, each subset $H \subseteq \mathcal{Y}$ will now be a seed. The benefit of this is that, if $S \subseteq \mathcal{Y}$ is shattered and contains test selection elements $y_{i_1}, \ldots, y_{i_t}$, then at least 2^t seeds must participate in the shattering of S. This is because, for each $H \subseteq \mathcal{Y}$, the intersection of S with any concept corresponding to H, when restricted to $\mathcal{Y}$, is always $H \cap \{y_{i_1}, \ldots, y_{i_t}\}$. Hence, each subset of $\{y_{i_1}, \ldots, y_{i_t}\}$ must come a from different seed.

The only other change from the candidate reduction is that each H will test multiple matchings rather than one matching. This is due to a technical reason: we need the number of matchings, ℓ, to be large in order get the approximation ratio down to $1/2 + o(1)$; in our proof, if $\ell = 1$, then we can only achieve a factor of $1 - \varepsilon$ to some $\varepsilon > 0$. The full details of the reduction are shown in Figure 14.1.

Before we proceed to the proof, let us define some additional notation that will be used throughout.

- Every assignment element of the form x_{i,σ_i} is called an *i-assignment element*; we denote the set of all i-assignment elements by $\mathcal{X}_i$, i.e., $\mathcal{X}_i = \{x_{i,\sigma_i} \mid \sigma_i \in \Sigma^{U_i}\}$. Let $\mathcal{X}$ denote all the assignment elements, i.e., $\mathcal{X} = \bigcup_i \mathcal{X}_i$.

- For every $S \subseteq \mathcal{U}$, let $I(S)$ denote the set of all $i \in [r]$ such that S contains an i-assignment element, i.e., $I(S) = \{i \in [r] \mid S \cap \mathcal{X}_i \neq \varnothing\}$.

- We call a set $S \subseteq \mathcal{X}$ *non-repetitive* if, for each $i \in [r]$, S contains at most one i-assignment element, i.e., $|S \cap \mathcal{X}_i| \leq 1$. Each non-repetitive set S canonically induces a partial assignment $\phi(S) : \bigcup_{i \in I(S)} U_i \to \Sigma$. This is the unique partial assignment that satisfies $\phi(S)|_{U_i} = \sigma_i$ for every $x_{i,\sigma_i} \in S$

- Even though we define each concept as C_{I,H,σ_H} where σ_H is a partial assignment to a subset $T_H \subset A \cup B$, it will be more convenient to view each concept as $C_{I,H,\sigma}$ where $\sigma \in \Sigma^V$ is the assignment to the entire Label Cover instance. This is just a notational change: the actual definition of the concept does not depend on the assignment outside T_H.

- For each $I \subseteq [r]$, let U_I denote $\bigcup_{i \in I} U_i$. For each $\sigma_I \in \Sigma^{U_I}$, we say that (I, σ_I) *passes* $H \subseteq \mathcal{Y}$ if σ_I does not violate any constraint within T_H. Denote the collection of H's that (I, σ_I) passes by $\mathcal{H}(I, \sigma_I)$.

- Finally, for any non-repetitive set $S \subseteq \mathcal{X}$ and any $H \subseteq \mathcal{Y}$, we say that S *passes* H if $(I(S), \phi(S))$ passes H. We write $\mathcal{H}(S)$ as a shorthand for $\mathcal{H}(I(S), \phi(S))$.

The output size of the reduction and the completeness follow almost immediately from definition.

Output Size of the Reduction. Clearly, the size of $\mathcal{U}$ is $\sum_{i \in [r]} |\Sigma|^{|U_i|} \leq r \cdot |\Sigma|^{n/r} \leq |\Sigma|^{O(|E|\mathsf{poly}(1/\delta)/r)}$. As for $|\mathcal{C}|$, note first that the number of choices for I and H are both 2^r. For fixed I and H, Lemma 2.7.7 implies that, for each matching $\pi_H^{(t)}$, the number of vertices from each U_i with at least one constraint to the matched partition in $\pi_H^{(t)}$ is at most $O(|E|/r^2)$. Since there are ℓ matchings, the number of vertices in $T_H = \mathcal{N}_1(M_H(1)) \cup \cdots \cup \mathcal{N}_r(M_H(r))$ is at most $O(|E|\ell/r)$. Hence, the number of choices for the partial assignment σ_H is at most $|\Sigma|^{O(|E|\mathsf{poly}(1/\delta)/r)}$. In total, we can conclude that $\mathcal{C}$ contains at most $|\Sigma|^{O(|E|\mathsf{poly}(1/\delta)/r)}$ concepts.

Completeness. If $\mathcal{L}$ has a satisfying assignment $\sigma^* \in \Sigma^V$, then the set $S_{\sigma^*} = \{x_{i,\sigma_i^*} \mid i \in [r]\} \cup \mathcal{Y}$ is shattered because, for any $S \subseteq S_{\sigma^*}$, we have $S = S_{\sigma^*} \cap C_{I(S),S \cap \mathcal{Y},\sigma^*}$. Hence, VC-dim$(\mathcal{C},\mathcal{U}) \geq 2r$.

The rest of this section is devoted to the soundness analysis.

14.1.3 Soundness

In this subsection, we will prove the following lemma, which, combined with the completeness and output size arguments above, imply Theorem 14.1.1. For brevity, we will assume throughout this subsection that r is sufficiently large, and leave it out of the lemmas' statements.

Lemma 14.1.2. *Let $(\mathcal{C},\mathcal{U})$ be the output from the reduction in Figure 14.1 on input $\mathcal{L}$. If* val$(\mathcal{L}) \leq \delta^2/100$*, then* VC-dim$(\mathcal{C},\mathcal{U}) \leq (1+\delta)r$ *with high probability.*

At a high level, the proof of Lemma 14.1.2 has two steps:

1. Given a shattered set $S \subseteq \mathcal{U}$, we extract a maximal non-repetitive set $S^{\text{NO-REP}} \subseteq S$ such that $S^{\text{NO-REP}}$ passes many ($\geq 2^{|S|-|S^{\text{NO-REP}}|}$) H's. If $|S^{\text{NO-REP}}|$ is small, the trivial upper bound of 2^r on the number of different H's implies that $|S|$ is also small. As a result, we are left to deal with the case that $|S^{\text{NO-REP}}|$ is large.

2. When $|S^{\text{NO-REP}}|$ is large, $S^{\text{NO-REP}}$ induces a partial assignment on a large fraction of vertices of $\mathcal{L}$. Since we assume that $\text{val}(\mathcal{L})$ is small, this partial assignment must violate many constraints. We will use this fact to argue that, with high probability, $S^{\text{NO-REP}}$ only passes very few H's, which implies that $|S|$ must be small.

The two parts of the proof are presented in Subsection 14.1.3.1 and 14.1.3.2 respectively. We then combine them in Subsection 14.1.3.3 to prove Lemma 14.1.2.

14.1.3.1 Part I: Finding Non-Repetitive Set That Passes Many Tests

The goal of this subsection is to prove the following lemma, which allows us to, given a shattered set $S \subseteq \mathcal{U}$, find a non-repetitive set $S^{\text{NO-REP}}$ that passes many H's.

Lemma 14.1.3. *For any shattered $S \subseteq \mathcal{U}$, there is a non-repetitive set $S^{\text{NO-REP}}$ of size $|I(S)|$ s.t. $|\mathcal{H}(S^{\text{NO-REP}})| \geq 2^{|S|-|I(S)|}$.*

We will start by proving the following lemma, which will be a basis for the proof of Lemma 14.1.3.

Lemma 14.1.4. *Let $C, C' \in \mathcal{C}$ correspond to the same H (i.e. $C = C_{I,H,\sigma}$ and $C' = C_{I',H,\sigma'}$ for some $H \subseteq \mathcal{Y}, I, I' \subseteq [r], \sigma, \sigma' \in \Sigma^V$).*

For any subset $S \subseteq \mathcal{U}$ and any maximal non-repetitive subset $S^{\text{NO-REP}} \subseteq S$ such that $I(S^{\text{NO-REP}}) = I(S)$, if $S^{\text{NO-REP}} \subseteq C$ and $S^{\text{NO-REP}} \subseteq C'$, then $S \cap C = S \cap C'$.

The most intuitive interpretation of this lemma is as follows. Recall that if S is shattered, then, for each $\tilde{S} \subseteq S$, there must be a concept $C_{I_{\tilde{S}},H_{\tilde{S}},\sigma_{\tilde{S}}}$ such that $\tilde{S} = S \cap C_{I_{\tilde{S}},H_{\tilde{S}},\sigma_{\tilde{S}}}$. The above lemma implies that, for each $\tilde{S} \supseteq S^{\text{NO-REP}}$, $H_{\tilde{S}}$ must be different. This means that at least $2^{|S|-|S^{\text{NO-REP}}|}$ different H's must be involved in shattering S. Indeed, this will be the argument we use when we prove Lemma 14.1.3.

Proof of Lemma 14.1.4. Let $S, S^{\text{NO-REP}}$ be as in the lemma statement. Suppose for the sake of contradiction that there exists $H \subseteq \mathcal{Y}, I, I' \subseteq [r], \sigma, \sigma' \in \Sigma^V$ such that $S^{\text{NO-REP}} \subseteq C_{I,H,\sigma}, S^{\text{NO-REP}} \subseteq C_{I',H,\sigma'}$ and $S \cap C_{I,H,\sigma} \neq S \cap C_{I',H,\sigma'}$.

First, note that $S \cap C_{I,H,\sigma} \cap \mathcal{Y} = S \cap H \cap \mathcal{Y} = S \cap C_{I',H,\sigma'} \cap \mathcal{Y}$. Since $S \cap C_{I,H,\sigma} \neq S \cap C_{I',H,\sigma'}$, we must have $S \cap C_{I,H,\sigma} \cap \mathcal{X} \neq S \cap C_{I',H,\sigma'} \cap \mathcal{X}$. Assume w.l.o.g. that there exists $x_{i,\sigma_i} \in (S \cap C_{I,H,\sigma}) \setminus (S \cap C_{I',H,\sigma'})$.

Note that $i \in I(S) = I(S^{\text{NO-REP}})$ (where the equality follows by maximality of $S^{\text{NO-REP}}$). Thus there exists $\sigma'_i \in \Sigma^{U_i}$ such that $x_{i,\sigma'_i} \in S^{\text{NO-REP}} \subseteq C_{I,H,\sigma} \cap C_{I',H,\sigma'}$. Since x_{i,σ'_i} is in both $C_{I,H,\sigma}$ and $C_{I',H,\sigma'}$, we have $i \in I \cap I'$ and

$$\sigma|_{\mathcal{N}_i(M_H(i))} = \sigma'_i|_{\mathcal{N}_i(M_H(i))} = \sigma'|_{\mathcal{N}_i(M_H(i))}. \tag{14.1}$$

However, since $x_{i,\sigma_i} \in (S \cap C_{I,H,\sigma}) \setminus (S \cap C_{I',H,\sigma'})$, we have $x_{i,\sigma_i} \in C_{I,H,\sigma} \setminus C_{I',H,\sigma'}$. This implies that

$$\sigma|_{\mathcal{N}_i(M_H(i))} = \sigma_i|_{\mathcal{N}_i(M_H(i))} \neq \sigma'|_{\mathcal{N}_i(M_H(i))},$$

which contradicts to (14.1). □

In addition to the above lemma, we will also need the following observation, which states that, if a non-repetitive $S^{\text{NO-REP}}$ is contained in a concept C_{I,H,σ_H}, then $S^{\text{NO-REP}}$ must pass H. This observation follows definitions.

Observation 14.1.5. If a non-repetitive set $S^{\text{NO-REP}}$ is a subset of some concept C_{I,H,σ_H}, then $H \in \mathcal{H}(S^{\text{NO-REP}})$.

With Lemma 14.1.4 and Observation 14.1.5 ready, it is now easy to prove Lemma 14.1.3.

Proof of Lemma 14.1.3. Pick $S^{\text{NO-REP}}$ to be any maximal non-repetitive subset of S such that $I(S^{\text{NO-REP}}) = I(S)$. Clearly, $|S^{\text{NO-REP}}| = |I(S)|$. To see that $|\mathcal{H}(S^{\text{NO-REP}})| \geq 2^{|S|-|I(S)|}$, consider any $\tilde{S}$ such that $S^{\text{NO-REP}} \subseteq \tilde{S} \subseteq S$. Since S is shattered, there exists $I_{\tilde{S}}, H_{\tilde{S}}, \sigma_{\tilde{S}}$ such that $S \cap C_{I_{\tilde{S}},H_{\tilde{S}},\sigma_{\tilde{S}}} = \tilde{S}$. Since $\tilde{S} \supseteq S^{\text{NO-REP}}$, Observation 14.1.5 implies that $H_{\tilde{S}} \in \mathcal{H}(S^{\text{NO-REP}})$. Moreover, from Lemma 14.1.4, $H_{\tilde{S}}$ is distinct for every $\tilde{S}$. As a result, $|\mathcal{H}(S^{\text{NO-REP}})| \geq 2^{|S|-|I(S)|}$ as desired. □

14.1.3.2 Part II: No Large Non-Repetitive Set Passes Many Tests

The goal of this subsection is to show that, if $\text{val}(\mathcal{L})$ is small, then w.h.p. (over the randomness in the construction) every large non-repetitive set passes only few H's. This is formalized as Lemma 14.1.6 below.

Lemma 14.1.6. *If* $\text{val}(\mathcal{L}) \leq \delta^2/100$*, then, with high probability, for every non-repetitive set* $S^{\text{NO-REP}}$ *of size at least* δr*,* $|\mathcal{H}(S^{\text{NO-REP}})| \leq 100n \log|\Sigma|$.

Note that the mapping $S^{\text{NO-REP}} \mapsto (I(S^{\text{NO-REP}}), \phi(S^{\text{NO-REP}}))$ is a bijection from the collection of all non-repetitive sets to $\{(I, \sigma_I) \mid I \subseteq [r], \sigma_I \in \Sigma^{U_I}\}$. Hence, the above lemma is equivalent to the following.

Lemma 14.1.7. *If* $\text{val}(\mathcal{L}) \leq \delta^2/100$*, then, with high probability, for every* $I \subseteq [r]$ *of size at least* δr *and every* $\sigma_I \in \Sigma^{U_I}$*,* $|\mathcal{H}(I, \sigma_I)| \leq 100n \log|\Sigma|$.

Here we use the language in Lemma 14.1.7 instead of Lemma 14.1.6 as it will be easier for us to reuse this lemma later. To prove the lemma, we first need to bound the probability that the assignment σ_I does not violate any constraint induced by a random matching. More precisely, we will prove the following lemma.

Lemma 14.1.8. *For any $I \subseteq [r]$ of size at least δr and any $\sigma_I \in \Sigma^{U_I}$, if $\pi : [r] \to [r]$ is a random permutation of $[r]$, then the probability that σ_I does not violate any constraint in $\bigcup_{i\in[r]} \mathcal{N}_i(M(i))$ is at most $(1-0.1\delta^2)^{\delta r/8}$ where $M(i)$ denote the index that i is matched with in the matching $\Big(\pi(1),\pi(2)\Big),\dots,\Big(\pi(r-1),\pi(r)\Big)$.*

Proof. Let p be any positive odd integer such that $p \le \delta r/2$ and let $i_1,\dots,i_{p-1} \in [r]$ be any $p-1$ distinct elements of $[r]$. We will first show that conditioned on $\pi(1) = i_1,\dots,\pi(p-1) = i_{p-1}$, the probability that σ_I violates a constraint induced by $\pi(p),\pi(p+1)$ (i.e. in $\mathcal{N}_{\pi(p)}(\pi(p+1)) \cup \mathcal{N}_{\pi(p+1)}(\pi(p))$) is at least $0.1\delta^2$.

To see that this is true, let $I_{\ge p} = I \setminus \{i_1,\dots,i_{p-1}\}$. Since $|I| \ge \delta r$, we have $|I_{\ge p}| = |I| - p + 1 \ge \delta r/2 + 1$. Consider the partial assignment $\sigma_{\ge p} = \sigma_I|_{U_{I_{\ge p}}}$. Since $\mathrm{val}(\mathcal{L}) \le 0.01\delta^2$, $\sigma_{\ge p}$ can satisfy at most $0.01\delta^2|E|$ constraints. From Lemma 2.7.7, we have, for every $i \ne j \in I_{\ge p}$, the number of constraints between U_i and U_j are at least $|E|/r^2$. Hence, there are at most $0.01\delta^2 r^2$ pairs of $i < j \in I_{\ge p}$ such that $\sigma_{\ge p}$ does not violate any constraint between U_i and U_j. In other words, there are at least $\binom{|I_{\ge p}|}{2} - 0.01\delta^2 r^2 \ge 0.1\delta^2 r^2$ pairs $i < j \in I_{\ge p}$ such that $\sigma_{\ge p}$ violates some constraints between U_i and U_j. Now, if $\pi(p) = i$ and $\pi(p+1) = j$ for some such pair i, j, then $\phi(S^{\text{NO-REP}})$ violates a constraint induced by $\pi(p),\pi(p+1)$. Thus, we have

$$\Pr\left[\sigma_I \text{ doesn't violate a constraint induced by } \pi(p),\pi(p+1) \;\middle|\; \bigwedge_{t=1}^{p-1} \pi(t) = i_t\right] \le 1 - 0.1\delta^2. \tag{14.2}$$

Let E_p denote the event that σ_I does not violate any constraints induced by $\pi(p)$ and $\pi(p+1)$. We can now bound the desired probability as follows.

$$\begin{aligned}
\Pr\left[\begin{array}{c}\sigma_I \text{ doesn't violate}\\ \text{any constraint in } \bigcup_{i\in[r]}\mathcal{N}_i(M(i))\end{array}\right] &\le \Pr\left[\bigwedge_{\text{odd } p\in[\delta r/2+1]} E_p\right] \\
&= \prod_{\text{odd } p\in[\delta r/2+1]} \Pr\left[E_p \;\middle|\; \bigwedge_{\text{odd } t\in[p-1]} E_t\right] \\
(\text{From (14.2)}) &\le \prod_{\text{odd } p\in[\delta r/2+1]} (1-0.1\delta^2) \\
&\le (1-0.1\delta^2)^{\delta r/4-1},
\end{aligned}$$

which is at most $(1-0.1\delta^2)^{\delta r/8}$ for sufficiently large r (i.e. $r \geq 8/\delta$). □

We can now prove our main lemma.

Proof of Lemma 14.1.7. For a fixed $I \subseteq [r]$ of size at least δr and a fixed $\sigma_I \in \Sigma^{U_I}$, Lemma 14.1.8 tells us that the probability that σ_I does not violate any constraint induced by a single matching is at most $(1-0.1\delta^2)^{\delta r/8}$. Since for each $H \subseteq \mathcal{Y}$ the construction picks ℓ matchings at random, the probability that (I, σ_I) passes each H is at most $(1-0.1\delta^2)^{\delta \ell r/8}$. Recall that we pick $\ell = 80/\delta^3$; this gives the following upper bound on the probability:

$$\Pr[(I,\sigma_I) \text{ passes } H] \leq (1-0.1\delta^2)^{\delta \ell r/8} = (1-0.1\delta^2)^{10r/\delta^2} \leq \left(\frac{1}{1+0.1\delta^2}\right)^{10r/\delta^2} \leq 2^{-r} \tag{14.3}$$

where the last inequality comes from Bernoulli's inequality.

Inequality (14.3) implies that the expected number of H's that (I, σ_I) passes is less than 1. Since the matchings M_H are independent for all H's, we can apply Chernoff bound which implies that

$$\Pr[|\mathcal{H}(I,\sigma_I)| \geq 100n \log|\Sigma|] \leq 2^{-10n\log|\Sigma|} = |\Sigma|^{-10n}.$$

Finally, note that there are at most $2^r|\Sigma|^n$ different (I, σ_I)'s. By union bound, we have

$$\Pr\left[\exists I \subseteq [r], \sigma_I \in \Sigma^{U_I} \text{ s.t. } |I| \geq \delta r \text{ AND } |\mathcal{H}(I,\sigma_I)| \geq 100n\log|\Sigma|\right] \leq (2^r|\Sigma|^n)\left(|\Sigma|^{-10n}\right) \leq |\Sigma|^{-8n},$$

which concludes the proof. □

14.1.3.3 Putting Things Together

Proof of Lemma 14.1.2. From Lemma 14.1.6, every non-repetitive set $S^{\text{NO-REP}}$ of size at least δr, $|\mathcal{H}(S^{\text{NO-REP}})| \leq 100n\log|\Sigma|$. Conditioned on this event happening, we will show that VC-dim$(\mathcal{U}, \mathcal{C}) \leq (1+\delta)r$.

Consider any shattered set $S \subseteq \mathcal{U}$. Lemma 14.1.3 implies that there is a non-repetitive set $S^{\text{NO-REP}}$ of size $|I(S)|$ such that $|\mathcal{H}(S^{\text{NO-REP}})| \geq 2^{|S|-|I(S)|}$. Let us consider two cases:

1. $|I(S)| \leq \delta r$. Since $\mathcal{H}(S^{\text{NO-REP}}) \subseteq \mathcal{P}(\mathcal{Y})$, we have $|S| - |I(S)| \leq |\mathcal{Y}| = r$. This implies that $|S| \leq (1+\delta)r$.

2. $|I(S)| > \delta r$. From our assumption, $|\mathcal{H}(S^{\text{NO-REP}})| \leq 100n \log |\Sigma|$. Thus, $|S| \leq |I(S)| + \log(100n \log |\Sigma|) \leq r + o(r)$; when r is sufficiently large, the latter expression is at most $(1+\delta)r$.

Hence, for sufficiently large r, VC-dim$(\mathcal{U}, \mathcal{C}) \leq (1+\delta)r$ with high probability. □

14.2 Inapproximability of Littlestone's Dimension

We next proceed to Littlestone's Dimension. The main theorem of this section is stated below. Again, note that this theorem, together with Theorem 2.4.4 implies Theorem 14.0.2.

Theorem 14.2.1. *There exists $\varepsilon > 0$ such that there is a randomized reduction from a bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$ to a ground set $\mathcal{U}$ and a concept classes $\mathcal{C}$ such that, if $n \triangleq |A| + |B|, r \triangleq \sqrt{n}/\log n$ and $k \triangleq 10^{10}|E| \log |\Sigma|/r^2$, then the following conditions hold for every sufficiently large n.*

- *(Size) The reduction runs in time $2^{rk} \cdot |\Sigma|^{O(|E|/r)}$ and $|\mathcal{C}|, |\mathcal{U}| \leq 2^{rk} \cdot |\Sigma|^{O(|E|/r)}$.*
- *(Completeness) If $\mathcal{L}$ is satisfiable, then* L-dim$(\mathcal{C}, \mathcal{U}) \geq 2rk$.
- *(Soundness) If* val$(\mathcal{L}) \leq 0.001$*, then* L-dim$(\mathcal{C}, \mathcal{U}) \leq (2-\varepsilon)rk$ *with high probability.*

14.2.1 Why the VC Dimension Reduction Fails for Littlestone's Dimension

It is tempting to think that, since our reduction from the previous section works for VC Dimension, it may also work for Littlestone's Dimension. In fact, thanks to Fact 2.5.5, completeness for that reduction even translates for free to Littlestone's Dimension. Alas, the soundness property does not hold. To see this, let us build a depth-$2r$ mistake tree for $\mathcal{C}, \mathcal{U}$, even when val$(\mathcal{L})$ is small, as follows.

- We assign the test-selection elements to the first r levels of the tree, one element per level. More specifically, for each $s \in \{0,1\}^{<r}$, we assign $y_{|s|+1}$ to s.
- For every string $s \in \{0,1\}^r$, the previous step of the construction gives us a subset of $\mathcal{Y}$ corresponding to the path from root to s; this subset is simply $H_s = \{y_i \in \mathcal{Y} \mid s_i = 1\}$. Let T_{H_s} denote the set of vertices tested by this seed

H_s. Let $\phi_s \in \Sigma^V$ denote an assignment that satisfies all the constraints in T_{H_s}. Note that, since T_{H_s} is of small size (only $\tilde{O}(\sqrt{n})$), even if $\mathrm{val}(\mathcal{L})$ is small, ϕ_s is still likely to exist (and we can decide whether it exists or not in time $2^{\tilde{O}(\sqrt{n})}$).

We then construct the subtree rooted at s that corresponds to ϕ_s by assigning each level of the subtree $x_{i,\phi_s|_{U_i}}$. Specifically, for each $t \in \{0,1\}^{\geq r}$, we assign $x_{|t|-r+1,\phi_{t_{\leq r}}|_{U_{|t|-r+1}}}$ to node t of the tree.

It is not hard to see that the constructed tree is indeed a valid mistake tree. This is because the path from root to each leaf $l \in \{0,1\}^{2r}$ agrees with $C_{I(l),H_{l_{\leq r}},\phi_{l\leq r}}$ (where $I(l) = \{i \in [r] \mid l_i = 1\}$).

14.2.2 The Final Reduction

The above counterexample demonstrates the main difference between the two dimensions: order does not matter in VC Dimension, but it does in Littlestone's Dimension. By moving the test-selection elements up the tree, the tests are chosen before the assignments, which allows an adversary to "cheat" by picking different assignments for different tests. We would like to prevent this, i.e., we would like to make sure that, in the mistake tree, the upper levels of the tree are occupied with the assignment elements whereas the lower levels are assigned test-selection elements. As in the VC Dimension argument, our hope here is that, given such a tree, we should be able to decode an assignment that passes tests on many different tests. Indeed we will tailor our construction to achieve such property.

Recall that, if we use the same reduction as VC Dimension, then, in the completeness case, we can construct a mistake tree in which the first r layers consist solely of assignment elements and the rest of the layers consist of only test-selection elements. Observe that there is no need for different nodes on the r-th layer to have subtrees composed of the same set of elements; the tree would still be valid if we make each test-selection element only work with a specific $s \in \{0,1\}^r$ and create concepts accordingly. In other words, we can modify our construction so that our test-selection elements are $\mathcal{Y} = \{y_{I,i} \mid I \subseteq [r], i \in [r]\}$ and the concept class is $\{C_{I,H,\sigma_H} \mid I \subseteq [r], H \subseteq \mathcal{Y}, \sigma_H \in \Sigma^{T_H}\}$ where the condition that an assignment element lies in C_{I,H,σ_H} is the same as in the VC Dimension reduction, whereas for $y_{I',i}$ to be in C_{I,H,σ_H}, we require not only that $i \in H$ but also that $I = I'$. Intuitively, this should help us, since each $y_{I,i}$ is now only in a small fraction ($\leq 2^{-r}$) of concepts; hence, one would hope that any subtree rooted at any $y_{I,i}$ cannot be too deep, which would indeed implies that the test-selection elements cannot appear in the first few layers of the tree.

Alas, for this modified reduction, it is not true that a subtree rooted at any $y_{I,i}$ has small depth; specifically, we can bound the depth of a subtree $y_{I,i}$ by the log of the number of concepts containing $y_{I,i}$ plus one (for the first layer). Now, note that $y_{I,i} \in C_{I',H,\sigma_H}$ means that $I' = I$ and $i \in H$, but there can be still as many as $2^{r-1} \cdot |\Sigma|^{|T_H|} = |\Sigma|^{O(|E|/r)}$ such concepts. This gives an upper bound of $r + O(|E| \log |\Sigma|/r)$ on the depth of the subtree rooted at $y_{I,i}$. However, $|E| \log |\Sigma|/r = \Theta(\sqrt{n} \log n) = \omega(r)$; this bound is meaningless here since, even in the completeness case, the depth of the mistake tree is only $2r$.

Fortunately, this bound is not useless after all: if we can keep this bound but make the intended tree depth much larger than $|E| \log |\Sigma|/r$, then the bound will indeed imply that no $y_{I,i}$-rooted tree is deep. To this end, our reduction will have one more parameter $k = \Theta(|E| \log |\Sigma|/r)$ where $\Theta(\cdot)$ hides a large constant and the intended tree will have depth $2rk$ in the completeness case; the top half of the tree (first rk layers) will again consist of assignment elements and the rest of the tree composes of the test-selection elements. The rough idea is to make k "copies" of each element: the assignment elements will now be $\{x_{i,\sigma_i,j} \mid i \in [r], \sigma_i \in \Sigma^{U_i}, j \in [k]\}$ and the test-selection elements will be $\{y_{I,i,j} \mid I \subseteq [r] \times [k], j \in [k]\}$. The concept class can then be defined as $\{C_{I,H,\sigma_H} \mid I \subseteq [r] \times [k], H \subseteq [r] \times [k], \sigma_H \in \Sigma^{T_H}\}$ naturally, i.e., H is used as the seed to pick the test set T_H, $y_{I',i,j} \in C_{I,H,\sigma_H}$ iff $I' = I$ and $(i,j) \in H$ whereas $x_{i,\sigma_i,j} \in C_{I,H,\sigma_H}$ iff $(i,j) \in I$ and $\sigma_i|_{(I,\sigma_I)} = \sigma_H|_{(I,\sigma_I)}$. For this concept class, we can again bound the depth of $y_{I,i}$-rooted tree to be $rk + O(|E| \log |\Sigma|/r)$; this time, however, rk is much larger than $|E| \log |\Sigma|/r$, so this bound is no more than, say, $1.001rk$. This is indeed the desired bound, since this means that, for any depth-$1.999rk$ mistake tree, the first $0.998rk$ layers must consist solely of assignment elements.

Unfortunately, the introduction of copies in turn introduces another technical challenge: it is not true any more that a partial assignment to a large set only passes a few tests w.h.p. (i.e. an analogue of Lemma 14.1.7 does not hold). By Inequality (14.3), each H is passed with probability at most 2^{-r}, but now we want to take a union bound there are $2^{rk} \gg 2^r$ different H's. To circumvent this, we will define a map $\tau : \mathcal{P}([r] \times [k]) \to \mathcal{P}([r])$ and use $\tau(H)$ to select the test instead of H itself. The map τ we use in the construction is the *threshold projection* where i is included in H if and only if, for at least half of $j \in [k]$, H contains (i,j). To motivate our choice of τ, recall that our overall proof approach is to first find a node that corresponds to an assignment to a large subset of the Label Cover instance; then argue that it can pass only a few tests, which we hope would imply that the subtree rooted there cannot be too deep. For this implication to be true, we need the following to also hold: for any small subset $\mathcal{H} \subseteq \mathcal{P}([r])$ of $\tau(H)$'s, we have that L-dim$(\tau^{-1}(\mathcal{H}), [r] \times [k])$ is small. This property indeed holds for our choice of τ (see

Lemma 14.2.9).

With all the moving parts explained, we state the full reduction formally in Figure 14.2.

Similar to our VC Dimension proof, we will use the following notation:

- For every $i \in [r]$, let $\mathcal{X}_i \triangleq \{x_{i,\sigma_i,j} \mid \sigma_i \in \Sigma^{U_i}, j \in [k]\}$; we refer to these elements as the i-assignment elements. Moreover, for every $(i,j) \in [r] \times [k]$, let $\mathcal{X}_{i,j} \triangleq \{x_{i,\sigma_i,j} \mid \sigma_i \in \Sigma^{U_i}\}$; we refer to these elements as the (i,j)-assignment elements.
- For every $S \subseteq \mathcal{U}$, let $I(S) = \{i \in [r] \mid S \cap \mathcal{X}_i \neq \varnothing\}$ and $IJ(S) = \{(i,j) \in [r] \times [k] \mid S \cap \mathcal{X}_{i,j} \neq \varnothing\}$.
- A set $S \subseteq \mathcal{X}$ is *non-repetitive* if $|S \cap \mathcal{X}_{i,j}| \leq 1$ for all $(i,j) \in [r] \times [k]$.
- We say that S *passes* $\tilde{H}$ if the following two conditions hold:
  - For every $i \in [r]$ such that $S \cap \mathcal{X}_i \neq \varnothing$, all i-assignment elements of S are consistent on $T_{\tilde{H}}|_{U_i}$, i.e., for every $(i,\sigma_i,j),(i,\sigma_i',j') \in S$, we have $\sigma_i|_{U_i} = \sigma_i'|_{U_i}$.
  - The canonically induced assignment on $T_{\tilde{H}}$ does not violate any constraint (note that the previous condition implies that such assignment is unique).

  We use $\mathcal{H}(S)$ to denote the collection of all seeds $\tilde{H} \subseteq [r]$ that S passes.

We also use the following notation for mistake trees:

- For any subset $S \subseteq \mathcal{U}$ and any function $\rho : S \to \{0,1\}$, let $\mathcal{C}[\rho] \triangleq \{C \in \mathcal{C} \mid \forall a \in S, a \in C \Leftrightarrow \rho(a) = 1\}$ be the collections of all concept that agree with ρ on S. We sometimes abuse the notation and write $\mathcal{C}[S]$ to denote the collection of all the concepts that contain S, i.e., $\mathcal{C}[S] = \{C \in \mathcal{C} \mid S \subseteq C\}$.
- For any binary string s, let $\mathrm{pre}(s) \triangleq \{\varnothing, s_{\leq 1}, \ldots, s_{\leq |s|-1}\}$ denote the set of all proper prefixes of s.
- For any depth-d mistake tree $\mathcal{T}$, let $v_{\mathcal{T},s}$ denote the element assigned to the node $s \in \{0,1\}^{\leq d}$, and let $P_{\mathcal{T},s} \triangleq \{v_{\mathcal{T},s'} \mid s' \in \mathrm{pre}(s)\}$ denote the set of all elements appearing from the path from root to s (excluding s itself). Moreover, let $\rho_{\mathcal{T},s} : P_{\mathcal{T},s} \to \{0,1\}$ be the function corresponding to the path from root to s, i.e., $\rho_{\mathcal{T},s}(s') = s_{|s'|+1}$ for every $s' \in \mathrm{pre}(s)$.

Output Size of the Reduction The output size of the reduction follows immediately from a similar argument as in the VC Dimension reduction. The only different here is that there are 2^{rk} choices for I and H, instead of 2^r choices as in the previous construction.

Completeness. If $\mathcal{L}$ has a satisfying assignment $\sigma^* \in \Sigma^V$, we can construct a depth-rk mistake $\mathcal{T}$ as follows. For $i \in [r], j \in [k]$, we assign $x_{i,\sigma_i^*,j}$ to every node in the $((i-1)k+j)$-th layer of $\mathcal{T}$. Note that we have so far assigned every node in the first rk layers. For the rest of the vertices s's, if s lies in layer $rk+(i-1)k+j$, then we assign $y_{I(\rho^{-1}_{\mathcal{T},s}(1)),i,j}$ to it. It is clear that, for a leaf $s \in \{0,1\}^{rk}$, the concept $C_{I(\rho^{-1}_{\mathcal{T},s}(1)),H_{\mathcal{T},s},\sigma^*}$ agrees with the path from root to s where $H_{\mathcal{T},s}$ is defined as $\{(i,j) \mid y_{I(\rho^{-1}_{\mathcal{T},s}(1)),i,j} \in \rho^{-1}_{\mathcal{T},s}(1)\}$. Hence, L-dim$(\mathcal{C},\mathcal{U}) \geq 2rk$.

14.2.3 Soundness

Next, we will prove the soundness of our reduction, stated more precisely below. Note that this lemma, together with completeness and output size properties we argue above, implies Theorem 14.2.1 with $\varepsilon = 0.001$.

Lemma 14.2.2. *Let $(\mathcal{C},\mathcal{U})$ be the output from the reduction in Figure 14.2 on input $\mathcal{L}$. If* val$(\mathcal{L}) \leq 0.001$*, then* L-dim$(\mathcal{C},\mathcal{U}) \leq 1.999rk$ *with high probability.*

Roughly speaking, the overall strategy of our proof of Lemma 14.2.2 is as follows:

1. First, we will argue that any subtree rooted at any test-selection element must be shallow (of depth $\leq 1.001rk$). This means that, if we have a depth-$1.999rk$ mistake tree, then the first $0.998rk$ levels must be assigned solely assignment elements.

2. We then argue that, in this $0.998rk$-level mistake tree of assignment elements, we can always extract a leaf s such that the path from root to s indicates inclusion of a large non-repetitive set. In other words, the path to s can be decoded into a (partial) assignment for the Label Cover instance $\mathcal{L}$.

3. Let the leaf from the previous step be s and the non-repetitive set be $S^{\text{NO-REP}}$. Our goal now is to show that the subtree rooted as s must have small depth. We start working towards this by showing that, with high probability, there are few tests that agree with $S^{\text{NO-REP}}$. This is analogous to Part II of the VC Dimension proof.

4. With the previous steps in mind, we only need to argue that, when $|\mathcal{H}(S^{\text{NO-REP}})|$ is small, the Littlestone's dimension of all the concepts that contains $S^{\text{NO-REP}}$

(i.e. L-dim($\mathcal{C}[S^{\text{NO-REP}}],\mathcal{U}$)) is small. Thanks to Fact 2.5.6, it is enough for us to bound L-dim($\mathcal{C}[S^{\text{NO-REP}}],\mathcal{X}$) and L-dim($\mathcal{C}[S^{\text{NO-REP}}],\mathcal{Y}$) separately. For the former, our technique from the second step also gives us the desired bound; for the latter, we prove that L-dim($\mathcal{C}[S^{\text{NO-REP}}],\mathcal{Y}$) is small by designing an algorithm that provides correct predictions on a constant fraction of the elements in $\mathcal{Y}$.

Let us now proceed to the details of the proofs.

14.2.3.1 Part I: Subtree of a Test-Selection Assignment is Shallow

Lemma 14.2.3. *For any $y_{I,i,j} \in \mathcal{Y}$,* L-dim($\mathcal{C}[\{y_{I,i,j}\}],\mathcal{U}$) $\le rk + (4|E|\ell/r)\log|\Sigma| \le 1.001rk$.

Note that the above lemma implies that, in any mistake tree, the depth of the subtree rooted at any vertex s assigned to some $y_{I,i,j} \in \mathcal{Y}$ is at most $1+1.001rk$. This is because every concept that agrees with the path from the root to $s1$ must be in $\mathcal{C}[\{y_{I,i,j}\}]$, which has depth at most $1.001rk$.

Proof of Lemma 14.2.3. Consider any $C_{I',H,\sigma_{\tau(H)}} \in \mathcal{C}[\{y_{I,i,j}\}],\mathcal{U})$. Since $y_{I,i,j} \in C_{I',H,\sigma_{\tau(H)}}$, we have $I = I'$. Moreover, from Lemma 2.7.7, we know that $\left|\mathcal{N}_i\left(M_{\tau(H)}(i)\right)\right| \le 4|E|\ell/r^2$, which implies that $|T_{\tau(H)}| \le 4|E|\ell/r$. This means that there are only at most $|\Sigma|^{4|E|\ell/r}$ choices of $\sigma_{\tau(H)}$. Combined with the fact that there are only 2^{rk} choices of H, we have $|\mathcal{C}[\{y_{I,i,j}\}]| \le 2^{rk}\cdot|\Sigma|^{4|E|\ell/r}$. Fact 2.5.5 then implies the lemma. □

14.2.3.2 Part II: Every Deep Mistake Tree Contains a Non-Repetitive Set

The goal of this part of the proof is to show that, for mistake tree of $\mathcal{X},\mathcal{C}$ of depth slightly less than rk, there exists a leaf s such that the corresponding path from root to s indicates an inclusion of a large non-repetitive set; in our notation, this means that we would like to identify a leaf s such that $IJ(\rho^{-1}_{\mathcal{T},s}(1))$ is large. Since we will also need a similar bound later in the proof, we will prove the following lemma, which is a generalization of the stated goal that works even for the concept class $\mathcal{C}[S^{\text{NO-REP}}]$ for any non-repetitive $S^{\text{NO-REP}}$. To get back the desired bound, we can simply set $S^{\text{NO-REP}} = \varnothing$.

Lemma 14.2.4. *For any non-repetitive set $S^{\text{NO-REP}}$ and any depth-d mistake tree $\mathcal{T}$ of $\mathcal{X},\mathcal{C}[S^{\text{NO-REP}}]$, there exists a leaf $s \in \{0,1\}^d$ such that $|IJ(\rho^{-1}_{\mathcal{T},s}(1)) \setminus IJ(S^{\text{NO-REP}})| \ge d-r$.*

The proof of this lemma is a double counting argument where we count a specific class of leaves in two ways, which ultimately leads to the above bound. The leaves that we focus on are the leaves $s \in \{0,1\}^d$ such that, for every (i,j) such that an (i,j)-assignment element appears in the path from root to s but not in $S^{\text{NO-REP}}$, the first appearance of (i,j)-assignment element in the path is included. In other words, for every $(i,j) \in IJ(P_{\mathcal{T},s}) \setminus IJ(S^{\text{NO-REP}})$, if we define $u_{i,j} \triangleq \inf_{s' \in \text{pre}(s), v_{\mathcal{T},s'} \in \mathcal{X}_{i,j}} |s'|$, then $s_{u_{i,j}}$ must be equal to 1. We call these leaves the *good* leaves. Denote the set of good leaves of $\mathcal{T}$ by $\mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$.

Our first way of counting is the following lemma. Informally, it asserts that different leaves agree with different sets $\tilde{H} \subseteq [r]$. This can be thought of as an analogue of Lemma 14.1.4 in our proof for VC Dimension. Note that this lemma immediately gives an upper bound of 2^r on $|\mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}|$.

Lemma 14.2.5. *For any depth-d mistake tree $\mathcal{T}$ of $\mathcal{X}, \mathcal{C}[S^{\text{NO-REP}}]$ and any different $s_1, s_2 \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$, if C_{I_1,H_1,σ_1} agrees with s_1 and C_{I_2,H_2,σ_2} agrees with s_2 for some $I_1, I_2, H_1, H_2, \sigma_1, \sigma_2$, then $\tau(H_1) \neq \tau(H_2)$.*

Proof. Suppose for the sake of contradiction that there exist $s_1 \neq s_2 \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$, $H_1, H_2, I_1, I_2, \sigma_1, \sigma_2$ such that C_{I_1,H_1,σ_1} and C_{I_2,H_2,σ_2} agree with s_1 and s_2 respectively, and $\tau(H_1) = \tau(H_2)$. Let s be the common ancestor of s_1, s_2, i.e., s is the longest string in $\text{pre}(s_1) \cap \text{pre}(s_2)$. Assume w.l.o.g. that $(s_1)_{|s|+1} = 0$ and $(s_2)_{|s|+1} = 1$. Consider the node $v_{\mathcal{T},s}$ in tree $\mathcal{T}$ where the paths to s_1, s_2 split; suppose that this is $x_{i,\sigma_i,j}$. Therefore $x_{i,\sigma_i,j} \in C_{I_2,H_2,\sigma_2} \setminus C_{I_1,H_1,\sigma_1}$.

We now argue that there is some $x_{i,\sigma_i',j}$ (with the same i,j but a different assignment σ_i') that is in both concepts, i.e. $x_{i,\sigma_i',j} \in C_{I_2,H_2,\sigma_2} \cap C_{I_1,H_1,\sigma_1}$. We do this by considering two cases:

- If $(i,j) \in IJ(S^{\text{NO-REP}})$, then there is $x_{i,\sigma_i',j} \in S^{\text{NO-REP}} \subseteq C_{I_1,H_1,\sigma_1}, C_{I_2,H_2,\sigma_2}$ for some $\sigma_i' \in \Sigma^{U_i}$.
- Suppose, that $(i,j) \notin IJ(S^{\text{NO-REP}})$. Since s_1 is a good leaf, there is some $t \in \text{pre}(s)$ such that $v_{\mathcal{T},t} = x_{i,\sigma_i',j}$ for some $\sigma_i' \in \Sigma^{U_i}$ and t is included by the path (i.e. $s_{|t|+1} = 1$). This also implies that $x_{i,\sigma_i',j}$ is in both C_{I_1,H_1,σ_1} and C_{I_2,H_2,σ_2}.

Now, since both $x_{i,\sigma_i,j}$ and $x_{i,\sigma_i',j}$ are in the concept C_{I_2,H_2,σ_2}, we have $(i,j) \in I_2$ and

$$\sigma_i|_{\mathcal{N}_i(M_{\tau(H_1)})} = \sigma_2|_{\mathcal{N}_i(M_{\tau(H_1)})} = \sigma_i'|_{\mathcal{N}_i(M_{\tau(H_1)})}. \tag{14.4}$$

On the other hand, since C_{I_1,H_1,σ_1} contains $x_{i,\sigma_i',j}$ but not $x_{i,\sigma_i,j}$, we have $(i,j) \in I_1$ and

$$\sigma_i|_{\mathcal{N}_i(M_{\tau(H_2)})} \neq \sigma_1|_{\mathcal{N}_i(M_{\tau(H_2)})} = \sigma_i'|_{\mathcal{N}_i(M_{\tau(H_2)})}. \tag{14.5}$$

which contradicts (14.4) since $\tau(H_1) = \tau(H_2)$. □

Next, we will present another counting argument which gives a lower bound on the number of good leaves, which, together with Lemma 14.2.5, yields the desired bound.

Proof of Lemma 14.2.4. For any depth-d mistake tree $\mathcal{T}$ of $\mathcal{C}[S^{\text{NO-REP}}], \mathcal{X}$, let us consider the following procedure which recursively assigns a weight λ_s to each node s in the tree. At the end of the procedure, all the weight will be propagated from the root to good leaves.

1. For every non-root node $s \in \{0,1\}^{\geq 1}$, set $\lambda_s \leftarrow 0$. For root $s = \varnothing$, let $\lambda_\varnothing \leftarrow 2^d$.
2. While there is an internal node $s \in \{0,1\}^{<d}$ such that $\lambda_s > 0$, do the following:
   a) Suppose that $v_s = x_{i,\sigma_i,j}$ for some $i \in [r], \sigma_i \in \Sigma^{U_i}$ and $j \in [k]$.
   b) If so far no (i,j)-element has appeared in the path or in $S^{\text{NO-REP}}$, i.e., $(i,j) \notin IJ(P_{\mathcal{T},s}) \cup IJ(S^{\text{NO-REP}})$, then $\lambda_{s1} \leftarrow \lambda_s$. Otherwise, set $\lambda_{s0} = \lambda_{s1} = \lambda_s/2$.
   c) Set $\lambda_s \leftarrow 0$.

The following observations are immediate from the construction:

- The total of λ's over all the tree, $\sum_{s\in\{0,1\}^{\leq d}} \lambda_d$ always remain 2^d.
- At the end of the procedure, for every $s \in \{0,1\}^{\leq d}$, $\lambda_s \neq 0$ if and only if $s \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$.
- If $s \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$, then $\lambda_s = 2^{|IJ(\rho^{-1}_{\mathcal{T},s}(1)) \setminus IJ(S^{\text{NO-REP}})|}$ at the end of the execution.

Note that the last observation comes from the fact that λ always get divides in half when moving down one level of the tree unless we encounter an (i,j)-assignment element for some i,j that never appears in the path or in $S^{\text{NO-REP}}$ before. For any good leaf s, the set of such (i,j) is exactly the set $IJ(\rho^{-1}_{\mathcal{T},s}(1)) \setminus IJ(S^{\text{NO-REP}})$.

As a result, we have $2^d = \sum_{s\in\mathcal{G}_\mathcal{T}} 2^{|IJ(\rho^{-1}_{\mathcal{T},s}(1)) \setminus IJ(S^{\text{NO-REP}})|}$. Since Lemma 14.2.5 implies that $|\mathcal{G}_{\mathcal{T},S}| \leq 2^r$, we can conclude that there exists $s \in \mathcal{G}_{\mathcal{T},S^{\text{NO-REP}}}$ such that $|IJ(\rho^{-1}_{\mathcal{T},s}(1)) \setminus IJ(S^{\text{NO-REP}})| \geq d - r$ as desired. □

14.2.3.3 Part III: No Large Non-Repetitive Set Passes Many Test

The main lemma of this subsection is the following, which is analogous to Lemma 14.1.6

Lemma 14.2.6. *If* $\mathrm{val}(\mathcal{L}) \le 0.001$*, then, with high probability, for every non-repetitive set* $S^{\text{NO-REP}}$ *of size at least* $0.99rk$*,* $|\mathcal{H}(S^{\text{NO-REP}})| \le 100n \log|\Sigma|$*.*

Proof. For every $I \subseteq [r]$, let $U_I \triangleq \bigcup_{i\in I} U_i$. For every $\sigma_I \in \Sigma^{U_I}$ and every $\tilde{H} \subseteq \mathcal{Y}$, we say that (I, σ_I) *passes* $\tilde{H}$ if σ_I does not violate any constraint in $T_{\tilde{H}}$. Note that this definition and the way the test is generated in the reduction is the same as that of the VC Dimension reduction. Hence, if we can apply Lemma 14.1.7 with $\delta = 0.99$, which implies the following: with high probability, for every $I \subseteq [r]$ of size at least $0.99r$ and every $\sigma_I \in \Sigma^{U_I}$, $|\mathcal{H}(I, \sigma_I)| \le 100n \log|\Sigma|$. Conditioned on this event happening, we will show that, for every non-repetitive set $S^{\text{NO-REP}}$ of size at least $0.99rk$, $|\mathcal{H}(S^{\text{NO-REP}})| \le 100n \log|\Sigma|$.

Consider any non-repetitive set $S^{\text{NO-REP}}$ of size $0.99rk$. Let $\sigma_{I(S^{\text{NO-REP}})}$ be an assignment on $U_{I(S^{\text{NO-REP}})}$ such that, for each $i \in I(S^{\text{NO-REP}})$, we pick one $x_{i,\sigma_i,j} \in S^{\text{NO-REP}}$ (if there are more than one such x's, pick one arbitrarily) and let $\sigma_{I(S^{\text{NO-REP}})}|_{U_i} = \sigma_i$. It is obvious that $\mathcal{H}(S^{\text{NO-REP}}) \subseteq \mathcal{H}(I(S^{\text{NO-REP}}), \sigma_{I(S^{\text{NO-REP}})})$. Since $S^{\text{NO-REP}}$ is non-repetitive and of size at least $0.99rk$, we have $|I(S^{\text{NO-REP}})| \ge 0.99r$, which means that $|\mathcal{H}(I(S^{\text{NO-REP}}), \sigma_{I(S^{\text{NO-REP}})})| \le 100n \log|\Sigma|$ as desired. □

14.2.3.4 Part IV: A Subtree Containing $S^{\text{no-rep}}$ Must be Shallow

In this part, we will show that, if we restrict ourselves to only concepts that contain some non-repetitive set $S^{\text{NO-REP}}$ that passes few tests, then the Littlestone's Dimension of this restrictied concept class is small. Therefore when we build a tree for the whole concept class $\mathcal{C}$, if a path from root to some node indicates an inclusion of a non-repetitive set that passes few tests, then the subtree rooted at this node must be shallow.

Lemma 14.2.7. *For every non-repetitive set* $S^{\text{NO-REP}}$*,*

$$\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{U}) \le 1.75rk - |S^{\text{NO-REP}}| + r + 1000k\sqrt{r}\log(|\mathcal{H}(S^{\text{NO-REP}})| + 1).$$

We prove the above lemma by bounding $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{X})$ and $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y})$ separately, and combining them via Fact 2.5.6. First, we can bound $\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{X})$ easily by applying Lemma 14.2.4 coupled with the fact that $|IJ(S^{\text{NO-REP}})| = |S^{\text{NO-REP}}|$ for every non-repetitive $S^{\text{NO-REP}}$. This gives the following corollary.

Corollary 14.2.8. *For every non-repetitive set* $S^{\text{NO-REP}}$*,*

$$\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{X}) \le rk - |S^{\text{NO-REP}}| + r.$$

We will next prove the following bound on L-dim$(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y})$. Note that Corollary 14.2.8, Lemma 14.2.9, and Fact 2.5.6 immediately imply Lemma 14.2.7.

Lemma 14.2.9. *For every non-repetitive set $S^{\text{NO-REP}}$,*

$$\text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}], \mathcal{Y}) \leq 0.75rk + 500k\sqrt{r}\log(|\mathcal{H}(S^{\text{NO-REP}})| + 1).$$

The overall outline of the proof of Lemma 14.2.9 is that we will design a prediction algorithm whose mistake bound is at most $0.75rk + 1000k\sqrt{r}\log|\mathcal{H}(S^{\text{NO-REP}})|$. Once we design this algorithm, Lemma 2.5.4 immediately implies Lemma 14.2.9. To define our algorithm, we will need the following lemma, which is a general statement that says that, for a small collection of H's, there is a some $\tilde{H}^* \subset [r]$ that agrees with almost half of every H in the collection.

Lemma 14.2.10. *Let $\mathcal{H} \subseteq \mathcal{P}([r])$ be any collections of subsets of $[r]$, there exists $\tilde{H}^* \subseteq [r]$ such that, for every $\tilde{H} \in \mathcal{H}$, $|\tilde{H}^* \Delta \tilde{H}| \leq 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)$ where Δ denotes the symmetric difference between two sets.*

Proof. We use a simple probabilistic method to prove this lemma. Let $\tilde{H}^r$ be a random subset of $[r]$ (i.e. each $i \in [r]$ is included independently with probability 0.5). We will show that, with non-zero probability, $|\tilde{H}^r \Delta \tilde{H}| \leq 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)$ for all $\tilde{H} \in \mathcal{H}$, which immediately implies that a desired $\tilde{H}^*$ exists.

Fix $\tilde{H} \in \mathcal{H}$. Observe that $|\tilde{H}^r \Delta \tilde{H}|$ can be written as $\sum_{i \in [r]} \mathbb{1}[i \in (\tilde{H}^r \Delta \tilde{H})]$. For each i, $\mathbb{1}[i \in (\tilde{H}^r \Delta \tilde{H})]$ is a $0,1$ random variable with mean 0.5 independent of other $i' \in [r]$. Applying Chernoff bound here yields

$$\Pr[|\tilde{H}^r \Delta \tilde{H}| > 0.5r + 1000\sqrt{r}\log(|cH| + 1)] \leq 2^{-\log^2(|\mathcal{H}|+1)} \leq \frac{1}{|\mathcal{H}| + 1}.$$

Hence, by union bound, we have

$$\Pr[\exists \tilde{H} \in \mathcal{H}, |\tilde{H}^r \Delta \tilde{H}| > 0.5r + 1000\sqrt{r}\log(|cH| + 1)] \leq \frac{|\mathcal{H}|}{|\mathcal{H}| + 1} < 1.$$

In other words, $|\tilde{H}^r \Delta \tilde{H}| \leq 0.5r + 1000\sqrt{r}\log(|\mathcal{H}| + 1)$ for all $\tilde{H} \in \mathcal{H}$ with non-zero probability as desired. □

We also need the following observation, which is an analogue of Observation 14.1.5 in the VC Dimension proof; it follows immediately from definition of $\mathcal{H}(S)$.

Observation 14.2.11. If a non-repetitive set $S^{\text{NO-REP}}$ is a subset of some concept $C_{I,H,\sigma_{\tau(H)}}$, then $\tau(H) \in \mathcal{H}(S^{\text{NO-REP}})$.

With Lemma 14.2.10 and Observation 14.2.11 in place, we are now ready to prove Lemma 14.2.9.

Proof of Lemma 14.2.9. Let $\tilde{H}^* \subseteq [r]$ be the set guaranteed by applying Lemma 14.2.10 with $\mathcal{H} = \mathcal{H}(S^{\text{NO-REP}})$. Let $H^* \triangleq \tilde{H}^* \times [k]$.

Our prediction algorithm will be very simple: it always predicts according to H^*; i.e., on an input[2] $y \in \mathcal{Y}$, it outputs $\mathbb{1}[y \in H^*]$. Consider any sequence $(y_1, h_1), \ldots, (y_w, h_w)$ that agrees with a concept $C_{I,H,\sigma_{\tau(H)}} \in \mathcal{C}[S^{\text{NO-REP}}]$. Observe that the number of incorrect predictions of our algorithm is at most $|H^* \Delta H|$.

Since $C_{I,H,\sigma_{\tau(H)}} \in \mathcal{C}[S^{\text{NO-REP}}]$, Observation 14.2.11 implies that $\tau(H) \in \mathcal{H}(S^{\text{NO-REP}})$. This means that $|\tau(H)\Delta\tilde{H}^*| \leq 0.5r + 1000\sqrt{r}\log(|\mathcal{H}|+1)$. Now, let us consider each $i \in [r] \setminus (\tau(H)\Delta\tilde{H}^*)$. Suppose that $i \in \tau(H) \cap \tilde{H}^*$. Since $i \in \tau(H)$, at least $k/2$ elements of $\mathcal{Y}_i$ are in H and, since $i \in \tilde{H}^*$, we have $\mathcal{Y}_i \subseteq H^*$. This implies that $|(H^*\Delta H) \cap Y_i| \leq k/2$. A similar bound can also be derived when $i \notin \tau(H) \cap \tilde{H}^*$. As a result, we have

$$\begin{aligned}
|H^*\Delta H| &= \sum_{i\in[r]} |(H^*\Delta H) \cap Y_i| \\
&= \sum_{i\in\tau(H)\Delta\tilde{H}^*} |(H^*\Delta H) \cap Y_i| + \sum_{i\in[r]\setminus(\tau(H)\Delta\tilde{H}^*)} |(H^*\Delta H) \cap Y_i| \\
&\leq (|\tau(H)\Delta\tilde{H}^*|)(k) + (r - |\tau(H)\Delta\tilde{H}^*|)(k/2) \\
&\leq 0.75rk + 500k\sqrt{r}\log(|\mathcal{H}|+1),
\end{aligned}$$

concluding our proof of Lemma 14.2.9. □

14.2.3.5 Putting Things Together

Proof of Lemma 14.2.2. Suppose for the sake of contradiction that $\text{val}(\mathcal{L}) \leq 0.001$ but $\text{L-dim}(\mathcal{C},\mathcal{U}) \geq 1.999rk$. Consider any depth-$1.999rk$ mistake tree $\mathcal{T}$ of $\mathcal{C},\mathcal{U}$. From Lemma 14.2.3, we know that no test-selection element is assigned to any node in the first $1.999rk - 1.001rk - 1 \geq 0.997rk$ levels. In other words, the tree induced by the first $0.997rk$ levels is simply a mistake tree of $\mathcal{C}, \mathcal{X}$. By Lemma 14.2.4 with $S^{\text{NO-REP}} = \varnothing$, there exists $s \in \{0,1\}^{0.997rk}$ such that $|IJ(\rho^{-1}_{\mathcal{T},s}(1))| \geq 0.997rk - r \geq 0.996rk$.

Since $|IJ(\rho^{-1}_{\mathcal{T},s}(1))| \geq 0.996rk$, there exists a non-repetitive set $S^{\text{NO-REP}} \subseteq \rho^{-1}_{\mathcal{T},s}(1)$ of size $0.996rk$. Consider the subtree rooted at s. This is a mistake tree of $\mathcal{C}[\rho_{\mathcal{T},s}],\mathcal{U}$ of depth $1.002rk$. Since $S^{\text{NO-REP}} \subseteq \rho^{-1}_{\mathcal{T},s}(1)$, we have $\mathcal{C}[\rho_{\mathcal{T},s}] \subseteq \mathcal{C}[S^{\text{NO-REP}}]$. However, this

[2] We assume w.l.o.g. that input elements are distinct; if an element appears multiple times, we know the correct answer from its first appearance and can always correctly predict it afterwards.

implies

$$\begin{aligned}
1.002rk &\le \text{L-dim}(\mathcal{C}[\rho_{\mathcal{T},s}],\mathcal{U}) \\
&\le \text{L-dim}(\mathcal{C}[S^{\text{NO-REP}}],\mathcal{U}) \\
(\text{From Lemma 14.2.7}) &\le 1.75rk - 0.996rk + r + 100k\sqrt{r}\log(|\mathcal{H}(S^{\text{NO-REP}})| + 1) \\
(\text{From Lemma 14.2.6}) &\le 0.754rk + r + 100k\sqrt{r}\log(100n\log|\Sigma| + 1) \\
&= 0.754rk + o(rk),
\end{aligned}$$

which is a contradiction when r is sufficiently large. □

14.3 Quasi-polynomial Algorithm for Littlestone's Dimension

Theorem 14.3.1 (Quasi-polynomial Time Algorithm for Littlestone's Dimension)**.** *Littlestone's Dimension can be computed (exactly) in time*

$$O\left(|\mathcal{C}|\cdot(2|\mathcal{U}|)^{\text{L-dim}(\mathcal{C},\mathcal{U})}\right) \le |\mathcal{C}|\cdot|\mathcal{U}|^{O\left(\log|\mathcal{C}|\right)}.$$

Proof. We prove by induction on $\text{L-dim}(\mathcal{C},\mathcal{U})$. Assume by induction that the bound on the running time holds for all concept classes of smaller dimension. Enumerate over all posible roots $x \in \mathcal{U}$ for the mistake tree. For each root x, partition $\mathcal{C}$ into $\mathcal{C}_{x=1}$, concepts that include x, and $\mathcal{C}_{x=0}$ (this takes time $O(|\mathcal{C}|)$). Now any mistake tree rooted at x has depth exactly $1 + \min\{\text{L-dim}(\mathcal{C}_{x=0},\mathcal{U}), \text{L-dim}(\mathcal{C}_{x=1},\mathcal{U})\}$. We attempt to compute both $\text{L-dim}(\mathcal{C}_{x=0},\mathcal{U}), \text{L-dim}(\mathcal{C}_{x=1},\mathcal{U})$ until one of the computations terminates. Since one of $\mathcal{C}_{x=0},\mathcal{C}_{x=1}$ has a lower Littlestone's Dimension, it follows by our induction hypothesis that we can compute it in time $O\left(|\mathcal{C}|\cdot(2|\mathcal{U}|)^{\text{L-dim}(\mathcal{C},\mathcal{U})-1}\right)$. Multiplying by 2 because we're computing for two classes and by $\mathcal{U}$ because we're performing this computation for every root x, the total upper bound on the running time follows. □

Input: A bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e\in E})$ and a parameter $\delta > 0$.
Output: A ground set $\mathcal{U}$ and a concept class $\mathcal{C}$.
The procedure to generate $(\mathcal{U}, \mathcal{C})$ works as follows:

- Let r be $\sqrt{n}/\log n$ where $n = |A| + |B|$. Use Lemma 2.7.7 to partition $A \cup B$ into r blocks $U_1, \ldots, U_r$.
- For convenience, we assume that r is even. Moreover, for $i \neq j \in [r]$, let $\mathcal{N}_i(j) \subseteq U_i$ denote the set of all vertices in U_i with at least one neighbor in U_j (w.r.t. the graph (A, B, E)). We also extend this notation naturally to a set of j's; for $J \subseteq [r]$, $\mathcal{N}_i(J)$ denotes $\bigcup_{j\in J} \mathcal{N}_i(j)$.
- The universe $\mathcal{U}$ consists of two types of elements, as described below.
  - *Assignment elements*: for every $i \in [r]$ and every partial assignment $\sigma_i \in \Sigma^{U_i}$, there is an assignment element x_{i,σ_i} corresponding to it. Let $\mathcal{X}$ denote all the assignment elements, i.e., $\mathcal{X} = \{x_{i,\sigma_i} \mid i \in [r], \sigma_i \in \Sigma^{U_i}\}$.
  - *Test selection elements*: there are r test selection elements, which we will call $y_1, \ldots, y_r$. Let $\mathcal{Y}$ denote the set of all test selection elements.
- The concepts in $\mathcal{C}$ are defined by the following procedure.
  - Let $\ell \triangleq 80/\delta^3$ be the number of matchings to be tested.
  - For each $H \subseteq \mathcal{Y}$, we randomly select ℓ permutations $\pi_H^{(1)}, \ldots, \pi_H^{(\ell)} : [r] \to [r]$; this gives us ℓ matchings (i.e. the t-th matching is $\left(\pi_H^{(t)}(1), \pi_H^{(t)}(2)\right), \ldots, \left(\pi_H^{(t)}(r-1), \pi_H^{(t)}(r)\right)$). For brevity, let us denote the set of (up to ℓ) elements that i is matched with in the matchings by $M_H(i)$. Let $T_H = \bigcup_i \mathcal{N}_i(M_H(i))$
  - For every $I \subseteq [r], H \subseteq \mathcal{Y}$ and for every partial assignment $\sigma_H \in \Sigma^{T_H}$ that does not violate any constraints, we create a concept C_{I,H,σ_H} such that each $x_{i,\sigma_i} \in \mathcal{X}$ is included in C_{I,H,σ_H} if and only if $i \in I$ and σ_i is consistent with σ_H, i.e., $\sigma_i|_{\mathcal{N}_i(M_H(i))} = \sigma_H|_{\mathcal{N}_i(M_H(i))}$ whereas $y_i \in \mathcal{Y}$ in included in C_{I,H,σ_H} if and only if $y \in H$.

Figure 14.1: Reduction from Label Cover to VC Dimension

Input: A bi-regular Label Cover instance $\mathcal{L} = (A, B, E, \Sigma, \{\pi_e\}_{e \in E})$.
Output: A ground set $\mathcal{U}$ and a concept class $\mathcal{C}$.
The procedure to generate $(\mathcal{U}, \mathcal{C})$ works as follows:

- Let $r, U_1, \ldots, U_r, \mathcal{N}$ be defined in the same manner as in Reduction 14.1 and let $k \triangleq 10^{10}|E| \log |\Sigma|/r^2$.
- The universe $\mathcal{U}$ consists of two types of elements, as described below.
  - *Assignment elements*: for every $i \in [r]$, every partial assignment $\sigma_i \in \Sigma^{U_i}$ and every $j \in [k]$, there is an assignment element $x_{i,\sigma_i,j}$ corresponding to it. Let $\mathcal{X}$ denote all the assignment elements, i.e., $\mathcal{X} = \{x_{i,\sigma_i,j} \mid i \in [r], \sigma_i \in \Sigma^{U_i}, j \in [k]\}$.
  - *Test-selection elements*: there are $rk(2^{rk})$ test-selection elements, which we will call $y_{I,i,j}$ for every $i \in [r], j \in [k], I \subseteq [r] \times [k]$. Let $\mathcal{Y}$ denote the set of all test-selection elements. Let $\mathcal{Y}_i$ denote $\{y_{I,i,j} \mid I \subseteq [r] \times [k], j \in [k]\}$. We call the elements of $\mathcal{Y}_i$ *i-test-selection elements*.
- The concepts in $\mathcal{C}$ are defined by the following procedure.
  - Let $\ell \triangleq 1000$ be the number of matchings to be tested.
  - For each $\tilde{H} \subseteq [r]$, we randomly select ℓ permutations $\pi^{(1)}_{\tilde{H}}, \ldots, \pi^{(\ell)}_{\tilde{H}} : [r] \to [r]$; this gives us ℓ matchings (i.e. the t-th matching is $\left(\pi^{(t)}_{\tilde{H}}(1), \pi^{(t)}_{\tilde{H}}(2)\right), \ldots, \left(\pi^{(t)}_{\tilde{H}}(r-1), \pi^{(t)}_{\tilde{H}}(r)\right)$). Denote the set of elements that i is matched with in the matchings by $M_{\tilde{H}}(i)$. Let $T_H = \bigcup_i \mathcal{N}_i(M_{\tilde{H}}(i))$
  - Let $\tau : \mathcal{P}([r] \times [k]) \to \mathcal{P}([r])$ denote the *threshold projection* operation where each $i \in [r]$ is included in $\tau(H)$ if and only if H contains at least half of the i-test-selection elements, i.e., $\tau(H) = \{i \in [r] \mid |H \cap \mathcal{Y}_i| \geq k/2\}$.
  - For every $I \subseteq [r] \times [k], H \subseteq [r] \times [k]$ and for every partial assignment $\sigma_{\tau(H)} \in \Sigma^{T_{\tau(H)}}$ that does not violate any constraints, we create a concept $C_{I,H,\sigma_{\tau(H)}}$ such that each $x_{i,\sigma_i,j} \in \mathcal{X}$ is included in $C_{I,H,\sigma_{\tau(H)}}$ if and only if $(i,j) \in I$ and σ_i is consistent with $\sigma_{\tau(H)}$, i.e., $\sigma_i|_{\mathcal{N}_i(M_{\tau(H)}(i))} = \sigma_{\tau(H)}|_{\mathcal{N}_i(M_{\tau(H)}(i))}$ whereas each $y_{I',i,j} \in \mathcal{Y}$ in included in $C_{I,H,\sigma_{\tau(H)}}$ if and only if $(i,j) \in H$ and $I' = I$.

Figure 14.2: Reduction from Label Cover to Littlestone's Dimension

Chapter 15

Signaling

Many classical questions in economics involve extracting information from strategic agents. Lately, there has been growing interest within algorithmic game theory in *signaling*: the study of how to *reveal information* to strategic agents (see e.g. [MS12; DIR13; Eme+14; Dug14; Che+15b] and references therein). Signaling has been studied in many interesting economic and game theoretic settings. Among them, Zero-Sum Signaling proposed by Dughmi [Dug14] stands out as a canonical problem that cleanly captures the computational nature of signaling. In particular, focusing on zero-sum games clears away issues of equilibrium selection and computational tractability of finding an equilibrium.

Definition 15.0.1 (Zero-Sum Signaling [Dug14])**.** Alice and Bob play a Bayesian zero-sum game where the payoff matrix M is drawn from a publicly known prior. The signaler Sam privately observes the state of nature (i.e. the payoff matrix), and then publicly broadcasts a signal $\varphi(M)$ to both Alice and Bob. Alice and Bob Bayesian-update their priors according to $\varphi(M)$'s and play the Nash equilibrium of the expected game; but they receive payoffs according to the true M. Sam's goal is to design an efficient signaling scheme φ (a function from payoff matrices to strings) that maximizes Alice's expected payoff.

Dughmi's [Dug14] main result proves that assuming the hardness of the Planted Clique problem, there is no additive FPTAS for Zero-Sum Signaling. The main open question left by [Dug14] is whether there exists an additive PTAS. Here we answer this question in the negative: we prove that assuming the Exponential Time Hypothesis (ETH), obtaining an additive-ϵ-approximation (for some constant $\epsilon > 0$) requires quasi-polynomial time $(n^{\tilde{\Omega}(\lg n)})$. This result is tight thanks to a recent quasi-polynomial $(n^{\frac{\lg n}{\mathrm{poly}(\epsilon)}})$ time algorithm by Cheng et al. [Che+15b]. Another important

advantage of our result is that it replaces the hardness of Planted Clique with a more believable worst-case hardness assumption (see e.g. the discussion in [BKW15]).

Theorem 15.0.2. *There exists a constant $\epsilon > 0$, such that assuming ETH, approximating* Zero-Sum Signaling *with payoffs in $[-1, 1]$ to within an additive ϵ requires time $n^{\tilde{\Omega}(\lg n)}$.*

NP-hard variants

This chapter is based on [Rub15] where we also proved two NP-hardness of approximation result: first, we show that obtaining a multiplicative approximation is NP-hard. Then, we introduce a new variant of the signaling problem where the signaler may lie, i.e. he commits to one signaling scheme, but uses a different scheme. In the lying variant even an additive $1 - 2^{-\Omega(n)}$-approximation is NP-hard.

Concurrent work of Bhaskar et al.

In independent concurrent work by Bhaskar et al. [Bha+16], quasi-polynomial time hardness for additive approximation of Zero-Sum Signaling was obtained assuming the hardness of the Planted Clique problem (among other interesting results[1] about network routing games and security games). Although we are not aware of a formal reduction, hardness of Planted Clique is a qualitatively stronger assumption than ETH in the sense that it requires average case instances to be hard. Hence in this respect, our result is stronger.

15.0.1 Techniques

Our starting point for this reduction is "birthday repetition" We reduce from a 2-ary constraint satisfaction problem (2-CSP) over n variables to a distribution over N zero-sum $N \times N$ games, with $N = 2^{\Theta(\sqrt{n})}$. Alice and Bob's strategies correspond to assignments to tuples of $\sqrt{n}$ variables. By the birthday paradox, the two $\sqrt{n}$-tuples chosen by Alice and Bob share a constraint with constant probability. If a constant fraction of the constraints are unsatisfiable, Alice's payoff will suffer with constant probability. Assuming ETH, approximating the value of the CSP requires time $2^{\tilde{\Omega}(n)} = N^{\tilde{\Omega}(\lg N)}$.

[1] For zero-sum games, [Bha+16] also rule out an additive FPTAS assuming P ≠ NP. This result follows immediately the NP-hardness for multiplicative approximation in [Rub15].

The challenge The main difficulty is that once the signal is public, the zero-sum game is tractable. Thus we would like to force the signaling scheme to output a satisfying assignment. Furthermore, if the scheme would output partial assignments on different states of nature (aka different zero-sum games in the support), it is not clear how to check consistency between different signals. Thus we would like each signal to contain an entire satisfying assignment. The optimal scheme may be very complicated and even require randomization, yet by an application of the Caratheodory Theorem the number of signals is, wlog, bounded by the number of states of nature [Dug14]. If the state of nature can be described using only $\lg N = \tilde{\Theta}\left(\sqrt{n}\right)$ bits[2], how can we force the scheme to output an entire assignment?

To overcome this obstacle, we let the state of nature contain a partial assignment to a random $\sqrt{n}$-tuple of variables. We then check the consistency of Alice's assignment with nature's assignment, Bob's assignment with nature's assignment, and Alice and Bob's assignments with each other; let $\tau^{A,Z}, \tau^{B,Z}, \tau^{A,B}$ denote the outcomes of those consistency checks, respectively. Alice's payoff is given by:

$$U = \delta\tau^{A,Z} - \delta^2\tau^{B,Z} + \delta^3\tau^{A,B}$$

for some small constant $\delta \in (0,1)$. Now, both Alice and Bob want to maximize their chances of being consistent with nature's partial assignment, and the signaling scheme gains by maximizing $\tau^{A,B}$.

Of course, if nature outputs a random assignment, we have no reason to expect that it can be completed to a full satisfying assignment. Instead, the state of nature consists of N assignments, and the signaling scheme helps Alice and Bob play with the assignment that can be completed.

15.1 Near-optimal signaling is hard

Theorem 15.1.1. *There exists a constant $\epsilon > 0$, such that assuming ETH, approximating* Zero-Sum Signaling *with payoffs in $[-1,1]$ to within an additive ϵ requires time $n^{\tilde{\Omega}(\lg n)}$.*

Construction overview

Our reduction begins with a 2CSP ψ over n variables from alphabet Σ. We partition the variables into n/k disjoint subsets $\left\{S_1, \ldots, S_{n/k}\right\}$, each of size at most $2k$ for $k = \sqrt{n}$ such that every two subsets share at most a constant number of constraints.

[2]In other words, N, the final size of the reduction, is an upper bound on the number of states of nature.

Nature chooses a random subset S_i from the partition, a random assignment $\vec{u} \in \Sigma^{2k}$ to the variables in S_i, and an auxiliary vector $\hat{b} \in \{0,1\}^{\Sigma \times [2k]}$. As mentioned in Section 15.0.1, $\vec{u}$ may not correspond to any satisfying assignment. Alice and Bob participate in one of $|\Sigma|^{2k}$ subgames; for each $\vec{v} \in \Sigma^{2k}$, there is a corresponding subgame where all the assignments are XOR-ed with $\vec{v}$. The goal of the auxiliary vector $\hat{b}$ is to force Alice and Bob to participate in the right subgame, i.e. the one where the XOR of $\vec{v}$ and $\vec{u}$ can be completed to a full satisfying assignment. In particular, the optimum signaling scheme reveals partial information about $\hat{b}$ in a way that guides Alice and Bob to participate in the right subgame. The scheme also outputs the full satisfying assignment, but reveals no information about the subset S_i chosen by nature.

Each player has $\left(|\Sigma|^{2k} \times 2\right) \times \left(n/k \times \binom{n/k}{n/2k} \times |\Sigma|^{2k}\right) = 2^{\Theta(\sqrt{n})}$ strategies. The first $|\Sigma|^{2k}$ strategies correspond to a Σ-ary vector $\vec{v}$ that the scheme will choose after observing the random input. The signaling scheme forces both players to play (w.h.p.) the strategy corresponding to $\vec{v}$ by controlling the information that corresponds to the next 2 strategies. Namely, for each $\vec{v}' \in \Sigma^{2k}$, there is a random bit $b(\vec{v}')$ such that each player receives a payoff of 1 if they play $(\vec{v}', b(\vec{v}'))$ and 0 for $(\vec{v}', 1 - b(\vec{v}'))$. The b's are part of the state of nature, and the optimal signaling scheme will reveal only the bit corresponding to the special $\vec{v}$. Since there are $|\Sigma|^{2k}$ bits, nature cannot choose them independently, as that would require $2^{|\Sigma|^{2k}}$ states of nature. Instead we construct a pairwise independent distribution.

The next n/k strategies correspond to the choice of a subset S_i from the specified partition of variables. The $\binom{n/k}{n/2k}$ strategies that follow correspond to a gadget due to Althofer [Alt94] whereby each player forces the other player to randomize (approximately) uniformly over the choice of subset.

The last $|\Sigma|^{2k}$ strategies correspond to an assignment to S_i. The assignment to each S_i is XOR-ed entry-wise with $\vec{v}$. Then, the players are paid according to checks of consistency between their assignments, and a random assignment to a random S_i picked by nature. (The scheme chooses $\vec{v}$ so that nature's random assignment is part of a globally satisfying assignment.) Each player wants to pick an assignment that passes the consistency check with nature's assignment. Alice also receives a small bonus if her assignment agrees with Bob's; thus her payoff is maximized when there exists a globally satisfying assignment.

See formal construction below, as well as summary in Table 15.1.

Table 15.1: Variables in proof of Theorem 15.1.1

Variable	Role in reduction	Chosen by...
Nature		
$i \in [n/k]$	Specifies subset S_i	Uniformly at random
$\vec{u} \in \Sigma^{2k}$	Specifies (shifted) assignment for S_i	
$\hat{b} \in \{0,1\}^{\Sigma \times [2k]}$	Force Alice and Bob into right subgame	
Alice[3] (expected behvaior)		
$\vec{v}^A, c^A$	Force Alice into right subgame	$c^A = b(\vec{v}^A)$
$j^A \in [n/k]$	Specifies subset S_{j^A}	Uniformly at random
$T^A \subset [n/k]$	Force Bob to pick j^B at random	Random subset of size $n/2k$
$\vec{w}^A \in \Sigma^{2k}$	Assignment for S_{j^A}	$\vec{\alpha}$ restricted to S_{j^A}
Signaler (completeness)		
$\vec{\alpha} \in \Sigma^n$	Assignment to entire 2CSP	Satisfying assignment
$\vec{v} \in \Sigma^{2k}$	Correct shift to Nature's assignment	$\vec{v} \oplus_\Sigma \vec{u}$ is $\vec{\alpha}$ restricted to S_i
$b(\vec{v}) \in \{0,1\}$	Force Alice and Bob into $\vec{v}$-subgame	Extracted from $\hat{b}$

Formal construction

Let ψ be a 2CSP-d over n variables from alphabet Σ, as guaranteed by Theorem 2.4.1. In particular, ETH implies that distinguishing between a completely satisfiable instance and $(1-\eta)$-satisfiable requires time $2^{\tilde{\Omega}(n)}$. By Lemma 2.7.6, we can (deterministically and efficiently) partition the variables into n/k subsets $\left\{S_1, \ldots, S_{n/k}\right\}$ of size at most $2k = 2\sqrt{n}$, such that every two subsets share at most $8d^2k^2/n = O(1)$ constraints.

States of nature Nature chooses a state $\left(\hat{b}, i, \vec{u}\right) \in \{0,1\}^{\Sigma \times [2k]} \times [n/k] \times \Sigma^{2k}$ uniformly at random. For each $\vec{v}$, $b(\vec{v})$ is the XOR of bits from $\hat{b}$ that correspond to entries of $\vec{v}$:

$$\forall \vec{v} \in \Sigma^{2k} \;\; b(\vec{v}) \triangleq \left(\bigoplus_{(\sigma,\ell):[\vec{v}]_\ell = \sigma} \left[\hat{b}\right]_{(\sigma,\ell)} \right).$$

Notice that the $b(\vec{v})$'s are pairwise independent and each marginal distribution is uniform over $\{0,1\}$.

Strategies Alice and Bob each choose a strategy $(\vec{v}, c, j, T, \vec{w}) \in \Sigma^{2k} \times \{0,1\} \times [n/k] \times \binom{[n/k]}{n/2k} \times \Sigma^{2k}$. We use $\vec{v}^A, c^A$, etc. to denote the strategy Alice plays, and similarly $\vec{v}^B, c^B$, etc. for Bob. For $\sigma, \sigma' \in \Sigma$, we denote $\sigma \oplus_\Sigma \sigma' \triangleq \sigma + \sigma' \pmod{|\Sigma|}$, and for vectors $\vec{v}, \vec{v}' \in \Sigma^{2k}$, we let $\vec{v} \oplus_\Sigma \vec{v}' \in \Sigma^{2k}$ denote the entry-wise $\oplus_\Sigma$.

Payoffs Consider state of nature $\left(\hat{b}, i, \vec{u}\right)$ and players' strategies $(\vec{v}^A, c^A, j^A, T^A, \vec{w}^A)$ and $(\vec{v}^B, c^B, j^B, T^B, \vec{w}^B)$.

When $\vec{v}^A = \vec{v}^B = \vec{v}$, we set $\tau^{A,Z} = 1$ if assignments $\vec{w}^A$ and $(\vec{v} \oplus_\Sigma \vec{u})$ to subsets S_{j^A} and S_i, respectively, satisfy all the constraints in ψ that are determined by $\left(S_i \cup S_{j^A}\right)$, and $\tau^{A,Z} = 0$ otherwise. Similarly, $\tau^{B,Z} = 1$ iff $\vec{w}^B$ and $(\vec{v} \oplus_\Sigma \vec{u})$ satisfy the corresponding constraints in ψ; and $\tau^{A,B}$ checks $\vec{w}^A$ and $\vec{w}^B$. When $\vec{v}^A \neq \vec{v}^B$, we set $\tau^{A,Z} = \tau^{B,Z} = \tau^{A,B} = 0$.

We decompose Alice's payoff as:

$$U^A \triangleq U_b^A + U_{\text{Althofer}}^A + U_\psi^A,$$

where

$$U_b^A \triangleq \mathbf{1}\left\{c^A = b\left(\vec{v}^A\right)\right\} - \mathbf{1}\left\{c^B = b\left(\vec{v}^B\right)\right\},$$
$$U_{\text{Althofer}}^A \triangleq \mathbf{1}\left\{j^B \in T^A\right\} - \mathbf{1}\left\{j^A \in T^B\right\},$$

and

$$U_\psi^A \triangleq \delta\tau^{A,Z} - \delta^2\tau^{B,Z} + \delta^3\tau^{A,B}, \tag{15.1}$$

for a sufficiently small constant $0 < \delta \ll \sqrt{\eta}$.

Completeness

Lemma 15.1.2. *If ψ is satisfiable, there exists a signaling scheme and a mixed strategy for Alice that guarantees expected payoff $\delta - \delta^2 + \delta^3$.*

Proof. Fix a satisfying assignment $\vec{\alpha} \in \Sigma^n$. Given state of nature $\left(\hat{b}, i, \vec{u}\right)$, let $\vec{v}$ be such that $(\vec{v} \oplus_\Sigma \vec{u}) = [\vec{\alpha}]_{S_i}$. The scheme outputs the signal $(\vec{v}, b(\vec{v}), \vec{\alpha})$. Alice's mixed strategy sets $(\vec{v}^A, c^A) = (\vec{v}, b(\vec{v}))$, picks j^A and T^A uniformly at random, and sets $\vec{w}^A = [\vec{\alpha}]_{S_{j^A}}$.

Because Bob has no information about $b(\vec{v}')$ for any $\vec{v}' \neq \vec{v}$, he has probability $1/2$ of losing whenever he picks $\vec{v}^B \neq \vec{v}$, i.e. $\mathsf{E}\left[U_b^A\right] \geq \frac{1}{2}\Pr\left[\vec{v}^B \neq \vec{v}\right]$. Furthermore, because Alice chooses T^A and j^A uniformly, $\mathsf{E}\left[U_{\text{Althofer}}^A\right] = 0$.

Since $\vec{\alpha}$ completely satisfies ψ, we have that $\tau^{A,Z} = 1$ as long as $\vec{v}^B = \vec{v}$ (regardless of the rest of Bob's strategy). Bob's goal is thus to maximize $\mathsf{E}\left[\delta^2\tau^{B,Z} - \delta^3\tau^{A,B}\right]$. Notice that $\vec{w}^A$ and $(\vec{v} \oplus_\Sigma \vec{u})$ are two satisfying partial assignments to uniformly random subsets from the partition. In particular, they are both drawn from the same distribution, so we have that for any mixed strategy that Bob plays, $\mathsf{E}\left[\tau^{B,Z}\right] = \mathsf{E}\left[\tau^{A,B}\right]$. Therefore Alice's payoff is at least

$$\left(\delta - \delta^2 + \delta^3\right)\Pr\left[\vec{v}^B = \vec{v}\right] + \frac{1}{2}\Pr\left[\vec{v}^B \neq \vec{v}\right] \geq \delta - \delta^2 + \delta^3.$$

□

Soundness

Lemma 15.1.3. *If at most a $(1-\eta)$-fraction of the constraints are satisfiable, Alice's maxmin payoff is at most $\delta - \delta^2 + (1 - \Omega_\eta(1))\,\delta^3$, for any signaling scheme.*

Proof. Fix any mixed strategy by Alice; we show that Bob can guarantee a payoff of at least $-(\delta - \delta^2 + (1 - \Omega_\eta(1))\,\delta^3)$. On any signal, Bob chooses $(\vec{v}^B, c^B)$ from the same distribution that Alice uses for $(\vec{v}^A, c^A)$. He chooses j^B uniformly, and picks T^B so as to minimize $\mathsf{E}\left[U^A_{\text{Althofer}}\right]$. Finally, for each j^B, he draws $\vec{w}^B$ from the same marginal distribution that Alice uses for $\vec{w}^A$ conditioning on $j^A = j^B$ (and uniformly at random if Alice never plays $j^A = j^B$). By symmetry, $\mathsf{E}\left[U^A_b\right] = 0$ and $\mathsf{E}\left[U^A_{\text{Althofer}}\right] \leq 0$.

In this paragraph, we use Althoefer's gadget to argue that, wlog, Alice's distribution over the choice of j^A is approximately uniform. In Althofer's gadget, Alice can guarantee an (optimal) expected payoff of 0 by randomizing uniformly over her choice of j^A and T^A. By Lemma 2.7.8, if Alice's marginal distribution over the choice of j^A is $8\delta^2$-far from uniform (in total variation distance), then Bob can guess that j^A is in some subset $T^B \in \binom{[n/k]}{n/2k}$ with advantage (over guessing at random) of at least $2\delta^2$. Therefore $\mathsf{E}\left[U^A_{\text{Althofer}}\right] \leq -2\delta^2$; but this would imply $\mathsf{E}\left[U^A\right] \leq -2\delta^2 + \mathsf{E}\left[U^A_\psi\right] \leq \delta - 2\delta^2 + \delta^3$. So henceforth we assume wlog that Alice's marginal distribution over the choice of j^A is $O(\delta^2)$-close to uniform (in total variation distance).

Since Alice's marginal distribution over j^A is $O(\delta^2)$-close to uniform, we have that Bob's distribution over $(j^B, \vec{w}^B)$ is $O(\delta^2)$-close to Alice's distribution over $(j^A, \vec{w}^A)$. Therefore $\mathsf{E}\left[\tau^{B,Z}\right] \geq \mathsf{E}\left[\tau^{A,Z}\right] - O(\delta^2)$, and so we also get:

$$\mathsf{E}\left[U^A\right] \leq \mathsf{E}\left[U^A_\psi\right] \leq \delta - \delta^2 + \delta^3\mathsf{E}\left[\tau^{A,B}\right] + O\left(\delta^4\right). \tag{15.2}$$

Bounding $\mathsf{E}[\tau^{A,B}]$ To complete the proof, it remains to show an upper bound on $\mathsf{E}[\tau^{A,B}]$. In particular, notice that it suffices to bound the probability that Alice's and Bob's induced assignments agree. Intuitively, if they gave assignments to uniformly random (and independent) subsets of variables, the probability that their assignments agree cannot be much higher than the value of the 2CSP; below we formalize this intuition.

By the premise, any assignment to all variables violates at least an η-fraction of the constraints. In particular, this is true in expectation for assignments drawn according to Alice's and Bob's mixed strategy. This is a bit subtle: in general, it is possible that Alice's assignment alone doesn't satisfy many constraints and neither does Bob's, but when we check constraints between Alice's and Bob's assignments everything satisfied (for example, think of the 3-Coloring CSP, where Alice colors all her vertices blue, and Bob colors all his vertices red). Fortunately, this subtlety is irrelevant for our construction since we explicitly defined Bob's mixed strategy so that conditioned on each set S_j of variables, Alice and Bob have the same distribution over assignments.

The expected number of violations between pairs directly depends on the value of the 2CSP. To bound the *probability* of observing at least one violations, recall that every pair of subsets shares at most a constant number of constraints, so this probability is within a constant factor of the expected number of violations. In particular, an $\Omega(\eta)$-fraction of the pairs of assignments chosen by Alice and Bob violate ψ.

Finally, Alice doesn't choose j^A uniformly at random; but her distribution is $O(\delta^2)$-close to uniform. Therefore, we have $\mathsf{E}[\tau^{A,B}] \leq 1 - \Omega(\eta) + O(\delta^2)$. Plugging into (15.2) completes the proof. □

Part V

Approximate Nash Equilibrium

Chapter 16

2-Player approximate Nash Equilibrium

For the past decade, the central open problem in equilibrium computation has been whether two-player Nash equilibrium admits a PTAS. We had good reasons to be hopeful: there was a series of improved approximation ratios [KPS09; DMP09; DMP07; BBM10; TS08] and several approximation schemes for special cases [KT07; DP09; Alo+13; Bar15]. Yet most interesting are two inefficient algorithms for two-player Nash:

- the classic Lemke-Howson algorithm [LH64] finds an exact Nash equilibrium in exponential time; and
- a simple algorithm by Lipton, Markakis, and Mehta [LMM03] finds an ϵ-Approximate Nash Equilibrium in time $n^{O(\log n)}$.

Although the Lemke-Howson algorithm takes exponential time, it has a special structure which places the problem inside the complexity class PPAD [Pap94]. Proving hardness for problems in PPAD is notoriously challenging because they are *total*, i.e. they always have a solution, so the standard techniques from NP-hardness do not apply. By now, however, we know that exponential and polynomial approximations for two-player Nash are PPAD-complete [DGP09; CDT09]. However, ϵ-approximation for two-player Nash is unlikely to have the same fate: otherwise, the quasi-polynomial algorithm of [LMM03] would refute the Exponential Time Hypothesis for PPAD. Thus the strongest hardness result we can hope to prove (given our current understanding of complexity[1]) is a quasi-polynomial hardness that sits inside

[1]Given our current understanding of complexity, refuting ETH for PPAD seems unlikely: there are matching black-box lower bounds [HPV89; Bea+98]. Recall that the NP-analogue ETH [IPZ01]

PPAD:

Theorem (Theorem 1.3.1 restated)**.** *There exists a constant $\epsilon > 0$ such that, assuming ETH for* PPAD*, finding an ϵ-Approximate Nash Equilibrium in a two-player $n \times n$ game requires time $T(n) = n^{\log^{1-o(1)} n}$.*

16.0.1 Additional related work

Previous attempts to show lower bounds for approximate Nash in two player games have mostly focused on limited models of computation [DP09] and lower bounding the support required for obtaining approximate equilibria [Alt94; FNS07; Anb+13; Anb+15] (in contrast, [LMM03]'s algorithm runs in quasi-polynomial time because there exist approximate equilibria with support size at most $O\left(\frac{\log n}{\epsilon^2}\right)$).

Best Nash equilibrium Hazan and Krauthgamer [HK11] showed that finding an ϵ-Approximate Nash Equilibrium with ϵ-optimal welfare is as hard as the Planted-Clique problem; Austrin et al. [ABC13] later showed that the optimal-welfare constraint can be replaced by other decision problems. Braverman et al. [BKW15] recently showed that the hardness Planted-Clique can be replaced by the Exponential Time Hypothesis, the NP-analog of the ETH for PPAD we use here. (See also Theorem 11.1.1.)

Multiplicative hardness of approximation Daskalakis [Das13] and our recent work [**Rub15˙simpler-games**] show that finding an ϵ-relative Well-Supported Nash Equilibrium in two-player games is PPAD-hard. The case of ϵ-relative Approximate Nash Equilibrium is still open: our main theorem implies that it requires at least quasi-polynomial time, but it is not known whether it is PPAD-hard, or even if it requires a large support (see also discussion in [BPR16]).

Approximation algorithms The state of the art for games with arbitrary payoffs is ≈ 0.339 for two-player games due to Tsaknakis and Spirakis [TS08] and $0.5 + \epsilon$ for polymatrix games due to Deligkas et al. [Del+17]. For two-player games, PTAS have been given for the special cases of constant rank games by Kannan and Theobald [KT07], small-probability games by Daskalakis and Papadimitriou [DP09], positive semi-definite games by Alon et al. [Alo+13], and sparse games by Barman [Bar15].

is widely used (e.g. [KS10; LMS11; AIM14; BKW15; CPP16],as well as the previous part of this thesis), often in stronger variants such as SETH [IP01; CIP09] and NSETH [Car+16].

16.1 Technical overview

Given an End-of-a-Line instance of size n, we construct a two-player $N \times N$ game for $N = 2^{n^{1/2+o(1)}}$ whose approximate equilibria correspond to solutions to the End-of-a-Line instance. Thus, assuming the "ETH for PPAD", finding an approximate equilibrium requires time $2^n = N^{\log^{1-o(1)} N}$.

The main steps of the final construction are: (i) reducing End-of-a-Line to a new discrete problem which we call Local End-of-a-Line; (ii) reducing Local End-of-a-Line to a problem of finding an approximate Brouwer fixed point (this step uses our reduction from Section 4.2); (iii) reducing from Brouwer fixed point to finding an approximate Nash equilibrium in a multiplayer game over $n^{1/2+o(1)}$ players with $2^{n^{1/2+o(1)}}$ actions each; and (iv) reducing to the two-player game.

The main novelty in the reduction is the use of techniques such as error correcting codes and probabilistically checkable proofs (PCPs) inside PPAD. In particular, the way we use PCPs in our proof is very unusual.

Constructing the first gap: showing hardness of Euclidean Brouwer

The first step in all known PPAD-hardness results for (approximate) Nash equilibrium is reducing End-of-a-Line to the problem of finding an (approximate) Brouwer fixed point of a continuous, Lipschitz function $f:[0,1]^n \to [0,1]^n$. Even Theorem 4.1.1, which shows hardness of approximation in ℓ_∞-norm does not suffice for our purposes. In particular, it only implies that it is hard to find an $\mathbf{x}$ such that $f(\mathbf{x})$ is approximately equal to $\mathbf{x}$ *on every coordinate*. The first step in our proof is to strengthen this result to obtain hardness of approximation with respect to 2-norm (Theorem 4.2.1). Now, even finding an $\mathbf{x}$ such that $f(\mathbf{x})$ is approximately equal to $\mathbf{x}$ on *most of the coordinates* is already PPAD-hard.

Theorem 4.1.1 was obtained by adapting a construction due to Hirsch, Papadimitriou, and Vavasis [HPV89] via the use of *error correcting code*. The first obstacle to using PCP-like techniques for problems in PPAD is totality: problems in PPAD always have a solution. For NP-hard problems, the PCP verifier expects the proof to be encoded in some error correcting code. If the proof is far from any codeword, the verifier detects that (with high probability), and immediately rejects. For problems in PPAD (more generally, in TFNP) this is always tricky because it is not clear what does it mean "to reject". Hirsch et al.'s original construction has the following useful property: for the vast majority of $\mathbf{x}$'s (in particular, all $\mathbf{x}$'s far from the embedding of the paths) the displacement $f(\mathbf{x}) - \mathbf{x}$ is the same default displacement. Thus, when an $\mathbf{x}$ is too far from any codeword to faithfully decode it, we can simply apply the default displacement.

The main challenge: locality.

Our ultimate goal is to construct a two-player game that simulates the Brouwer function from Theorem 4.2.1. This is done via an *imitation gadget*: Alice's mixed strategy induces a point $\mathbf{x}^{(\mathcal{A})} \in [0,1]^n$; Bob's strategy induces $\mathbf{x}^{(\mathcal{B})} \in [0,1]^n$; Alice wants to minimize $\|\mathbf{x}^{(\mathcal{A})} - \mathbf{x}^{(\mathcal{B})}\|_2$, whereas Bob wants to minimize $\|f(\mathbf{x}^{(\mathcal{A})}) - \mathbf{x}^{(\mathcal{B})}\|_2$. Alice and Bob are both satisfied at a fixed point, where $\mathbf{x}^{(\mathcal{A})} = \mathbf{x}^{(\mathcal{B})} = f(\mathbf{x}^{(\mathcal{A})})$. (Recall also our construction of communicationally-hard games for Theorem 3.0.1.)

The main obstacle is that we want to incentivize Bob to minimize $\|f(\mathbf{x}^{(\mathcal{A})}) - \mathbf{x}^{(\mathcal{B})}\|_2$ via *local constraints* (payoffs - each depends on one pure strategy), while $f(\mathbf{x}^{(\mathcal{A})})$ has a *global dependency* on Alice's entire mixed strategy.

Our goal is thus to construct a hard Brouwer function that can be *locally computed*. How local does the computation need to be? In a game of size $2^{\sqrt{n}} \times 2^{\sqrt{n}}$, each strategy can faithfully store information about $\sqrt{n}$ bits. Specifically, our construction will be $n^{1/2+o(1)}$-local.

We haven't yet defined exactly what it means for our construction to be "$n^{1/2+o(1)}$-local"; the exact formulation is quite cumbersome as the query access needs to be partly adaptive, robust to noise, etc. Eventually (Section 16.4), we formalize the "locality" of our Brouwer function via a statement about multiplayer games. On a high level, however, our goal is to show that for any $j \in \{1,\ldots,n\}$, the j-th output $f_j(\mathbf{x})$ can be approximately computed, with high probability, by accessing $\mathbf{x}$ at only $n^{1/2+o(1)}$ coordinates.

This is a good place to note that achieving any sense of "local computation" in our setting is surprising, even if we consider just the error correcting encoding for our Brouwer function: in order to maintain constant relative distance, an average bit of the output must depend on a constant fraction of the input bits!

Local End-of-a-Line

In order to introduce locality, we go back to the End-of-a-Line problem. "Wishful thinking": imagine that we could replace the arbitrary predecessor and successor circuits in End-of-a-Line with NC^0 (constant depth and constant fan-in) circuits $S^{\text{LOCAL}}, P^{\text{LOCAL}} : \{0,1\}^n \to \{0,1\}^n$, so that each output bit only depends on a constant number of input bits. Imagine further that we had the guarantee that for each input, the outputs of $S^{\text{LOCAL}}, P^{\text{LOCAL}}$ differ from the input on just a constant number of bits. Additionally, it would be really nice if we had a succinct pointer that immediately told us which bits are about to be replaced. (We later call this succinct pointer the *counter*, because it also cycles through its possible values in a fixed order.)

Suppose all our wishes came true, and furthermore the hard Brouwer function from Theorem 4.2.1 used a *linear* error correcting code. Then, we could use the encoding of the counter, henceforth $C(u)$, to read only the bits that are about to be replaced, and the inputs that determine the new values of those bits. Thus, using only local access to a tiny fraction of the bits ($|C(u)| + O(1)$), we can construct a difference vector $u - S^{\text{LOCAL}}(u)$ (which is 0 almost everywhere). As we discussed above, the encodings $E(u), E(S^{\text{LOCAL}}(u))$ must differ on a constant fraction of the bits - but because the code is linear, we can also locally construct the difference vector $E(u) - E(S^{\text{LOCAL}}(u)) = E(u - (S^{\text{LOCAL}}(u)))$. Given $E(u) - E(S^{\text{LOCAL}}(u))$, we can locally compute any bit of $E(S^{\text{LOCAL}}(u))$ by accessing only the corresponding bit of $E(u)$.

Back to reality: unfortunately we do not know of a reduction to such a restricted variant of End-of-a-Line. Surprisingly, we can almost do that. The problem Local End-of-a-Line (formally defined in Section 16.2) satisfies all the guarantees defined above, is linear-time reducible from End-of-a-Line, but has one caveat: it is only defined on a strict subset V^{LOCAL} of the discrete hypercube ($V^{\text{LOCAL}} \subsetneq \{0,1\}^n$). Verifying that a vertex belongs to V is quite easy - it can be done in AC^0. Let us take a brief break to acknowledge this new insight about the canonical problem of PPAD:

Theorem 16.1.1. *The predecessor and successor circuits of* End-of-a-Line *are, wlog,* AC^0 *circuits.*

The class AC^0 is quite restricted, but the outputs of its circuits are not local functions of the inputs. Now, we want to represent u in a way that will make it possible to locally determine whether $u \in V^{\text{LOCAL}}$ or not. To this end we augment the linear error correcting encoding $E(u)$ with a *probabilistically checkable proof* (PCP) $\pi(u)$ of the statement $(u \in V^{\text{LOCAL}})$.

Our holographic proof system

Some authors distinguish between PCPs and holographic proofs[2]: a PCP verifier has unrestricted access to the instance, and queries the proof locally; whereas the holographic proof verifier has restricted, local access to both the proof and (an error correcting encoding of) the instance. In this sense, what we actually want is a holographic proof.

[2]In a nutshell, PCPs or holographic proofs are proofs that can be verified "locally" (with high probability) by reading only a small (random) portion of the proof; see e.g. [AB09, Chapter 18] for many more details.

We construct a holographic proof system with some very unusual properties. We are able to achieve these properties thanks to our modest locality desideratum: $n^{1/2+o(1)}$, as opposed to the typical $\mathsf{polylog}(n)$ or $O(1)$. We highlight here a few of these properties; see Section 16.3 for details.

- **(Local proof construction)** The most surprising property of our holographic proof system is that the proof $\pi(u)$ can be constructed from local access to the encoding $E(u)$. In particular, note that we can locally compute $E(S^{\text{LOCAL}}(u))$ because $E(\cdot)$ is linear - but $\pi(\cdot)$ is not. Once we obtain $E(S^{\text{LOCAL}}(u))$, we can use local proof construction to compute $\pi(S^{\text{LOCAL}}(u))$ locally.

- **(Very low random-bit complexity)** Our verifier is only allowed to use $(1/2 + o(1)) \log_2 n$ random bits - this is much lower even than the $\log_2 n$ bits necessary to choose one entry at random. In related works, similar random-bit complexity was achieved by bundling the entries together via "birthday repetition". To some extent, something similar happens here, but our locality is already $n^{1/2+o(1)}$ so no bundling (or repetition) is necessary. To achieve nearly optimal random-bit complexity, we use λ-biased sets over large finite fields together with the Sampling Lemma of Ben-Sasson et al. [Ben+03].

- **(Tolerant verifier)** Typically, a verifier must reject (with high probability) whenever the input is far from valid, but it is allowed to reject even if the input is off by only one bit. Our verifier, however, is *required* to accept (with high probability) inputs that are close to valid proofs. (This is related to the notion of "tolerant testing", which was defined in [PRR06] and discussed in [GR05] for locally testable codes.)

- **(Local decoding)** We make explicit use of the property that our holographic proof system is also a locally decodable code. While the relations between PCPs and locally testable codes have been heavily explored (see e.g. Goldreich's survey [Gol10]), the connection to locally decodable codes is not as immediate. Nevertheless, related ideas of Locally Decode/Reject Codes [MR10] and decodable PCP [DH13] have been used before in order to facilitate composition of tests (our holographic proof system, in contrast, is essentially composition-free). Fortunately, as noted by [DH13] many constructions of PCPs are already implicitly locally decodable.

- **(Robust everything)** Ben-Sasson et al. [Ben+06] introduce a notion of *robust soundness*, where on an invalid proof, the string read by the verifier must be far from any acceptable string. (Originally the requirement is far in expectation, but we want far with high probability.) Another way of looking at the

same requirement, is that even if a malicious prover *adaptively* changes a small fraction of the bits queried by the verifier, the test is still sound. In this sense, we require that all our guarantees, not just soundness, continue to hold (with high probability) even if a malicious entity adaptively changes a small fraction of the bits queried by the verifier.

How is local proof construction possible? At a high level, our holographic proof system expects an encoding of u as a low-degree t-variate polynomial, and a few more low-degree t-variate polynomials, that encode the proof of $u \in V^{\text{LOCAL}}$. (This is essentially the standard "arithmetization", dating back at least to [BF91; Sha92], although our construction is most directly inspired by [PS94; Spi95].) In our actual proof, t is a small super-constant, e.g. $t \triangleq \sqrt{\log n}$; but for our exposition here, let us consider $t = 2$, i.e. we have bivariate polynomials.

The most interesting part of the proof verification is testing that a certain low-degree polynomial $\Psi: \mathcal{G}^2 \to \mathcal{G}$, for some finite field $\mathcal{G}$ of size $|\mathcal{G}| = \Theta\left(n^{1/2+o(1)}\right)$, is identically zero over all of $\mathcal{F}^2$, for some subset $\mathcal{F} \subsetneq \mathcal{G}$ of cardinality $|\mathcal{F}| = |\mathcal{G}| / \mathsf{polylog}(n)$. This can be done by expecting the prover to provide the following low-degree polynomials:

$$\begin{aligned} \Psi'(x,y) &\triangleq \sum_{f_i \in \mathcal{F}} \Psi(x, f_i)\, y^i \\ \Psi''(x,y) &\triangleq \sum_{f_j \in \mathcal{F}} \Psi'(f_j, y)\, x^j. \end{aligned}$$

Then, $\Psi''(x,y) = \sum_{f_i, f_j \in \mathcal{F}} \Psi(f_j, f_i)\, x^j y^i$ is the zero polynomial if and only if Ψ is indeed identically zero over all of $\mathcal{F}^2$. $\Psi(x,y)$ can be computed by accessing $E(u)$ on just a constant number of entries. Thus, computing $\Psi'(x,y)$ requires $\Psi(x, f_i)$ for all $f_i \in \mathcal{F}$, so a total of $\Theta\left(n^{1/2+o(1)}\right)$ queries to $E(u)$. However, computing even one entry of $\Psi''(\cdot)$ requires $\Omega(n)$ queries to $E(u)$. The crucial observation is that we don't actually need the prover to provide Ψ''. Instead, it suffices that the prover provide Ψ', and the verifier checks that $\sum_{f_j \in \mathcal{F}} \Psi'(f_j, y)\, x^j = 0$ for sufficiently many (x,y).

Putting it all together via polymatrix games

The above arguments suffice to construct a hard Brouwer function (in the sense of Theorem 4.2.1) that can be computed "$n^{1/2+o(1)}$-locally". We formalize this statement in terms of approximate Nash equilibria in a polymatrix game.

Definition 16.1.2 (Polymatrix games)**.** In a polymatrix game, each pair of players simultaneously plays a separate two-player subgame. Every player has to play the

same strategy in every two-player subgame, and her utility is the sum of her subgame utilities. The game is given in the form of the payoff matrix for each two-player subgame.

We construct a bipartite polymatrix game between $n^{1/2+o(1)}$ players with $2^{n^{1/2+o(1)}}$ actions each. By "bipartite", we mean that each player on Alice's side only interacts with players on Bob's side and vice versa. The important term here is "polymatrix": it means that when we compute the payoffs in each subgame, they can only depend on the $n^{1/2+o(1)}$ coordinates described by the two players' strategies. It is in this sense that we guarantee "local computation".

The mixed strategy profile $\mathcal{A}$ of all the players on Alice's side of the bipartite game induces a vector $\mathbf{x}^{(\mathcal{A})} \in [0,1]^m$, for some $m = n^{1+o(1)}$. The mixed strategy profile $\mathcal{B}$ of all the players on Bob's side induces a vector $\mathbf{x}^{(\mathcal{B})} \in [0,1]^m$. Our main technical result is:

Proposition 16.1.3 (Informal)**.** *If all but an ϵ-fraction of the players play ϵ-optimally, then $\|\mathbf{x}^{(\mathcal{A})} - \mathbf{x}^{(\mathcal{B})}\|_2^2 = O(\epsilon)$ and $\|f(\mathbf{x}^{(\mathcal{A})}) - \mathbf{x}^{(\mathcal{B})}\|_2^2 = O(\epsilon)$.*

Each player on Alice's side corresponds to one of the PCP verifier's random string. Her strategy corresponds to an assignment to the bits queried by the verifier given this random string. On Bob's side, we consider a partition of $\{1, \ldots, m\}$ into $n^{1/2+o(1)}$ tuples of $n^{1/2+o(1)}$ indices each. Each player on Bob's side assigns values to one such tuple.

On each two-player subgame, the player on Alice's side is incentivized to imitate the assignment of the player on Bob's side on the few coordinates where they intersect. The player on Bob's side, uses Alice's strategy to locally compute $f_j(\mathbf{x}^{(\mathcal{A})})$ on a few j's in his $(n^{1/2+o(1)})$-tuple of coordinates. This computation may be inaccurate, but we can guarantee that for most coordinates it is approximately correct most of the time.

From polymatrix to bimatrix

The final reduction from the polymatrix game to two-player game follows more or less from known techniques for hardness of Nash equilibria [Alt94; DGP09; BPR16]. We let each of Alice and Bob control one side of the bipartite polymatrix game. In particular, each strategy in the two-player game corresponds to picking a player of the polymatrix game, and a strategy for that player. We add a gadget due to Althofer [Alt94] to guarantee that Alice and Bob mix approximately uniformly across all their players. See Section 16.5 for details.

16.1.1 The PCP Conjecture for PPAD

Recall our conjecture from Chapter 5:

Conjecture (PCP for PPAD; [BPR16])**.** *There exist constants $\epsilon, \delta > 0$ such that finding an (ϵ, δ)-WeakNash in a bipartite, degree three polymatrix game with two actions per player is* PPAD*-complete.*

The main original motivation was an approach to prove our main theorem given this conjecture. As pointed out by [BPR16], it turns out that resolving this conjecture would also have interesting consequences for relative approximations of two-player Nash equilibrium, as well as applications to inapproximability of market equilibrium.

More importantly, this question is interesting in its own right: how far can we extend the ideas from the PCP Theorem (for NP) to the wold of PPAD? The $\mathsf{PCP}[r(n), q(n)]$ characterization [AS98] is mainly concerned with two parameters: $r(n)$, the number of random bits, and $q(n)$, the number of bits read from the proof. A major tool in all proofs of the PCP Theorem is *verifier composition*: in the work of Polishchuk and Spielman [PS94; Spi95], for example, it is first shown that $\mathsf{NP} \subseteq \mathsf{PCP}[O(\log n), n^{1/2+o(1)}]$, and then via composition it is eventually obtained that $\mathsf{NP} = \mathsf{PCP}[O(\log n), O(1)]$. In some **informal sense**, one may think of our main technical result as something analogous[3] to $\mathsf{PPAD} \subseteq \mathsf{PCP}[(1/2 + o(1)) \log_2 n, n^{1/2+o(1)}]$. Furthermore, our techniques in Section 16.3 build on many existing ideas from the PCP literature [Bab+91; PS94; Spi95; Ben+03; Ben+06] that have been used to show similar statements for NP. It is thus natural to ask: is there a sense in which our "verifier" can be composed? can such composition eventually resolve the PCP Conjecture for PPAD?

More generally, some of the tools we use here, even as simple as error correcting codes, have been the basic building blocks in hardness of approximation for decades, yet to the best of our knowledge have not been used before for any problem in PPAD. We hope to see other applications of similar ideas in this regime.

16.1.2 Organization

In Section 16.2 we introduce the restricted variant Local End-of-a-Line and prove that it is also PPAD-complete. In Section 16.3 we construct our holographic proof system. In Section 16.4 we bring together ideas from Sections 4.2, 16.2, and 16.3 to prove our hardness for polymatrix games of subexponential size. Finally, in Section 16.5 we reduce from polymatrix to two-player games.

[3] We stress that our analogy is very loose. For example, we are not aware of any formal extension of PCP to function problems, and it is well known that $\mathsf{NP} \subseteq \mathsf{PCP}\left[(1/2 + o(1)) \log_2 n, n^{1/2+o(1)}\right]$.

Small constants

Our proof uses several arbitrary small constants that satisfy:

$$0 < \epsilon_{\text{PRECISION}} \ll \epsilon_{\text{NASH}} \ll \delta \ll h \ll \epsilon_{\text{COMPLETE}} \ll \epsilon_{\text{SOUND}} \ll \epsilon_{\text{DECODE}} \ll 1.$$

By $\ll$ we mean that we pick them such that $\epsilon_{\text{PRECISION}}$ is arbitrarily smaller than any polynomial in ϵ_{NASH}, and ϵ_{NASH} is arbitrarily smaller than any polynomial in δ, etc. Although their significance will be fully understood later in the paper, we briefly mention that $\epsilon_{\text{PRECISION}}$ is the precision with which the players can specify real values; ϵ_{NASH} is the approximation factor of Nash equilibrium in Proposition 16.3.1; δ and h are parameters of the Brouwer function construction; finally, $\epsilon_{\text{COMPLETE}}, \epsilon_{\text{SOUND}}, \epsilon_{\text{DECODE}}$, are parameters of our holographic proof system.

16.2 END-OF-A-LINE with local computation

In this section we introduce a local variant of END-OF-A-LINE with very simple successor and predecessor circuits.

Definition 16.2.1 (LOCAL END-OF-A-LINE)**.** The problem LOCAL END-OF-A-LINE is similar to END-OF-A-LINE, but the graph is defined on a subset $V^{\text{LOCAL}} \subseteq \{0,1\}^n$. The input consists of a membership circuit $M_{V^{\text{LOCAL}}}: \{0,1\}^n \to \{0,1\}$ that determines whether a string $u \in \{0,1\}^n$ corresponds to a vertex of the graph (i.e. $M_{V^{\text{LOCAL}}}(u) = \begin{cases} 1 & u \in V^{\text{LOCAL}} \\ 0 & u \notin V^{\text{LOCAL}} \end{cases}$), a special vertex $u_{\mathbf{0}} \in V^{\text{LOCAL}}$, and successor and predecessor circuits $S^{\text{LOCAL}}, P^{\text{LOCAL}} : \{0,1\}^n \to \{0,1\}^n$ with the promise that every output bit depends only on a constant number of input bits. (I.e. S^{LOCAL} and P^{LOCAL} are in NC^0.) Furthermore, we require that the outputs of S^{LOCAL} and P^{LOCAL} are identical to the respective inputs, except at a constant number of coordinates. Similarly to END-OF-A-LINE, we are guaranteed that $P^{\text{LOCAL}}(u_{\mathbf{0}}) = u_{\mathbf{0}} \neq S^{\text{LOCAL}}(u_{\mathbf{0}})$.

The goal is to find an input $u \in V^{\text{LOCAL}}$ that satisfies any of the following:

- End-of-a-line: $P^{\text{LOCAL}}(S^{\text{LOCAL}}(u)) \neq u$ or $S^{\text{LOCAL}}(P^{\text{LOCAL}}(u)) \neq u \neq u_{\mathbf{0}}$; or
- Boundary conditions: $S^{\text{LOCAL}}(u) \notin V^{\text{LOCAL}}$ or $P^{\text{LOCAL}}(u) \notin V^{\text{LOCAL}}$.

Notice that each vertex $u \in V^{\text{LOCAL}}$, only defers from $S^{\text{LOCAL}}(u)$ and $P^{\text{LOCAL}}(u)$ at a constant number of bits, and the value of each of those only depends on a constant number of bits of u. We will refer to all those bits as the *criticial bits* of u, and denote their cardinality q_{CRITICAL}.

We now reduce End-of-a-Line to Local End-of-a-Line, proving that the latter is PPAD-complete. In the following sections we use this reduction "white-box", and revisit some of its specific properties.

Theorem 16.2.2. *There is a linear-time reduction from* End-of-a-Line *to* Local End-of-a-Line.

Notice that there is a trivial reduction in the other direction: add self-loops to all strings not in V^{LOCAL}.

Proof of Theorem 16.2.2. Given circuits $S, P: \{0,1\}^n \to \{0,1\}^n$, we construct new circuits $S^{\text{LOCAL}}, P^{\text{LOCAL}} : \{0,1\}^m \to \{0,1\}^m$ that satisfy Definition 16.2.1. We assume wlog that the circuits have fan-out 2 (otherwise, replace gates with larger fan-outs with a binary tree of equality gates; this only blows up the size of the circuit by a constant factor). We set $m = 4(|S| + |P|)$, where by $|S|$ and $|P|$ we mean the number of lines (i.e. number of inputs + number of logic gates) in each circuit, respectively. We think of the m bits as representing two copies of each circuit (S_1, P_1 and S_2, P_2), with each line represented by two bits, representing three possible states: value 0, value 1, and inactive.

The path begins with circuit S_1 containing the computation from $\mathbf{0}_n$ to $S(\mathbf{0}_n)$, and circuit P_1 containing the reverse computation back to $\mathbf{0}_n$; the lines of circuits S_2, P_2 are inactive. This is the special vertex, $u_{\mathbf{0}}$.

Over the next n steps, the output bits of S_1 are copied (one-by-one) to the input bits of S_2 (and the corresponding lines are activated). Over the next $|S| - n$ steps, the values on the lines of S_2 are updated -one-by-one, starting from the inputs and propagating to the outputs in a fixed order- until all the output bits are activated and set to $S(S(\mathbf{0}_n))$. Then, over the next $|P|$ steps, the input of P_2 is set to $S(S(\mathbf{0}_n))$, the values on the gates are updated (propagating in a fixed order from output to input), and finally setting the output of P_2 to $S(\mathbf{0}_n)$.

Notice that so far, if we want to trace back a step (a.k.a. implement P^{LOCAL}), we simply need to deactivate the last activated line. We now start erasing values when going forward, but we must do so carefully so that we can reconstruct them when going backward. We proceed by deactivating the lines in S_1: line by line, in reverse order from outputs to inputs. Indeed, in order to go back we can reconstruct the output of any gate from its inputs. If we want to reconstruct an input to S_1 - this is no problem as it is saved as the output of P_1. Once we finish deactivating S_1, we deactivate P_1, also in reverse order. Notice again that the input to P_1 is saved as the input to S_2 and output of P_2. Now, we copy the values from S_2 and P_2 to S_1 and P_1 in the same order that we used to deactivate the lines, starting from the outputs of S_1, to inputs, to outputs of P_1, to inputs. Then we deactivate the lines in S_2 and

P_2; again, we use the same order in which they were activated, starting from inputs of S_2, to outputs, to inputs of P_2, to outputs. So now we have S_1 as a witness to the computation of $S(S(\mathbf{0}_n))$ from $S(\mathbf{0}_n)$ and P_1 goes in the reverse order, while all the lines of S_2 and P_2 are inactive.

We repeat the process from the last two paragraphs to find $S(S(S(\mathbf{0}_n)))$, etc.

We let V^{LOCAL} to be the set of strings that correspond to a legal partial computation as described above. $M_{V^{\text{LOCAL}}}$ verifies that a string corresponds to one of the following scenarios:

- S_1 holding the computation from some $x \in \{0,1\}^n$ to $S(x)$, P_1 holding the computation from $S(x)$ back to x, and S_2 and P_2 holding part of the computation from $S(x)$ to $S(S(x))$ and back to $S(x)$;
- S_2 and P_2 holding the computation from some x to $S(x)$ and back to x, and P_1 and S_1 are in the process of erasing the computation from x to $P(x)$ and back to x;
- S_2 and P_2 holding the computation from some x to $S(x)$ and back to x, and S_1 and P_1 are in the process of copying it; or
- S_1 and P_1 holding the computation from some x to $S(x)$ and back to x, are S_2 and P_2 in the process of erasing the same computation.

Notice that the joint active/inactive state of all the lines always belongs to one of $O(m)$ different states (that can be described by $\log_2 m + O(1)$ bits. Verifying that a string corresponds to one of the above scenarios is equivalent to checking that the joint active/inactive state is valid, and that the values on all the lines satisfy all of the following *local conditions*:

- The i-th input of S_1 is either inactive, equal to the i-th output of P_1, or, if the latter is inactive (during the copy phase), equal to the i-th input of S_2.
- The i-th input of P_1 is either inactive, equal to the i-th output of S_1, or, if the latter is inactive (during the erase phase), equal to the i-th input of S_2.
- The output of every gate in S_1, P_1 is either inactive, equal to the gate applied to its inputs, or, if either of the inputs is inactive (during the copy phase), equal to the respective line in S_2, P_2.
- The i-th input of S_2 is either inactive, equal to the i-th output of P_2, or, if the latter is inactive (during the compute phase), equal to the i-th input of P_2.

- The i-th input of P_2 is either inactive, equal to the i-th output of S_2, or, if the latter is inactive (during the erase phase), equal to the i-th input of P_1.
- The output of every gate in S_2, P_2 is either inactive, equal to the gate applied to its inputs, or, if either of the inputs is inactive (during the erase phase), equal to the respective line in S_1, P_1.

Notice that each line participates in at most three local constraints. Note also that each of those conditions only depends on a constant number of bits so $M_{V^{\text{LOCAL}}}$, which simply needs to compute their AND, is in AC^0 (this proves Theorem 16.1.1).

In every step a line is either activated or deactivated, so there are no fixed points; but fixed points of the original circuit lead to violations of the form $P^{\text{LOCAL}}\left(S^{\text{LOCAL}}\left(u\right)\right) \neq u$ and $S^{\text{LOCAL}}\left(P^{\text{LOCAL}}\left(u\right)\right) \neq u \neq u_{\mathbf{0}}$. In particular, every solution to the new Local End-of-a-Line instance corresponds to a valid solution to the original End-of-a-Line instance.

Also, notice that in every step we change at most two bits (and at least one), and deciding the next value of each bit can be done by looking only at a constant number of bits (conditioned on being a legal vertex in V^{LOCAL}): which is the next line to be activated/deactivated? if activated, what are the inputs to the corresponding gate? □

16.2.1 The Local End-of-a-Line counter

Consider the instance produced in the reduction above. Notice that joint active/inactive state of all the lines rotates among $O\left(m\right)$ possible vectors. I.e. there is a $(\log m + O\left(1\right))$-bit vector, henceforth called the *counter* which describes precisely which lines should be active. Furthermore, given the counter of some $u \in V^{\text{LOCAL}}$, it is easy to compute the counter of $S^{\text{LOCAL}}\left(u\right)$, i.e. we always know which line should be activated/deactivated next - regardless of the values on the lines. Similarly, for all $u \neq u_{\mathbf{0}} \in V^{\text{LOCAL}}$, it is also easy to compute the counter of $P^{\text{LOCAL}}\left(u\right)$. An additional useful property is that given the counter of u, we also know the coordinates of the critical bits of u.

16.3 Holographic Proof

In this section we construct our holographic proof system for statements of the form $u \in V^{\text{LOCAL}}$. Note that we use the term "verifier" loosely, as our verifier's responsibilities extend far beyond checking the validity of a proof - it is also expected to perform local decoding, "local proof construction", etc.

Proposition 16.3.1. *For sufficiently small constants* $\epsilon_{\text{DECODE}} \gg \epsilon_{\text{SOUND}} \gg \epsilon_{\text{COMPLETE}} > 0$*, the following holds. Fix any instance* $(S^{\text{LOCAL}}, P^{\text{LOCAL}}, V^{\text{LOCAL}})$ *of* LOCAL END-OF-A-LINE*, generated via the reduction in Section 16.2 from an instance* (S, P) END-OF-A-LINE*. Given a vertex* $u \in V^{\text{LOCAL}}$*, there is a polynomial time deterministic algorithm that takes as input* (S, P) *and* u*, and outputs a* holographic proof $\Pi(u) = (E(u), C(u), \pi(u))$ *where:*

- $E(u)$ *is an* ***encoding*** *of* u *with a* ***linear*** *error correcting code of constant relative distance.*
- $C(u)$ *is an encoding of the* ***counter*** *of* u *(with a good error correcting code).*
- $\pi(u)$ *is a* ***proof*** *that* $u \in V^{\text{LOCAL}}$*.*
- *Let* $n \triangleq |u|$*; then the total length of* $\Pi(u)$ *is* $n^{1+o(1)}$*.*
- *Let* $t \triangleq \sqrt{\log n}$ *(in particular,* $t = \omega(1)$*, but also* $t = o\left(\frac{\log n}{\log\log n}\right)$*). Let* $\mathcal{G}$ *be a finite field of size* $O(n^{1/t+o(1)})$*.* $E(u)$ *and* $\pi(u)$ *can be written as functions over domain* $\mathcal{G}^t$*. In particular, we can talk about accessing* $E(u)$*,* $\pi(u)$*, or* $\Pi(u)$ *at a point* $\mathbf{g} \in \mathcal{G}^t$*. (*$C(u)$ *is much shorter and the verifier reads it entirely.)*
- *There is a quasi-polynomial*[4] *time probabilistic verifier such that:*
  - ***(Local access)*** *The verifier's access to* $\Pi(u)$ *is restricted as follows:*
    - ***(Querying subspaces)*** *The verifier reads* $n^{o(1)}$ *axis-parallel* $(t/2)$*-dimensional subspaces of* $\Pi(u)$*. Each subspace is defined by a restriction of a* $(t/2)$*-tuple of coordinates. The tuple of coordinates which is restricted is always one of a constant number of possibilities (either* $\{2, \ldots, t/2+1\}$*, or the union of any two of:* $\{1, \ldots, t/4\}$*,* $\{t/4+1, \ldots, t/2\}$*,* $\{t/2+1, \ldots, 3t/4\}$*, and* $\{3t/4+1, \ldots, t\}$*). We denote the union of all the subspaces read by the verifier* G_{ALL}*. It will be useful to decompose* $G_{\text{ALL}} = G_{\text{PCP}} \cup G_{\text{LTC}} \cup G_{\text{CRITICAL}}$*; another subset of interest is* $G_{\text{SAMPLE}} \subset G_{\text{PCP}}$*.*
    - ***(Partially adaptive)*** $G_{\text{SAMPLE}}, G_{\text{PCP}}, G_{\text{LTC}}$ *are chosen non-adaptively, i.e. they depend only on the verifier's internal randomness.* G_{CRITICAL} *is a union of affine translations of a subspace that is also chosen non-adaptively, but the affine translations depend on* $C(u)$*.*

[4]The verifier of our holographic proof actually runs in polynomial time, except for the derandomization of the construction of λ-biased sets. As we discuss in Section 2.7.3 this could also be obtained (with slightly worse parameters) in deterministic polynomial time, but quasi-polynomial time suffices for the purpose of our main theorem.

  - ***(Randomness)*** *The verifier uses only* $(1/2 + o(1)) \log_2 n$ *bits of randomness to decide which subspaces to query.*
  - ***(Good Sample)*** *For any function* $B: \mathcal{G}^t \to [0,1]$*, and any* $G \in \{G_{\text{SAMPLE}}, G_{\text{PCP}}, G_{\text{LTC}}, G_{\text{CRITICA}}$ *with probability* $1 - o(1)$ *over the verifier's randomness,*

$$\left| \mathsf{E}_{\mathbf{g} \in G} \left[B(\mathbf{g}) \right] - \mathsf{E}_{\mathbf{h} \in \mathcal{G}^t} \left[B(\mathbf{h}) \right] \right| = o(1).$$

- ***(Soundness)*** *If the verifier is given a string* Π' *which is* ϵ_{SOUND}*-far from* $\Pi(u)$ *for every* $u \in V^{\text{LOCAL}}$*, the verifier rejects with probability* $1 - o(1)$*.*
  - ***(Robust soundness)*** *Furthermore, with probability* $1 - o(1)$*, the bits read by the verifier are* $\epsilon^2_{\text{SOUND}}$*-far from any string that would make the verifier accept. In other words, the verifier continues to reject with probability* $1 - o(1)$ *even in the presence of an adaptive adversary, who observes the queries made by the verifier, and then corrupts an* $\epsilon^2_{\text{SOUND}}/2$*-fraction of each of* $G_{\text{PCP}}, G_{\text{LTC}}$*.*
- ***(Completeness)*** *If the verifier is given access to a string* $\widehat{\Pi}$ *that is* $\epsilon_{\text{COMPLETE}}$*-close to a valid* $\Pi(u)$ *for some* $u \in V^{\text{LOCAL}}$*, then the verifier accepts with probability* $1 - o(1)$*.*
  - ***(Robust completeness)*** *Furthermore, the verifier continues to accept with probability at least* $1 - o(1)$ *even in the presence of an adaptive adversary, who observes the queries made by the verifier, and then corrupts a* $\sqrt{\epsilon_{\text{COMPLETE}}}$*-fraction of* $G_{\text{PCP}}, G_{\text{LTC}}$*.*
- ***(Decoding)*** *If the verifier is given access to a string* $\widehat{\Pi}$ *that is* ϵ_{DECODE}*-close to a valid* $\Pi(u)$ *for some* $u \in V^{\text{LOCAL}}$*:*
  - ***(Error-correction on a sample)*** *With probability at least* $1 - o(1)$*, the verifier correctly decodes the entries of* $E(u)$ *and* $\pi(u)$ *on* G_{SAMPLE}*. In particular, with probability* $1 - o(1)$*, the verifier can estimate the distance between* $\widehat{\Pi}$ *and* $\Pi(u)$ *to within an additive error of* $\pm o(1)$*.*
  - ***(Error-correction on the critical bits)*** *The verifier can* adaptively *decode and correct any* q_{CRITICAL}*-tuple of symbols from the proof with success probability at least* $1 - o(1)$*. Let* $G_{\text{CRITICAL}} \subset \mathcal{G}^t$ *denote the union of all* $O\left(|\mathcal{G}|^2\right)$ *axis-parallel subspaces queried in this process.*
  - ***(Proof-construction)*** *Given access as above to the counter* $C(u)$ *and the linear code* $E(u)$ *(but not to the proof* $\pi(u)$*), the verifier can output (with probability at least* $1 - o(1)$*) the correct values of* $\pi(u)$ *on* G_{SAMPLE}*.*

* ***(Robust decoding)*** *Both types of error-correction and the proof-construction continue to hold even in the presence of an adaptive adversary, who observes the queries made by the verifier, and then corrupts a $\sqrt{\epsilon_{\text{DECODE}}}$-fraction of each of $G_{\text{SAMPLE}}, G_{\text{PCP}}, G_{\text{LTC}}, G_{\text{CRITICAL}}$.*

The rest of this section is devoted to the proof of Proposition 16.3.1.

16.3.1 From LOCAL END-OF-A-LINE to coloring graphs

Let $t \triangleq \sqrt{\log n}$. Set $\ell \triangleq \frac{\log_2 n}{t-1} = \Theta\left(\sqrt{\log n}\right)$, and $\ell' \triangleq \ell + c \log\log n$ for some sufficiently large constant c. Let $p(z)$ be an $\mathbb{F}_2$-irreducible polynomial of degree ℓ', and let $\mathcal{G} \triangleq \mathbb{F}_2[z] / \langle p(z) \rangle$ denote the field of size $2^{\ell'}$. We will focus our attention on a subset $\mathcal{F} \subset \mathcal{G}$ of polynomials (modulo $p(z)$) of degree ℓ; notice that $|\mathcal{G}| / |\mathcal{F}| = \mathsf{poly}\log n$.

Let α be a generator of $\mathcal{G}$, and let $\mathcal{E} \triangleq \{1, \alpha, \ldots, \alpha^{5t\ell}\}$. We define the *Extended Shuffle-$\mathcal{F}$-Exchange graph* on $\mathcal{F}^{t-1} \times \mathcal{E}$ to be the directed graph such that each vertex $(x_1, \ldots, x_{t-1}, \alpha^i) \in \mathcal{F}^{t-1} \times (\mathcal{E} \setminus \{\alpha^{5t\ell}\})$, has a *static edge* to vertex $(x_1, \ldots, x_{t-1}, \alpha^{i+1})$, a *shuffle edge* to vertex $(x_2, \ldots, x_{t-1}, x_1, \alpha^{i+1})$, and ℓ *exchange edges* to all vertices of the form $(x_1 + [z^j], \ldots, x_{t-1}, \alpha^{i+1})$, for every $j \in \{0, \ldots, \ell-1\}$. Here $[z^j]$ is the element of $\mathcal{G}$ representing the equivalence class $\{z^j + q(z)\, p(z) : q(z) \in \mathbb{F}_2[z]\}$. Notice that $x \in \mathcal{F}$ iff it can be written as $x = \sum_{j \in S} [z^j]$ for some subset $S \subseteq \{0, \ldots, \ell-1\}$. In particular, for any $(x_1, \ldots, x_{t-1}), (y_1, \ldots, y_{t-1}) \in \mathcal{F}^{t-1}$, the Extended Shuffle-$\mathcal{F}$-Exchange graph contains a path of length $(t-1)(\ell+1)$ from $(x_1, \ldots, x_{t-1}, 1)$ to $(y_1, \ldots, y_{t-1}, \alpha^{(t-1)(\ell+1)})$.

For each $\alpha^i \in \mathcal{E}$, we call the set of vertices $\{(x_1, \ldots, x_{t-1}, \alpha^i)\}$ a *layer*; in particular, $\{(x_1, \ldots, x_{t-1}, 1)\}$ form the *first layer*, and $\{(x_1, \ldots, x_{t-1}, \alpha^{5t\ell})\}$ form the *last layer*.

We encode each vertex $u \in V^{\text{LOCAL}}$ as a partial coloring of the Extended Shuffle-$\mathcal{F}$-Exchange graph, representing the assignments to the lines in the four circuits S_1, P_1, S_2, P_2. Recall that $|\mathcal{F}| = 2^\ell = n^{1/(t-1)}$. We associate each line of S_1, P_1, S_2, P_2 (including input, gates, and output lines) with a vector $\mathbf{x} \in \mathcal{F}^{t-1}$ arbitrarily. For the first- and last-layer vertices $(\mathbf{x}, 1)$ and $(\mathbf{x}, \alpha^{5t\ell})$, we assign colors that represent the assignments (from $\{0, 1, \perp\}$, where $\perp$ represents an inactive line) to the corresponding line. Recall from the "local conditions" in Section 16.2 that every line only participates in three constraints.

We would like the inputs to the line in $(x_1, \ldots, x_{t-1}, \alpha^{5t\ell})$ to be propagated by the Extended Shuffle-$\mathcal{F}$-Exchange graph from the first layer. One important feature of the Extended Shuffle-$\mathcal{F}$-Exchange graph is that we can do exactly that: By using standard packet-routing techniques, we can route the assignments from the first layer to the last layer so that every vertex on the way needs to remember at most three assignments [Lei92, Theorem 3.16]. Fix one such routing. Our coloring now assigns

the same color from $\{0,1,\perp\}\times\{\perp\}^2$ to each of the corresponding vertices on the first and last layers, and propagates this assignment through the middle layers according to the routing scheme; each vertex has a color in $\{0,1,\perp\}^3$.

It is important to note that the middle layers do not perform any computation - they only copy symbols according to a fixed routing. In particular, if we change one entry in the representation of Local End-of-a-Line vertex u, we can locally compute the difference of the old and new assignments for every middle layer node, even if we don't know the old assignment itself (which may correspond to up to three values routed through that node).

Notice that in order to verify that $u \in V^{\text{LOCAL}}$, we need to check that the active/inactive pattern matches the counter, and the "local conditions". Given a coloring of the Extended Shuffle-$\mathcal{F}$-Exchange graph, checking the local conditions reduces to checking that the color for each first-layer vertex is equal to the color on the respective vertex on the last layer, and that the color of every vertex not in the first layer is computed correctly from its incoming neighbors.

16.3.2 Arithmetization

We now want to represent the verification of $u \in V^{\text{LOCAL}}$ as constraints on polynomials over $\mathcal{G}$ (do not confuse with polynomials over $\mathbb{F}_2$!).

We represent each vertex $u \in V^{\text{LOCAL}}$ as two polynomials: $T_u\colon \mathcal{F}^{t-1}\times\mathcal{E}\to\mathcal{C}$ and $T_{\text{COUNTER}}\colon \mathcal{F}^{t-1}\to\mathcal{C}$, where $\mathcal{C}\subset\mathcal{F}$ is a subset of $\mathcal{F}$ of constant size. For each $(x_1,\ldots,x_{t-1},\alpha^i)\in\mathcal{F}^{t-1}\times\mathcal{E}$, we set $T_u(x_1,\ldots,x_{t-1},\alpha^i)$ to be equal to the color assigned to the respective vertex. $T_{\text{COUNTER}}(x_1,\ldots,x_{t-1})$ is assigned 1 if the corresponding line should be active, and 0 otherwise. Additionally, we construct a polynomial $T_{S,P}\colon \mathcal{F}^{t-1}\times\mathcal{E}\to\mathcal{C}'$, for $\mathcal{C}\subset\mathcal{C}'\subset\mathcal{F}$ of size $|\mathcal{C}'|=\Theta\left(\log^{3/2}n\right)$; $T_{S,P}(\cdot)$ represents the instance of End-of-a-Line: for each $(x_1,\ldots,x_{t-1},\alpha^{5t\ell})\in\mathcal{F}^{t-1}\times\{\alpha^{5t\ell}\}$, it specifies the gate functionality, whereas for $(x_1,\ldots,x_{t-1},\alpha^i)$ $(i<5t\ell)$ it specifies the routing through this vertex. Note that each vertex takes its inputs from at most 3 out of $\Theta\left(\sqrt{\log n}\right)$ incoming edges.

Notice that $T_u, T_{\text{COUNTER}}, T_{S,P}$ are all polynomials of degree at most $|\mathcal{F}|, |\mathcal{E}|$ in their respective variables. The verifier will expect low degree extensions of $T_u, T_{S,P}$ on all of $\mathcal{G}^t$, and analogously for T_{COUNTER} over $\mathcal{G}^{t-1}$.

In the following, keep in mind that T_{COUNTER} is constructed implicitly (once we decode all $\log_2 n + O(1)$ bits of the counter), and $T_{S,P}$ is part of the input of the problem, so we only need to locally decode and verify the validity of T_u.

First, we construct a polynomial

$$\Psi(\mathbf{x}) \triangleq \psi(T_u(\mathbf{x})) \triangleq \prod_{c\in\mathcal{C}}(T_u(\mathbf{x})-c), \tag{16.1}$$

such that $\psi\left(T_u\left(\mathbf{x}\right)\right) = 0$ iff $T_u\left(\mathbf{x}\right) \in \mathcal{C}$.

Similarly, we would like to construct a polynomial that verifies that the colors are propagated correctly through the middle layers; for the last layer, it should also verify that the gate functionality is implemented correctly. First, we describe the edges of the Extended Shuffle-$\mathcal{F}$-Exchange graph with the following affine transformations:

$$\begin{aligned}
\rho_{\textsc{Static}}\left(x_1,\ldots,x_{t-1},\alpha^{i+1}\right) &\triangleq \left(x_1,\ldots,x_{t-1},\alpha^{i}\right)\\
\rho_{\textsc{Shuffle}}\left(x_1,\ldots,x_{t-1},\alpha^{i+1}\right) &\triangleq \left(x_{t-1},x_1,\ldots,x_{t-2},\alpha^{i}\right)\\
\rho_{0}\left(x_1,\ldots,x_{t-1},\alpha^{i+1}\right) &\triangleq \left(x_1+\left[z^0\right],\ldots,x_{t-1},\alpha^{i}\right)\\
&\vdots\\
\rho_{\ell-1}\left(x_1,\ldots,x_{t-1},\alpha^{i+1}\right) &\triangleq \left(x_1+\left[z^{\ell-1}\right],\ldots,x_{t-1},\alpha^{i}\right)
\end{aligned}$$

(So each vertex $\mathbf{x} \in \mathcal{F}^{t-1}\times(\mathcal{E}\setminus\{1\})$ has incoming edges from $\rho_{\textsc{Static}}\left(\mathbf{x}\right), \rho_{\textsc{Shuffle}}\left(\mathbf{x}\right), \rho_0\left(\mathbf{x}\right), \ldots, \rho_{\ell-1}\left(\mathbf{x}\right)$.)

Now we can define a polynomial

$$\Phi\left(\mathbf{x}\right) \triangleq \phi\left(T_u\left(\mathbf{x}\right), T_u\left(\rho_{\textsc{Static}}\left(\mathbf{x}\right)\right) T_u\left(\rho_{\textsc{Shuffle}}\left(\mathbf{x}\right)\right), T_u\left(\rho_0\left(\mathbf{x}\right)\right), \ldots, T_u\left(\rho_{\ell-1}\left(\mathbf{x}\right)\right), T_{(S,P)}\left(\mathbf{x}\right)\right), \tag{16.2}$$

where for $\mathbf{x} \in \mathcal{F}^{t-1} \times (\mathcal{E}\setminus\{1,\alpha^{5t\ell}\})$, $\phi\left(\cdots\right) = 0$ iff the colors are propagated correctly to the corresponding vertex, and for $\mathbf{x} \in \mathcal{F}^{t} \times \{\alpha^{5t\ell}\}$, $\phi\left(\cdots\right)$ further checks that the gate functionality is implemented correctly.

Finally, we add a third polynomial over $\mathcal{F}^{t-1}$,

$$\Xi\left(\mathbf{x}\right) \triangleq \xi\left(T_u\left(x_1,\ldots,x_{t-1},1\right), T_u\left(x_1,\ldots,x_{t-1},\alpha^{5t\ell}\right), T_{\textsc{counter}}\left(x_1,\ldots,x_{t-1}\right)\right), \tag{16.3}$$

which is zero iff $T_u\left(x_1,\ldots,x_{t-1},1\right) = T_u\left(x_1,\ldots,x_{t-1},\alpha^{5t\ell}\right)$ and the corresponding line is active or inactive as dictated by $T_{\textsc{counter}}\left(x_1,\ldots,x_{t-1}\right)$.

We now have that $u \in V^{\textsc{local}}$ is equivalent to AND of the following conditions:

- Values-in-domain: $\Psi\left(\mathbf{x}\right) = 0$ for all $\mathbf{x} \in \mathcal{F}^{t-1} \times \mathcal{E}$;
- Value-propagation and computation:$\Phi\left(\mathbf{x}\right) = 0$ for all $\mathbf{x} \in \mathcal{F}^{t-1} \times (\mathcal{E}\setminus\{1\})$; and
- Active-vs-inactive: $\Xi\left(\mathbf{x}\right) = 0$ for all $\mathbf{x} \in \mathcal{F}^{t-1}$.

Notice also that ψ, ξ are constant total degree polynomials, whereas ϕ has total degree $\mathsf{polylog}n$; thus Ψ, Φ, Ξ are of degrees $|\mathcal{F}|\cdot\mathsf{polylog}n, |\mathcal{E}|\cdot\mathsf{polylog}n \leq |\mathcal{G}|/2$ in their respective variables.

16.3.3 PCP

Our verifier should test that Ψ, Φ, Ξ are zero everywhere on the respective domains $(\mathcal{F}^{t-1} \times \mathcal{E}, \mathcal{F}^{t-1} \times (\mathcal{E} \setminus \{1\}), \mathcal{F}^{t-1})$. We describe the test for Ψ; the corresponding tests for Φ and Ξ follow analogously.

In order to test that $\Psi(\mathbf{x})$ is zero over all of $\mathcal{F}^{t-1} \times \mathcal{E}$, consider the following polynomials:

$$\Psi'\left(x_1, \ldots, x_{t-1}, \alpha^j\right) \triangleq \sum_{\substack{f_{i_{t/2+1}}, \ldots, f_{i_{t-1}} \in \mathcal{F} \\ j' \in \{1, \ldots, 5t\ell\}}} \Psi\left(x_1, \ldots, x_{t/2}, f_{i_{t/2+1}}, \ldots, f_{i_{t-1}}, \alpha^{j'}\right) \prod_{k=t/2+1}^{t-1} x_k^{i_k - 1} \cdot \left(\alpha^j\right)^{j'} \tag{16.4}$$

$$\Psi''\left(x_1, \ldots, x_{t-1}, \alpha^j\right) \triangleq \sum_{f_{i_1}, \ldots, f_{i_{t/2}} \in \mathcal{F}} \Psi'\left(f_{i_1}, \ldots, f_{i_{t/2}}, x_{t/2+1}, \ldots, x_{t-1}, \alpha^j\right) \prod_{k=1}^{t/2} x_k^{i_k - 1}. \tag{16.5}$$

In particular, observe that

$$\Psi''\left(x_1, \ldots, x_{t-1}, \alpha^j\right) = \sum_{\substack{f_{i_1}, \ldots, f_{i_{t-1}} \in \mathcal{F} \\ j' \in \{1, \ldots, 5t\ell\}}} \Psi\left(f_{i_1}, \ldots, f_{i_{t-1}}, \alpha^{j'}\right) \cdot \prod_{k=1}^{t-1} x_k^{i_k - 1} \cdot \left(\alpha^j\right)^{j'}.$$

The main point of this construction is that Ψ'' is the zero polynomial if and only if $\Psi(\mathbf{x}) = 0$ for every $\mathbf{x} \in \mathcal{F}^{t-1} \times \mathcal{E}$. Now, by definition, Ψ'' has degrees at most $|\mathcal{F}|, |\mathcal{E}|$ in each variable, respectively, so it suffices to test that it is indeed zero at one random point from $\mathcal{G}^t$.

The verifier should also ensure that Ψ' and Ψ'' are constructed correctly. This is done by picking a uniformly random $\mathbf{x} \in \mathcal{G}^t$ and verifying that (16.4) and(16.5) hold. If Ψ is constructed correctly, then Ψ' also has low degrees in each variable (compared to $|\mathcal{G}|$), so it suffices to test on a single $\mathbf{x} \in \mathcal{G}^t$. Similarly, the verifier should also check that Ψ is constructed correctly as in (16.1); thanks to the low-degree guarantee, a single $\mathbf{x} \in \mathcal{G}^t$ suffices here as well.

Notice that the verifier does not need to receive Ψ'' explicitly. (This is crucial later for the proof construction!) Testing that Ψ'' is constructed correctly at $(x_1, \ldots, x_{t-1}, \alpha^j)$ and that it is zero at the same point is equivalent to simply testing that

$$\sum_{f_{i_1}, \ldots, f_{i_{t/2}} \in \mathcal{F}} \Psi'\left(f_{i_1}, \ldots, f_{i_{t/2}}, x_{t/2+1}, \ldots, x_{t-1}, \alpha^j\right) \prod_{k=1}^{t/2} x_k^{i_k - 1} = 0. \tag{16.6}$$

We also need to test that all polynomials are indeed low degree. We do this later, in Subsection 16.3.7.

Define and test Φ', Ξ' analogously to Ψ'. We conclude that if $T_u, \Psi, \Phi, \Xi, \Psi', \Phi', \Xi'$ are all indeed of low degrees as promised and the verifier accepts with probability greater than ϵ, then T_u, T_{COUNTER} represent a real vertex $u \in V^{\text{LOCAL}}$. Notice that all of our tests choose $\mathbf{x} \in \mathcal{G}^{t/2}$ with uniform marginal probabilities.

For simplicity of notation, we henceforth treat $\Xi(\cdot)$ as t-variate polynomial (with the t-th variable being a dummy variable).

16.3.4 The final encoding

Our proof $\Pi(u)$ consists of the following strings.

- $C(u)$ consists of a good encoding of the "counter" of u. (Notice that we require a succinct encoding that can be fully decoded, and then the entire T_{COUNTER} can be computed locally.)
- $E(u)$ consists of the polynomial $T_u: \mathcal{G}^t \to \mathcal{G}$.
- $\pi(u)$ consists of the polynomials $\Psi, \Psi', \Phi, \Phi', \Xi, \Xi': \mathcal{G}^t \to \mathcal{G}$.

Although they have different lengths (in particular, $C(u)$ is much shorter), we think of them as having equal "weights", i.e. we allow only a constant fraction of errors in each. Also, we think of symbols in larger alphabets as encoded with a good binary error correcting code.

16.3.5 Local decoding and error correction

When a typical PCP verifier is given a proof that is wrong even at a single bit, it is already allowed to reject the proof. For our purpose, we want a more lenient verifier that, when given a proof that is wrong at a small fraction of the bits, corrects those bits and accepts. Furthermore, we want to ensure that our verifier only uses very little randomness: only $(1/2 + o(1)) \log_2 n$ random bits. A second goal of this subsection is to establish the desideratum that the verifier can adaptively decode an arbitrary q_{CRITICAL}-tuple of entries from the proof.

We describe how to locally decode and correct Ψ; $T_u, \Psi', \Phi, \Xi, \Phi', \Xi'$ follow with minor modifications, whereas the counter has a succinct representation and can be fully decoded. Specifically, the verifier needs to decode Ψ in order to test (16.1), (16.4). For (16.1) the verifier only needs to decode Ψ at one uniformly random point, so this is the easy case. For (16.4), the verifier wants to decode an entire axis-parallel $(t/2)$-dimensional affine subspace, $Q_{(x_1,\ldots,x_{t/2})} \triangleq \left\{\left(x_1, \ldots, x_{t/2}\right)\right\} \times \mathcal{G}^{t/2}$, for a uniformly random choice of $\left(x_1, \ldots, x_{t/2}\right) \in \mathcal{G}^{t/2}$.

Let $S \subset \mathcal{G}^{t/2}$ be a $(1/(t^2 \log|\mathcal{G}|))$-biased set of cardinality $|S| = \mathsf{poly}(t, \log|\mathcal{G}|)$ (as guaranteed by Section 2.7.3). We pick a uniformly random $\mathbf{x} = (x_1, \ldots, x_{t/2}) \in \mathcal{G}^{t/2}$, and a uniformly random $\mathbf{y} = (y_1, \ldots, y_{t/2}) \in S$. We then consider the $(t/2+1)$-dimensional affine subspace $R_{\mathbf{x},\mathbf{y}} \triangleq \bigcup_{\beta \in \mathcal{G}} Q_{(\mathbf{x}+\beta\mathbf{y})}$. If $\widehat{\Psi}$ is $O(\epsilon_{\text{DECODE}})$-close to Ψ, we have by Lemma 2.7.5 that with probability $1 - o(1)$, the restriction of $\widehat{\Psi}$ to $R_{\mathbf{x},\mathbf{y}}$, denoted $\widehat{\Psi}|_{R_{\mathbf{x},\mathbf{y}}}$, is also $O(\epsilon_{\text{DECODE}})$-close to $\Psi|_{R_{\mathbf{x},\mathbf{y}}}$. Whenever this is the case, the verifier correctly decodes Ψ on $R_{\mathbf{x},\mathbf{y}}$.

Notice that so far, our verifier queries $|\mathcal{G}|^{t/2+1} = n^{1/2+o(1)}$ symbols, and uses $\log_2|\mathcal{G}^{t/2}| + \log_2|S| = (1/2 + o(1)) \log_2 n$ random bits.

The local decoding for $T_u, \Psi', \Phi, \Xi, \Phi', \Xi'$ follows with minor modifications which we now describe. Notice that we can reuse the same random bits from the decoding of Ψ to decode each of the other polynomials. For (16.5) the verifier wants to decode Ψ' (equivalently for Φ', Ξ') on a subspace that fixes the second half of the coordinates, $Q'_{(x_{t/2+1},\ldots,x_t)} \triangleq \mathcal{G}^{t/2} \times \{(x_{t/2+1}, \ldots, x_t)\}$.

For (16.1) and (16.3), the verifier also needs to decode one or two entries of T_u, but this is no harder than decoding an entire plane. For (16.2), there is some subtlety: $\Phi(\mathbf{x})$ is a function of the values of T_u on a super-constant number ($\ell + 3 = \Theta(\sqrt{\log n})$) of points $\mathbf{x}, \rho_{\text{STATIC}}(\mathbf{x}), \rho_{\text{SHUFFLE}}(\mathbf{x}), \rho_0(\mathbf{x}), \ldots, \rho_{\ell-1}(\mathbf{x})$. Observe that $\mathbf{x}, \rho_{\text{STATIC}}(\mathbf{x}), \rho_0(\mathbf{x}), \ldots, \rho_{\ell-1}(\mathbf{x})$ all belong to the same $(t/2)$-dimensional subspace (only the first and last coordinates change); thus we only need to apply subspace-decoding twice.

Notice that our decoding mechanism also suffices to locally correct errors (with probability at least $(1 - o(1))$, and assuming that we start with a function that is close to low-degrees). Furthermore, by Lemma 2.7.5, the distance $\left|\widehat{\Psi}|_{R_{\mathbf{x},\mathbf{y}}} - \Psi|_{R_{\mathbf{x},\mathbf{y}}}\right|$ is within $\pm o(1)$ of the global distance $\left|\widehat{\Psi} - \Psi\right|$. Handling the rest of the polynomials as described in the previous paragraphs, we can thus estimate $\left|\widehat{\Pi}|_{E,\pi} - \Pi(u)|_{E,\pi}\right|$ to within $\pm o(1)$; for the bits that correspond to the counter, we can compute the distance exactly.

Decoding a particular subspace

Above, we described how to locally decode a random subspace. Suppose instead that we want to locally decode $\Psi|_{Q_{(w_1,\ldots,w_{t/2})}}$ for some particular $\mathbf{w} \in \mathcal{G}^{t/2}$ (specifically, when $Q_{(w_1,\ldots,w_{t/2})}$ contains information about one of the critical bits). Since we did not waste any random bits on picking $\mathbf{w}$, we can afford to pick $\mathbf{y}$ uniformly from all of $\mathcal{G}^{t/2}$. On the other hand, since $\mathbf{w}$ is arbitrary, there may still be an $\Theta(\epsilon_{\text{DECODE}})$ chance that $\left|\widehat{\Psi}|_{R_{\mathbf{w},\mathbf{y}}} - \Psi|_{R_{\mathbf{w},\mathbf{y}}}\right|$ is too large ($> 1/2$) to guarantee correct decoding. Instead, we

pick a third vector $\mathbf{z} \in S$, and query $\widehat{\Psi}$ on the $(t/2+2)$-dimensional affine subspace $P_{\mathbf{w},\mathbf{y},\mathbf{z}} \triangleq \bigcup_{\beta,\gamma\in\mathcal{G}} Q_{(\mathbf{w}+\beta(\mathbf{y}+\gamma\mathbf{z}))}$; notice that $P_{\mathbf{w},\mathbf{y},\mathbf{z}} \triangleq \bigcup_{\gamma\in\mathcal{G}} R_{\mathbf{w},\mathbf{y}+\gamma\mathbf{z}}$. In expectation, for each $(\mathbf{y}+\gamma\mathbf{z})$'s, the restriction $\widehat{\Psi}\mid_{R_{\mathbf{w},\mathbf{y}+\gamma\mathbf{z}}}$ is at distance at most $O(\epsilon_{\text{DECODE}})$ from $\Psi\mid_{R_{\mathbf{w},\mathbf{y}+\gamma\mathbf{z}}}$. Therefore, by Lemma 2.7.5, with probability $1-o(1)$, we have that $\left|\widehat{\Psi}\mid_{P_{\mathbf{w},\mathbf{y},\mathbf{z}}} - \Psi\mid_{P_{\mathbf{w},\mathbf{y},\mathbf{z}}}\right| = O(\epsilon_{\text{DECODE}})$.

16.3.6 Local proof-construction

We now want to establish the following unusual property of our holographic proof scheme: we can, with high probability, locally construct any subspace $Q_{(x_1,\ldots,x_{t/2})}$ of the proof $\pi(u)$, by reading only a small part of the encoding $E(u)$ (in particular, we need to decode a constant number of $(t/2+1)$-dimensional subspaces from $E(u)$). Here, it is important that $Q_{(x_1,\ldots,x_{t/2})}$ is defined via a restriction of the *first* $t/2$ coordinates, but we assume access to decoded subspaces of $E(u)$ with restriction of any $(t/2)$-tuple of coordinates.

Recall that in order to compute $\Psi(\mathbf{x})$ or $\Xi(\mathbf{x})$, it suffices to know the values of T_u at a constant number of locations ((16.1) and (16.3)), so here this property trivially holds. For $\Phi(\mathbf{x})$, we need to know T_u at $\Theta\left(\sqrt{\log n}\right)$ locations (16.2); however, as we argue in Subsection 16.3.5, they belong to two axis-parallel subspaces, so it again suffices to decode T_u on a constant number of subspaces.

In order to compute $\Psi'(\mathbf{x}), \Phi'(\mathbf{x}), \Xi'(\mathbf{x})$, we need to correctly decode the values of Ψ, Φ, Ξ on (almost) an entire axis-parallel subspace $Q_{(x_1,\ldots,x_{t/2})}$. The values of Ψ, Φ, Ξ are not given as part of $E(u)$, but (as we argued in the previous paragraph), we can reconstruct them from T_u. Specifically, in order to know the values of Ψ or Ξ on $Q_{(x_1,\ldots,x_{t/2})}$, we simply need to decode T_u on $Q_{(x_1,\ldots,x_{t/2})}$. In order to construct Φ on an entire subspace, observe that for every $\left(x_1,\ldots,x_{t/2},f_{i_{t/2+1}},\ldots,f_{i_{t-1}},\alpha^j\right) \in Q_{(x_1,\ldots,x_{t/2})}$, all the vectors $\rho_{\text{STATIC}}\left(x_1,\ldots,x_{t/2},f_{i_{t/2+1}},\ldots,f_{i_{t-1}},\alpha^j\right), \rho_0\left(x_1,\ldots,x_{t/2},f_{i_{t/2+1}},\ldots,f_{i_{t-1}},\alpha^j\right),\ldots,$ $\rho_{\ell-1}\left(x_1,\ldots,x_{t/2},f_{i_{t/2+1}},\ldots,f_{i_{t-1}},\alpha^j\right)$ belong to the same subspace $Q_{(x_1,\ldots,x_{t/2})}$. Furthermore, all the vectors $\rho_{\text{SHUFFLE}}\left(x_1,\ldots,x_{t/2},f_{i_{t/2+1}},\ldots,f_{i_{t-1}},\alpha^j\right) = \left(f_{i_{t-1}},x_1,\ldots,x_{t/2},f_{i_{t/2+1}},\ldots,f_{i_{t-2}},\alpha^j\right)$ belong to the subspace $Q^{\text{SHUFFLE}}_{(x_1,\ldots,x_{t/2})} \triangleq \mathcal{G}\times\left(x_1,\ldots,x_{t/2}\right)\times\mathcal{G}^{t/2-1}$. Therefore the value of $\Phi'(\mathbf{x})$ can also be computed from the values of T_u on a constant number of subspaces.

16.3.7 Local robust testing

So far, we assumed that we are given functions that are low degree (Subsection 16.3.3), or approximately low degree (Subsection 16.3.5). However, we must verify that this is indeed the case.

We will use results for locally testing tensor codes [Vid15]. In order to describe them, let us briefly introduce some minimal background and notation. Given linear codes C_1, C_2 with respective generator matrices M_1, M_2, their *tensor product*, denoted $C_1 \otimes C_2$ is the linear code generated by the tensor product of the matrices $M_1 \otimes M_2$. We say that $C_1 \otimes C_2$ is a *tensor code*. In general, we can talk about the tensor product of k codes, $C_1 \otimes \cdots \otimes C_k$, which is defined recursively. We say that $C_1 \otimes \cdots \otimes C_k$ is the k-th power of C (denoted C^k), if $C_i = C$ for all $i \in [k]$. In our case, for example, the code $\mathcal{C}$ of all $(|\mathcal{G}|/2)$-individual degrees polynomials $T : \mathcal{G}^t \to \mathcal{G}$ is the t-th power of the code of all $(|\mathcal{G}|/2)$-degree polynomials $t : \mathcal{G} \to \mathcal{G}$.

Alternatively, we can think of any product code as a function from some product domain to some range. The (axis-parallel) *hyperplane tester* picks a uniformly random index $i \in [k]$, and a uniformly random value v_i from the i-th domain. It reads the assignment of the function for all vectors that have v_i in their i-th coordinate, and tests whether they form (or are close to) a codeword from $C_1 \otimes \cdots \otimes C_{i-1} \otimes C_{i+1} \otimes \cdots \otimes C_k$. Going back to our setting, this could mean selecting a random $i \in \{1, \ldots, t\}$, and a random value $v_i \in \mathcal{G}$; reading $T(\mathbf{x})$ for all $\mathbf{x}$ such that $x_i = v_i$, and verifying that the restricted $(t-1)$-variate function is close to a low degrees polynomial.

To optimize query complexity, it will be more convenient for us to think of the code of $(t/4)$-variate $(|\mathcal{G}|/2)$-individual degrees polynomials $T_{t/4} : \mathcal{G}^{t/4} \to \mathcal{G}$, henceforth denoted $\mathcal{C}_{1/4}$. The code $\mathcal{C}$ can now be written $\mathcal{C} = \left(\mathcal{C}_{1/4}\right)^4$. The hyperplane tester now chooses a random $i \in \{1, 2, 3, 4\}$ and a random value $v_i \in \mathcal{G}^{t/4}$, and reads $T(\mathbf{x})$ for all $\mathbf{x}$ whose restriction to the i-th $(t/4)$-tuple of coordinates is equal to v_i.

We can now apply the following theorem due to Viderman [Vid15]:

Theorem 16.3.2. *[Vid15, Theorem 4.4] Let $\mathcal{C} = C_1 \otimes \cdots \otimes C_k$ be the tensor product of $k > 2$ linear codes of relative distances $\delta_1, \ldots, \delta_k$. Let M be a string which is ϵ-far from the tensor code $\mathcal{C}$. Then the restriction of M to a random hyperplane drawn from the above distribution is, in expectation, at least $\left(\epsilon \cdot \frac{\prod_{j=1}^{k} \delta_j}{2k^2}\right)$-far from the restricted code $C_1 \otimes \cdots \otimes C_{i-1} \otimes C_{i+1} \otimes \cdots \otimes C_k$.*

Let $S_{1/4} \subset \mathcal{G}^{t/4}$ be a $(1/(t^2 \log |\mathcal{G}|))$-biased set of cardinality $\left|S_{1/4}\right| = \mathsf{poly}(t, \log |\mathcal{G}|)$ (for example, take the restriction of each $\mathbf{y} \subset S$ to its first $t/4$ entries). Sample $\mathbf{x} \in \mathcal{G}^{t/4}$ and $\mathbf{y} \in S_{1/4}$ uniformly at random. Run the hyperplane tester for $\left(\mathcal{C}_{1/4}\right)^4$ on each alleged polynomial $4|\mathcal{G}|$ times, taking all the possibilities for $i \in \{1, 2, 3, 4\}$ and $v_i = (\mathbf{x} + \beta \mathbf{y})$ for all $\beta \in \mathcal{G}$. By Theorem 16.3.2, if the polynomial is ϵ_{SOUND}-far from $\left(\mathcal{C}_{1/4}\right)^4$, then the expected distance on each of the $4|\mathcal{G}|$ tests is $\left(\epsilon_{\text{SOUND}} \cdot \frac{\prod_{j=1}^{k} \delta_j}{2k^2}\right) = \epsilon_{\text{SOUND}}/512$ (since $k = 4$ and $\delta_j \geq 1/2$). By Lemma 2.7.5, the average over all the tests is, with probability $1 - o(1)$, within $\pm o(1)$ of this expectation.

So far the tester requires $(1/4 + o(1)) \log_2 n$ random bits to sample a hyperplane, but has prohibitively high query complexity: $n^{3/4+o(1)}$. In order to drive down the query complexity, we can recurse on the test using fresh $(1/4 + o(1)) \log_2 n$ random bits: we re-partition $\left(\mathcal{C}_{1/4}\right)^3$ as a tensor product of three linear codes, and reapply Viderman's theorem. Our query complexity is now $n^{1/2+o(1)}$, the total random-bits complexity is $(1/2 + o(1)) \log_2 n$, and the robustness guarantee is $\left(\epsilon_{\text{SOUND}} \cdot \frac{2^{-4}}{2 \cdot 4^2} \cdot \frac{2^{-3}}{2 \cdot 3^2}\right) - o(1) > \epsilon_{\text{SOUND}}/10^5$.

Finally, our tester accepts iff the average distance across all $(t/2)$-dimensional subspaces from the nearest low-degrees polynomial is less than $2\sqrt{\epsilon_{\text{COMPLETE}}} < \epsilon_{\text{SOUND}}/(2 \cdot 10^5)$. Thus our tester accepts with high probability whenever $\widehat{\Psi}$ is $\epsilon_{\text{COMPLETE}}$-close to the true Ψ, and rejects with high probability whenever $\widehat{\Psi}$ is ϵ_{SOUND}-far from any low degrees polynomial. (Testing $\widehat{T_u}, \widehat{\Phi}, \widehat{\Xi}, \widehat{\Psi'}, \widehat{\Phi'}, \widehat{\Xi'}$ follows analogously.)

16.3.8 Summary

The verifier, on input $\widehat{\Pi} = \left(\widehat{E}, \widehat{C}, \widehat{\pi}\right)$, does the following.

Randomness

In a preprocessing step, the verifier fixes in advance a $(1/(t^2 \log |\mathcal{G}|))$-biased set S of cardinality $|S| = \mathsf{poly}(t, \log |\mathcal{G}|)$. Let $S_{1/4}$ denote the restriction of S to the first $t/4$ coordinates.

The tester then:

- draws $\mathbf{x}$ uniformly at random from $\mathcal{G}^{t/2}$, and $\mathbf{y}$ uniformly at random from S; and
- reusing the same randomness, draws $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{G}^{t/4}$ and $\mathbf{y}_1, \mathbf{y}_2 \in S_{1/4}$.

Queries

The verifier reads $\widehat{T_u}$ on all vectors $\mathbf{z} \in \mathcal{G}^t$ such that one of $\left\{\left(z_1, \ldots, z_{t/2}\right), \left(z_2, \ldots z_{t/2+1}\right)\right\}$ is of the form $(\mathbf{x} + \beta\mathbf{y})$ for some $\beta \in \mathcal{G}$. It also reads $\widehat{\Psi}, \widehat{\Phi}, \widehat{\Xi}, \widehat{\Psi'}, \widehat{\Phi'}, \widehat{\Xi'}$ on all $\mathbf{z} \in \mathcal{G}^t$ such that $\left(z_1, \ldots z_{t/2}\right) = (\mathbf{x} + \beta\mathbf{y})$ for some $\beta \in \mathcal{G}$ (denote those vectors $R_{\mathbf{x},\mathbf{y}}$). Similarly, it also reads Ψ', Φ', and Ξ' on all $\mathbf{z} \in \mathcal{G}^t$ such that $\left(z_{t/2+1}, \ldots, z_t\right) = (\mathbf{x} + \beta\mathbf{y})$ (denote those vectors $R'_{\mathbf{x},\mathbf{y}}$). The union of all those subspaces (i.e. $R_{\mathbf{x},\mathbf{y}} \cup R'_{\mathbf{x},\mathbf{y}}$) forms G_{PCP}, while $G_{\text{SAMPLE}} \triangleq R_{\mathbf{x},\mathbf{y}}$.

For each choice of $i_1 \neq i_2 \in \{1, 2, 3, 4\}$ and each $\beta_1, \beta_2 \in \mathcal{G}$, the verifier reads each polynomial on all vectors $\mathbf{z} \in \mathcal{G}^t$ such that the i_1-th $(t/4)$-tuple of $\mathbf{z}$ is equal to

$\mathbf{x}_1 + \beta_1 \mathbf{y}_1$, and the i_2-th $(t/4)$-tuple of $\mathbf{z}$ is equal to $\mathbf{x}_2 + \beta_2 \mathbf{y}_2$. The union of all those subspaces forms G_{LTC}.

Finally, the verifier also reads the entire encoding $C(u)$ of the counter of u. Given $C(u)$, the verifier also picks q_{CRITICAL} vectors $\mathbf{w}^1, \ldots, \mathbf{w}^{q_{\text{CRITICAL}}} \in \mathcal{G}^{t/2}$ that correspond to each of the critical bits, and reads T_u on all vectors $\mathbf{z} \in \mathcal{G}^t$ such that $\left(z_1, \ldots, z_{t/2}\right) = (\mathbf{w}^i + \beta(\mathbf{x} + \beta\mathbf{y}))$ for some $\beta, \gamma \in \mathcal{G}$ and $i \in \{1, \ldots, q_{\text{CRITICAL}}\}$. Those are G_{CRITICAL}.

Test

The verifier uses the hyperplane tester from Subsection 16.3.7 to verify that all the polynomials are indeed close to low-degree polynomials. (If not, it suffices to return an error message and abort.)

Correct & decode

For each subspace queried, the verifier finds the unique low-degree polynomials $T_u, \Psi, \Phi, \Xi, \Psi', \Phi', \Xi'$ that are close to the values read.

Verify

The verifier checks that the corrected polynomials $T_u, \Psi, \Phi, \Xi, \Psi', \Phi', \Xi'$ satisfy (16.1), (16.2), (16.3), and (16.4) on all of $R_{\mathbf{x},\mathbf{y}}$. It also checks that (16.6) (and its analogues for Φ', Ξ') are satisfied on all of $R'_{\mathbf{x},\mathbf{y}}$.

Estimate distance

The verifier also computes the hamming distance between the values of $\widehat{T_u}, \widehat{\Psi}, \widehat{\Phi}, \widehat{\Xi}, \widehat{\Psi'}, \widehat{\Phi'}, \widehat{\Xi'}$ and $T_u, \Psi, \Phi, \Xi, \Psi', \Phi', \Xi'$ on $R_{\mathbf{x},\mathbf{y}}$.

Extrapolate

Given $C(u)$, the verifier can locally reconstruct $C(S^{\text{LOCAL}}(u))$. Additionally, after the verifier decoded the values of T_u at the critical bits, it also knows the current and new values of all the entries that change between u and $S^{\text{LOCAL}}(u)$. Since $E(\cdot)$ is a linear encoding, the verifier can also completely construct the difference vector $E(S^{\text{LOCAL}}(u)) - E(u)$. In other words, given the value of $T_u(\mathbf{x})$ for any $\mathbf{x}$, the verifier can locally compute $T_{S^{\text{LOCAL}}(u)}(\mathbf{x})$.

The verifier can use the ideas from Subsection (16.3.6) to also construct the values of all the polynomials $(\Psi, \Phi, \Xi, \Psi', \Phi', \Xi')$ for $S^{\text{LOCAL}}(u)$, on all of $R_{\mathbf{x},\mathbf{y}}$.

For $P^{\text{LOCAL}}(u)$, the verifier must first check that u is not the special vertex $u_{\mathbf{0}}$. This is easy to do since T_u and the respective $T_{u_{\mathbf{0}}}$ are both low degree polynomials and must differ (if $u \neq u_{\mathbf{0}}$) on a constant fraction of their entries. If indeed $u \neq u_{\mathbf{0}}$, recovering information about $P^{\text{LOCAL}}(u)$ is completely analogous to the procedure for $S^{\text{LOCAL}}(u)$. Otherwise, we simply have $P^{\text{LOCAL}}(u_{\mathbf{0}}) = u_{\mathbf{0}}$.

16.4 Polymatrix WeakNash

In this section we prove our main technical result:

Theorem 16.4.1. *There exists a constant $\epsilon_{\text{NASH}} > 0$, such that there is a $2^{n^{1/2+o(1)}}$-time reduction from* End-of-a-Line *to finding an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-*Well-Supported-WeakNash *of a complete bipartite polymatrix game between $n^{1/2+o(1)}$ players with $2^{n^{1/2+o(1)}}$ actions each; the payoffs in each bimatrix subgame are in $[0, 1/n_B], [0, 1/n_A]$, where n_A, n_B denote the number of players on each side of the bipartite graph.*

We construct a subexponential polymatrix game that implements a variant of the hard Brouwer function constructed in Section 4.2. The new Brouwer function is very similar, but we make the following modifications:

- We use the vertices of a Local End-of-a-Line instance instead of End-of-a-Line.
- Instead of encoding each Local End-of-a-Line vertex u with an arbitrary error correcting code, the Brouwer vertices correspond to the holographic proofs $\Pi(u)$ from Section 16.3.
- For convenience of notation, instead of maintaining m coordinates (which we expect to be identical anyway), for each of the auxiliary compute-vs-copy bit and the special direction, we keep just one coordinate for each, but give each a constant relative weight.
- In particular, the "first" Brouwer line segment goes from $\mathbf{z}_2 \triangleq (\Pi(u_{\mathbf{0}}), \Pi(u_{\mathbf{0}}), 0, 2)$ to $(\Pi(u_{\mathbf{0}}), \Pi(u_{\mathbf{0}}), 0, 0)$.

The rest proof proceeds as follows: In Subsection 16.4.1, we define the strategy space for the players and relate it to the holographic proof system from Section 16.3. In Subsection 16.4.2 we introduce some important notation used throughout this section. In the next two subsections (16.4.3 and 16.4.4) we define the players' payoffs,

and implement an imitation gadget, assuming that the Brouwer function f can be locally computed. In Subsection 16.4.5, we translate the PCP guarantees from Proposition 16.3.1 to guarantees about approximate equilibria of our polymatrix game. This completes the setup of the argument. The rest of this section (Subsections 16.4.6-16.4.9) shows that we can indeed locally compute f in the different regions where it is defined (outside the picture, close to a Brouwer line segment, etc.).

16.4.1 Players and strategies

Let $\mathcal{R}_{\text{PCP}}$ denote the set of random strings used by the PCP verifier in Proposition 16.3.1. For each $r \in \mathcal{R}_{\text{PCP}}$ we construct a player (A, r) on one side of the bipartite game. On the opposite side, we construct a player $(B, \mathbf{q})$ for every $\mathbf{q} \in \mathcal{G}^{t/2}$. We refer to the respective sides of the bipartite game as Alice's players and Bob's players. (In Section 16.5 we will construct a bimatrix game between Alice and Bob where each player "controls" the vertices on her or his side of the polymatrix game.)

Each player has $(3/\epsilon_{\text{PRECISION}} + 1)^{n^{1/2+o(1)}+2}$ actions. We think of each action as an ordered tuple of:

- two vectors in $[-1, 2]^{n^{1/2+o(1)}}$, where the interval $[-1, 2]$ is discretized into $\{-1, -1 + \epsilon_{\text{PRECISION}}, \ldots, 2 - \epsilon_{\text{PR}}$ and
- two more variables in $[-1, 2]$.

Recall that each Brouwer vertex corresponds to a pair (u, v) of either identical or consecutive vertices from the Local End-of-a-Line instance. For player (A, r), the two vectors allegedly correspond to bits read from two holographic proofs (a-la Section 16.3) $(\Pi(u), \Pi(v))$ by the PCP verifier with random string r. For player $(B, \mathbf{q})$ the vectors represent (a binary encoding of) the counters $C(u), C(v)$, and the entries of $E(u), \pi(u), E(v), \pi(v)$ on a $(t/2)$-dimensional subspace $G^{(B,\mathbf{q})} \subset \mathcal{G}^t$ to be specified below.

The additional two variables represent the compute-vs-copy bit and the special direction.

In addition to the assignments described above, each player on Alice's side has an additional choice among $(3/\epsilon_{\text{PRECISION}} + 1)^{n^{1/2+o(1)}}$ actions. These additional actions allegedly correspond to $2q_{\text{CRITICAL}}$ (affine, $(t/2 + 2)$-dimensional) subspaces from which we can decode the q_{CRITICAL} critical bits of each of the vertices. (We know which entries are represented by reading the "counter" part of the proof.)

Bob's subspace $G^{(B,\mathbf{q})}$

Let $G^{(A,r)}_{\text{ALL}} \subset \mathcal{G}^t$ denote the set of entries of $E(\cdot), \pi(\cdot)$ read by the PCP verifier with random string r. Recall that the PCP verifier picks a set of $(t/2)$-dimensional axis-parallel affine subspaces; each subspace is an affine shift of a linear subspace, spanned by one of a constant number of $(t/2)$-tuples of standard basis vectors. We pick $G^{(B,\mathbf{q})}$ so that it is spanned by vectors that are linearly independent of all those tuples. In particular, $G^{(B,\mathbf{q})}$ intersects each of (A,r)'s subspaces at exactly one point.

For any $(t/2)$-tuple of standard basis vectors of $\mathcal{G}^t$, a uniformly random $(t/2)$-tuple of new vectors in $\mathcal{G}^t$ completes it to a basis with high probability. To see this, consider adding the new vectors one by one; at each step, there is at least a $(1 - 1/|\mathcal{G}|)$-probability that it is linearly independent of the previously chosen vectors; the claim follows by union bound and $t = o(|\mathcal{G}|)$. Therefore, a random $(t/2)$-tuple of vectors will also, with high probability, be simultaneously linearly independent of each of the relevant tuples of standard basis vectors. Fix any such a $(t/2)$-tuple (there are also efficient ways to deterministically find such a tuple, but in quasi-polynomial time we can simply brute-force enumerate through all tuples).

Finally, we let $G^{(B,\mathbf{0}_{t/2})}$ be the linear subspace spanned by those vectors; for general $\mathbf{q} \in \mathcal{G}^{t/2}$, we let $G^{(B,\mathbf{q})}$ be the same subspace shifted by $(\mathbf{q}, \mathbf{0}_{t/2})$.

16.4.2 Notation

We introduce some necessary additional notation for formally handling vectors in this section. The notation for (A,r)'s assignment to the critical bits is analogous in spirit but more subtle to formalize; we come back to it to Subsection 16.4.3.

Definition 16.4.2. We say that a player is *happy* if every strategy on her support is ϵ_{NASH}-optimal.

Rounding For $x \in \mathbb{R}$, let $\nu(x) \triangleq \begin{cases} 1 & x > 1/2 \\ 0 & \text{otherwise} \end{cases}$ denote the rounding of x to $\{0,1\}$; for $\mathbf{x} \in \mathbb{R}^n$, we compute $\nu(\mathbf{x})$ coordinate-wise.

Coordinates Let $M \triangleq \{1, \ldots, 2m+2\}$ denote the set of all coordinates. We partition M into: $M_1 \triangleq \{1, \ldots, m\}$ (the first proof), $M_2 \triangleq \{m+1, \ldots, 2m\}$ (the second proof), $M_3 \triangleq \{2m+1\}$ (the Compute-vs-Copy bit), and $M_4 \triangleq \{2m+2\}$ (the special direction). We will also use the abbreviations $M_{1,2} \triangleq M_1 \cup M_2$ and $M_{3,4} \triangleq M_3 \cup M_4$.

Fix some player (A,r) on Alice's side; we further detail (A,r)'s strategy space as follows. Let $G^{(A,r)}_{\text{SAMPLE}}, G^{(A,r)}_{\text{PCP}}, G^{(A,r)}_{\text{LTC}} \subset \mathcal{G}^t$ denote the sets of entries of $E(\cdot), \pi(\cdot)$ read by

the PCP verifier with random string r from the alleged proofs. Let $I^{(A,r)}_{\text{SAMPLE}}, I^{(A,r)}_{\text{PCP}}, I^{(A,r)}_{\text{LTC}} \subset M_{1,2}$ denote the indices of the corresponding bits, respectively. For player $(B, \mathbf{q})$ on Bob's side, let $I^{(B,\mathbf{q})} \subset M_{1,2}$ denote the indices of the bits corresponding to $G^{(B,\mathbf{q})}$. We also let $K \subset M_{1,2}$ (K for Kounter) denote the set of indices that correspond to the encodings of counters in both proofs. (Notice that K is the same for every player on both sides.)

Finally we will be particularly interested in the following sets:

$$\begin{aligned} J^{(A,r)} &\triangleq K \cup I^{(A,r)}_{\text{SAMPLE}} \cup M_{3,4} \\ J^{(B,\mathbf{q})} &\triangleq K \cup I^{(B,\mathbf{q})} \cup M_{3,4}. \end{aligned}$$

Partial vectors Player (A, r)'s strategy $\mathbf{a}$ naturally corresponds to a *partial vector* $\mathbf{x}^{(\mathbf{a})}_{\text{ALL}} \in ([-1, 2] \cup \{\perp\})^M$. For $i \in K \cup I^{(A,r)}_{\text{PCP}} \cup I^{(A,r)}_{\text{LTC}} \cup M_{3,4}$, we set the i-th coordinate of $\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}$ to the value that $\mathbf{a}$ assigns to the corresponding coordinate; for all other i's, we let $\mathbf{x}^{(\mathbf{a})}_{\text{ALL}} |_i \triangleq \perp$.

In particular we are interested in the partial vector $\mathbf{x}^{(\mathbf{a})} \in ([-1, 2] \cup \{\perp\})^M$ that describes (A, r)'s partial assignment to $J^{(A,r)}$. For $i \in J^{(A,r)}$, we set the i-th coordinate of $\mathbf{x}^{(\mathbf{a})}$ to the value that $\mathbf{a}$ assigns to the corresponding coordinate; for all other i's, we let $\mathbf{x}^{(\mathbf{a})} |_i \triangleq \perp$.

For action $\mathbf{b}$ taken by player $(B, \mathbf{q})$, we can simply define one partial vector $\mathbf{x}^{(\mathbf{b})} \in ([-1, 2] \cup \{\perp\})^M$, where $\mathbf{x}^{(\mathbf{b})} |_i \triangleq \perp$ for $i \notin J^{(B,\mathbf{q})}$.

Partial norms All norms in this section are 2-norm: we henceforth drop the 2 subscript from $\|\cdot\|_2$. Instead, we will use the subscript to denote the subset of coordinates over which we want to compute the norm. For example,

$$\|\mathbf{x}\|^2_K \triangleq \mathsf{E}_{i \in K}\left[x_i^2\right].$$

In fact, we are often interested in expectation with respect to a non-uniform distribution; for example:

$$\begin{aligned} \|\mathbf{x}\|^2_{J^{(A,r)}} &\triangleq \frac{1}{4}\mathsf{E}_{i \in I^{(A,r)}_{\text{SAMPLE}}}\left[x_i^2\right] + \frac{1}{4}\mathsf{E}_{i \in K}\left[x_i^2\right] + \frac{1}{2}\mathsf{E}_{i \in M_{3,4}}\left[x_i^2\right] \\ \|\mathbf{x}\|^2_{J^{(B,\mathbf{q})}} &\triangleq \frac{1}{4}\mathsf{E}_{i \in I^{(B,\mathbf{q})}}\left[x_i^2\right] + \frac{1}{4}\mathsf{E}_{i \in K}\left[x_i^2\right] + \frac{1}{2}\mathsf{E}_{i \in M_{3,4}}\left[x_i^2\right]. \end{aligned}$$

The distribution is set such that each part (e.g. K, $I^{(A,r)}_{\text{PCP}}$, $I^{(B,\mathbf{q})}$, or $M_1, \ldots, M_4$) receives an equal weight (up to constant factors). The distribution will henceforth be implicit, and only the subset of coordinates will be explicit.

This notation also allows us to talk about distances between partial vectors, e.g.

$$\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{J^{(A,r)}\cap J^{(B,\mathbf{q})}} \triangleq \frac{1}{4}\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathbf{b})}\right\|^2_K + \frac{1}{2}\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{M_{3,4}} + \frac{1}{4}\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{I^{(A,r)}_{\text{SAMPLE}}\cap I^{(B,\mathbf{q})}}.$$

Aggregate vectors Let $\mathcal{A}$ denote the joint (product) distribution over all Alice's players' mixed strategies, and define $\mathcal{B}$ analogously for Bob's players. We define $\mathbf{x}^{(\mathcal{A})}, \mathbf{x}^{(\mathcal{B})} \in [-1,2]^M$ to be the weighted entry-wise average of $\mathbf{x}^{(\mathbf{a})}, \mathbf{x}^{(\mathbf{b})}$, respectively, where for each coordinate we only take the expectation over vectors for which it is assigned a real (non-$\perp$) value.

16.4.3 Alice's imitation gadget

Our first goal is to incentivize player (A,r) to imitate $\mathbf{x}^{(\mathcal{B})}$ on her assigned coordinates. We break up her utility on each bimatrix subgame as follows[5]:

$$U^{(A,r)} \triangleq U^{(A,r)}_{\text{SAMPLE}} + U^{(A,r)}_{\text{PCP}} + U^{(A,r)}_{\text{LTC}} + U^{(A,r)}_{\text{COUNTER}} + U^{(A,r)}_{M_{3,4}} + U^{(A,r)}_{\text{CRITICAL}}.$$

For each player $(B,\mathbf{q})$ on Bob's side, we simply define the first five parts to incentivize copying $(B,\mathbf{q})$'s strategy on each shared coordinate. Let $\mathbf{a},\mathbf{b}$ denote (A,r) and $(B,\mathbf{q})$ respective strategies; then

$$\begin{aligned}
U^{(A,r)}_{\text{SAMPLE}} &\triangleq -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{I^{(A,r)}_{\text{SAMPLE}}\cap I^{(B,\mathbf{q})}}\\
U^{(A,r)}_{\text{PCP}} &\triangleq -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{I^{(A,r)}_{\text{PCP}}\cap I^{(B,\mathbf{q})}}\\
U^{(A,r)}_{\text{LTC}} &\triangleq -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{I^{(A,r)}_{\text{LTC}}\cap I^{(B,\mathbf{q})}}\\
U^{(A,r)}_{\text{COUNTER}} &\triangleq -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{K}\\
U^{(A,r)}_{M_{3,4}} &\triangleq -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{M_{3,4}}.
\end{aligned}$$

Notice that

$$\frac{1}{8}U^{(A,r)}_{\text{PCP}} + \frac{1}{8}U^{(A,r)}_{\text{LTC}} + \frac{1}{4}U^{(A,r)}_{\text{COUNTER}} + \frac{1}{2}U^{(A,r)}_{M_{3,4}} = -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{\left(K\cup I^{(A,r)}_{\text{PCP}}\cup I^{(A,r)}_{\text{LTC}}\cup M_{3,4}\right)\cap J^{(B,\mathbf{q})}},$$

[5]For convenience of notation, we define bimatrix subgames with payoffs in $[-54/n_B, 0], [-36/n_A, 0]$. Payoffs $[0, 1/n_B], [0, 1/n_A]$ can be obtained by scaling and shifting.

whereas

$$\frac{1}{4}U^{(A,r)}_{\textsc{Sample}} + \frac{1}{4}U^{(A,r)}_{\textsc{Counter}} + \frac{1}{2}U^{(A,r)}_{M_{3,4}} = -\left\|\mathbf{x}^{(\mathbf{a})} - \mathbf{x}^{(\mathbf{b})}\right\|^2_{J^{(A,r)}\cap J^{(B,\mathbf{q})}}.$$

As we mentioned earlier, we also want (A,r) to encode the critical bits for each Local End-of-a-Line vertex. The indices of this encoding depend on both:

- the indices of the critical bits - which depends on the "counter" of the two Local End-of-a-Line vertices, allegedly encoded in both $\mathbf{a}$ and $\mathbf{b}$; and
- the directions in $\mathcal{G}^{t/2}$ that the PCP verifier uses to locally decode those bits - which depend on the random string r.

We let $G^{(A,r)}_{\text{CRITICAL}}(\mathbf{a}) \subset \mathcal{G}^t$ denote the set of additional entries that encode the critical bits - as induced by (A,r)'s strategy $\mathbf{x}^{(\mathbf{a})}|_K$ and random string r; let $I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a}) \subset M_{1,2}$ denote the corresponding bits' indices. In order to define (A,r)'s utilities, it is actually more convenient to use the critical bits determined by $(B,\mathbf{q})$'s $\mathbf{b}$, henceforth denoted $I^{(A,r)}_{\text{CRITICAL}}(\mathbf{b})$. (The superscript remains (A,r) because we still decode those bits according to random string r.) This added level of indirection prevents (A,r) from manipulating her encoding of the counter to increase her utility from the critical bits. In other words, $\mathbf{a}$ includes an assignment to q_{CRITICAL} affine $(t/2+2)$-dimensional subspaces of $\mathcal{G}^t$. On each subgame, we subjectively interpret $\mathbf{a}$'s assignment as a partial vector $\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}$ with values in $[-1,2]$ on $I^{(A,r)}_{\text{CRITICAL}}(\mathbf{b})$. In particular, we define

$$U^{(A,r)}_{\text{CRITICAL}} \triangleq -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}} - \mathbf{x}^{(\mathbf{b})}\right\|^2_{I^{(A,r)}_{\text{CRITICAL}}(\mathbf{b})\cap I^{(B,\mathbf{q})}}.$$

For some choices of $\mathbf{b}$, the induced assignment to the counter may be ambiguous or just far from every codeword. However, in that case we are probably far from every Brouwer line segment, so we don't care what (A,r) assigns to the critical bits.

Claim 16.4.3. For every happy player (A,r) on Alice's side and any action $\mathbf{a}$ in her support, for each $I \in \left\{I^{(A,r)}_{\textsc{Sample}}, I^{(A,r)}_{\text{PCP}}, I^{(A,r)}_{\text{LTC}}\right\}$

$$\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}} - \mathbf{x}^{(\mathcal{B})}\right\|^2_I = O\left(\epsilon_{\text{NASH}}\right).$$

Proof. Recall that (A,r)'s total utility from her assignment to $I^{(A,r)}_{\text{LTC}}$ is given by

$$\begin{aligned} U^{(A,r)}_{\text{LTC}} &= -\mathsf{E}_{\mathbf{q}\in\mathcal{G}^{t/2}}\mathsf{E}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q})]}\left[\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}} - \mathbf{x}^{(\mathbf{b})}\right\|^2_{I^{(A,r)}_{\text{LTC}}\cap I^{(B,\mathbf{q})}}\right] \\ &= -\mathsf{E}_{i\in I^{(A,r)}_{\text{LTC}}}\left[\left(\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}|_i - \mathsf{E}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q}(i))]}\left[\mathbf{x}^{(\mathbf{b})}|_i\right]\right)^2\right] - \mathsf{E}_{i\in I^{(A,r)}_{\text{LTC}}}\left[\text{Var}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q}(i))]}\left[\mathbf{x}^{(\mathbf{b})}|_i\right]\right], \end{aligned} \tag{16.7}$$

where $(B, \mathbf{q}(i))$ is the player on Bob's side for which $i \in I^{(B,\mathbf{q}(i))}$ (i.e. $\mathbf{q}(i)$ denotes the value $\mathbf{q} \in \mathcal{G}^{t/2}$ such that bit i is part of the representation of some $\mathbf{g} \in G^{(B,\mathbf{q})}$).

Notice that the second term of (16.7) does not depend on (A, r)'s strategy, so she simply wants to maximize the first term. For every i, she can approximate $\mathsf{E}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q}(i))]}\left[\mathbf{x}^{(\mathbf{b})}\,|_i\right]$ to within $\pm\epsilon_{\text{PRECISION}}$. Therefore, if $\mathbf{a}$ is an ϵ_{NASH}-optimal strategy, we have

$$U^{(A,r)}_{\text{PCP}} \geq -\mathsf{E}_{i\in I^{(A,r)}_{\text{PCP}}}\left[\text{Var}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q}(i))]}\left[\mathbf{x}^{(\mathbf{b})}\,|_i\right]\right] - \underbrace{\epsilon_{\text{NASH}} - \epsilon^2_{\text{PRECISION}}}_{=O(\epsilon_{\text{NASH}})}.$$

An analogous argument works for $I^{(A,r)}_{\text{SAMPLE}}, I^{(A,r)}_{\text{PCP}}$. □

The following claims follow along the same lines:

Claim 16.4.4. For every happy player (A, r) on Alice's side and any action $\mathbf{a}$ in her support,

$$\left\|\mathbf{x}^{(\mathbf{a})} - \mathbf{x}^{(\mathcal{B})}\right\|^2_K = O\left(\epsilon_{\text{NASH}}\right).$$

Claim 16.4.5. For every happy player (A, r) on Alice's side and any action $\mathbf{a}$ in her support,

$$\left\|\mathbf{x}^{(\mathbf{a})} - \mathbf{x}^{(\mathcal{B})}\right\|^2_{J^{(A,r)}} = O\left(\epsilon_{\text{NASH}}\right).$$

Corollary 16.4.6. *Let $(\mathcal{A}, \mathcal{B})$ be an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-Well-Supported-WeakNash, then*

$$\left\|\mathbf{x}^{(\mathcal{A})} - \mathbf{x}^{(\mathcal{B})}\right\|^2_M = O\left(\epsilon_{\text{NASH}}\right).$$

16.4.4 Bob's imitation gadget

We would like to also have the players on Bob's side minimize $\left\|f\left(\mathbf{x}^{(\mathcal{A})}\right) - \mathbf{x}^{(\mathbf{b})}\right\|^2_{J^{(B,\mathbf{q})}}$. In order to implement this in a polymatrix game, we want to locally compute $f(\cdot)$ given access only to the partial information provided by player (A, r)'s strategy $\mathbf{a}$. In Section 4.2.3 we argued that we can locally compute $f_i(\mathbf{x})$ using only approximate, partial information about $\mathbf{x}$. The main goal of this section is to argue that for most $i \in J^{(A,r)}$, we can compute this partial information from $\mathbf{a}$, approximately and with high probability over r. Before we do that, however, let us assume that that we have some estimate $f^{(\mathbf{a})}\,|_i$ and derive the utility of player $(B, \mathbf{q})$.

For any action $\mathbf{a}$ that (A, r) plays, we define a partial target vector $f^{(\mathbf{a})} \in \left([-1, 2] \cup \{\perp\}\right)^M$ that we would like $(B, \mathbf{q})$ to imitate. In particular, $(B, \mathbf{q})$ is incentivized to play a strategy that is close to $f^{(\mathbf{a})}$ on the coordinates where both $f^{(\mathbf{a})}$ and $\mathbf{x}^{(\mathbf{b})}$ are defined. We define,

$$U^{(B,\mathbf{q})} \triangleq -\left\|f^{(\mathbf{a})} - \mathbf{x}^{(\mathbf{b})}\right\|^2_{J^{(A,r)}\cap J^{(B,\mathbf{q})}}.$$

Similarly to $\mathbf{x}^{(\mathcal{A})}$ and $\mathbf{x}^{(\mathcal{B})}$, let $f^{(\mathcal{A})} \in [-1,2]^M$ denote the weighted entry-wise average of $f^{(\mathbf{a})}$, where for each i we take an average over $f^{(\mathbf{a})}|_i$ for all $\mathbf{a}$ in the support of (A,r), for r such that $i \in J^{(A,r)}$.

Claim 16.4.7. For every happy player $(B,\mathbf{q})$, on Bob's side and any action $\mathbf{b}$ in his support,

$$\left\|\mathbf{x}^{(\mathbf{b})} - f^{(\mathcal{A})}\right\|^2_{J(B,\mathbf{q})} = O\left(\epsilon_{\text{NASH}}\right).$$

Proof. Analogous to Claim 16.4.3. □

Corollary 16.4.8. *Let $(\mathcal{A},\mathcal{B})$ be an $(\epsilon_{\text{NASH}},\epsilon_{\text{NASH}})$-Well-Supported-WeakNash, then*

$$\left\|\mathbf{x}^{(\mathcal{B})} - f^{(\mathcal{A})}\right\|^2_{M} = O\left(\epsilon_{\text{NASH}}\right).$$

For the rest of this section, our goal is to prove that $f^{(\mathcal{A})}$ is also close to $f\left(\mathbf{x}^{(\mathcal{B})}\right)$ - this would prove that $\mathbf{x}^{(\mathcal{B})}$ is an approximate fixed point of f.

16.4.5 Reading the holographic proofs

In this subsection we translate the desiderata from Proposition 16.3.1 to guarantees about approximate equilibria in our game. Our first task in approximating $f\left(\mathbf{x}^{(\mathcal{B})}\right)$ is deciding whether $\mathbf{x}^{(\mathcal{B})}$ is close to a Brouwer line segment, close to a Brouwer vertex, or far from both. Close to any Brouwer vertex, both restrictions $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ and $\mathbf{x}^{(\mathcal{B})}|_{M_2}$ are close to some $\Pi(u), \Pi(v)$, for some $u,v \in V^{\text{LOCAL}}$; close to a Brouwer line, at least one of $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ and $\mathbf{x}^{(\mathcal{B})}|_{M_2}$ is close to some $\Pi(u)$.

The first step of this first task is to decide whether $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ is close to some $\Pi(u)$. Formally, we have a *game verifier* that encapsulates rounding assignments in $[-1,2]^M$ to $\{0,1\}^M$ and feeding them to the PCP verifier. Given that player (A,r) chooses strategy $\mathbf{a}$, we say that the game verifier *accepts* $\mathbf{a}|_{M_1}$ if:

- $\mathbf{x}^{(\mathbf{a})}|_{M_1}$ is close to binary, i.e.

$$\left\|\mathbf{x}^{(\mathbf{a})} - \nu\left(\mathbf{x}^{(\mathbf{a})}|_{M_1}\right)\right\|^2_{J(A,r)} < \sqrt{\epsilon_{\text{COMPLETE}}};$$

- and the PCP verifier accepts the rounded bits it reads from the first proof, $\nu\left(\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}|_{M_1}\right)$.

If either of those does not hold, we say that the game verifier *rejects* $\mathbf{a}|_{M_1}$.

The main idea in all the lemmata below is that by the Good Sample desideratum in Proposition 16.3.1, $G^{(A,r)}_{\text{SAMPLE}}$, $G^{(A,r)}_{\text{PCP}}$, $G^{(A,r)}_{\text{LTC}}$ and $G^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})$ are all representing samples of $\mathcal{G}^t$. Therefore, if $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ is close to $\{0,1\}^{M_1}$, then we expect that

$\mathbf{x}^{(\mathcal{B})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}$ is also close to $\{0,1\}^{M_1 \cap I_{\text{PCP}}^{(A,r)}}$. By Claim 16.4.3, we can then also expect that $\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}$ is close to the *same* vector in $\{0,1\}^{M_1 \cap I_{\text{PCP}}^{(A,r)}}$. Furthermore, if $\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}$ is ϵ-close to some binary vector $\nu\left(\mathbf{x}^{(\mathcal{B})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}\right)$, then its rounding, $\nu\left(\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}\right)$, is also $O(\epsilon)$-close to $\nu\left(\mathbf{x}^{(\mathcal{B})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}\right)$: On each coordinate i, if the two binary vectors disagree (if $\left|\nu\left(\mathbf{x}^{(\mathcal{B})}|_i\right) - \nu\left(\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_i\right)\right| = 1$), then we must have $|\nu\left(\mathbf{x}^{(\mathcal{B})}|_i\right) - \mathbf{x}^{(\mathcal{B})}|_i| + \left|\mathbf{x}^{(\mathcal{B})}|_i - \mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_i\right| \geq 1/2$. Finally, if $\nu\left(\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}\right)$ is close to $\nu\left(\mathbf{x}^{(\mathcal{B})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}\right)$ for most $\mathbf{a}$, then we can argue that reading $\nu\left(\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}\right)$ is almost like sampling $\nu\left(\mathbf{x}^{(\mathcal{B})}|_{M_1}\right)$, and then letting an adaptive adversary corrupt a small fraction of the entries. Therefore, we can use the guarantees of the PCP verifier with robust soundness / decoding / completeness.

16.4.5.1 Soundness

Lemma 16.4.9. *Let $(\mathcal{A},\mathcal{B})$ be an $(\epsilon_{\text{NASH}},\epsilon_{\text{NASH}})$-Well-Supported-WeakNash. If $\left\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\right\|_{M_1}^2 > 4\epsilon_{\text{SOUND}}$ for every $u \in V^{\text{LOCAL}}$, then for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of r's, the game verifier rejects $\mathbf{a}|_{M_1}$ for every strategy $\mathbf{a}$ in the support of (A,r).*

Proof. Suppose that $\left\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\right\|_{M_1}^2 > 4\epsilon_{\text{SOUND}}$. Then by triangle inequality we have that at least one of the following holds:

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1}^2 > \epsilon_{\text{SOUND}}^3 \tag{16.8}$$

$$\left\|\nu\left(\mathbf{x}^{(\mathcal{B})}\right) - \Pi(u)\right\|_{M_1}^2 > \epsilon_{\text{SOUND}} \tag{16.9}$$

Far from $\{0,1\}^{M_1}$

Assume first that (16.8) holds. We can further break (16.8) into its K-component and (E,π)-component:

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1}^2 = \frac{1}{2}\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1 \cap K}^2 + \frac{1}{2}\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1 \setminus K}^2 \tag{16.10}$$

When restricted to a random $r \in \mathcal{R}_{\text{PCP}}$, we can learn the first term exactly, and estimate the second term via the proxy:

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1 \cap I_{\text{SAMPLE}}^{(A,r)}}^2.$$

By the Good Sample guarantee from Proposition 16.3.1, the $\left(M_1 \cap I^{(A,r)}_{\text{SAMPLE}}\right)$-restricted distance between $\mathbf{x}^{(\mathcal{B})}$ and $\nu\left(\mathbf{x}^{(\mathcal{B})}\right)$ concentrates (to within $\pm o(1)$, with high probability) around its expectation (the last term of (16.10)). Therefore, with probability $(1-o(1))$ over the choice of $r \in \mathcal{R}_{\text{PCP}}$,

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{J^{(A,r)}} \geq \frac{1}{4}\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{\underbrace{M_1 \cap J^{(A,r)}}_{=M_1 \cap \left(I^{(A,r)}_{\text{SAMPLE}} \cup K\right)}} = \Omega\left(\epsilon^3_{\text{SOUND}}\right).$$

For any such r, if player (A,r) is also happy, we have by Claim 16.4.5 that for every strategy $\mathbf{a}$ in her support,

$$\left\|\mathbf{x}^{(\mathbf{a})} - \nu\left(\mathbf{x}^{(\mathbf{a})}\right)\right\|^2_{J^{(A,r)}} = \Omega\left(\epsilon^3_{\text{SOUND}}\right) \gg \sqrt{\epsilon_{\text{COMPLETE}}}. \tag{16.11}$$

Close to $\{0,1\}^{M_1}$, but far from every $\Pi(u)$

Alternatively, assume that $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ is close to $\{0,1\}^{M_1}$, but (16.9) holds: we have a binary vector $\nu\left(\mathbf{x}^{(\mathcal{B})}|_{M_1}\right)$ which is ϵ_{SOUND}-far from a valid proof.

We assume wlog that (16.8) is false, namely

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{M_1} \leq \epsilon^3_{\text{SOUND}}.$$

By the Good Sample guarantee in Proposition (16.3.1), we have that for $(1-o(1))$-fraction of r's, also

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{M_1 \cap I^{(A,r)}_{\text{PCP}}} = O\left(\epsilon^3_{\text{SOUND}}\right). \tag{16.12}$$

By Claim 16.4.3, for every happy (A,r) and every strategy $\mathbf{a}$ in her support,

$$\left\|\mathbf{x}^{(\mathbf{a})}_{\text{ALL}} - \mathbf{x}^{(\mathcal{B})}\right\|^2_{M_1 \cap I^{(A,r)}_{\text{PCP}}} = O\left(\epsilon^3_{\text{SOUND}}\right).$$

Therefore, combining with (16.12), we have:

$$\left\|\nu\left(\mathbf{x}^{(\mathbf{a})}_{\text{ALL}}\right) - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{M_1 \cap I^{(A,r)}_{\text{PCP}}} = O\left(\epsilon^3_{\text{SOUND}}\right),$$

and analogous arguments hold for $M_1 \cap K$ and $M_1 \cap I^{(A,r)}_{\text{LTC}}$. Therefore, by the Robust soundness guarantee in Proposition 16.3.1, the game verifier rejects every $\mathbf{a}|_{M_1}$ for all but an $o(1)$-fraction of happy (A,r)'s. □

16.4.5.2 Completeness

Lemma 16.4.10. *Let $(\mathcal{A}, \mathcal{B})$ be an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-Well-Supported-WeakNash. If $\left\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\right\|_{M_1}^2 \leq \frac{1}{4}\epsilon_{\text{COMPLETE}}$ for some $u \in V^{\text{LOCAL}}$, then for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of r's, and every strategy $\mathbf{a}$ in the support of (A, r), the game verifier accepts $\mathbf{a}|_{M_1}$.*

Proof. First, notice that the premise implies that,

$$\left\|\nu\left(\mathbf{x}^{(\mathcal{B})}\right) - \Pi(u)\right\|_{M_1}^2 \leq \epsilon_{\text{COMPLETE}}. \tag{16.13}$$

(Since for any $i \in M_1$, $\left\|\nu\left(\mathbf{x}^{(\mathcal{B})}\right) - \Pi(u)\right\|_i^2 \leq 4\left\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\right\|_i^2$.)

Furthermore, we also have that

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1}^2 \leq \left\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\right\|_{M_1}^2 \leq \frac{1}{4}\epsilon_{\text{COMPLETE}}.$$

Therefore, by the Good Sample guarantee from Proposition 16.3.1, we also have that for a $(1 - o(1))$-fraction of r's,

$$\left\|\mathbf{x}^{(\mathcal{B})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}^2 = O\left(\epsilon_{\text{COMPLETE}}\right).$$

Combining with Claim 16.4.3, we get that whenever (A, r) is also happy,

$$\left\|\mathbf{x}_{\text{ALL}}^{(\mathbf{a})} - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}^2 = O\left(\epsilon_{\text{COMPLETE}}\right).$$

Thus, as in (16.13), we have

$$\left\|\nu\left(\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}\right) - \nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M_1 \cap I_{\text{PCP}}^{(A,r)}}^2 = O\left(\epsilon_{\text{COMPLETE}}\right).$$

Analogous arguments hold for $M_1 \cap K$ and $M_1 \cap I_{\text{LTC}}^{(A,r)}$. Therefore, for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of (A, r)'s, sampling $\nu\left(\mathbf{x}_{\text{ALL}}^{(\mathbf{a})}|_{M_1}\right)$ satisfies the distance criteria for the Completeness and Robust completeness in Proposition 16.3.1. □

16.4.5.3 Decoding

Lemma 16.4.11. *Let $(\mathcal{A}, \mathcal{B})$ be an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-Well-Supported-WeakNash. If $\left\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\right\|_{M_1}^2 \leq \frac{1}{4}\epsilon_{\text{DECODE}}$ for some $u \in V^{\text{LOCAL}}$, then for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of r's, and every strategy $\mathbf{a}$ in the support of (A, r):*

- ***(Error correction)*** *The game verifier can compute the restriction of* $\Pi(u)$ *to* $M_1 \cap J^{(A,r)}$ *correctly.*
- ***(Critical bits)*** *The induced assignment on the critical bits is approximately correct:*
$$\left\| \mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}} - \Pi(u) \right\|^2_{M_1 \cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})} = O\left(\epsilon_{\text{DECODE}}\right).$$
- ***(Extrapolation)*** *The game verifier can also compute the restrictions of* $\Pi\left(S^{\text{LOCAL}}(u)\right)$ *and* $\Pi\left(P^{\text{LOCAL}}(u)\right)$ *to* $M_1 \cap J^{(A,r)}$ *correctly.*

Proof. By analogous argument to Lemma 16.4.10 (replace $\epsilon_{\text{COMPLETE}}$ with ϵ_{DECODE}), we have that, for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of (A, r)'s, sampling $\nu\left(\mathbf{x}^{(\mathbf{a})}_{\text{ALL}} \mid_{M_1}\right)$ satisfies the distance criteria for the Decoding and Robust decoding in Proposition 16.3.1. Thus the decoding of $\Pi(u)$ on $M_1 \cap J^{(A,r)}$ follows from the "Error-correction on a sample" desideratum of Proposition 16.3.1.

Critical bits

By the premise, we have that

$$\left\| \mathbf{x}^{(\mathcal{B})} - C(u) \right\|^2_{M_1 \cap K} = O\left(\epsilon_{\text{DECODE}}\right). \tag{16.14}$$

Thus, by Corollary 16.4.8, also

$$\left\| f^{(\mathcal{A})} - C(u) \right\|^2_{M_1 \cap K} = O\left(\epsilon_{\text{DECODE}}\right).$$

By Claims 16.4.4 and 16.4.7, for every ϵ_{NASH}-optimal $\mathbf{a}$ and $\mathbf{b}$ we also have that

$$\left\| \mathbf{x}^{(\mathbf{a})} - C(u) \right\|^2_{M_1 \cap K} = O\left(\epsilon_{\text{DECODE}}\right). \tag{16.15}$$

$$\left\| \mathbf{x}^{(\mathbf{b})} - C(u) \right\|^2_{M_1 \cap K} = O\left(\epsilon_{\text{DECODE}}\right). \tag{16.16}$$

Therefore, we have that for every ϵ_{NASH}-optimal $\mathbf{a}$ and $\mathbf{b}$, the counter's encoding $C(u)$ is decoded correctly; and in particular,

$$M_1 \cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a}) = M_1 \cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{b}). \tag{16.17}$$

Let $U^{(A,r)}_{\text{CRITICAL}} \mid_{M_1}$ denote the portion of $U^{(A,r)}_{\text{CRITICAL}}$ that is derived from assignments to the critical bits in the first proof. For every happy player (A, r), and every strategy

$\mathbf{a}$ in her support, we have

$$\begin{aligned}
U^{(A,r)}_{\text{CRITICAL}}\,|_{M_1} &= -\mathsf{E}_{\mathbf{q}\in\mathcal{G}^{t/2}}\mathsf{E}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q})]}\left[\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{b})\cap I^{(B,\mathbf{q})}}\right]\\
&= -\mathsf{E}_{\mathbf{q}\in\mathcal{G}^{t/2}}\mathsf{E}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q})]}\left[\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}-\mathbf{x}^{(\mathbf{b})}\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})\cap I^{(B,\mathbf{q})}}\right]\pm O\left(\epsilon_{\text{NASH}}\right)\\
&= -\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}-\mathbf{x}^{(\mathcal{B})}\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}-\mathsf{E}_{i\in M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}\left[\text{Var}_{\mathbf{b}\sim\mathcal{B}[(B,\mathbf{q}(i))]}\left[\mathbf{x}^{(\mathbf{b})}\,|_i\right]\right]\pm O\left(\epsilon_{\text{NASH}}\right),
\end{aligned}$$

where the second equality follows from (16.17). Therefore, for every ϵ_{NASH}-optimal strategy $\mathbf{a}$,

$$\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}-\mathbf{x}^{(\mathcal{B})}\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}=O\left(\epsilon^2_{\text{PRECISION}}+\epsilon_{\text{NASH}}\right)=O\left(\epsilon_{\text{NASH}}\right). \tag{16.18}$$

By the Good Sample guarantee from Proposition 16.3.1, we have that for a $(1-o(1))$-fraction of r's,

$$\left\|\mathbf{x}^{(\mathcal{B})}-\Pi\left(u\right)\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}=O\left(\epsilon_{\text{DECODE}}\right), \tag{16.19}$$

Combining with (16.18), we have that for a $(1-O(\epsilon_{\text{NASH}}))$-fraction of r's,

$$\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}-\Pi\left(u\right)\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}=O\left(\epsilon_{\text{DECODE}}\right).$$

Extrapolation

Notice that (16.19) in particular implies

$$\left\|\mathbf{x}^{(\mathcal{B})}-\nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}=O\left(\epsilon_{\text{DECODE}}\right).$$

Combining with (16.18), we have that

$$\left\|\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}-\nu\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{M_1\cap I^{(A,r)}_{\text{CRITICAL}}(\mathbf{a})}=O\left(\epsilon_{\text{DECODE}}\right).$$

Therefore, for a $(1-O(\epsilon_{\text{NASH}}))$-fraction of (A,r)'s, sampling $\nu\left(\mathbf{x}^{(\mathbf{a})}_{\text{CRITICAL}}\,|_{M_1}\right)$ satisfies the distance criteria for the Decoding and Robust decoding in Proposition 16.3.1. Therefore, by the "Error-correction on the critical bits" desideratum, for a $(1-O(\epsilon_{\text{NASH}}))$-fraction of (A,r)'s, the PCP verifier correctly decodes the assignment to all the critical bits.

From the assignment to the critical bits, the PCP verifier can reconstruct the complete difference vector $E(u)-E(S^{\text{LOCAL}}(u))$. So now, accessing $E(S^{\text{LOCAL}}(u))$ is equivalent to accessing $E(u)$ (with the same error guarantees). Using the local proof construction guarantee from Proposition 16.3.1, the PCP verifier can correctly compute $\pi(S^{\text{LOCAL}}(u))$ on $I^{(A,r)}_{\text{SAMPLE}}$. Finally, recall that once the PCP verifier successfully decoded $C(u)$, it can also locally compute $C(S^{\text{LOCAL}}(u))$. □

16.4.6 Default displacement

The simplest case is when the restrictions of $\mathbf{x}^{(\mathcal{B})}$ to both proofs are far from any $\Pi(u), \Pi(v)$. In this case $\mathbf{x}^{(\mathcal{B})}$ is far from all Brouwer line segments, and $f^{(\mathcal{A})}$ needs to apply the default displacement.

Lemma 16.4.12. *If $(\mathcal{A}, \mathcal{B})$ is an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-Well-Supported-WeakNash, and both $\|\mathbf{x}^{(\mathcal{B})} - \Pi(u)\|^2_{M_1} > 4\epsilon_{\text{SOUND}}$ and $\|\mathbf{x}^{(\mathcal{B})} - \Pi(v)\|^2_{M_2} > 4\epsilon_{\text{SOUND}}$ for every $u, v \in V^{\text{LOCAL}}$, then $\|f^{(\mathcal{A})} - f(\mathbf{x}^{(\mathcal{B})})\|^2_M = O(\epsilon_{\text{NASH}})$.*

Proof. By Lemma 16.4.9, for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of r's and every strategy $\mathbf{a}$ in their support, the game verifier rejects both proofs. In this case, for all those $\mathbf{a}$'s, $f^{(\mathbf{a})}$ implements the default displacement. If $\mathbf{a}$ is also an ϵ_{NASH}-optimal strategy, then $\|\mathbf{x}^{(\mathbf{a})} - \mathbf{x}^{(\mathcal{B})}\|^2_{J(A,r)} = O(\epsilon_{\text{NASH}})$, in which case also

$$\left\|f^{(\mathbf{a})} - f\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{J(A,r)} = O(\epsilon_{\text{NASH}}).$$

Since the rest of Alice's players can have at most an $O(\epsilon_{\text{NASH}})$ effect, we have that

$$\left\|f^{(\mathcal{A})} - f\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{M} = O(\epsilon_{\text{NASH}}).$$

□

The following cases are handled with minor modifications:

- $\mathbf{x}^{(\mathcal{B})}$ is outside the picture and far from the first line $(\mathbf{z}_2 \to (\Pi(u_{\mathbf{0}}), \Pi(u_{\mathbf{0}}), 0, 0))$.
- $\mathbf{x}^{(\mathcal{B})}$ is inside the picture, both $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ and $\mathbf{x}^{(\mathcal{B})}|_{M_2}$ are close to valid proofs, but $\mathbf{x}^{(\mathcal{B})}|_{M_{3,4}}$ doesn't match any Brouwer line segment or vertex.
- $\mathbf{x}^{(\mathcal{B})}$ is inside the picture, both $\mathbf{x}^{(\mathcal{B})}|_{M_1}$ and $\mathbf{x}^{(\mathcal{B})}|_{M_2}$ are close to valid proofs $\Pi(u)$ and $\Pi(v)$, but u and v are not consecutive vertices in the LOCAL END-OF-A-LINE graph.

16.4.7 Close to a line (1)

Now suppose that $\mathbf{x}^{(\mathcal{B})}$ is close to some Brouwer line segment $(\mathbf{s} \to \mathbf{t})$ from $\mathbf{s} = (\Pi(u), \Pi(u), 0, 0)$ to $\mathbf{t} = (\Pi(u), \Pi(v), 0, 0)$, for $u, v \in V^{\text{LOCAL}}$ such that v is the LOCAL END-OF-A-LINE-successor of u. (The case of a line from $(\Pi(u), \Pi(v), 1, 0)$ to $(\Pi(v), \Pi(v), 0, 1)$ follows with minor modifications.) Now, the line $(\mathbf{s} \to \mathbf{t})$ consists of points of the form $\beta\mathbf{s} + (1 - \beta)\mathbf{t}$ (for $\beta \in [0, 1]$). From player (A, r)'s assignment to M_1, we can locally decode and verify $\Pi(u)$ on $I^{(A,r)}_{\text{PCP}} \cup I^{(A,r)}_{\text{LTC}} \cup K$. Furthermore, we

can reconstruct both $\Pi(u)$ and $\Pi(v)$ on $I^{(A,r)}_{\text{SAMPLE}} \cup K$. Let $\mathbf{s}^{(\mathbf{a})}, \mathbf{t}^{(\mathbf{a})}$ denote the locally reconstructed restrictions of $\mathbf{s}, \mathbf{t}$, respectively, to $J^{(A,r)}$.

If all the tests passed, then we want to locally apply the displacement close to a line, as defined in (4.2). The assignments of $\mathbf{x}^{(\mathbf{a})}, \mathbf{s}^{(\mathbf{a})}, \mathbf{t}^{(\mathbf{a})}$ also induce a partial vector $\mathbf{z}^{(\mathbf{a})}$, which is the point closest to $\mathbf{x}^{(\mathbf{a})}$ on the line segment $(\mathbf{s}^{(\mathbf{a})} \to \mathbf{t}^{(\mathbf{a})})$. We can also use $\|\mathbf{x}^{(\mathbf{a})} - \mathbf{z}^{(\mathbf{a})}\|_{J^{(A,r)}}, \|\mathbf{t}^{(\mathbf{a})} - \mathbf{s}^{(\mathbf{a})}\|_{J^{(A,r)}}$ as estimates of $\|\mathbf{x} - \mathbf{z}\|_M, \|\mathbf{t} - \mathbf{s}\|_M$.

Lemma 16.4.13. *Let $(\mathcal{A}, \mathcal{B})$ be an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-Well-Supported-WeakNash. Suppose that $\mathbf{x}^{(\mathcal{B})}$ is somewhat close to some Brouwer line segment,*

$$\epsilon^2_{\text{COMPLETE}} < \min_{\beta} \left\|\mathbf{x}^{(\mathcal{B})} - (\Pi(u), \beta\Pi(u) + (1-\beta)\Pi(v), 0, 0)\right\|^2_M < \epsilon_{\text{DECODE}}, \tag{16.20}$$

but far from its endpoints,

$$\begin{aligned} \left\|\mathbf{x}^{(\mathcal{B})} - (\Pi(u), \Pi(u), 0, 0)\right\|_M &> 2\sqrt{h} \\ \left\|\mathbf{x}^{(\mathcal{B})} - (\Pi(u), \Pi(v), 0, 0)\right\|_M &> 2\sqrt{h}, \end{aligned} \tag{16.21}$$

where $u, v \in V^{\text{LOCAL}}$ satisfy $v = S^{\text{LOCAL}}(u)$ and $u = P^{\text{LOCAL}}(v)$.

Then $\|f^{(\mathcal{A})} - f(\mathbf{x}^{(\mathcal{B})})\|^2_M = O(\epsilon_{\text{NASH}})$.

Proof. By Lemma 16.4.11, for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of r's, and every strategy $\mathbf{a}$ in the support of (A, r), the game verifier can compute $\Pi(u)\,|_{I^{(A,r)}_{\text{SAMPLE}} \cup K}$ and $\Pi(v)\,|_{I^{(A,r)}_{\text{SAMPLE}} \cup K}$ correctly; let $\widehat{\Pi(u)}$ and $\widehat{\Pi(v)}$ denote the respective results of those computations.

Consider

$$\beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})} \triangleq \arg\min_{\beta} \left\|\mathbf{x}^{(\mathbf{a})} - \left(\beta\widehat{\Pi(u)} + (1-\beta)\widehat{\Pi(v)}\right)\right\|^2_{M_2 \cap \left(I^{(A,r)}_{\text{SAMPLE}} \cup K\right)}. \tag{16.22}$$

and

$$\mathbf{z}^{\left(\beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})}\right)} \triangleq \left(\Pi(u), \beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})}\Pi(u) + \left(1 - \beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})}\right)\Pi(v), 0, 0\right) \in [-1, 2]^M;$$

By (16.20), we have

$$\left\|\mathbf{x}^{(\mathcal{B})} - \mathbf{z}^{\left(\beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})}\right)}\right\|^2_M > \epsilon^2_{\text{COMPLETE}}. \tag{16.23}$$

By the Good Sample guarantee from Proposition 16.3.1, whenever $\Pi(u)\,|_{I^{(A,r)}_{\text{SAMPLE}} \cup K}$ and $\Pi(v)\,|_{I^{(A,r)}_{\text{SAMPLE}} \cup K}$ are indeed decoded correctly,

$$\left\|\mathbf{x}^{(\mathbf{a})} - \mathbf{z}^{(\mathbf{a})}\right\|^2_{J^{(A,r)}} > \epsilon^2_{\text{COMPLETE}} - O(\epsilon_{\text{NASH}}) \gg (3h)^2.$$

Therefore, for all those $\mathbf{a}$'s, we have that $f^{(\mathbf{a})}$ correctly applies the default displacement. □

Lemma 16.4.14. *Let $(\mathcal{A},\mathcal{B})$ be an $(\epsilon_{\text{NASH}},\epsilon_{\text{NASH}})$-Well-Supported-WeakNash. Suppose that $\mathbf{x}^{(\mathcal{B})}$ is close to some Brouwer line segment,*

$$\min_{\beta}\left\|\mathbf{x}^{(\mathcal{B})}-\left(\Pi\left(u\right),\beta\Pi\left(u\right)+\left(1-\beta\right)\Pi\left(v\right),0,0\right)\right\|_{M}^{2}\leq\epsilon_{\text{COMPLETE}}^{2}, \tag{16.24}$$

but far from its endpoints,

$$\begin{aligned}\left\|\mathbf{x}^{(\mathcal{B})}-\left(\Pi\left(u\right),\Pi\left(u\right),0,0\right)\right\|_{M} &> 2\sqrt{h}\\ \left\|\mathbf{x}^{(\mathcal{B})}-\left(\Pi\left(u\right),\Pi\left(v\right),0,0\right)\right\|_{M} &> 2\sqrt{h},\end{aligned} \tag{16.25}$$

where $u,v\in V^{\text{LOCAL}}$ satisfy $v=S^{\text{LOCAL}}\left(u\right)$ and $u=P^{\text{LOCAL}}\left(v\right)$.

Then $\left\|f^{(\mathcal{A})}-f\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|_{M}^{2}=O\left(\epsilon_{\text{NASH}}\right)$.

Proof. We show that $f^{(\mathbf{a})}$ correctly implements the displacement close to a line (Subsection 16.4.7) for $\mathbf{s}=\left(\Pi\left(u\right),\Pi\left(u\right),0,0\right)$ and $\mathbf{t}=\left(\Pi\left(u\right),\Pi\left(v\right),0,0\right)$.

By Lemma 16.4.10, for a $\left(1-O\left(\epsilon_{\text{NASH}}\right)\right)$-fraction of r's, and every strategy $\mathbf{a}$ in the support of (A,r), the game verifier accepts $\mathbf{a}\left|_{\{1,\ldots,m\}}\right.$. Furthermore, by Lemma 16.4.11, it can compute $\Pi\left(u\right)\left|_{I_{\text{SAMPLE}}^{(A,r)}\cup K}\right.$ and $\Pi\left(v\right)\left|_{I_{\text{SAMPLE}}^{(A,r)}\cup K}\right.$ correctly; we again denote the partial proofs locally computed by the game verifier $\widehat{\Pi\left(u\right)}$ and $\widehat{\Pi\left(v\right)}$. Denote the induced partial vectors $\mathbf{s}^{(\mathbf{a})}$ and $\mathbf{t}^{(\mathbf{a})}$. Recall that for every ϵ-optimal $\mathbf{a}$, $\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathcal{B})}\right\|_{J^{(A,r)}}^{2}=O\left(\epsilon_{\text{NASH}}\right)$, and also,$\left|\left\|\mathbf{s}^{(\mathbf{a})}-\mathbf{t}^{(\mathbf{a})}\right\|_{J^{(A,r)}}-\left\|\mathbf{s}-\mathbf{t}\right\|_{M}\right|^{2}=o\left(1\right)$.

Let

$$\beta_{(\mathbf{s}\to\mathbf{t})}^{(\mathbf{a})}\triangleq\frac{\left(\mathbf{t}^{(\mathbf{a})}-\mathbf{s}^{(\mathbf{a})}\right)}{\left\|\mathbf{s}^{(\mathbf{a})}-\mathbf{t}^{(\mathbf{a})}\right\|_{J^{(A,r)}}}\cdot\left(\mathbf{x}^{(\mathbf{a})}-\mathbf{s}^{(\mathbf{a})}\right),$$

where the $(\cdot)$ denotes a $J^{(A,r)}$-restricted dot-product.

For every ϵ-optimal $\mathbf{a}$, $\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathcal{B})}\right\|_{J^{(A,r)}}^{2}=O\left(\epsilon_{\text{NASH}}\right)$; for most of them $\Pi\left(u\right)\left|_{I_{\text{SAMPLE}}^{(A,r)}\cup K}\right.$ and $\Pi\left(v\right)\left|_{I_{\text{SAMPLE}}^{(A,r)}\cup K}\right.$ are also computed correctly, so $\mathbf{s}^{(\mathbf{a})},\mathbf{t}^{(\mathbf{a})}$ are the correct restrictions of $\mathbf{s},\mathbf{t}$ to $J^{(A,r)}$. Then, by the Good Sample guarantee from Proposition 16.3.1, we have that with probability $\left(1-o\left(1\right)\right)$, the $J^{(A,r)}$-restricted dot-product is a good approximation. Namely,

$$\left|\left(\mathbf{t}^{(\mathbf{a})}-\mathbf{s}^{(\mathbf{a})}\right)\cdot\left(\mathbf{x}^{(\mathcal{B})}\left|_{J^{(A,r)}}\right.-\mathbf{s}^{(\mathbf{a})}\right)-\left(\mathbf{t}-\mathbf{s}\right)\cdot\left(\mathbf{x}^{(\mathcal{B})}-\mathbf{s}\right)\right|=o\left(1\right)$$

and

$$\left|\left(\mathbf{t}^{(\mathbf{a})}-\mathbf{s}^{(\mathbf{a})}\right)\cdot\left(\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathcal{B})}\left|_{J^{(A,r)}}\right.\right)\right|=O\left(\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{x}^{(\mathcal{B})}\right\|_{J^{(A,r)}}\right)=O\left(\sqrt{\epsilon_{\text{NASH}}}\right),$$

and thus also

$$\left|\left(\mathbf{t}^{(\mathbf{a})}-\mathbf{s}^{(\mathbf{a})}\right)\cdot\left(\mathbf{x}^{(\mathbf{a})}-\mathbf{s}^{(\mathbf{a})}\right)-\left(\mathbf{t}-\mathbf{s}\right)\cdot\left(\mathbf{x}^{(\mathcal{B})}-\mathbf{s}\right)\right|=O\left(\sqrt{\epsilon_{\text{NASH}}}\right).$$

Because $\Pi(\cdot)$ is a constant relative distance error correcting code, $\|\mathbf{s}-\mathbf{t}\|_M = \Theta(1)$. Therefore,

$$\left|\beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})} - \beta_{(\mathbf{s}\to\mathbf{t})}\left(\mathbf{x}^{(\mathcal{B})}\right)\right|^2 = O\left(\epsilon_{\text{NASH}}\right).$$

In particular, let

$$\begin{aligned}\mathbf{z}^{(\mathbf{a})} &\triangleq \beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})}\mathbf{s}^{(\mathbf{a})} + \left(1-\beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{t})}\right)\mathbf{t}^{(\mathbf{a})}\\ \mathbf{z} &\triangleq \beta_{(\mathbf{s}\to\mathbf{t})}\left(\mathbf{x}^{(\mathcal{B})}\right)\mathbf{s} + \left(1-\beta_{(\mathbf{s}\to\mathbf{t})}\left(\mathbf{x}^{(\mathcal{B})}\right)\right)\mathbf{t};\end{aligned}$$

then for a $(1-O(\epsilon_{\text{NASH}}))$-fraction of r's, we have

$$\left\|\mathbf{z}^{(\mathbf{a})}-\mathbf{z}\right\|^2_{J(A,r)} = O\left(\epsilon_{\text{NASH}}\right).$$

By the triangle inequality, also:

$$\left|\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{z}^{(\mathbf{a})}\right\|_{J(A,r)} - \left\|\mathbf{x}^{(\mathcal{B})}-\mathbf{z}\right\|_{J(A,r)}\right|^2 = O\left(\epsilon_{\text{NASH}}\right),$$

and thus by the Good Sample guarantee also

$$\left|\left\|\mathbf{x}^{(\mathbf{a})}-\mathbf{z}^{(\mathbf{a})}\right\|_{J(A,r)} - \left\|\mathbf{x}^{(\mathcal{B})}-\mathbf{z}\right\|_{M}\right|^2 = O\left(\epsilon_{\text{NASH}}\right).$$

Finally, whenever $\mathbf{a}$ is ϵ_{NASH}-optimal and satisfies all the above, we have by Lipschitz continuity that:

$$\left\|f^{(\mathbf{a})} - f\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{J(A,r)} = O\left(\epsilon_{\text{NASH}}\right).$$

□

16.4.8 Close to a line (2)

There are a few different scenarios where we expect the game verifier to accept both proofs:

- $\mathbf{x}^{(\mathcal{B})}$ may still be far from every Brouwer line segment; e.g. because the corresponding vertices are not neighbors in the Local End-of-a-Line graph, or the values on $M_{3,4}$ don't match.
- $\mathbf{x}^{(\mathcal{B})}$ may be close to the "first" Brouwer line segment, where only the special direction (M_4) changes.
- $\mathbf{x}^{(\mathcal{B})}$ may be close to a Brouwer line segment where only the Compute-vs-Copy bit (M_3) changes.

- or $\mathbf{x}^{(\mathcal{B})}$ may be close to a Brouwer vertex.

In the first case, simply apply the default displacement. In this subsection we briefly describe the two cases corresponding to $\mathbf{x}^{(\mathcal{B})}$ close to a single Brouwer line segment. Finally, the case where $\mathbf{x}^{(\mathcal{B})}$ is close to a Brouwer vertex is deferred to Subsection 16.4.9.

Close to the first line

Near the "first" Brouwer line segment, locally computing the displacement is relatively simple. First observe that this case is easy to recognize by local access: by the Good Sample guarantee from Proposition 16.3.1, we can estimate the hamming distance of the first $2m+1$ coordinates to $(\Pi(u_{\mathbf{0}}), \Pi(u_{\mathbf{0}}), 0)$. Furthermore we know $\mathbf{s} = \mathbf{z}_2$ and $\mathbf{t} = (\Pi(u_{\mathbf{0}}), \Pi(u_{\mathbf{0}}), 0, 0)$ exactly, and therefore also $\|\mathbf{t} - \mathbf{s}\|_M = 1$. Finally, let $\mathbf{z}^{(\mathbf{a})}$ be equal to $(\Pi(u_{\mathbf{0}}), \Pi(u_{\mathbf{0}}), 0)$ on its first $2m+1$ coordinates, and $\mathbf{x}^{(\mathbf{a})}|_{M_4}$ on the last one.

Close to a line that updates the auxiliary compute-vs-copy bit

Suppose that $\mathbf{x}^{(\mathcal{B})}$ is close to a Brouwer line segment from $\mathbf{s} = (\Pi(u), \Pi(v), 0, 0)$ to $\mathbf{t} = (\Pi(u), \Pi(v), 1, 0)$, for $u, v \in V^{\text{LOCAL}}$ such that v is the Local End-of-a-Line-successor of u. (The case of a Brouwer line segment from $(\Pi(v), \Pi(v), 1, 0)$ to $(\Pi(v), \Pi(v), 0, 0)$ follows with minor modifications.)

By Lemma 16.4.11, we can locally decode both $\Pi(u)$ and $\Pi(v)$ (and verify that they are valid proofs of consecutive vertices in the Local End-of-a-Line graph). We can therefore construct partial vectors $\mathbf{s}^{(\mathbf{a})}, \mathbf{t}^{(\mathbf{a})}$ which are equal, for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of (A, r)'s, to the restrictions of the true $\mathbf{s}, \mathbf{t}$ to $J^{(A,r)}$. Finally, let

$$\mathbf{z}^{(\mathbf{a})} \triangleq \left(\mathbf{x}^{(\mathbf{a})}|_{M_3}\right)\mathbf{t}^{(\mathbf{a})} + \left(1 - \mathbf{x}^{(\mathbf{a})}|_{M_3}\right)\mathbf{s}^{(\mathbf{a})}.$$

16.4.9 Close to a vertex

In this subsection we consider the case where $\mathbf{x}^{(\mathcal{B})}$ is close to a Brouwer vertex representing two different Local End-of-a-Line vertices. In particular, we assume that it is close to a Brouwer vertex of the form $\mathbf{y} = (\Pi(u), \Pi(v), 1, 0)$. The case of $(\Pi(u), \Pi(v), 0, 0)$ follows with minor modifications; we will return to the cases of $(\Pi(u), \Pi(u), 0, 0)$ and $(\Pi(v), \Pi(v), 1, 0)$ in a couple of paragraphs.

After the game verifier accepts both proofs and that v is the successor of u in the Local End-of-a-Line graph, we are assured that $\mathbf{x}^{(\mathcal{B})}$ is indeed close to some Brouwer vertex $\mathbf{y}$. Furthermore, we know that there is an incoming Brouwer

line segment from $\mathbf{s} = (\Pi(u), \Pi(v), 1, 0)$ and an outgoing Brouwer line segment to $\mathbf{t} = (\Pi(v), \Pi(v), 1, 0)$. We can locally compute $\mathbf{s}, \mathbf{y}, \mathbf{t}$ on $J^{(A,r)}$ with high probability; denote the resulting partial vectors $\mathbf{s}^{(\mathbf{a})}, \mathbf{y}^{(\mathbf{a})}, \mathbf{t}^{(\mathbf{a})}$.

Alternatively, consider the case where $\mathbf{y} = (\Pi(v), \Pi(v), 1, 0)$ (similarly for $(\Pi(u), \Pi(u), 0, 0)$), with an incoming Brouwer line segment from $\mathbf{s} = (\Pi(P^{\text{LOCAL}}(v)), \Pi(v), 1, 0)$ and outgoing Brouwer line segment to $\mathbf{t} = (\Pi(v), \Pi(v), 0, 0)$. By the Error-correction guarantee in Lemma 16.4.11, we can (with high probability over r) locally decode partial vectors $\mathbf{y}^{(\mathbf{a})}$ and $\mathbf{t}^{(\mathbf{a})}$ on $J^{(A,r)}$; furthermore, by the Extrapolation guarantee in the same lemma, we can locally compute $\mathbf{s}^{(\mathbf{a})}$.

It is left to show that we can also locally compute the more involved construction of displacement next to a Brouwer vertex.

We know $\|\mathbf{y} - \mathbf{t}\|_M = 1/2$ exactly, and we can estimate $\|\mathbf{s} - \mathbf{y}\|_M$ from $\|\mathbf{s}^{(\mathbf{a})} - \mathbf{y}^{(\mathbf{a})}\|_{J^{(A,r)}}$. Now, we can locally compute $\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}$ and $\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}$ on the coordinates $J^{(A,r)}$ for which we know the value of $\mathbf{s}, \mathbf{y}, \mathbf{t}$; denote those partial vectors $\mathbf{z}^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})}$ and $\mathbf{z}^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}$, respectively.

Recall that

$$\begin{aligned}
\Delta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}) &= \frac{(\mathbf{y}-\mathbf{s})}{\|\mathbf{s}-\mathbf{y}\|_M} \cdot (\mathbf{x}-\mathbf{s}) - \left(1 - \sqrt{h}\right) \\
\Delta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) &= \sqrt{h} - \frac{(\mathbf{t}-\mathbf{y})}{\|\mathbf{y}-\mathbf{t}\|_M} \cdot (\mathbf{x}-\mathbf{y}).
\end{aligned}$$

We can use $\mathbf{x}^{(\mathbf{a})}, \mathbf{s}^{(\mathbf{a})}, \mathbf{y}^{(\mathbf{a})}, \mathbf{t}^{(\mathbf{a})}$ to locally compute estimates $\Delta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x})$ and $\Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x})$, and therefore also $\alpha^{(\mathbf{a})} \approx \alpha(\mathbf{x}^{(\mathcal{B})})$ and $\mathbf{z}^{(\mathbf{a})} \triangleq \alpha^{(\mathbf{a})}\mathbf{z}^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})} + (1-\alpha^{(\mathbf{a})})\mathbf{z}^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}$.

Lemma 16.4.15. *Let $(\mathcal{A}, \mathcal{B})$ be an $(\epsilon_{\text{NASH}}, \epsilon_{\text{NASH}})$-Well-Supported-WeakNash, and let $\|\mathbf{x}^{(\mathcal{B})} - (\Pi(u), \Pi(v), 1, 0)\|_M \le 2\sqrt{h}$ for $u, v \in V^{\text{LOCAL}}$ such that $v = S^{\text{LOCAL}}(u)$ and $u = P^{\text{LOCAL}}(v)$. Then $\|f^{(\mathbf{a})} - f(\mathbf{x}^{(\mathcal{B})})\|^2_M = O(\epsilon_{\text{NASH}}/h)$.*

Proof. By Lemmata 16.4.11 and 16.4.10, we have that for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of (A, r)'s and every strategy $\mathbf{a}$ in their supports, the game verifier recognizes that $\mathbf{x}^{(\mathbf{a})}$ is close to some Brouwer vertex $\mathbf{y} \triangleq (\Pi(u), \Pi(v), 1, 0)$. In particular, by the Error-correction desideratum in Lemma 16.4.11, the game verifier can construct $\mathbf{y}^{(\mathbf{a})}, \mathbf{s}^{(\mathbf{a})}, \mathbf{t}^{(\mathbf{a})} \in \{0, 1, \perp\}^M$ that, for a $(1 - O(\epsilon_{\text{NASH}}))$-fraction of (A, r)'s and every strategy $\mathbf{a}$ in their supports, are equal to $\mathbf{y}, \mathbf{s}, \mathbf{t}$ on $J^{(A,r)}$.

We also know $\|\mathbf{y} - \mathbf{t}\|_M = 1/2$ exactly, and by the Good Sample guarantee from Proposition 16.3.1,

$$\left| \left\| \mathbf{y}^{(\mathbf{a})} - \mathbf{t}^{(\mathbf{a})} \right\|_{J^{(A,r)}} - \|\mathbf{y} - \mathbf{t}\|_M \right|^2 = o(1).$$

Furthermore, as we argue in the proof of Lemma 16.4.14,

$$\begin{aligned}\left|\beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})} - \beta_{(\mathbf{s}\to\mathbf{y})}\left(\mathbf{x}^{(\mathcal{B})}\right)\right|^2 &= O\left(\epsilon_{\text{NASH}}\right)\\ \left|\beta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})} - \beta_{(\mathbf{y}\to\mathbf{t})}\left(\mathbf{x}^{(\mathcal{B})}\right)\right|^2 &= O\left(\epsilon_{\text{NASH}}\right). \end{aligned} \tag{16.26}$$

Let $\Delta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})} \triangleq \beta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})} - \left(1-\sqrt{h}\right)$, and $\Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})} \triangleq \sqrt{h} - \beta_{(\mathbf{y}\to\mathbf{t})}$. If either quantity is negative, continue with the displacement close to a line as in Subsection 16.4.7. Recall that f is $O(1)$-Lipschitz on $[-1,2]^M$, and in particular near the interface between the line and vertex displacements (the hyperplanes defined by $\beta_{(\mathbf{s}\to\mathbf{y})}(\mathbf{x}) = 0$ and $\beta_{(\mathbf{y}\to\mathbf{t})}(\mathbf{x}) = 0$). Therefore, whenever all the parameters are computed approximately correctly, the displacement is also approximately correct - even if near the interface we use the vertex displacement instead of the line displacement or vice versa. We henceforth focus on $\Delta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})}, \Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})} \geq 0$.

Define

$$\alpha^{(\mathbf{a})} \triangleq \frac{\Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}}{\Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})} + \Delta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})}},$$

and finally also

$$\begin{aligned}\mathbf{z}^{(\mathcal{B})} &\triangleq \alpha\left(\mathbf{x}^{(\mathcal{B})}\right)\mathbf{z}_{(\mathbf{s}\to\mathbf{y})} + \left(1-\alpha\left(\mathbf{x}^{(\mathcal{B})}\right)\right)\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}\\ \mathbf{z}^{(\mathbf{a})} &\triangleq \alpha^{(\mathbf{a})}\mathbf{z}^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})} + \left(1-\alpha^{(\mathbf{a})}\right)\mathbf{z}^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}. \end{aligned} \tag{16.27}$$

We now consider two different cases depending on the value of $\left\|\mathbf{x}^{(\mathcal{B})} - \mathbf{z}^{(\mathcal{B})}\right\|_M$:

Case $\left\|\mathbf{x}^{(\mathcal{B})} - \mathbf{z}^{(\mathcal{B})}\right\|_M > 10h$:

The challenge is that when $\left\|\mathbf{x}^{(\mathcal{B})} - \mathbf{z}^{(\mathcal{B})}\right\|_M$ is huge, $\Delta_{(\mathbf{y}\to\mathbf{t})} + \Delta_{(\mathbf{s}\to\mathbf{y})}$ may be very small, which could lead to $\alpha^{(\mathbf{a})}$ being far from the true $\alpha\left(\mathbf{x}^{(\mathcal{B})}\right)$. Fortunately, in this case the true $\hat{g}\left(\mathbf{x}^{(\mathcal{B})}\right)$ is just the default displacement, $\delta\left(\mathbf{0}_{2m+1}, 1\right)$ - so we only have to argue that $f^{(\mathbf{a})}$ also applies the default displacement (for most players (A,r)).

Observe that if $\alpha\left(\mathbf{x}^{(\mathcal{B})}\right) = 1/2$, the point on $L_{\mathbf{y}}$ which is closest to $\mathbf{x}^{(\mathcal{B})}$ is indeed $\mathbf{z}^{(\mathcal{B})}$. In general, this is not true, but $\left\|\mathbf{x}^{(\mathcal{B})} - \mathbf{z}^{(\mathcal{B})}\right\|_M$ is at most $\sqrt{2}$-times larger than the distance from $\mathbf{x}^{(\mathcal{B})}$ to $L_{\mathbf{y}}$. In particular, for any $\mathbf{z}^{\left(\alpha^{(\mathbf{a})}\right)} \triangleq \alpha^{(\mathbf{a})}\mathbf{z}_{(\mathbf{s}\to\mathbf{y})} + \left(1-\alpha^{(\mathbf{a})}\right)\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}$, we have that $\left\|\mathbf{x}^{(\mathcal{B})} - \mathbf{z}^{\left(\alpha^{(\mathbf{a})}\right)}\right\|_M > 7h$.

Therefore by the Good Sample guarantee from Proposition 16.3.1 and Claim 16.4.5, for a $\left(1 - O\left(\epsilon_{\text{NASH}}\right)\right)$-fraction of (A,r)'s and every strategy $\mathbf{a}$ in their supports, we have

$$\left\|\mathbf{x}^{(\mathbf{a})} - \mathbf{z}^{(\mathbf{a})}\right\|_{J^{(A,r)}} > 7h - O\left(\epsilon_{\text{NASH}}\right);$$

For all those $\mathbf{a}$'s, $f^{(\mathbf{a})}$ correctly implements the default displacement.

Case $\left\|\mathbf{x}^{(\mathcal{B})}-\mathbf{z}^{(\mathcal{B})}\right\|_M \leq 10h$:

From (16.26),

$$\begin{aligned}\left|\Delta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})}-\Delta_{(\mathbf{s}\to\mathbf{y})}\left(\mathbf{x}^{(\mathcal{B})}\right)\right|^2 &= O\left(\epsilon_{\text{NASH}}\right)\\ \left|\Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}-\Delta_{(\mathbf{y}\to\mathbf{t})}\left(\mathbf{x}^{(\mathcal{B})}\right)\right|^2 &= O\left(\epsilon_{\text{NASH}}\right).\end{aligned}$$

Plugging into (4.5), we have that

$$\Delta^{(\mathbf{a})}_{(\mathbf{s}\to\mathbf{y})}+\Delta^{(\mathbf{a})}_{(\mathbf{y}\to\mathbf{t})}\geq\sqrt{h}-O\left(h\right),$$

and therefore also

$$\left|\alpha^{(\mathbf{a})}-\alpha\left(\mathbf{x}^{(\mathcal{B})}\right)\right|^2=O\left(\epsilon_{\text{NASH}}/h\right).$$

Similarly, also[6]

$$\left\|\mathbf{z}^{(\mathbf{a})}-\mathbf{z}^{(\mathcal{B})}\right\|^2_{J(A,r)}=O\left(\epsilon_{\text{NASH}}/h\right).$$

Finally, whenever $\mathbf{a}$ is ϵ_{NASH}-optimal and satisfies all the above, we have

$$\left\|f^{(\mathbf{a})}-f\left(\mathbf{x}^{(\mathcal{B})}\right)\right\|^2_{J(A,r)}=O\left(\epsilon_{\text{NASH}}/h\right).$$

□

16.5 From polymatrix to bimatrix

In this section we complete the proof of our main result, Theorem 1.3.1, by reducing from multiplayer polymatrix games (a-la Theorem 16.4.1) to two-player games.

16.5.1 From $\left(\sqrt{\epsilon}+\epsilon,\delta\right)$-Well-Supported-WeakNash to (ϵ,δ)-WeakNash

Lemma 16.5.1. *Consider a complete bipartite polymatrix game such that every two adjacent vertices play a bimatrix subgame with payoffs in $[0,1/n_B]$, $[0,1/n_A]$, where n_A,n_B are respective the numbers of players on the two sides of the bipartite game. Given an (ϵ,δ)-WeakNash, we can construct in polynomial time a $\left(\sqrt{\epsilon}\cdot\left(\sqrt{\epsilon}+5\right),\delta\right)$-Well-Supported-WeakNash.*

[6] In fact, we actually have $\left\|\mathbf{z}^{(\mathbf{a})}-\mathbf{z}^{(\mathcal{B})}\right\|^2_{J(A,r)}=O\left(\epsilon_{\text{NASH}}\right)$, because α interpolates between $\mathbf{z}_{(\mathbf{s}\to\mathbf{y})}$ and $\mathbf{z}_{(\mathbf{y}\to\mathbf{t})}$, which are already at distance $O\left(\sqrt{h}\right)$ from each other.

Proof. Let $(V_A; V_B)$ be the sets of players, where each $v \in V_A \cup V_B$ has utility U^v and action set $\mathcal{S}^v$. Let $\mathbf{x} = (x_s^v) \in \Delta\left(\times_{v \in V_A \cup V_B} \mathcal{S}^v\right)$ be an (ϵ, δ)-WeakNash. Finally, let $U^v_{\max}\left(\mathbf{x}^{-v}\right) = \max_{s \in \mathcal{S}^v} U^v_s\left(\mathbf{x}^{-v}\right)$. Since this is a polymatrix game, we can write

$$\begin{aligned} \forall v \in V_A \;\; U^v_s(\mathbf{x}) &= \sum_{u \in V_B} U^{v,u}_s\left(\mathbf{x}^u\right), \\ \forall v \in V_B \;\; U^v_s(\mathbf{x}) &= \sum_{u \in V_A} U^{v,u}_s\left(\mathbf{x}^u\right). \end{aligned}$$

Let $k = k(\epsilon) > 0$ be some large number do be specified later. We construct our new approximate equilibrium as follows. For each of the $(1-\delta)$ players who play ϵ-optimally in $\mathbf{x}$, we take only the strategies that are within ϵk of the optimum:

$$\hat{x}^v_s = \begin{cases} \frac{x^v_s}{1-z^v} & \text{if } U^v_s\left(\mathbf{x}^{-v}\right) \geq U^v_{\max}\left(\mathbf{x}^{-v}\right) - \epsilon k \\ 0 & \text{otherwise} \end{cases}$$

where z^v is the total probability that player v assigns to strategies that are more than ϵk away from the optimum.

The above division is well-defined because for $k > 1$ and v who plays ϵ-optimally, z^v is bounded away from 1. Moreover, the following claim from [DGP09] formalizes the intuition that when k is sufficiently large, the total weight on actions removed is small, so $\hat{\mathbf{x}}^v$ is close to $\mathbf{x}^v$:

Claim 16.5.2. ([DGP09, Claim 6])

$$\forall v \in V_A \cup V_B \quad \sum_{s \in \mathcal{S}^v} |\hat{x}^v_s - x^v_s| \leq \frac{2}{k-1}$$

Now, the total change to the expected payoff of player v for each action s, is bounded by the total change in mixed strategies of its neighbors. In the following let $v \in V_A$; the analogous argument for the utility of players in V_B follows with minor modifications.

$$\begin{aligned} \left|U^v_s\left(\mathbf{x}^{-v}\right) - U^v_s\left(\hat{\mathbf{x}}^{-v}\right)\right| &\leq \sum_{u \in V_B} \left|U^{v,u}_s\left(\mathbf{x}^u\right) - U^{v,u}_s\left(\hat{\mathbf{x}}^u\right)\right| \\ &\leq \frac{1}{n_B} \cdot \sum_{u \in V_B} \sum_{s \in \mathcal{S}^u} |\hat{x}^u_s - x^u_s| \leq \frac{2}{k-1} \end{aligned}$$

It follows that $\hat{\mathbf{x}}$ is a $\left(k\epsilon + \frac{2}{k-1}, \delta\right)$-Well-Supported-WeakNash:

$$U^v_s\left(\hat{\mathbf{x}}^{-v}\right) \geq U^v_s\left(\mathbf{x}^{-v}\right) - \frac{2}{k-1} \geq U^v_{\max}\left(\mathbf{x}^{-v}\right) - \epsilon k - \frac{2}{k-1} \geq U^v_{\max}\left(\hat{\mathbf{x}}^{-v}\right) - \epsilon k - \frac{4}{k-1}$$

Finally, take $k = 1 + 1/\sqrt{\epsilon}$ to get that

$$k\epsilon + \frac{4}{k-1} \leq \sqrt{\epsilon} \cdot \left(\sqrt{\epsilon} + 5\right)$$

□

16.5.2 From (ϵ, δ)-WeakNash to $\Theta(\epsilon \cdot \delta)$-ANE in a bimatrix game

Lemma 16.5.3. *Suppose we are given a complete bipartite polymatrix game between $n^{1/2+o(1)}$ vertices with $2^{n^{1/2+o(1)}}$ actions each; the payoffs in each bimatrix game are in $[0, 1/n_B], [0, 1/n_A]$, where n_A, n_B denote the number of players on each side of the bipartite graph. Then we can construct a bimatrix game of size $2^{n^{1/2+o(1)}}$ such that every ϵ-ANE of the new bimatrix game corresponds to a $(\delta, \epsilon_{\text{POLYMATRIX}})$-WeakNash of the polymatrix game, for sufficiently small $\epsilon = \Theta(\delta^2 \cdot \epsilon^2_{\text{POLYMATRIX}})$*

Proof. The two players play three games simultaneously: the main game, which is the heart of the reduction; and two games based on a construction due to Althofer [Alt94], which impose structural properties of any approximate Nash equilibrium.

Main game We let each of the two players "control" the vertices on one side of the bipartite graphical game.

Alice's actions correspond to a choice of vertex on her side of the polymatrix game, and an action for that player. Similarly, Bob's actions correspond to a a choice of vertex on his side, and a choice of action for that vertex. The utility for each player is the utility to the corresponding vertex from the induced subgame, scaled by a factor of $\lambda \cdot n_B$ or $\lambda \cdot n_A$ for Alice or Bob, respectively. Here $\lambda \gg \epsilon$ is a small constant do be defined later.

Althofer games In addition to the main game, we use a construction due to Althofer [Alt94] to introduce two auxiliary games that force the player to spread their mixed strategies approximately evenly across all vertices. Althofer's original game is a one-sum game with payoffs in $\{0, 1\}$ and size $k \times \binom{k}{k/2}$. In each column, exactly half of the entries are 1's, and the rest are 0's. For example for $k = 4$, the payoff matrices are given by:

$$R = 1 - C = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

The value of this game is 1/2.

For our purposes, we consider two instantiations of Althofer's gadget: a "primal" Althofer game of size $n_A \times \binom{n_A}{n_A/2}$. and a "dual" Althofer game of size $\binom{n_B}{n_B/2} \times n_B$. In any (approximate) Nash, both Alice and Bob must mix (approximately) evenly among (almost) all of their actions.

The final construction Finally, we compose all three games together by identifying Alice's choice of vertex in the main game with her choice of row in the primal Althofer game, and Bob's choice of vertex in the main game with his choice of column in the dual Althofer game.

Analysis Alice's and Bob's respective mixed strategies induce a mixed strategy profile $(\mathcal{A}, \mathcal{B})$ on their vertices: Alice's vertex (A, i)'s action is drawn from Alice's action conditioned on picking (A, i), and analogously for Bob's (B, j). For any vertex that is never picked by its player, fix an arbitrary strategy. Our goal is to show that $(\mathcal{A}, \mathcal{B})$ is a $(\delta, \epsilon_{\text{POLYMATRIX}})$-WeakNash of the polymatrix game.

Let $U^{(A,i)}\left(\mathcal{A}\left[(A,i)\right], \mathcal{B}\right)$ denote (A, i)'s expected utility when all players draw their strategies according to $(\mathcal{A}, \mathcal{B})$, and analogously for $U^{(B,j)}\left(\mathcal{B}\left[(B,j)\right], \mathcal{A}\right)$.

By Lemma 2.7.8, in every λ-ANE (and in particular in any ϵ-ANE), Alice's and Bob's respective marginal distributions over their vertices are $O(\lambda)$-close to uniform. Therefore, Alice's expected utilities from the main game satisfy

$$\left|U^A_{\text{MAIN}} - \lambda \mathsf{E}_{i\epsilon[n_A]} U^{(A,i)}\left(\mathcal{A}\left[(A,i)\right], \mathcal{B}\right)\right| \;=\; O\left(\lambda^2\right).$$

Suppose by contradiction that Alice and Bob are in an ϵ-ANE in the bimatrix game, but a δ-fraction of Alice's vertices have an $\epsilon_{\text{POLYMATRIX}}$-deviating strategy in the polymatrix game. Let $\left(\widehat{\mathcal{A}}, \mathcal{B}\right)$ denote the deviating mixed strategy profile (notice that the deviation of a player on Alice's side does not affect the utilities of other player on the same side). $\left(\widehat{\mathcal{A}}, \mathcal{B}\right)$ induces, without changing the marginal distributions over vertices, a profile of mixed strategies for Alice and Bob; Alice's new expected utility from the main game, $\widehat{U^A_{\text{MAIN}}}$, also satisfies

$$\left|\widehat{U^A_{\text{MAIN}}} - \lambda \mathsf{E}_{i\epsilon[n_A]} U^{(A,i)}\left(\widehat{\mathcal{A}}\left[(A,i)\right], \mathcal{B}\right)\right| \;=\; O\left(\lambda^2\right)$$

Therefore,

$$\begin{aligned}
\widehat{U^A_{\text{MAIN}}} &\geq \lambda \mathsf{E}_{i\epsilon[n_A]} U^{(A,i)}\left(\widehat{\mathcal{A}}\left[(A,i)\right], \widehat{\mathcal{B}}\right) - O\left(\lambda^2\right) \\
&\geq \lambda \mathsf{E}_{i\epsilon[n_A]} U^{(A,i)}\left(\mathcal{A}\left[(A,i)\right], \mathcal{B}\right) + \lambda\delta \cdot \epsilon_{\text{POLYMATRIX}} - O\left(\lambda^2\right) \\
&\geq U^A_{\text{MAIN}} + \lambda\delta \cdot \epsilon_{\text{POLYMATRIX}} - O\left(\lambda^2\right).
\end{aligned}$$

Set $\lambda\delta \cdot \epsilon_{\text{POLYMATRIX}} - O(\lambda^2) > \epsilon$. Since Alice and Bob have not changed their marginal vertex utilities, this is an ϵ-deviating strategy. □

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. ISBN: 978-0-521-42426-4. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

[ABC13] Per Austrin, Mark Braverman, and Eden Chlamtac. "Inapproximability of NP-Complete Variants of Nash Equilibrium". In: *Theory of Computing* 9 (2013), pp. 117–142. DOI: 10.4086/toc.2013.v009a003. URL: http://dx.doi.org/10.4086/toc.2013.v009a003.

[ABI86] Noga Alon, László Babai, and Alon Itai. "A Fast and Simple Randomized Parallel Algorithm for the Maximal Independent Set Problem". In: *J. Algorithms* 7.4 (1986), pp. 567–583. DOI: 10.1016/0196-6774(86)90019-2. URL: http://dx.doi.org/10.1016/0196-6774(86)90019-2.

[AIM14] Scott Aaronson, Russell Impagliazzo, and Dana Moshkovitz. "AM with Multiple Merlins". In: *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*. 2014, pp. 44–55. DOI: 10.1109/CCC.2014.13. URL: http://dx.doi.org/10.1109/CCC.2014.13.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. "Finding a Large Hidden Clique in a Random Graph". In: *SODA*. 1998, pp. 594–598.

[Alo+07] Noga Alon et al. "Testing k-wise and almost k-wise independence". In: *STOC*. 2007, pp. 496–505.

[Alo+11] Noga Alon et al. "Inapproximability of densest κ-subgraph from average case hardness". In: *Unpublished manuscript* (2011).

[Alo+13] Noga Alon et al. "The approximate rank of a matrix and its algorithmic applications: approximate rank". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 2013, pp. 675–684. DOI: 10.1145/2488608.2488694. URL: http://doi.acm.org/10.1145/2488608.2488694.

[Alo+92] Noga Alon et al. "Simple Construction of Almost k-wise Independent Random Variables". In: *Random Struct. Algorithms* 3.3 (1992), pp. 289–304. DOI: 10.1002/rsa.3240030308. URL: http://dx.doi.org/10.1002/rsa.3240030308.

[Alt94] Ingo Althofer. "On sparse approximations to randomized strategies and convex combinations". In: *Linear Algebra and its Applications* 199 (1994), pp. 339 –355.

[AMN98] Yossi Azar, Rajeev Motwani, and Joseph Naor. "Approximating Probability Distributions Using Small Sample Spaces". In: *Combinatorica* 18.2 (1998), pp. 151–171. DOI: 10.1007/PL00009813. URL: http://dx.doi.org/10.1007/PL00009813.

[Anb+13] Yogesh Anbalagan et al. "Polylogarithmic Supports Are Required for Approximate Well-Supported Nash Equilibria below 2/3". In: *Web and Internet Economics - 9th International Conference, WINE 2013, Cambridge, MA, USA, December 11-14, 2013, Proceedings*. 2013, pp. 15–23. DOI: 10.1007/978-3-642-45046-4_2. URL: http://dx.doi.org/10.1007/978-3-642-45046-4_2.

[Anb+15] Yogesh Anbalagan et al. "Large Supports are Required for Well-Supported Nash Equilibria". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*. 2015, pp. 78–84. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2015.78. URL: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2015.78.

[Ans+17] Anurag Anshu et al. "Lifting randomized query complexity to randomized communication complexity". In: *arXiv preprint arXiv:1703.07521* (2017).

[Aro+12] Sanjeev Arora et al. "Finding overlapping communities in social networks: toward a rigorous approach". In: *ACM Conference on Electronic Commerce, EC '12, Valencia, Spain, June 4-8, 2012*. 2012, pp. 37–54. DOI: 10.1145/2229012.2229020. URL: http://doi.acm.org/10.1145/2229012.2229020.

[Aro+98] Sanjeev Arora et al. "Proof Verification and the Hardness of Approximation Problems". In: *J. ACM* 45.3 (1998), pp. 501–555. DOI: 10.1145/278298.278306. URL: http://doi.acm.org/10.1145/278298.278306.

[AS98] Sanjeev Arora and Shmuel Safra. "Probabilistic Checking of Proofs: A New Characterization of NP". In: *J. ACM* 45.1 (1998), pp. 70–122. DOI: 10.1145/273865.273901. URL: http://doi.acm.org/10.1145/273865.273901.

[Ast+15] Megasthenis Asteris et al. "Sparse PCA via Bipartite Matchings". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 2015, pp. 766–774. URL: http://papers.nips.cc/paper/5901-sparse-pca-via-bipartite-matchings.

[Aum87] Robert J. Aumann. "game theory". In: *The New Palgrave: A Dictionary of Economics*. Ed. by John Eatwell, Murray Milgate, and Peter Newman. Basingstoke: Palgrave Macmillan, 1987.

[AUY83] Alfred V. Aho, Jeffrey D. Ullman, and Mihalis Yannakakis. "On Notions of Information Transfer in VLSI Circuits". In: *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*. 1983, pp. 133–139. DOI: 10.1145/800061.808742. URL: http://doi.acm.org/10.1145/800061.808742.

[BA08] Shai Ben-David and Margareta Ackerman. "Measures of Clustering Quality: A Working Set of Axioms for Clustering". In: *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*. 2008, pp. 121–128. URL: http://papers.nips.cc/paper/3491-measures-of-clustering-quality-a-working-set-of-axioms-for-clustering.

[Bab+91] László Babai et al. "Checking Computations in Polylogarithmic Time". In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*. 1991, pp. 21–31. DOI: 10.1145/103418.103428. URL: http://doi.acm.org/10.1145/103418.103428.

[Bab12] Yakov Babichenko. "Completely uncoupled dynamics and Nash equilibria". In: *Games and Economic Behavior* 76.1 (2012), pp. 1–14. DOI: 10.1016/j.geb.2012.06.004. URL: http://dx.doi.org/10.1016/j.geb.2012.06.004.

[Bab16] Yakov Babichenko. "Query Complexity of Approximate Nash Equilibria". In: vol. 63. 4. 2016, 36:1–36:24. DOI: 10.1145/2908734. URL: http://doi.acm.org/10.1145/2908734.

[Bad+16] Ashwinkumar Badanidiyuru et al. "Locally Adaptive Optimization: Adaptive Seeding for Monotone Submodular Functions". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. 2016, pp. 414–429. DOI: 10.1137/1.9781611974331.ch31. URL: http://dx.doi.org/10.1137/1.9781611974331.ch31.

[Bal+13] Maria-Florina Balcan et al. "Finding Endogenously Formed Communities". In: *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*. 2013, pp. 767–783. DOI: 10.1137/1.9781611973105.55. URL: http://dx.doi.org/10.1137/1.9781611973105.55.

[Ban+16] Jess Banks et al. "Information-theoretic thresholds for community detection in sparse networks". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 383–416. URL: http://jmlr.org/proceedings/papers/v49/banks16.html.

[Bar+16] Boaz Barak et al. "A Nearly Tight Sum-of-Squares Lower Bound for the Planted Clique Problem". In: *FOCS*. 2016.

[Bar04] Boaz Barak. "Non-Black-Box Techniques in Cryptography". PhD thesis. Weizmann Institute of Science, 2004.

[Bar15] Siddharth Barman. "Approximating Nash Equilibria and Dense Bipartite Subgraphs via an Approximate Version of Caratheodory's Theorem". In: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*. 2015, pp. 361–369. DOI: 10.1145/2746539.2746566. URL: http://doi.acm.org/10.1145/2746539.2746566.

[BB15] Yakov Babichenko and Siddharth Barman. "Query Complexity of Correlated Equilibrium". In: *ACM Trans. Economics and Comput.* 3.4 (2015), p. 22. DOI: 10.1145/2785668. URL: http://doi.acm.org/10.1145/2785668.

[BBM10] Hartwig Bosse, Jaroslaw Byrka, and Evangelos Markakis. "New algorithms for approximate Nash equilibria in bimatrix games". In: *Theor. Comput. Sci.* 411.1 (2010), pp. 164–173. DOI: 10.1016/j.tcs.2009.09.023. URL: http://dx.doi.org/10.1016/j.tcs.2009.09.023.

[BBV16] Afonso S. Bandeira, Nicolas Boumal, and Vladislav Voroninski. "On the low-rank approach for semidefinite programs arising in synchronization and community detection". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 361–382. URL: http://jmlr.org/proceedings/papers/v49/bandeira16.html.

[BDX11] Benjamin E. Birnbaum, Nikhil R. Devanur, and Lin Xiao. "Distributed algorithms via gradient descent for fisher markets". In: *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011), San Jose, CA, USA, June 5-9, 2011*. 2011, pp. 127–136. DOI: 10.1145/1993574.1993594. URL: http://doi.acm.org/10.1145/1993574.1993594.

[Bea+98] Paul Beame et al. "The Relative Complexity of NP Search Problems". In: *J. Comput. Syst. Sci.* 57.1 (1998), pp. 3–19. DOI: 10.1006/jcss.1998.1575. URL: http://dx.doi.org/10.1006/jcss.1998.1575.

[Ben+03] Eli Ben-Sasson et al. "Randomness-efficient low degree tests and short PCPs via epsilon-biased sets". In: *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*. 2003, pp. 612–621. DOI: 10.1145/780542.780631. URL: http://doi.acm.org/10.1145/780542.780631.

[Ben+06] Eli Ben-Sasson et al. "Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding". In: *SIAM J. Comput.* 36.4 (2006), pp. 889–974. DOI: 10.1137/S0097539705446810. URL: http://dx.doi.org/10.1137/S0097539705446810.

[BF91] László Babai and Lance Fortnow. "Arithmetization: A New Method in Structural Complexity Theory". In: *Computational Complexity* 1 (1991), pp. 41–66. DOI: 10.1007/BF01200057. URL: http://dx.doi.org/10.1007/BF01200057.

[BFS16] Cristina Bazgan, Florent Foucaud, and Florian Sikora. "On the Approximability of Partial VC Dimension". In: *Combinatorial Optimization and Applications - 10th International Conference, COCOA 2016, Hong Kong, China, December 16-18, 2016, Proceedings*. 2016, pp. 92–106. DOI: 10.1007/978-3-319-48749-6_7. URL: http://dx.doi.org/10.1007/978-3-319-48749-6_7.

[Bha+10] Aditya Bhaskara et al. "Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph". In: *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. 2010, pp. 201–210. DOI: 10.1145/1806689.1806718. URL: http://doi.acm.org/10.1145/1806689.1806718.

[Bha+12] Aditya Bhaskara et al. "Polynomial Integrality Gaps for Strong SDP Relaxations of Densest K-subgraph". In: *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '12. Kyoto, Japan: SIAM, 2012, pp. 388–405. URL: http://dl.acm.org/citation.cfm?id=2095116.2095150.

[Bha+16] Umang Bhaskar et al. "Hardness Results for Signaling in Bayesian Zero-Sum and Network Routing Games". In: *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*. 2016, pp. 479–496. DOI: 10.1145/2940716.2940753. URL: http://doi.acm.org/10.1145/2940716.2940753.

[BKK91] Raymond C. Battalio, John H. Kagel, and Carl A. Kogut. "Experimental Confirmation of the Existence of a Giffen Good". In: *The American Economic Review* 81.4 (1991), pp. 961–970.

[BKW15] Mark Braverman, Young Kun-Ko, and Omri Weinstein. "Approximating the best Nash Equilibrium in $n^{o(\log n)}$-time breaks the Exponential Time Hypothesis". In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. 2015, pp. 970–982. DOI: 10.1137/1.9781611973730.66. URL: http://dx.doi.org/10.1137/1.9781611973730.66.

[BLP15] Siddharth Barman, Katrina Ligett, and Georgios Piliouras. "Approximating Nash Equilibria in Tree Polymatrix Games". In: *Algorithmic Game Theory - 8th International Symposium, SAGT 2015, Saarbrücken, Germany, September 28-30, 2015, Proceedings*. 2015, pp. 285–296. DOI: 10.1007/978-3-662-48433-3_22. URL: http://dx.doi.org/10.1007/978-3-662-48433-3_22.

[Blu94] Avrim Blum. "Separating Distribution-Free and Mistake-Bound Learning Models over the Boolean Domain". In: *SIAM J. Comput.* 23.5 (1994), pp. 990–1000. DOI: 10.1137/S009753979223455X. URL: http://dx.doi.org/10.1137/S009753979223455X.

[Bor+16] Christian Borgs et al. "An Axiomatic Approach to Community Detection". In: *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*. 2016, pp. 135–146. DOI: 10.1145/2840728.2840748. URL: http://doi.acm.org/10.1145/2840728.2840748.

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. "On the Cryptographic Hardness of Finding a Nash Equilibrium". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. 2015, pp. 1480–1498. DOI: 10.1109/FOCS.2015.94. URL: https://doi.org/10.1109/FOCS.2015.94.

[BPR16] Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinstein. "Can Almost Everybody be Almost Happy?" In: *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*. 2016, pp. 1–9. DOI: 10.1145/2840728.2840731. URL: http://doi.acm.org/10.1145/2840728.2840731.

[BPS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. "Agnostic Online Learning". In: *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*. 2009. URL: http://www.cs.mcgill.ca/~colt2009/papers/032.pdf#page=1.

[BR13] Quentin Berthet and Philippe Rigollet. "Complexity theoretic lower bounds for sparse principal component detection". In: *Conference on Learning Theory*. 2013, pp. 1046–1066.

[Bra+17] Mark Braverman et al. "ETH Hardness for Densest-k-Subgraph with Perfect Completeness". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*. 2017, pp. 1326–1341. DOI: 10.1137/1.9781611974782.86. URL: http://dx.doi.org/10.1137/1.9781611974782.86.

[Bro51] George W. Brown. "Iterative Solutions of Games by Fictitious Play". In: *Activity Analysis of Production and Allocation* (1951).

[BS00] William C. Brainard and Herbet Scarf. *How to Compute Equilibrium Prices in 1891*. Cowles Foundation Discussion Papers 1272. Cowles Foundation for Research in Economics, Yale University, 2000.

[Bud+14] Eric Budish et al. "Course Match: A Large-Scale Implementation of Approximate Competitive Equilibrium from Equal Incomes for Combinatorial Allocation". Working paper. 2014.

[Bud11] Eric Budish. "The Combinatorial Assignment Problem: Approximate Competitive Equilibrium from Equal Incomes". In: *Journal of Political Economy* 119.6 (2011), pp. 1061 –1103. URL: http://EconPapers.repec.org/RePEc:ucp:jpolec:doi:10.1086/664613.

[Car+16] Marco L. Carmosino et al. "Nondeterministic Extensions of the Strong Exponential Time Hypothesis and Consequences for Non-reducibility". In: *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*. 2016, pp. 261–270. DOI: 10.1145/2840728.2840746. URL: http://doi.acm.org/10.1145/2840728.2840746.

[CCT17] Xi Chen, Yu Cheng, and Bo Tang. "Well-Supported versus Approximate Nash Equilibria: Query Complexity of Large Games". In: *To appear in ITCS*. 2017.

[CD09] Xi Chen and Xiaotie Deng. "On the complexity of 2D discrete fixed point problem". In: *Theor. Comput. Sci.* 410.44 (2009), pp. 4448–4456. DOI: 10.1016/j.tcs.2009.07.052. URL: https://doi.org/10.1016/j.tcs.2009.07.052.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. "Settling the complexity of computing two-player Nash equilibria". In: *J. ACM* 56.3 (2009).

[CF08] Richard Cole and Lisa Fleischer. "Fast-converging tatonnement algorithms for one-time and ongoing market problems". In: *STOC*. 2008, pp. 315–324.

[Che+09] Xi Chen et al. "Settling the Complexity of Arrow-Debreu Equilibria in Markets with Additively Separable Utilities". In: *FOCS*. 2009, pp. 273–282.

[Che+15a] Wei Chen et al. "Combining Traditional Marketing and Viral Marketing with Amphibious Influence Maximization". In: *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*. 2015, pp. 779–796. DOI: 10.1145/2764468.2764480. URL: http://doi.acm.org/10.1145/2764468.2764480.

[Che+15b] Yu Cheng et al. "Mixture Selection, Mechanism Design, and Signaling". In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. 2015, pp. 1426–1445. DOI: 10.1109/FOCS.2015.91. URL: http://dx.doi.org/10.1109/FOCS.2015.91.

[Che+16] Yuxin Chen et al. "Community Recovery in Graphs with Locality". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. 2016, pp. 689–698. URL: http://jmlr.org/proceedings/papers/v48/chena16.html.

[CIP09] Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. "The Complexity of Satisfiability of Small Depth Circuits". In: *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*. 2009, pp. 75–85. DOI: 10.1007/978-3-642-11269-0_6. URL: http://dx.doi.org/10.1007/978-3-642-11269-0_6.

[CMV05] Bruno Codenotti, Benton McCune, and Kasturi Varadarajan. "Market Equilibrium via the Excess Demand Function". In: *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*. STOC '05. Baltimore, MD, USA: ACM, 2005, pp. 74–83. ISBN: 1-58113-960-8. DOI: 10.1145/1060590.1060601. URL: http://doi.acm.org/10.1145/1060590.1060601.

[CPP16] Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. "Known Algorithms for Edge Clique Cover are Probably Optimal". In: *SIAM J. Comput.* 45.1 (2016), pp. 67–83. DOI: 10.1137/130947076. URL: http://dx.doi.org/10.1137/130947076.

[CPR16] Siu On Chan, Dimitris Papailliopoulos, and Aviad Rubinstein. "On the Approximability of Sparse PCA". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 623–646. URL: http://jmlr.org/proceedings/papers/v49/chan16.html.

[CPY13] Xi Chen, Dimitris Paparas, and Mihalis Yannakakis. "The complexity of non-monotone markets". In: *STOC*. Preprint of full version is available at http://arxiv.org/abs/1211.4918v1. 2013, pp. 181–190.

[CS04] Vincent Conitzer and Tuomas Sandholm. "Communication complexity as a lower bound for learning in games". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 24.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[Czu+15] Artur Czumaj et al. "Distributed Methods for Computing Approximate Equilibria". In: *CoRR* abs/1512.03315 (2015). URL: http://arxiv.org/abs/1512.03315.

[DA54] Gerard Debreu and Kenneth J. Arrow. "Existence of an Equilibrium for a Competitive Economy". In: *Econometrica* 22.3 (1954), pp. 265–90.

[Dan16] Amit Daniely. "Complexity theoretic limitations on learning halfspaces". In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. 2016, pp. 105–117. DOI: 10.1145/2897518.2897520. URL: http://doi.acm.org/10.1145/2897518.2897520.

[Das08] Constantinos Daskalakis. "The Complexity of Nash Equilibria". PhD thesis. University of California, Berkeley, 2008.

[Das13] Constantinos Daskalakis. "On the Complexity of Approximating a Nash Equilibrium". In: *ACM Transactions on Algorithms* 9.3 (2013), p. 23.

[DD08] Xiaotie Deng and Ye Du. "The computation of approximate competitive equilibrium is PPAD-hard". In: *Inf. Process. Lett.* 108.6 (2008), pp. 369–373. DOI: 10.1016/j.ipl.2008.07.011. URL: https://doi.org/10.1016/j.ipl.2008.07.011.

[Del+14] Holger Dell et al. "Exponential Time Complexity of the Permanent and the Tutte Polynomial". In: *ACM Trans. Algorithms* 10.4 (2014), 21:1–21:32. DOI: 10.1145/2635812. URL: http://doi.acm.org/10.1145/2635812.

[Del+17] Argyrios Deligkas et al. "Computing Approximate Nash Equilibria in Polymatrix Games". In: vol. 77. 2. 2017, pp. 487–514. DOI: 10.1007/s00453-015-0078-7. URL: http://dx.doi.org/10.1007/s00453-015-0078-7.

[Dev+08] Nikhil R. Devanur et al. "Market Equilibrium via a Primal–dual Algorithm for a Convex Program". In: *J. ACM* 55.5 (Nov. 2008), 22:1–22:18. ISSN: 0004-5411. DOI: 10.1145/1411509.1411512. URL: http://doi.acm.org/10.1145/1411509.1411512.

[DGGP10] Yael Dekel, Ori Gurel-Gurevich, and Yuval Peres. "Finding Hidden Cliques in Linear Time with High Probability". In: *CoRR* abs/1010.2997 (2010).

[DGP09] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. "The complexity of computing a Nash equilibrium". In: *SIAM Journal on Computing* 39.1 (2009), pp. 195–259.

[DH13] Irit Dinur and Prahladh Harsha. "Composition of Low-Error 2-Query PCPs Using Decodable PCPs". In: *SIAM J. Comput.* 42.6 (2013), pp. 2452–2486. DOI: 10.1137/100788161. URL: http://dx.doi.org/10.1137/100788161.

[Din07] Irit Dinur. "The PCP theorem by gap amplification". In: *J. ACM* 54.3 (2007), p. 12. DOI: 10.1145/1236457.1236459. URL: http://doi.acm.org/10.1145/1236457.1236459.

[DIR13] Shaddin Dughmi, Nicole Immorlica, and Aaron Roth. "Constrained signaling for welfare and revenue maximization". In: *SIGecom Exchanges* 12.1 (2013), pp. 53–56. DOI: 10.1145/2509013.2509022. URL: http://doi.acm.org/10.1145/2509013.2509022.

[DM15] Yash Deshpande and Andrea Montanari. "Improved Sum-of-Squares Lower Bounds for Hidden Clique and Hidden Submatrix Problems". In: *CoRR* abs/1502.06590 (2015). URL: http://arxiv.org/abs/1502.06590.

[DMP07] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. "Progress in approximate nash equilibria". In: *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*. 2007, pp. 355–358. DOI: 10.1145/1250910.1250962. URL: http://doi.acm.org/10.1145/1250910.1250962.

[DMP09] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. "A note on approximate Nash equilibria". In: *Theor. Comput. Sci.* 410.17 (2009), pp. 1581–1588. DOI: 10.1016/j.tcs.2008.12.031. URL: https://doi.org/10.1016/j.tcs.2008.12.031.

[DP09] Constantinos Daskalakis and Christos H. Papadimitriou. "On oblivious PTAS's for nash equilibrium". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. Full version available at http://arxiv.org/abs/1102.2280. 2009, pp. 75–84. DOI: 10.1145/1536414.1536427. URL: http://doi.acm.org/10.1145/1536414.1536427.

[DP15] Constantinos Daskalakis and Christos H. Papadimitriou. "Approximate Nash equilibria in anonymous games". In: *J. Economic Theory* 156 (2015), pp. 207–245. DOI: 10.1016/j.jet.2014.02.002. URL: http://dx.doi.org/10.1016/j.jet.2014.02.002.

[DS16] Amit Daniely and Shai Shalev-Shwartz. "Complexity Theoretic Limitations on Learning DNF's". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 815–830. URL: http://jmlr.org/proceedings/papers/v49/daniely16.html.

[Dug14] Shaddin Dughmi. "On the Hardness of Signaling". In: *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. 2014, pp. 354–363. DOI: 10.1109/FOCS.2014.45. URL: http://dx.doi.org/10.1109/FOCS.2014.45.

[Edg09] Francis Ysidro Edgeworth. "Review of Free Trade in Being". In: *Economic Journal* 19 (1909), pp. 104–5.

[Eme+14] Yuval Emek et al. "Signaling Schemes for Revenue Maximization". In: *ACM Trans. Economics and Comput.* 2.2 (2014), p. 5. DOI: 10.1145/2594564. URL: http://doi.acm.org/10.1145/2594564.

[EY10] Kousha Etessami and Mihalis Yannakakis. "On the Complexity of Nash Equilibria and Other Fixed Points". In: *SIAM J. Comput.* 39.6 (2010), pp. 2531–2597. DOI: 10.1137/080720826. URL: https://doi.org/10.1137/080720826.

[Fea+13] John Fearnley et al. "Learning equilibria of games via payoff queries." In: *EC*. 2013, pp. 397–414.

[Fei+96] Uriel Feige et al. "Interactive proofs and the hardness of approximating cliques". In: *Journal of the ACM (JACM)* 43.2 (1996), pp. 268–292.

[Fei02] Uriel Feige. "Relations between average case complexity and approximation complexity". In: *STOC*. Montreal, Quebec, Canada: ACM Press, 2002, pp. 534–543. ISBN: 1-58113-495-9. DOI: http://doi.acm.org/10.1145/509907.509985.

[Fel+06] Vitaly Feldman et al. "New Results for Learning Noisy Parities and Halfspaces". In: *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*. 2006, pp. 563–574. DOI: 10.1109/FOCS.2006.51. URL: http://dx.doi.org/10.1109/FOCS.2006.51.

[Fel+13] Vitaly Feldman et al. "Statistical algorithms and a lower bound for detecting planted cliques". In: *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 2013, pp. 655–664. DOI: 10.1145/2488608.2488692. URL: http://doi.acm.org/10.1145/2488608.2488692.

[FK00] Uriel Feige and Robert Krauthgamer. "Finding and certifying a large hidden clique in a semirandom graph". In: *Random Struct. Algorithms* 16.2 (2000), pp. 195–208.

[FKP01] Uriel Feige, Guy Kortsarz, and David Peleg. "The Dense k-Subgraph Problem". In: *Algorithmica* 29.3 (2001), pp. 410–421. DOI: 10.1007/s004530010050. URL: http://dx.doi.org/10.1007/s004530010050.

[FKS95] Joan Feigenbaum, Daphne Koller, and Peter W. Shor. "A Game-Theoretic Classification of Interactive Complexity Classes". In: *Structure in Complexity Theory Conference*. 1995, pp. 227–237.

[FL98] Moti Frances and Ami Litman. "Optimal Mistake Bound Learning is Hard". In: *Inf. Comput.* 144.1 (1998), pp. 66–82. DOI: 10.1006/inco.1998.2709. URL: http://dx.doi.org/10.1006/inco.1998.2709.

[FNS07] Tomás Feder, Hamid Nazerzadeh, and Amin Saberi. "Approximating nash equilibria using small-support strategies". In: *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*. 2007, pp. 352–354. DOI: 10.1145/1250910.1250961. URL: http://doi.acm.org/10.1145/1250910.1250961.

[Fol67] D.K. Foley. "Resource Allocation and the Public Sector". In: *Yale Economic Essays* 7.1 (1967), pp. 45–98.

[For+08] Lance Fortnow et al. "On the Complexity of Succinct Zero-Sum Games". In: *Computational Complexity* 17.3 (2008), pp. 353–376.

[For10] Santo Fortunato. "Community detection in graphs". In: *Physics Reports* 486 (2010), pp. 75–174.

[FP16] Laura Florescu and Will Perkins. "Spectral thresholds in the bipartite stochastic block model". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 943–959. URL: http://jmlr.org/proceedings/papers/v49/florescu16.html.

[FS97] Uriel Feige and Michael Seltser. *On the densest k-subgraph problem*. Citeseer, 1997.

[FY06] Dean P. Foster and H. Peyton Young. "Regret testing: learning to play Nash equilibrium without knowing you have an opponent". In: *Theoretical Economics* 1.3 (Sept. 2006), pp. 341–367. URL: http://econtheory.org/ojs/index.php/te/article/view/20060341.

[Gar+15] Jugal Garg et al. "A Complementary Pivot Algorithm for Market Equilibrium under Separable, Piecewise-Linear Concave Utilities". In: *SIAM J. Comput.* 44.6 (2015), pp. 1820–1847. DOI: 10.1137/140971002. URL: https://doi.org/10.1137/140971002.

[Gar+17] Jugal Garg et al. "Settling the complexity of Leontief and PLC exchange markets under exact and approximate equilibria". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. 2017, pp. 890–901. DOI: 10.1145/3055399.3055474. URL: http://doi.acm.org/10.1145/3055399.3055474.

[Gho+11] Ali Ghodsi et al. "Dominant Resource Fairness: Fair Allocation of Multiple Resource Types". In: *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*. NSDI'11. Boston, MA: USENIX Association, 2011, pp. 323–336. URL: http://dl.acm.org/citation.cfm?id=1972457.1972490.

[GK06] Rahul Garg and Sanjiv Kapoor. "Auction Algorithms for Market Equilibrium". In: *Math. Oper. Res.* 31.4 (2006), pp. 714–729. DOI: 10.1287/moor.1060.0216. URL: https://doi.org/10.1287/moor.1060.0216.

[GL07] Fabrizio Germano and Gabor Lugosi. "Global Nash convergence of Foster and Young's regret testing". In: *Games and Economic Behavior* 60.1 (2007), pp. 135–154.

[Gol10] Oded Goldreich, ed. *Property Testing - Current Research and Surveys [outgrow of a workshop at the Institute for Computer Science (ITCS) at Tsinghua University, January 2010]*. Vol. 6390. Lecture Notes in Computer Science. Springer, 2010. ISBN: 978-3-642-16366-1. DOI: 10.1007/978-3-642-16367-8. URL: http://dx.doi.org/10.1007/978-3-642-16367-8.

[GP14] Paul W Goldberg and Arnoud Pastink. "On the communication complexity of approximate Nash equilibria". In: *Games and Economic Behavior* 85 (2014), pp. 19–31.

[GPS16] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. "Revisiting the Cryptographic Hardness of Finding a Nash Equilibrium". In: *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*. 2016, pp. 579–604. DOI: 10.1007/978-3-662-53008-5_20. URL: https://doi.org/10.1007/978-3-662-53008-5_20.

[GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. "Query-to-communication lifting for BPP". In: *arXiv preprint arXiv:1703.07666* (2017).

[GR05] Venkatesan Guruswami and Atri Rudra. "Tolerant Locally Testable Codes". In: *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*. 2005, pp. 306–317. DOI: 10.1007/11538462_26. URL: http://dx.doi.org/10.1007/11538462_26.

[GR16] Paul W. Goldberg and Aaron Roth. "Bounds for the Query Complexity of Approximate Equilibria". In: *ACM Trans. Economics and Comput.* 4.4 (2016), 24:1–24:25. DOI: 10.1145/2956582. URL: http://doi.acm.org/10.1145/2956582.

[GS17] Anat Ganor and Karthik C. Srikanta. "Communication complexity of correlated equilibrium in two-player games". In: *arXiv preprint arXiv:1704.01104* (2017).

[GZ89] I. Gilboa and E. Zemel. "Nash and correlated equilibria: Some complexity considerations." In: Games and Economic Behavior, 1989.

[HK11] Elad Hazan and Robert Krauthgamer. "How Hard Is It to Approximate the Best Nash Equilibrium?" In: *SIAM J. Comput.* 40.1 (2011), pp. 79–91.

[HLL83] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. "Stochastic blockmodels: First steps". In: *Social Networks* 5 (1983), pp. 109–137.

[HM10] Sergiu Hart and Yishay Mansour. "How long to equilibrium? The communication complexity of uncoupled equilibrium procedures". In: *Games and Economic Behavior* 69.1 (2010), pp. 107–126.

[HMC03] Sergiu Hart and Andreu Mas-Colell. "Uncoupled dynamics do not lead to Nash equilibrium". In: *American Economic Review* 93.5 (2003), pp. 1830–1836.

[HMC06] Sergiu Hart and Andreu Mas-Colell. "Stochastic uncoupled dynamics and Nash equilibrium". In: *Games and Economic Behavior* 57.2 (2006), pp. 286–303.

[HMC13] Sergiu Hart and Andreu Mas-Colell. *Simple adaptive strategies: from regret-matching to uncoupled dynamics*. Vol. 4. World Scientific, 2013.

[HN13] Sergiu Hart and Noam Nisan. "The Query Complexity of Correlated Equilibria". In: *CoRR* abs/1305.4874 (2013).

[Hop+16] Samuel B. Hopkins et al. "On the Integrality Gap of Degree-4 Sum of Squares for Planted Clique". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. 2016, pp. 1079–1095. DOI: 10.1137/1.9781611974331.ch76. URL: http://dx.doi.org/10.1137/1.9781611974331.ch76.

[HPV89] Michael D. Hirsch, Christos H. Papadimitriou, and Stephen A. Vavasis. "Exponential lower bounds for finding Brouwer fix points". In: *J. Complexity* 5.4 (1989), pp. 379–416.

[HWX16] Bruce E. Hajek, Yihong Wu, and Jiaming Xu. "Semidefinite Programs for Exact Recovery of a Hidden Community". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 1051–1095. URL: http://jmlr.org/proceedings/papers/v49/hajek16.html.

[HY17] Pavel Hubácek and Eylon Yogev. "Hardness of Continuous Local Search: Query Complexity and Cryptographic Lower Bounds". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*. 2017, pp. 1352–1371. DOI: 10.1137/1.9781611974782.88. URL: https://doi.org/10.1137/1.9781611974782.88.

[Hås99] Johan Håstad. "Clique is Hard to Approximate Within n$^{1-epsilon}$". In: *Acta Mathematica* 182.1 (1999), pp. 105–142.

[Iem16] Rosalie Iemhoff. "Intuitionism in the Philosophy of Mathematics". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University, 2016.

[IP01] Russell Impagliazzo and Ramamohan Paturi. "On the Complexity of k-SAT". In: *J. Comput. Syst. Sci.* 62.2 (2001), pp. 367–375.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. "Which Problems Have Strongly Exponential Complexity?" In: *J. Comput. Syst. Sci.* 63.4 (2001), pp. 512–530. DOI: 10.1006/jcss.2001.1774. URL: http://dx.doi.org/10.1006/jcss.2001.1774.

[Jai04] Kamal Jain. "A Polynomial Time Algorithm for Computing the Arrow-Debreu Market Equilibrium for Linear Utilities". In: *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*. 2004, pp. 286–294. DOI: 10.1109/FOCS.2004.6. URL: http://dx.doi.org/10.1109/FOCS.2004.6.

[Jer92] Mark Jerrum. "Large Cliques Elude the Metropolis Process". In: *Random Struct. Algorithms* 3.4 (1992), pp. 347–360.

[Jev66] William Stanley Jevons. "Brief Account of a General Mathematical Theory of Political Economy". In: *History of Economic Thought Articles* 29 (1866), pp. 282–287. URL: http://EconPapers.repec.org/RePEc:hay:hetart:jevons1866.

[JLB15] Albert Xin Jiang and Kevin Leyton-Brown. "Polynomial-time computation of exact correlated equilibrium in compact games". In: *Games and Economic Behavior* 91 (2015), pp. 347–359.

[JV10] Kamal Jain and Vijay V. Vazirani. "Eisenberg-Gale markets: Algorithms and game-theoretic properties". In: *Games and Economic Behavior* 70.1 (2010), pp. 84–106. DOI: 10.1016/j.geb.2008.11.011. URL: https://doi.org/10.1016/j.geb.2008.11.011.

[Kal+08] Adam Tauman Kalai et al. "Agnostically Learning Halfspaces". In: *SIAM J. Comput.* 37.6 (2008), pp. 1777–1805. DOI: 10.1137/060649057. URL: http://dx.doi.org/10.1137/060649057.

[Kar72] Richard M. Karp. "Reducibility Among Combinatorial Problems". In: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.* 1972, pp. 85–103. URL: http://www.cs.berkeley.edu/~luca/cs172/karp.pdf.

[Kea07] Michael Kearns. "Graphical games". In: *Algorithmic Game Theory*. Ed. by Noam Nisan et al. Cambridge University Press, 2007. Chap. 7, pp. 159–180.

[Kha93] Michael Kharitonov. "Cryptographic hardness of distribution-specific learning". In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*. 1993, pp. 372–381. DOI: 10.1145/167088.167197. URL: http://doi.acm.org/10.1145/167088.167197.

[Kha95] Michael Kharitonov. "Cryptographic Lower Bounds for Learnability of Boolean Functions on the Uniform Distribution". In: *J. Comput. Syst. Sci.* 50.3 (1995), pp. 600–610. DOI: 10.1006/jcss.1995.1046. URL: http://dx.doi.org/10.1006/jcss.1995.1046.

[Kho01] Subhash Khot. "Improved Inaproximability Results for MaxClique, Chromatic Number and Approximate Graph Coloring". In: *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. 2001, pp. 600–609. DOI: 10.1109/SFCS.2001.959936. URL: http://dx.doi.org/10.1109/SFCS.2001.959936.

[Kho06] Subhash Khot. "Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique". In: *SIAM Journal on Computing* 36.4 (2006), pp. 1025–1071.

[KL93] Ehud Kalai and Ehud Lehrer. "Rational Learning Leads to Nash Equilibrium". In: *Econometrica* 61.5 (1993), pp. 1019–1045.

[Kle02] Jon M. Kleinberg. "An Impossibility Theorem for Clustering". In: *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*. 2002, pp. 446–453. URL: http://papers.nips.cc/paper/2340-an-impossibility-theorem-for-clustering.

[Kli16] Adam R. Klivans. "Cryptographic Hardness of Learning". In: *Encyclopedia of Algorithms*. 2016, pp. 475–477. DOI: 10.1007/978-1-4939-2864-4_96. URL: http://dx.doi.org/10.1007/978-1-4939-2864-4_96.

[KPS09] Spyros C. Kontogiannis, Panagiota N. Panagopoulou, and Paul G. Spirakis. "Polynomial algorithms for approximating Nash equilibria of bimatrix games". In: *Theor. Comput. Sci.* 410.17 (2009), pp. 1599–1606. DOI: 10.1016/j.tcs.2008.12.033. URL: http://dx.doi.org/10.1016/j.tcs.2008.12.033.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. "Super-logarithmic depth lower bounds via the direct sum in communication complexity". In: *Computational Complexity* 5.3-4 (1995), pp. 191–204.

[KS09] Adam R. Klivans and Alexander A. Sherstov. "Cryptographic hardness for learning intersections of halfspaces". In: *J. Comput. Syst. Sci.* 75.1 (2009), pp. 2–12. DOI: 10.1016/j.jcss.2008.07.008. URL: http://dx.doi.org/10.1016/j.jcss.2008.07.008.

[KS10] Marek Karpinski and Warren Schudy. "Faster Algorithms for Feedback Arc Set Tournament, Kemeny Rank Aggregation and Betweenness Tournament". In: *Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*. 2010, pp. 3–14. DOI: 10.1007/978-3-642-17517-6_3. URL: http://dx.doi.org/10.1007/978-3-642-17517-6_3.

[KT07] Ravi Kannan and Thorsten Theobald. "Games of fixed rank: a hierarchy of bimatrix games". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*. 2007, pp. 1124–1132. URL: http://dl.acm.org/citation.cfm?id=1283383.1283504.

[Kuc95] Ludek Kucera. "Expected Complexity of Graph Partitioning Problems". In: *Discrete Applied Mathematics* 57.2-3 (1995), pp. 193–212.

[KV94] Michael J. Kearns and Leslie G. Valiant. "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata". In: *J. ACM* 41.1 (1994), pp. 67–95. DOI: 10.1145/174644.174647. URL: http://doi.acm.org/10.1145/174644.174647.

[KZ11] Pascal Koiran and Anastasios Zouzias. "On the Certification of the Restricted Isometry Property". In: *CoRR* abs/1103.4984 (2011). URL: http://arxiv.org/abs/1103.4984.

[Lei92] F. Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Array, Trees, Hypercubes*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992. ISBN: 1-55860-117-1.

[LH64] C. E. Lemke and J. T. Howson. "Equilibrium Points of Bimatrix Games". In: *Journal of the Society for Industrial and Applied Mathematics* 12.2 (1964), pp. 413–423. ISSN: 03684245. URL: http://www.jstor.org/stable/2946376.

[Lit87] Nick Littlestone. "Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm". In: *Machine Learning* 2.4 (1987), pp. 285–318. DOI: 10.1007/BF00116827. URL: http://dx.doi.org/10.1007/BF00116827.

[LM14] Twan van Laarhoven and Elena Marchiori. "Axioms for graph clustering quality functions". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 193–215. URL: http://dl.acm.org/citation.cfm?id=2627441.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. "Playing Large Games Using Simple Strategies". In: *Proceedings of the 4th ACM Conference on Electronic Commerce*. EC '03. San Diego, CA, USA: ACM, 2003, pp. 36–41. ISBN: 1-58113-679-X. DOI: 10.1145/779928.779933. URL: http://doi.acm.org/10.1145/779928.779933.

[LMR91] Nathan Linial, Yishay Mansour, and Ronald L. Rivest. "Results on Learnability and the Vapnik-Chervonenkis Dimension". In: *Inf. Comput.* 90.1 (1991), pp. 33–49. DOI: 10.1016/0890-5401(91)90058-A. URL: http://dx.doi.org/10.1016/0890-5401(91)90058-A.

[LMS11] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. "Lower bounds based on the Exponential Time Hypothesis". In: *Bulletin of the EATCS* 105 (2011), pp. 41–72. URL: http://albcom.lsi.upc.edu/ojs/index.php/beatcs/article/view/96.

[Man17] Pasin Manurangsi. "Almost-Polynomial Ratio ETH-Hardness of Approximating Densest k-Subgraph". In: *Proceedings of the Fortieth-ninth Annual ACM Symposium on Theory of Computing*. STOC '17. To appear. 2017.

[Mar95] Alfred Marshall. *Principles of Economics*. III. Macmillan, 1895.

[Max97] R. R. Maxfield. "General equilibrium and the theory of directed graphs". In: *Journal of Mathematical Economics* 27.1 (1997), pp. 23–51.

[Men71] Carl Menger. *Principles of Economics*. Ludwig von Mises Institute, 1871.

[Mis+08] Nina Mishra et al. "Finding Strongly Knit Clusters in Social Networks". In: vol. 5. 1. 2008, pp. 155–174. DOI: 10.1080/15427951.2008.10129299. URL: http://dx.doi.org/10.1080/15427951.2008.10129299.

[MMV16] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. "Learning Communities in the Presence of Errors". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 1258–1291. URL: http://jmlr.org/proceedings/papers/v49/makarychev16.html.

[Mor03] Tsuyoshi Morioka. "The Relative Complexity of Local Search Heuristics and the Iteration Principle". In: *Electronic Colloquium on Computational Complexity (ECCC)* 051 (2003). URL: http://eccc.hpi-web.de/eccc-reports/2003/TR03-051/index.html.

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. "On Total Functions, Existence Theorems and Computational Complexity". In: *Theor. Comput. Sci.* 81.2 (1991), pp. 317–324. DOI: 10.1016/0304-3975(91)90200-L. URL: https://doi.org/10.1016/0304-3975(91)90200-L.

[MPW15] Raghu Meka, Aaron Potechin, and Avi Wigderson. "Sum-of-squares Lower Bounds for Planted Clique". In: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*. 2015, pp. 87–96. DOI: 10.1145/2746539.2746600. URL: http://doi.acm.org/10.1145/2746539.2746600.

[MPW16] Ankur Moitra, William Perry, and Alexander S. Wein. "How robust are reconstruction thresholds for community detection?" In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. 2016, pp. 828–841. DOI: 10.1145/2897518.2897573. URL: http://doi.acm.org/10.1145/2897518.2897573.

[MR10] Dana Moshkovitz and Ran Raz. "Two-query PCP with subconstant error". In: *J. ACM* 57.5 (2010), 29:1–29:29. DOI: 10.1145/1754399.1754402. URL: http://doi.acm.org/10.1145/1754399.1754402.

[MR16] Pasin Manurangsi and Prasad Raghavendra. "A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs". In: *CoRR* abs/1607.02986 (2016). URL: http://arxiv.org/abs/1607.02986.

[MS12] Peter Bro Miltersen and Or Sheffet. "Send mixed signals: earn more, work less". In: *ACM Conference on Electronic Commerce, EC '12, Valencia, Spain, June 4-8, 2012*. 2012, pp. 234–247. DOI: 10.1145/2229012.2229033. URL: http://doi.acm.org/10.1145/2229012.2229033.

[MT05] Andrew McLennan and Rabee Tourky. "From imitation games to Kakutani". In: *Manuscript, available at http://www.econ.umn.edu/ mclennan/Papers/papers.html* (2005).

[MU02] Elchanan Mossel and Christopher Umans. "On the complexity of approximating the VC dimension". In: *J. Comput. Syst. Sci.* 65.4 (2002), pp. 660–671. DOI: 10.1016/S0022-0000(02)00022-3. URL: http://dx.doi.org/10.1016/S0022-0000(02)00022-3.

[MX16] Elchanan Mossel and Jiaming Xu. "Density Evolution in the Degree-correlated Stochastic Block Model". In: *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*. 2016, pp. 1319–1356. URL: http://jmlr.org/proceedings/papers/v49/mossel16.html.

[Nas51] John Nash. "Non-Cooperative Games". In: *The Annals of Mathematics* 54 (1951), pp. 286–295.

[Nis09a] Noam Nisan. *Communication Complexity of Mixed-Nash Equilibria*. https://agtb.wordpress.co complexity-of-mixed-nash-equilibria/. Blog. 2009.

[Nis09b] Noam Nisan. *Economists and Complexity*. https://agtb.wordpress.com/2009/11/27/economist and-complexity/. Blog. 2009.

[OSB10] Abraham Othman, Tuomas Sandholm, and Eric Budish. "Finding approximate competitive equilibria: efficient and fair course allocation". In: *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*. 2010, pp. 873–880. DOI: 10.1145/1838206.1838323. URL: http://doi.acm.org/10.1145/1838206.1838323.

[Pap94] Christos H. Papadimitriou. "On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence". In: *J. Comput. Syst. Sci.* 48.3 (1994), pp. 498–532.

[PR08] Christos H. Papadimitriou and Tim Roughgarden. "Computing correlated equilibria in multi-player games". In: *J. ACM* 55.3 (2008).

[PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. "Tolerant property testing and distance approximation". In: *J. Comput. Syst. Sci.* 72.6 (2006), pp. 1012–1042. DOI: 10.1016/j.jcss.2006.03.002. URL: http://dx.doi.org/10.1016/j.jcss.2006.03.002.

[PS94] Alexander Polishchuk and Daniel A. Spielman. "Nearly-linear size holographic proofs". In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*. 1994, pp. 194–203. DOI: 10.1145/195058.195132. URL: http://doi.acm.org/10.1145/195058.195132.

[PY86] Christos H. Papadimitriou and Mihalis Yannakakis. "A note on succinct representations of graphs". In: *Information and Control* 71.3 (1986), pp. 181 –185. ISSN: 0019-9958. DOI: http://dx.doi.org/10.1016/S0019-9958(86)80009-2. URL: http://www.sciencedirect.com/science/article/pii/S0019995886800092.

[PY96] Christos H. Papadimitriou and Mihalis Yannakakis. "On Limited Nondeterminism and the Complexity of the V-C Dimension". In: *J. Comput. Syst. Sci.* 53.2 (1996), pp. 161–170. DOI: 10.1006/jcss.1996.0058. URL: http://dx.doi.org/10.1006/jcss.1996.0058.

[RM99] Ran Raz and Pierre McKenzie. "Separation of the Monotone NC Hierarchy". In: vol. 19. 3. 1999, pp. 403–435. DOI: 10.1007/s004930050062. URL: http://dx.doi.org/10.1007/s004930050062.

[Rob51] Julia Robinson. "An Iterative Method of Solving a Game". In: *Annals of Mathematics* 54 (1951), pp. 296–301.

[Rou14] Tim Roughgarden. "Barriers to Near-Optimal Equilibria". In: (2014), pp. 71–80. DOI: 10.1109/FOCS.2014.16. URL: http://dx.doi.org/10.1109/FOCS.2014.16.

[RS10] Prasad Raghavendra and David Steurer. "Graph expansion and the unique games conjecture". In: *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM. 2010, pp. 755–764.

[Rub15] Aviad Rubinstein. "ETH-Hardness for Signaling in Symmetric Zero-Sum Games". In: *CoRR* abs/1510.04991 (2015). URL: http://arxiv.org/abs/1510.04991.

[Rub16] Aviad Rubinstein. "Settling the Complexity of Computing Approximate Two-Player Nash Equilibria". In: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. 2016, pp. 258–265. DOI: 10.1109/FOCS.2016.35. URL: http://dx.doi.org/10.1109/FOCS.2016.35.

[RW16] Tim Roughgarden and Omri Weinstein. "On the Communication Complexity of Approximate Fixed Points." In: *Electronic Colloquium on Computational Complexity (ECCC)*. Vol. 23. 2016, p. 55.

[RW90] Ran Raz and Avi Wigderson. "Monotone Circuits for Matching Require Linear Depth". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. 1990, pp. 287–292. DOI: 10.1145/100216.100253. URL: http://doi.acm.org/10.1145/100216.100253.

[Sca67] Herbet Scarf. "The approximation of fixed points of a continuous mapping". In: *SIAM Journal of Applied Mathematics* 15 (1967), pp. 1328–1343.

[Sch00] Marcus Schaefer. "Deciding the K-Dimension is PSPACE-Complete". In: *Proceedings of the 15th Annual IEEE Conference on Computational Complexity, Florence, Italy, July 4-7, 2000*. 2000, pp. 198–203. DOI: 10.1109/CCC.2000.856750. URL: http://dx.doi.org/10.1109/CCC.2000.856750.

[Sch99] Marcus Schaefer. "Deciding the Vapnik-Cervonenkis Dimension in Sigma$^{\mathrm{p}}_3$-Complete". In: *J. Comput. Syst. Sci.* 58.1 (1999), pp. 177–182. DOI: 10.1006/jcss.1998.1602. URL: http://dx.doi.org/10.1006/jcss.1998.1602.

[Sha64] Lloyd Stowell Shapley. "Some Topics in Two-Person Games". In: *Annals of Mathematical Studies* 52 (1964), pp. 1–28.

[Sha92] Adi Shamir. "IP = PSPACE". In: *J. ACM* 39.4 (1992), pp. 869–877. DOI: 10.1145/146585.146609. URL: http://doi.acm.org/10.1145/146585.146609.

[Shm12] Eran Shmaya. *Brouwer Implies Nash Implies Brouwer*. http://theoryclass.wordpress.com/2012/implies-nash-implies-brouwer/. Blog. 2012.

[Spi95] Daniel A. Spielman. "Computationally Efficient Error-Correcting Codes and Holographic Proofs". PhD thesis. Massachusetts Institute of Technology, 1995.

[SSS95] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. "Chernoff-Hoeffding Bounds for Applications with Limited Independence". In: *SIAM J. Discrete Math.* 8.2 (1995), pp. 223–250. DOI: 10.1137/S089548019223872X. URL: http://dx.doi.org/10.1137/S089548019223872X.

[SSW04] Satinder P. Singh, Vishal Soni, and Michael P. Wellman. "Computing approximate bayes-nash equilibria in tree-games of incomplete information." In: *EC*. 2004, pp. 81–90.

[Sti47] George J. Stigler. "Notes on the History of the Giffen Paradox". In: *Journal of Political Economy* 55.2 (1947), pp. 152–156.

[SV12] Grant Schoenebeck and Salil P. Vadhan. "The Computational Complexity of Nash Equilibria in Concisely Represented Games". In: *TOCT* 4.2 (2012), p. 4.

[Tre+16] Nicolas Tremblay et al. "Compressive Spectral Clustering". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*. 2016, pp. 1002–1011. URL: http://jmlr.org/proceedings/papers/v48/tremblay16.html.

[TS08] Haralampos Tsaknakis and Paul G. Spirakis. "An Optimization Approach for Approximate Nash Equilibria". In: *Internet Mathematics* 5.4 (2008), pp. 365–382. DOI: 10.1080/15427951.2008.10129172. URL: https://doi.org/10.1080/15427951.2008.10129172.

[TV85] W. Thomson and H.R. Varian. "Theories of justice based on symmetry". In: *Social Goals and Social Organizations: Essays in Memory of Elisha Pazner* (1985).

[Var74] H. Varian. "Equity, envy, and efficiency". In: *Journal of Economic Theory* 9.1 (1974), pp. 63–91.

[VC71] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". In: *Theory of Probability & Its Applications* 16.2 (1971), pp. 264–280. DOI: 10.1137/1116025. eprint: http://dx.doi.org/10.1137/1116025. URL: http://dx.doi.org/10.1137/1116025.

[Vid15] Michael Viderman. "A combination of testability and decodability by tensor products". In: *Random Struct. Algorithms* 46.3 (2015), pp. 572–598. DOI: 10.1002/rsa.20498. URL: http://dx.doi.org/10.1002/rsa.20498.

[VY11] Vijay V. Vazirani and Mihalis Yannakakis. "Market equilibrium under separable, piecewise-linear, concave utilities". In: *J. ACM* 58.3 (2011), p. 10.

[Wal74] Léon Walras. *Éléments d'économie politique pure, ou, Théorie de la richesse sociale*. L. Corbaz Lausanne, 1874.

[Yan68] Elena B. Yanovskaya. "Equilibrium points in polymatrix games," in: *Litovskii Matematicheskii Sbornik* 8 (1968), pp. 381–384.

[You04] H Peyton Young. *Strategic learning and its limits*. OUP Oxford, 2004.

[You09] H Peyton Young. "Learning by trial and error". In: *Games and economic behavior* 65.2 (2009), pp. 626–643.

[Zuc07] David Zuckerman. "Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number". In: *Theory of Computing* 3.1 (2007), pp. 103–128. DOI: 10.4086/toc.2007.v003a006. URL: http://dx.doi.org/10.4086/toc.2007.v003a006.