New Directions in Approximation Algorithms and Hardness of Approximation

Yuan Zhou

CMU-CS-14-125

August 2014

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Venkatesan Guruswami, Co-chair
Ryan O’Donnell, Co-chair
Anupam Gupta
R. Ravi
Sanjeev Arora (Princeton University)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Copyright © 2014 Yuan Zhou

This research was sponsored by the National Science Foundation under grant numbers CCF-0915893, CCF-0953155, CCF-1115525; US Israel Binational Science Foundation under grant number 2008293; Simons Foundation under grant number 252545; and generous support from the Microsoft Corporation.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.
**Keywords:** Approximation Algorithms, Hardness of Approximation, Combinatorial Optimization, Convex Relaxation Hierarchies, Linear Programming, Semidefinite Programming
DEDICATION

To my parents.
Abstract

Combinatorial optimization encompasses a wide range of important computational tasks such as UNIFORMSPARSESTCUT (also known as NORMALIZEDCUT), MAXCUT, TRAVELINGSALESMANPROBLEM, and VERTEXCOVER. Most combinatorial optimization problems are NP-hard to be solved optimally. On one hand, a natural way to cope with this computational intractability is via designing approximation algorithms to efficiently approximate the optimal solutions with provable guarantees. On the other hand, given an NP-hard optimization problem, we are also interested in the best possible approximation guarantee that any polynomial-time algorithm could achieve, i.e. the hardness of approximation of the problem. Both approximation algorithms and hardness of approximation results contribute to understanding the approximability of combinatorial optimization problems.

In the last two decades, the research frontier of approximation algorithm design has been greatly advanced thanks to the convex optimization techniques such as linear and semidefinite programming. However, the exact approximability for many problems remains mysterious but some common barriers for progress revolving around a problem called Unique Games has been identified. The limitations of convex relaxation techniques answer the question that what is the best possible approximation guarantee to be achieved by the state-of-the-art algorithmic design tools, and shed light on the real approximability. Therefore, the study of the power of convex relaxations becomes a valuable new research direction to get around the current barrier on hardness proofs.

In this thesis, using constraint satisfaction problems, assignment problems, graph partitioning problems (BALANCEDSEPARATOR, UNIFORMSPARSESTCUT, DENSEkSUBGRAPH), and graph isomorphism as examples, we explore both the effectiveness and limitations of the most powerful convex relaxation techniques – convex relaxation hierarchies. We also use a proof complexity view of the convex relaxation hierarchies to analyze their performance on constraint satisfaction problems, and show that the so-called Parrilo–Lasserre semidefinite programming relaxation hierarchy succeeds on all hard instances constructed in literature for UNIQUEGAMES, MAXCUT, and BALANCEDSEPARATOR.
This thesis also contains a collection of approximation algorithms for almost satisfiable constraint satisfaction problems and MAXBISECTION, detection of almost isomorphic trees, and estimation of the $2 \rightarrow 4$ operator norm of random linear operators. There are also a few (conditional) hardness of approximation results for almost satisfiable linear systems over integers, almost satisfiable MAXHORN$3$-SAT, and detection of almost isomorphic graphs.
Acknowledgments

It would not have been possible to complete this doctoral thesis without the help and support of many people around me, to only some of whom it is possible to give particular mention here.

Above all, I owe my deepest gratitude to my Ph.D. advisors Venkat Guruswami and Ryan O’Donnell. When still in college, I read and was intrigued by a set of beautifully scribed online lecture notes on the PCP theorem, approximation algorithms, and hardness of approximation. A year later, I was fortunate enough to become a student of both of the lecturers in that course. In fact, Venkat and Ryan are the best advisors I could have asked for. I was always amazed by their brilliance, technical mastery, and wide knowledge on almost everything. Venkat could always patiently go through every ridiculous ideas of mine; and Ryan never failed to figure a new direction after we hitting thousands of dead-ends. Over the years, I enjoyed all the discussions through which I learned a lot from their techniques and research styles. For all these and the freedom in choosing research topics, the career advices, and a lot more left out – Thank you Venkat and Ryan!

I am very grateful to my thesis committee: Venkatesan Guruswami, Ryan O’Donnell, Sanjeev Arora, Anupam Gupta, and R. Ravi. Thank you for taking time from your busy schedules to make my thesis defense.

Many thanks to each of my paper co-authors through my Ph.D. : Boaz Barak, Aditya Bhaskara, Fernando Brandão, Moses Charikar, Xi Chen, Julia Chuzhoy, Parikshit Gopalan, Venkatesan Guruswami, Aram Harrow, Zhiyi Huang, Manuel Kauers, Jonathan Kelner, Gabor Kun, Jian Li, Yury Makarychev, Raghu Meka, Ryan O’Donnell, Prasad Raghvendra, Omer Reingold, Ali Kemal Sinop, David Steurer, Suguru Tamaki, Li-Yang Tan, Madhur Tulsiani, Salil Vadhan, Aravindan Vijayaraghavan, Lei Wang, John Wright, Chenggang Wu, Yi Wu, Yu Wu, Yuichi Yoshida, and Jiawei Zhang. It was great joy working with you; and this thesis would not be possible without your collaborations.

I had the rare opportunity to spend each of the four summers during graduate school visiting a number of places. When still getting familiar with theoretical research in my first
year, I had the pleasure of interning with Julia Chuzhoy and Yury Makarychev at Toyota Technological Institute at Chicago, as well as visiting Pinyan Lu at Microsoft Research Asia. In the following summers, I enjoyed collaborating with Boaz Barak at Microsoft Research New England, Parikshit Gopalan at Microsoft Research Silicon Valley, and Yury Makarychev and Madhur Tulsiani at Toyota Technological Institute at Chicago. During winter breaks, I also enjoyed a couple of visits to Shengyu Zhang at the Chinese University at Hong Kong, Yitong Yin at Nanjing University, and Xiaoming Sun at the Chinese Academy of Sciences. I am greatly indebted to all these institutions I have visited and the numerous people who hosted me. I would also like to thank the support from the Simons Graduate Fellowship in Theoretical Computer Science during the last two years.

I wish to thank my high school teacher Tao Jiang for introducing me to the world of algorithmic programming contest. I am also greatly indebted to Professor Andrew Yao and Elad Verbin at Tsinghua University and Wei Chen at Microsoft Research Asia who further guided my interest to theoretical computer science.

Many thanks go to the theory group faculty members at CMU from whom I learned a lot: Avrim Blum, Manuel Blum, Alan Frieze, Anupam Gupta, Gary Miller, and Danny Sleator.

Also, special thanks to the friends I met in Pittsburgh and during my summer internships for making my Ph.D. life colorful: Parinya Chalermsook, Zhuo Chen, Ricky Chis, Bin Fu, Edinah Gnang, Fan Guo, Favonia Hou, Jason Lee, Lei Li, Nan Li, Jian Peng, Richard Peng, Julian Shun, Xiaorui Sun, Carol Wang, John Wright, Chenye Wu, Yi Wu, Guangyu Xia, Guang Xiang, Lin Xiao, Ning Xie, Jia Xu, Kathy Zhang, Zeyuan Zhu, Mengjie Zou.

Finally, it would be difficult for me to express the debt of gratitude that I owe my parents Qihua Zhou and Hangli Wang, for their love and constant support.
Contents

1 Introduction 1

1.1 The notion of approximation algorithms 3
    1.1.1 Robust algorithms 4
    1.1.2 Hardness of approximation 4

1.2 The relaxation and rounding framework for designing approximation algorithms 5
    1.2.1 Convex relaxations 5
    1.2.2 Rounding schemes 6
    1.2.3 Integrality gaps and limitations of the relaxation 7

1.3 Linear and semidefinite programming relaxations, and methods of designing them 8
    1.3.1 Relaxation hierarchies and the implication of their limitations 9

1.4 A brief overview of contributions 10

2 Preliminaries 13

2.1 Problems studied in this thesis 13
    2.1.1 Constraint satisfaction problems 13
        2.1.1.1 The UNIQUEGAMES problem 14
    2.1.2 Graph partitioning problems 14
    2.1.3 Graph isomorphism and assignment problems 15
    2.1.4 Dense and locally-dense instances 17
# Study of the LP/SDP relaxation hierarchies

## Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems

### Introduction

4.1.1 Linear Programming (LP) relaxation and LP relaxation hierarchies

4.1.2 Our contributions

4.1.3 Proof overview

4.1.4 Comparison to previous works

4.1.5 Organization

### Preliminaries

4.2

### Conditioning operations for Sherali-Adams LP hierarchy

4.3.1 Conditioning operations

---

2.2 Approximation and hardness of approximation

2.2.1 Robust algorithms

2.3 The Unique Games Conjecture

3 Summary of contributions and organization of this thesis

3.1 Overview of Part I: study of the LP/SDP relaxation hierarchies

3.1.1 Algorithmic results

3.1.2 Integrality gaps

3.2 Overview of Part II: using the Parrilo–Lasserre hierarchy to solve hard instances for weaker hierarchies

3.3 Overview of Part III: robust algorithms

3.4 Overview of Part IV: other approximation and hardness of approximation results

3.4.1 Approximating the $2 \rightarrow 4$ norm of random linear operators

3.4.2 Hardness of $\text{MAX}^\Gamma\!-\!2\!-\!\text{LIN}$ and $\text{MAX}^\Gamma\!-\!3\!-\!\text{LIN}$ over integers

3.4.3 Hardness of approximating almost satisfiable MAXHORN3-SAT
## 6 Lasserre integrality gaps for BalancedSeparator and UniformSparsestCut

### 6.1 Introduction

- **6.1.1 Our contributions**
- **6.1.2 Our techniques**

### 6.2 Preliminaries on Lasserre SDPs for BalancedSeparator and UniformSparsestCut

- **6.2.1 Useful theorems about the Lasserre hierarchy**
- **6.2.2 Lasserre SDP for graph partitioning problems**
  - **6.2.2.1 BalancedSeparator**
  - **6.2.2.2 UniformSparsestCut**
- **6.2.3 Lasserre Gaps for 3-XOR from Schoenebeck**

### 6.3 Gaps for BalancedSeparator

- **6.3.1 Reduction**
- **6.3.2 Completeness: good SDP solution**
- **6.3.3 Soundness: bound for integral solutions**
- **6.3.4 Constant-degree integrality gap instance**

### 6.4 Gaps for UniformSparsestCut


## 7 Lasserre integrality gaps for RobustGISO

### 7.1 Introduction

- **7.1.1 Our contributions**
- **7.1.2 SOS/Lasserre gaps**
- **7.1.3 Robust graph isomorphism**
- **7.1.4 Robust asymmetry of random graphs**
- **7.1.5 Organization**

### 7.2 Preliminaries

- **7.2.1 SOS/Lasserre hierarchy**
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1</td>
<td>An SOS proof of small-set expansion in the noisy hypercube</td>
<td>179</td>
</tr>
<tr>
<td>10.2</td>
<td>The KKL Theorem</td>
<td>180</td>
</tr>
<tr>
<td>10.3</td>
<td>Analysis of the DKSV BALANCEDSEPARATOR instances</td>
<td>183</td>
</tr>
<tr>
<td>11</td>
<td>SOS proofs of the CLT, the $\frac{2}{\pi}$ Theorem, and the analysis of the KV MAXCut instances</td>
<td>187</td>
</tr>
<tr>
<td>11.1</td>
<td>An invariance theorem for polynomials of linear forms</td>
<td>187</td>
</tr>
<tr>
<td>11.2</td>
<td>An SOS proof of the $\frac{2}{\pi}$ Theorem</td>
<td>189</td>
</tr>
<tr>
<td>11.3</td>
<td>Analysis of the KV MAXCut instances</td>
<td>193</td>
</tr>
<tr>
<td>11.4</td>
<td>An approximator for the absolute-value function</td>
<td>199</td>
</tr>
<tr>
<td>III</td>
<td>Robust algorithms</td>
<td>203</td>
</tr>
<tr>
<td>12</td>
<td>Robust satisfiability algorithms for width 1 CSPs</td>
<td>205</td>
</tr>
<tr>
<td>12.1</td>
<td>Introduction</td>
<td>205</td>
</tr>
<tr>
<td>12.1.1</td>
<td>Our contributions</td>
<td>207</td>
</tr>
<tr>
<td>12.2</td>
<td>Preliminaries</td>
<td>208</td>
</tr>
<tr>
<td>12.2.1</td>
<td>CSP preliminaries</td>
<td>208</td>
</tr>
<tr>
<td>12.2.2</td>
<td>Algorithmic preliminaries</td>
<td>209</td>
</tr>
<tr>
<td>12.2.3</td>
<td>Algebraic preliminaries</td>
<td>210</td>
</tr>
<tr>
<td>12.3</td>
<td>Width 1 implies robust decidability by LP</td>
<td>212</td>
</tr>
<tr>
<td>12.3.1</td>
<td>Lattice CSPs: better quantitative dependence on $\epsilon$</td>
<td>213</td>
</tr>
<tr>
<td>13</td>
<td>An algorithm for ROBUSTMAXBISECTION</td>
<td>217</td>
</tr>
<tr>
<td>13.1</td>
<td>Introduction</td>
<td>217</td>
</tr>
<tr>
<td>13.1.1</td>
<td>Our contributions</td>
<td>220</td>
</tr>
<tr>
<td>13.1.2</td>
<td>Later development</td>
<td>220</td>
</tr>
<tr>
<td>13.2</td>
<td>Method overview</td>
<td>221</td>
</tr>
</tbody>
</table>
# 13.2.1 Integrality gap 221
# 13.2.2 Notations 222
# 13.2.3 Our approach 223
# 13.2.4 Organization 224

## 13.3 Preprocessing and partitioning graph $G$ 224
### 13.3.1 Preprocessing: Making $G$ bipartite and unweighted 224
### 13.3.2 Partitioning 225

## 13.4 Finding cuts in sets $U_i$ and $V_i$ 227
### 13.4.1 Candidate cuts in $V_i$ 227
### 13.4.2 Candidate cuts in $U_i$ 229

## 13.5 Combining candidate cuts 231

## 13.6 The bisection algorithm – proof of Theorem 13.1.3 234

## 14 A robust isomorphism algorithm for trees 237
### 14.1 Introduction 237
#### 14.1.1 Our contribution and overview of the proofs 238
### 14.2 The algorithm 239
#### 14.2.1 Robust isomorphism algorithm for $B$-trees 241
#### 14.2.2 Proofs of Lemma 14.2.3 and Lemma 14.2.4 243

## IV Other approximation and hardness of approximation results 247

## 15 Certifying the $2 \to 4$ norm of random linear operators 249
### 15.1 Introduction 249
### 15.2 The TensorSDP algorithm 250
### 15.3 Certifying the hypercontractivity of random operators 251

## 16 Hardness of $\text{Max}^1$-$2$-$\text{Lin}$ and $\text{Max}^1$-$3$-$\text{Lin}$ over integers 257
### 16.1 Introduction 257
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.1.1</td>
<td>Prior work on $\text{MAX}_3$-$\text{LIN}(\mathbb{Z})$</td>
<td>257</td>
</tr>
<tr>
<td>16.1.2</td>
<td>Prior work on $\text{MAX}_2$-$\text{LIN}$</td>
<td>258</td>
</tr>
<tr>
<td>16.1.3</td>
<td>Our contributions</td>
<td>259</td>
</tr>
<tr>
<td>16.2</td>
<td>Preliminaries</td>
<td>260</td>
</tr>
<tr>
<td>16.2.1</td>
<td>Notations and Definitions</td>
<td>260</td>
</tr>
<tr>
<td>16.3</td>
<td>Review of proofs of $\text{MAX}_1$-$2$-$\text{LIN}(\mathbb{Z}_q)$ and $\text{MAX}_1$-$3$-$\text{LIN}(\mathbb{Z}_q)$ hardness</td>
<td>263</td>
</tr>
<tr>
<td>16.3.1</td>
<td>$\text{MAX}_1$-$2$-$\text{LIN}$</td>
<td>263</td>
</tr>
<tr>
<td>16.3.2</td>
<td>$\text{MAX}_1$-$3$-$\text{LIN}$</td>
<td>263</td>
</tr>
<tr>
<td>16.4</td>
<td>Overview of our proofs</td>
<td>264</td>
</tr>
<tr>
<td>16.4.1</td>
<td>Active folding</td>
<td>264</td>
</tr>
<tr>
<td>16.4.2</td>
<td>Modifying the KKMO proof</td>
<td>265</td>
</tr>
<tr>
<td>16.4.3</td>
<td>Modifying the Håstad proof</td>
<td>266</td>
</tr>
<tr>
<td>16.5</td>
<td>Dictator Test details</td>
<td>267</td>
</tr>
<tr>
<td>16.5.1</td>
<td>Dictator Test for $\text{MAX}_1$-$2$-$\text{LIN}$</td>
<td>267</td>
</tr>
<tr>
<td>16.5.2</td>
<td>Matching Dictator Test for $\text{MAX}_1$-$3$-$\text{LIN}$</td>
<td>268</td>
</tr>
<tr>
<td>16.6</td>
<td>Reductions between $\text{MAX}_k$-$\text{LIN}(R)$ problems</td>
<td>272</td>
</tr>
<tr>
<td>16.7</td>
<td>From Dictator Tests to hardness of approximation</td>
<td>274</td>
</tr>
<tr>
<td>16.7.1</td>
<td>Proof of Lemma 16.5.2</td>
<td>274</td>
</tr>
<tr>
<td>16.7.2</td>
<td>Proof of Lemma 16.5.4</td>
<td>276</td>
</tr>
<tr>
<td>17</td>
<td>Hardness of approximating almost satisfiable $\text{MAXHORN}_3$-$\text{SAT}$</td>
<td>279</td>
</tr>
<tr>
<td>17.1</td>
<td>Introduction</td>
<td>279</td>
</tr>
<tr>
<td>17.2</td>
<td>Our contributions and previous work</td>
<td>281</td>
</tr>
<tr>
<td>17.3</td>
<td>Proof method</td>
<td>282</td>
</tr>
<tr>
<td>17.3.1</td>
<td>The canonical SDP for Boolean CSPs and UG-Hardness</td>
<td>283</td>
</tr>
<tr>
<td>17.3.2</td>
<td>Overview of construction of SDP gaps</td>
<td>284</td>
</tr>
<tr>
<td>17.3.3</td>
<td>Overview of algorithmic results</td>
<td>285</td>
</tr>
<tr>
<td>17.4</td>
<td>Approximability of $\text{HORN}_3$-$\text{SAT}$</td>
<td>286</td>
</tr>
<tr>
<td>17.4.1</td>
<td>SDP gap and UG hardness for $\text{HORN}_3$-$\text{SAT}$</td>
<td>286</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Permutations $\pi_1$ and $\pi_2$ over $[5]$ shown as mappings from $[5]$ to $[5]$.</td>
<td>54</td>
</tr>
<tr>
<td>6.1</td>
<td>IP and SDP relaxations for BALANCEDSEPARATOR.</td>
<td>91</td>
</tr>
<tr>
<td>6.2</td>
<td>IP and SDP relaxations for UNIFORMSPARSESTCUT.</td>
<td>92</td>
</tr>
<tr>
<td>6.3</td>
<td>The reduction for BALANCEDSEPARATOR.</td>
<td>95</td>
</tr>
<tr>
<td>16.1</td>
<td>Test $T$ with parameters $\epsilon, \gamma, q$ for functions on $\mathbb{Z}_Q^K$.</td>
<td>267</td>
</tr>
<tr>
<td>16.2</td>
<td>Test $U$ with parameters $\epsilon, q$ for $f$ on $\mathbb{Z}_Q^K$, $g$ on $\mathbb{Z}_Q^L$, $\pi : L \rightarrow K$.</td>
<td>269</td>
</tr>
</tbody>
</table>
## List of Tables

3.1 Table of contributions on convex relaxation hierarchies .......................... 24
3.2 Table of contributions on other approximation algorithms and hardness of approximation .......................... 25
## List of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Algorithm</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conditioning operation of Sherali-Adams solutions</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td>The TensorSDP algorithm for $2 \rightarrow 4$ norms.</td>
<td>251</td>
</tr>
<tr>
<td>3</td>
<td>Round any LP solution $f = {y_i}$ to a half-integral solution $f^<em>$ such that $\text{val}(f^</em>) \leq \text{val}(f)$</td>
<td>295</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Many important computational tasks can be modeled as combinatorial optimization problems, where the goal is to optimize a certain objective function on discrete variables subject to some constraints. To give the readers a flavor of the optimization problems studied in this thesis, we list a few examples as follows.

**Problem 1** (the $\frac{1}{3}$-**BALANCED SEPARATOR** problem). Given an undirected graph, partition the set of vertices into two parts so that each part contains at least $\frac{1}{3}$ of the total vertices and the number of edges across the partition is minimized.

There are many similar problems as **BALANCED SEPARATOR** (such as **MAX CUT**, **UNIFORM SPARSEST CUT**), which consist of the class of graph partitioning problems, and arise in many settings such as clustering, divide and conquer algorithms, VLSI layout, etc. There are also many other well-studied optimization problems, including the following examples.

**Problem 2** (the **TRAVELING SALESMAN PROBLEM** problem). Given a list of cities and the distances between each pair of cities, find out the shortest possible route (i.e. the one with minimum total distance) that visits each city exactly once and returns to the origin city.

**Problem 3** (the **VERTEX COVER** problem). Given an undirected graph, find out the smallest set of vertices (i.e. the one with minimum cardinality) such that each edge of the graph is incident to at least one vertex of the set.

**Problem 4** (solving overdetermined sparse linear systems). Given a system of linear equations over rational numbers, so that each equation contains at most 3 variables. If the linear system is consistent, it is easy to find a solution using Gaussian elimination. However,
suppose the system is not completely consistent, i.e. some of the equations are erroneous. The natural optimization problem here is to find a solution that satisfies the maximum number of equations.

The readers may refer to Section 2.1 where more combinatorial optimization problems (which are also the problems studied in this thesis) are defined.

For many combinatorial optimization problems, it is computationally intractable (NP-hard) to find the optimal solution. A popular and extensively studied way to deal with this intractability is via designing approximation algorithms to efficiently approximate the optimal solutions with provable guarantees. We will briefly introduce this notion in Section 1.1. We will also discuss (in Section 1.1.2) the notion of hardness of approximation, i.e. the limitation on approximation guarantee for polynomial-time algorithms. While it is desirable to design an approximation algorithm and prove hardness of approximation matching the approximation guarantee given by the algorithm, we are usually not able to achieve this goal due to the limited techniques on both algorithmic analysis and hardness proofs.

On the algorithmic side, convex relaxation hierarchies, which we will introduce in Section 1.2 and Section 1.3, are the most powerful framework for designing approximation algorithms. For many important combinatorial optimization problems, the state-of-the-art approximation algorithms only use a small portion of the power of convex relaxation hierarchies (as later discussed in the subsections). Therefore, the pursuit of better approximation algorithms greatly motivates the further study of convex relaxation hierarchies.

It is also worthwhile to explore the limitation of convex relaxation hierarchies, given the lack of sharp hardness of approximation results for many important problems. As discussed in Section 1.3.1, although such type of results does not work against all polynomial-time algorithms, they share a glimpse into the frontier of our approximation techniques, and help us understand the complexity of approximating optimization problems where no concrete NP-hardness is known. In the light of this, showing the limitation for the hierarchies can be viewed as a way of going beyond our limited NP-hardness results.

This thesis is a collection of results regarding the approximability and inapproximability of various combinatorial optimization problems, while a good portion of it is about the effectiveness and limitations of convex relaxation hierarchies. In Section 1.4, we will briefly talk about the results in the thesis to conclude this introductory section.
1.1 The notion of approximation algorithms

As we have seen, the class of combinatorial optimization problems encompasses many important computational tasks. However, for most of these interesting optimization problems (and indeed all the example problems provided above), finding the optimal solution is unfortunately computationally intractable (NP-hard).

One way to deal with this intractability is to use efficient algorithms to find approximately optimal solutions with provable guarantees – we call these approximation algorithms. For a maximization problem, we call an algorithm \( \alpha \)-approximation algorithm \((\alpha < 1)\) if the output value of the algorithm is at least \( \alpha \) times the optimal value. Similarly, for a minimization problem, an \( \alpha \)-approximation algorithm \((\alpha > 1)\) guarantees to output a solution with value at most \( \alpha \) times the value of the optimal solution. Clearly, for both maximization and minimization problems, when \( \alpha \) gets closer to 1, we get better approximation guarantee.

Sometimes we would love to talk about the approximation guarantee in a more precise manner. Let us take following three scenarios in the MAXCUT problem (the problem where we are given an undirected graph and the goal is to find a partition of the vertex set so that the fraction of edges across the partition is maximize), for example –

1. given an instance where the optimal solution cuts all the edges, and we would like to find a partition cutting all the edges (approximation ratio : 1);
2. given an instance where the optimal solution cuts .9999 fraction of the edges, and we would like to find a partition cutting .99 fraction of the edges (approximation ratio : \( \sim .9901 \));
3. given an instance where the optimal solution cuts .80 fraction of the edges, and we would like to find a partition cutting .77 fraction of the edges (approximation ratio : .9625).

The first task is obviously easy (i.e. in \( \mathcal{P} \)) since it is exactly to compute the bipartition of a bipartite graph. For the second task, since the optimal solution cuts almost all the edges, the input graph looks very close to a bipartite graph. It is conceivable that some bipartite graph recognition algorithm might be able to deal with a tiny fraction of the erroneous edges and output a quite good partition meeting the criteria in the task (and indeed the Geomans-Williamson algorithm [94], which will be heavily mentioned in the rest of this thesis, works in this way). For the third task, although the least quality of approximation is demanded in terms of the ratio, the task itself is NP-hard by [219]. The
hardness may be attributed to the low value of the optimal solution. These examples motivate the definition of the approximation quality according to the value of the optimal solution. For the MAXCUT problem and $0 < s \leq c \leq 1$, we say an algorithm is a $(c, s)$-approximation algorithm if the algorithm outputs a partition cutting at least $s$ fraction of the edges whenever the optimal solution has value at least $c$. An $\alpha$-approximation algorithm is always a $(c, \alpha c)$-approximation algorithm for all $0 < c \leq 1$. However, as we see from the examples above, for some particular $c$, we can get a solution with value much better than $\alpha c$. Please refer to Section 2.2 for the precise definitions of the approximation notions discussed here.

1.1.1 Robust algorithms

In the exemplary MAXCUT scenarios presented above, we had the intuition that if the task of recovering the perfect solution (i.e. the partition that cuts all the edges) can be done in polynomial time, then finding out an almost perfect solution if there is one might also be easy. This intuition is correct for MAXCUT, but might not be the case for other problems. However, the idea of generalizing an algorithm which works for perfect solutions to almost perfect solution motivates the notion of robust algorithms.

At a high level, robust algorithms extremely well-approximate an almost perfect solution when such solutions exist. To make this description more precise, let us fix a maximization problem. Usually we can normalize the objective of the maximization problem so that it always lies between $[0, 1]$ and having objective value 1 means a perfect solution. (In the MAXCUT example, one natural objective concerns about the number of edges across the partition. However, we chose to rescale this objective and made it the fraction of the desired edges.) We call an algorithm for the optimization problem robust, if there exists a function $r : [0, 1] \to [0, 1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever the input instance has objective value $(1 - \epsilon)$, the algorithm outputs a solution with value at least $(1 - r(\epsilon))$. The readers may refer to Section 2.2.1 for more definitions and discuss on robust algorithms.

1.1.2 Hardness of approximation

To complement algorithmic results for a problem, it is also interesting to study its hardness of approximation, i.e. the limitation of polynomial-time algorithms in terms of the approximation guarantee they can achieve. For a maximization problem, we say it is NP-hard to $\alpha$-approximate the problem if any polynomial-time algorithm with $\alpha$ approximation guar-
antee implies $P = \text{NP}$. We will also talk about hardness of approximation under other complexity assumptions (such as the Unique Games Conjecture which will be introduced in Section 2.3). We can define similar hardness of approximation notion for minimization problems.

Ideally, for each problem of interest, we would like to have both an approximation algorithm and hardness of approximation which implies that there is no algorithm with better approximation guarantee, therefore identifying the approximation threshold of the problem.

### 1.2 The relaxation and rounding framework for designing approximation algorithms

Convex programming relaxations and rounding schemes are a standard tool to design approximation algorithms. A vast majority of known approximation algorithms are designed using this approach with only a few exceptions (e.g. [14, 90]). A significant part of this thesis is devoted to exploring the effectiveness and limitations of convex relaxations. In this section, we provide a rudimentary introduction to this powerful framework. We suggest readers to refer to relevant chapters in [221] and surveys such as [66] for more information.

#### 1.2.1 Convex relaxations

In combinatorial optimization problems, the solution space is discrete and usually we can encode the solutions using variables that take value either 0 or 1. After such encoding, it is often straightforward to formulate the optimization problem as an integer program. However, it is NP-hard to exactly solve the integer program (as long as the original optimization problem is NP-hard). Such computational intractability stems from the non-convexity (or the integrality) of the solution space. The idea here is to relax the integral constraints to make the program tractable.

Specifically, we relax the condition that variables take values either 0 or 1 so that variables can be real numbers or even vectors. For example, a simple relaxation would be allow variables to take any real numbers in $[0, 1]$ (instead of $\{0, 1\}$ values). In this way, if the objective function and other constraints in the integer program are linear, we relax it to a tractable linear program. Other methods can be applied to deal with the integer programs when the objective function or constraints are not linear.

Above we have just shown one simple approach of deriving linear programming re-
laxations, while there are also many other relaxation techniques such as semidefinite programming relaxations and relaxation hierarchies. The main idea for all these approaches is to obtain a convex optimization problem by suitable relaxations so that the new problem is computational tractable.

### 1.2.2 Rounding schemes

It is clear that any solution for the original optimization problem is also a feasible solution for the convex relaxation (one may think about the linear programming relaxation example to get the intuition). Therefore, if the original optimization problem is a maximization problem, the optimum of the convex relaxation is always at least the optimum of the original problem. Fix a convex relaxation $\mathcal{R}$ and a specific problem instance $\mathcal{I}$, let $\text{opt}(\mathcal{I})$ be the value of the optimal solution to $\mathcal{I}$; let $\text{opt}_{\mathcal{R}}(\mathcal{I})$ be the value of the optimal solution to the convex relaxation of $\mathcal{I}$. We have just derived that $\text{opt}(\mathcal{I}) \leq \text{opt}_{\mathcal{R}}(\mathcal{I})$.

However, not every solution for the convex relaxation has a corresponding solution in the original problem. Therefore, when using convex relaxations to design approximation algorithms, there is usually a rounding step to convert the relaxation solution to a feasible solution of the original problem. We call this procedure “rounding” because in the linear programming relaxation setting, the goal is often to “round” the fractional assignments to the variables to integral values. However, when using other relaxation techniques such as semidefinite programs, one may have to convert vector-valued variables to integral values.

Formally, a rounding scheme is an algorithm that takes the problem instance $\mathcal{I}$ and the optimal solution $x^*$ to the convex relaxation $\mathcal{R}$ as input, and outputs a solution $x$ to the original problem. Let $\text{val}_{\mathcal{R}}(\mathcal{I}; x^*) = \text{opt}_{\mathcal{R}}(\mathcal{I})$ be the objective value of $x^*$; and let $\text{val}(\mathcal{I}; x)$ be the objective value of $x$. Now still assume that the original problem is a maximization problem (where the case of minimization problem can be similarly deduced). If the rounding algorithm can be proved to always output an $x$ such that

$$\text{val}(\mathcal{I}; x) \geq \alpha \cdot \text{opt}(\mathcal{I})$$

(1.1)

for some $\alpha \in [0, 1]$, then we get an $\alpha$-approximation algorithm by first solving the convex relaxation and then performing the rounding scheme. However, directly proving (1.1) is often quite difficult because computing $\text{opt}(\mathcal{I})$ itself is NP-hard and we do not know much about it. Alternatively we turn to prove

$$\text{val}(\mathcal{I}; x) \geq \alpha \cdot \text{val}_{\mathcal{R}}(\mathcal{I}; x^*)$$

(1.2)

which is usually easier and implies (1.1) since $\text{val}_{\mathcal{R}}(\mathcal{I}; x^*) = \text{opt}_{\mathcal{R}}(\mathcal{I}) \geq \text{opt}(\mathcal{I})$. 

6


1.2.3 Integrality gaps and limitations of the relaxation

As mentioned before, not every solution for the convex relaxation has a corresponding solution in the original problem. Therefore, the relaxation optimum might be way “better” than the optimum of the original problem.

We call $\mathcal{I}$ be the integral gap instance with gap ratio $\frac{\text{opt}(\mathcal{I})}{\text{opt}_{\mathcal{R}}(\mathcal{I})}$. When this ratio is far from 1, $\mathcal{I}$ is a hard instance for the relaxation.

We also define the integral gap ratio of $\mathcal{R}$ be the worst ratio between $\text{opt}(\mathcal{I})$ and $\text{opt}_{\mathcal{R}}(\mathcal{I})$, i.e. for maximization problems, let

$$IG(\mathcal{R}) \overset{\text{def}}{=} \inf_{\mathcal{I}} \frac{\text{opt}(\mathcal{I})}{\text{opt}_{\mathcal{R}}(\mathcal{I})} \leq 1.$$  

Then we have

$$IG(\mathcal{R}) \cdot \text{opt}_{\mathcal{R}}(\mathcal{I}) \leq \text{opt}(\mathcal{I}) \leq \text{opt}_{\mathcal{R}}(\mathcal{I}).$$

When $IG(\mathcal{R})$ is close to 1 (from below), we see that the relaxation optimum is a good estimation of the optimum of the original problem.

Similarly, for minimization problems, we let the integral gap ratio of a specific relaxation $\mathcal{R}$ be

$$IG(\mathcal{R}) \overset{\text{def}}{=} \sup_{\mathcal{I}} \frac{\text{opt}(\mathcal{I})}{\text{opt}_{\mathcal{R}}(\mathcal{I})} \geq 1,$$

and we have

$$\text{opt}_{\mathcal{R}}(\mathcal{I}) \leq \text{opt}(\mathcal{I}) \leq IG(\mathcal{R}) \cdot \text{opt}_{\mathcal{R}}(\mathcal{I}).$$

The relaxation optimum well approximates the optimum of the original problem when $IG(\mathcal{R})$ is close to 1 from above.

Integrality gaps serve as a measure of the quality of the relaxation $\mathcal{R}$. When the integrality gap of $\mathcal{R}$ is bad (i.e. far from 1), since we usually use (1.2) to prove the effectiveness of $\mathcal{R}$-based approximation algorithms, we tend not to get a good approximation algorithm. In this sense, designing the integrality gap instance with gap ratio far from 1 is a way to show the limitation of the relaxation.

On the other hand, whenever we establish (1.2), we also know that the integrality gap of relaxation $\mathcal{R}$ is no worse than $\alpha$. This is also the usual way of proving the good quality of the relaxation.

We summarize and extend the definitions made above as follows.

**Definition 1.2.1.** Fix an optimization problem $Q$ and an LP/SDP relaxation $\mathcal{R}$ for $Q$. An instance $\mathcal{I}$ of $Q$ is said to be a $(c, s)$-integrality gap instance for $\mathcal{R}$,
• (for maximization problems) if the optimal value of $R(I)$ is at least $c$ while $\text{val}(I)$ is less than $s$;

• (for minimization problems) if the optimal value of $R(I)$ is at most $c$ while $\text{val}(I)$ is more than $s$.

We also say that $I$ is an integrality gap instance with gap ratio $\frac{c}{s}$.

Given such an instance $I$, we say that $R$ has an integrality gap $(c,s)$. The integrality gap ratio of $R$ (namely $\text{IG}(R)$) is the infimum of the gap ratios of all integrality gap instances for maximization problems, and the supremum of the ratios for minimization problems.

1.3 Linear and semidefinite programming relaxations, and methods of designing them

A large number of approximation algorithms use a specific type of convex relaxation – linear programming (LP). While linear programs can be solved in polynomial time using interior point methods [6, 225, 224], the simplex method is used extensively in practice. The “basic linear programming relaxation” (i.e. the one derived from the simple exemplary approach described in Section 1.2.1) already succeeds in efficiently approximating problems such as VERTEXCOVER, SETCOVER, and a wide class of generalized covering problems [152]. There are also ways to add more constraints (and variables if necessary) to derive stronger linear programming relaxations for problems such as MULTICUT [57, 83], SPARSESTCUT [163], and MULTIWAYCUT [56, 168, 51, 206].

Semidefinite programming (SDP) relaxations are another class of powerful convex relaxations. In a semidefinite program, the variables are vector valued, while both the constraints and the objective are linear in terms of the inner products of the variables. Semidefinite programs can be solved in polynomial time using the interior point methods [6, 225, 224]. More precisely, these algorithms output a solution with the value which differs from the optimum by at most an additive error $\epsilon$ in time that is polynomial in the program description size and $\log \frac{1}{\epsilon}$. Semidefinite programming relaxations proved to be extremely successful in approximation algorithms design after being introduced by Goemans and Williamson [94] in the context of the MAXCUT problem. A few examples on the semidefinite programming relaxation-based approximation algorithms include the

\[1\] Indeed, the classic work by Lovász [165], known as the Lovász Theta function today, is essentially a semidefinite programming relaxation for the INDEPENDENTSET problem.
ones for constraint satisfaction problems [94, 89, 132, 227, 229, 81, 91, 59, 189], UNIFORMSPARSESTCUT [21], ordering problems [61], and discrete optimization problems [63, 9].

1.3.1 Relaxation hierarchies and the implication of their limitations

In this thesis, we are mainly interested in LP/SDP-based algorithms when talking about convex relaxations. When designing LP/SDP-based algorithms, one usually uses the most natural and simple LP/SDP relaxation of the integer programming formulation of the original problem. We refer to these relaxations as basic LP/SDP relaxations. As we have defined before, fix a combinatorial optimization problem and a convex relaxation, the integrality gap serves as a measure of the quality of the relaxation, which is a different quantity from the hardness of approximation factor for the problem. However, for many important problems, the integrality gaps of even the simplest basic LP/SDP relaxations interestingly correspond to the hardness of approximation results [189, 152, 168].

On the other hand, for some other problems, in order to strengthen the algorithmic power, one can add additional constraints into the basic relaxation, so that the resulting relaxation is tighter and gives better approximation guarantee. One notable example is the work by Arora, Rao, and Vazirani [21] which added the so-called \( \ell^2 \)-triangle inequalities to the basic SDP relaxation for UNIFORMSPARSESTCUT and improved the approximation ratio from \( \Theta(\log n) \) to \( O(\sqrt{\log n}) \) where \( n \) is the size of the input graph.

While analysis of the convex relaxations with such extra constraints are very problem specific, there are several systematic ways to add additional constraints without even looking at the problem. Such systematic sets of constraints include the ones defined by Lovász and Schrijver [166], and Sherali and Adams [207] for LP, and the one defined by Parrilo and Lasserre [185, 157] for SDP. There are also hierarchies with a mixture of linear and semidefinite constraints, such as the Sherali-Adams+SDP relaxation hierarchy. In each of these ways, we obtain a sequence of increasingly powerful relaxations, which we often refer to as the hierarchy of convex relaxations. The relaxation at the \( r \)-th level (also called \( \text{round} \)) in the hierarchy typically has \( n^{O(r)} \) additional constraints (and auxiliary variables), and can be solved in \( n^{O(r)} \) time. Please refer to [66] for a more comprehensive introduction and comparison of these relaxation hierarchies.

Among these relaxation hierarchies, the Parrilo–Lasserre SDP hierarchy is the most powerful. Most of the known LP/SDP relaxation-based algorithms can be derived from at most the 4th level of the hierarchy (including the Arora-Rao-Vazirani algorithm for UNIFORMSPARSESTCUT). Given this, it is natural to study the limitations of the Parrilo–
Lasserre hierarchy for optimization problems where no concrete hardness of approximation result is proven. In this thesis, we will prove several new integrality gap results for the Parrilo–Lasserre hierarchy for the problems where no concrete \( \text{NP} \)-hardness result is known. Such results will help us understand the complexity of approximating those problems.

Prior to our work, several of the known results on strong integrality gap results for many rounds of the Parrilo–Lasserre hierarchy, starting with the remarkable construction by Grigoriev \cite{97,99} and Schoenebeck \cite{204}, apply in situations where a corresponding \( \text{NP} \)-hardness result is already known. Thus they are not “prescriptive” of hardness. In fact, besides the results introduced later in this thesis, we are aware of only the following examples where a polynomial-round Lasserre integrality gap stronger than the corresponding \( \text{NP} \)-hardness result is known: \textsc{Max} \( k \)-\textsc{Csp} and \( k \)-\textsc{coloring} \cite{220}. Indeed, for all we know 4 rounds of the Parrilo–Lasserre hierarchy could improve the Goemans-Williamson algorithm for \textsc{Maxcut}, and therefore refute the famous Unique Games Conjecture \cite{136,141} (please refer to Section 2.3 for more on the conjecture).

A good portion of this thesis is devoted to the study of the power of the Parrilo–Lasserre SDP hierarchy. We will be using a crucial view of the Parrilo–Lasserre SDP hierarchy as the Sum-of-Squares algebraic proof system, and therefore also call the hierarchy as SOS hierarchy. Depending on the context, we will use Parrilo–Lasserre, Lasserre, SOS, and SOS/Lasserre interchangeably throughout the thesis.

### 1.4 A brief overview of contributions

A large part of this thesis is devoted to the study of the power the convex (LP and SDP) relaxation hierarchies – we explore both the effectiveness and limitations of this algorithmic framework. The remaining part of the thesis consists of a collection of approximation algorithms and hardness of approximation results, with an emphasis on the design of robust algorithms.

In this section, we briefly introduce the results included in this thesis. A more detailed list on these results can be found in Chapter 3.

In Part 1, we start off by showing the effectiveness of the Sherali-Adams LP relaxation hierarchy. Even given that the hierarchy does not use the power of semidefiniteness, we show that Sherali-Adams LP gives the state-of-the-art approximation guarantee for a large class of problems including the dense (and locally dense) constraint satisfaction problems and assignment problems.
Then we turn to study the limitations of the Parrilo–Lasserre hierarchy. We focus on the problems such as \textsc{DenseSubgraph}, \textsc{BalancedSeparator}, and \textsc{UniformSparsestCut}; and prove the Parrilo–Lasserre hierarchy integrality gaps which beats the known NP-hardness of approximation results. As mentioned in Section 1.3, such type of results are rare in literature and serve as an evidence that the problems are indeed hard to approximate.

To motivate the contributions in Part II we would like to mention another class of combinatorial optimization problems, including \textsc{UniqueGames} and \textsc{MaxCut}. These problems seem to be significantly easier than \textsc{DenseSubgraph}; however, the NP-hardness of approximation and the algorithmic bounds still do not match. While it is a general belief that \textsc{DenseSubgraph} is indeed very hard to approximate, whether the NP-hardness results for \textsc{UniqueGames} and \textsc{MaxCut} can be improved (or even an evidence for that) is of great interest in the field of approximations. Researchers have shown Sherali-Adams+SDP integrality gaps for these problems, which serve as the best evidence for the hardness of approximation. These integrality gap instances are the “hardest instances” in literature in the sense that they are resistant to the strong relaxation hierarchy (and perhaps the strongest excluding Parrilo–Lasserre). A natural question arise here is whether these gap instances are also resistant to the Parrilo–Lasserre hierarchy. An affirmative answer to this question would further consolidate our best evidence.

In Part II however, we show that these instances turn out to be easy for the Parrilo–Lasserre hierarchy, giving a negative answer to the question. This result is obtained by viewing the Parrilo–Lasserre hierarchy from a different perspective, namely as an algebraic proof system, instead of as a semidefinite programming. While this connection was brought up by Parrilo [185] and Lasserre [156][157] more than a decade ago, we first make use of it in the setting of combinatorial optimization problems. Our results are the first to separate the power of Parrilo–Lasserre from other hierarchies on \textsc{UniqueGames} and \textsc{MaxCut} and seriously question the possible optimality of the state-of-the-art algorithms for the two problems. We also hope that our proof techniques help to extend our (limited) understanding of the Parrilo–Lasserre hierarchy.

In the rest of this thesis, Part III consists of robust algorithms in different settings, such as a special class of constraint satisfaction problems (the ones with width-1), \textsc{MaxBisection}, and isomorphism detection for trees. Part IV is a collection of other approximation algorithms and hardness of approximation results. Please refer to Chapter 3 for a more detailed list of these results.
Chapter 2

Preliminaries

2.1 Problems studied in this thesis

In this section, we introduce the optimization problems studied in the thesis.

2.1.1 Constraint satisfaction problems

In a constraint satisfaction problem (CSP) with arity $k$ and alphabet set $\Sigma$, there is a set $V$ of $n$ variables, and a list of $m$ constraints, where each variable takes value from an finite set of alphabet $\Sigma$, while each constraint involves exactly $k$ variables (or at most $k$ variables). We also refer to this problem as $k$CSP.

We often talk about the special cases of a CSP where each constraint in the problem is from one of the several prefixed forms. For example, when the alphabet set is $\{\text{true}, \text{false}\}$, we can define the following CSPs.

- In 2-SAT, each constraint is of one of the forms $v_i \lor v_j$, $\bar{v}_i \lor v_j$, $v_i$, $\bar{v}_i \lor \bar{v}_j$.
- In MAXCUT, each constraint is of one of the forms $v_i \neq v_j$.
- In 3-SAT, each constraint is of one of the forms $v_i \lor v_j \lor v_k$, $v_i \lor v_j \lor \bar{v}_k$, $v_i \lor \bar{v}_j \lor v_k$, $\bar{v}_i \lor v_j \lor v_k$, $\bar{v}_i \lor \bar{v}_j \lor v_k$.
- In 2-LIN, each constraint is of one of the forms $x_i \oplus x_j$, $\bar{x}_i \oplus x_j$.
- In 3-LIN, each constraint is of one of the forms $x_i \oplus x_j \oplus x_k$, $\bar{x}_i \oplus x_j \oplus x_k$. 

13
• In HORN3-SAT, each constraint is of one the the forms $x_i, \bar{x}_i, x_i \rightarrow x_j, x_i \land x_j \rightarrow x_k$.

When the alphabet set becomes $\{0, 1, 2, \ldots, q - 1\}$, the following CSPs are also often used in this thesis.

• In 2-Lin($\mathbb{Z}_q$), each constraint is of one of the forms $x_i \pm x_j = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q - 1\}$).

• In 3-Lin($\mathbb{Z}_q$), each constraint is of one of the forms $x_i \pm x_j \pm x_k = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q - 1\}$).

• In $\Gamma$-2-Lin($\mathbb{Z}_q$), each constraint is of one of the forms $x_i - x_j = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q - 1\}$).

• In $\Gamma$-3-Lin($\mathbb{Z}_q$), each constraint is of one of the forms $x_i + x_j - x_k = c \mod q$ (for some $c \in \{0, 1, 2, \ldots, q - 1\}$).

Given a CSP $\Gamma$, the natural optimization task, called “Max$\Gamma$”, is to find an assignment to the variables such that the total weight of the satisfied constraints is maximized, where we assume that each constraint has a nonnegative weight, and the sum of all weights are 1. Fix a CSP instance $I$, we use $\text{val}(I)$ to denote the value of the optimal solution. We say that $I$ is “satisfiable” if $\text{val}(I) = 1$.

### 2.1.1 The UniqueGames problem

The UniqueGames (UG for short) problem is a special binary CSP with domain size $q$. For each constraint $e$, there is a bijection $\pi_e$ attached to it. The constraint $e$ is satisfied when the two corresponding variables $x_i$ and $x_j$ take values so that $\pi_e(x_i) = x_j$. The goal is to find an assignment to the variables so that the fraction of satisfied constraints is maximized. The famous Unique Games Conjecture informally states that UG is very hard to approximate, we will discuss a little more about the conjecture in Section 2.3.

### 2.1.2 Graph partitioning problems

Partitioning a graph into two (balanced) parts with few edges going across them is a fundamental optimization problem. Graph partitions or separators are widely used in many applications (such as clustering, divide and conquer algorithms, VLSI layout, etc). In this thesis, we will focus on the following prototypical objectives of graph partitioning.
**Definition 2.1.1 (BALANCEDSEPARATOR).** Given an undirected graph $G = (V, E)$ and $0 < \tau < .5$, the goal of the $\tau$ vs. $1 - \tau$ BALANCEDSEPARATOR problem is to find a set $A \subseteq V$ such that $\tau|V| \leq |A| \leq (1 - \tau)|V|$, while $\text{edges}(A, V \setminus A)$ is minimized. Here $\text{edges}(A, B)$ is the number of edges in $E$ that cross the cut $(A, B)$.

**Definition 2.1.2 (UNIFORMSPARSESTCUT).** Given an undirected graph $G = (V, E)$, the goal of the UNIFORMSPARSESTCUT problem is to find a set $\emptyset \subset A \subset V$ such that the sparsity

$$\frac{\text{edges}(A, V \setminus A)}{|A||V \setminus A|}$$

is minimized.

In some other cases, we are also interested in finding a partition that maximizes the number of edges those cross the partition (such as MAXCUT). In addition, we define the following problem which is MAXCUT plus a global cardinality constraint.

**Definition 2.1.3 (MAXBISECTION).** Given an undirected graph $G = (V, E)$, the goal of the MAXBISECTION problem is to find a set $A \subseteq |V|$ such that $|A| = \frac{1}{2}|V|$ (assume that $|V|$ is an even number), such that

$$\frac{\text{edges}(A, V \setminus A)}{|E|}$$

is maximized.

We may also use the density of the edges within a subset as the objective value (instead of counting the number of edges across the partition). In the light of this, we define

**Definition 2.1.4 (DENSEkSUBGRAPH).** Given an undirected graph $G = (V, E)$ and an integer $k$, the goal of the DENSEkSUBGRAPH problem is to find a subset $A \subseteq V$ such that $|A| = k$ and number of edges within $A$ is maximized.

### 2.1.3 Graph isomorphism and assignment problems

The GRAPHISOMORPHISM problem is one of the most intriguing and notorious problems in computational complexity (we will also refer to it as GISO for short); we refer to [149, 30, 24, 148, 69] for surveys. Together with FACTORING, it is one of the very rare problems in NP which is not known to be in P but which is believed to be not NP-hard [29, 49, 205] (according to standard complexity-theoretic assumptions).
The most well-known heuristic for GI is the Weisfeiler-Lehman (WL) algorithm \cite{222} and its “higher dimensional” generalizations. These heuristics, given two graphs, are always correct when the two graphs are isomorphic, but might misreport the nonisomorphic pairs of graphs as being isomorphic. The “k-dimensional generalization” WL$^k$ (see \cite{222,55} for discussion) runs in time $n^{O(k)}$ and is more and more powerful as $k$ grows larger (i.e. misreports less nonisomorphic pairs of graphs as being isomorphic). The $k$-dimensional generalization WL$^k$ runs in time $n^{O(k)}$ and is more and more powerful as $k$ grows larger (i.e. misreports less nonisomorphic pairs of graphs as being isomorphic). The WL$^k$ heuristic is very powerful. For example, it is known to work correctly in polynomial time for all graphs which exclude a fixed minor \cite{102}, a class which includes all graphs of bounded tree width or bounded genus. Spielman’s $2^{\tilde{O}(\sqrt{n})}$-time GI algorithm \cite{212} for strongly regular graphs is achieved by WL$^k$ with $k = \tilde{O}(\sqrt{n})$. The WL$^k$ algorithm with $k = O(\sqrt{n})$ is also a key component in the $2^{O(\sqrt{n} \log n)}$-time GI algorithm \cite{33}. Throughout the ’80s there was some speculation that GI might be solvable on all graphs by running the WL$^k$ algorithm with $k = O(\sqrt{n})$ of even $k = O(1)$. However this was disproved in the notable work of Cai, Frue, and Immerman \cite{55}, which showed the existence of pairs of nonisomorphic $n$-vertex graphs which are not distinguished by WL$^k$ unless $k = \Omega(n)$.

In this thesis proposal, we will study a potentially stronger algorithmic framework than the WL algorithm, and also study the GI problem from the approximation algorithms prospect and using tools from approximation algorithms design. In order to do this, we introduce several new definitions here.

**Definition 2.1.5.** Let $G$ and $H$ be nonempty $n$-vertex graphs. For $0 \leq \beta \leq 1$, we say that a permutation $\pi : V(G) \to V(H)$ is an $\alpha$-isomorphism if

$$\frac{|\{(u, v) \in E(G) : (\pi(u), \pi(v)) \in E(H)\}|}{\max\{|E(G)|, |E(H)|\}} \geq \alpha,$$

where $V(G)$ and $V(H)$ are the vertex sets of $G$ and $H$ respectively, and $E(G)$ and $E(H)$ are the edge sets of $G$ and $H$ respectively.

If there exists an $\alpha$-isomorphism between $G$ and $H$, we say that $G$ and $H$ are $\alpha$-isomorphic.

Observe that this definition is symmetric in $G$ and $H$. The two graphs are isomorphic if and only if they are 1-isomorphic. The classical GI problem is to check whether the two input graphs are 1-isomorphic. Now we introduce the following natural optimization version of the problem.

**Definition 2.1.6 (MAXGI).** Given two $n$-vertex graphs $G$ and $H$, the MAXGI problem is to find a permutation $\pi : V(G) \to V(H)$ such that $\pi$ is an $\alpha$-isomorphism and $\alpha$ is maximized.
The **Quadratic Assignment Problem** (QAP for short) is a natural generalization of the **MAXGISO** problem. We now define the even more generalized problem $k$AP, where the QAP problem is a special case when $k = 2$, as follows.

**Definition 2.1.7** ($k$AP). For an integer $k \geq 2$, an instance of the degree $k$ assignment problem ($k$AP) is given as $I = (V, \omega)$, where $V$ is the set of variables, $\omega$ is a distribution over $V^k \times V^k$. The goal is find a permutation $\pi$ of $V$ such that the value of $\pi$, defined as

$$
\text{val}(I, \pi) = n^k \Pr_{(U,W) \sim \omega}[\forall i \in \{1, 2, 3, \ldots, k\}: \pi(u_i) = w_i],
$$

is maximized. $U = (u_1, u_2, \ldots, u_k)$ and $W = (w_1, w_2, \ldots, w_k)$. We define the optimal value of $I$ to be $\text{val}(I) = \max_{\pi} \{\text{val}(I, \pi)\}$.

2.1.4 Dense and locally-dense instances

Given a CSP instance $I$ with arity $k$ and variable set $V$, let $\omega : V^k \to \mathbb{R}$ be the weights on the constraints. I.e. let $\omega(v_1, v_2, \ldots, v_k)$ be the weight on the constraint imposed on the $k$-tuple $(v_1, v_2, \ldots, v_k)$; let $\omega(v_1, v_2, \ldots, v_k) = 0$ if there is no such constraint. Since we assumed that the weights are nonnegative and sum up to 1, we can view $\omega$ as a probability distribution on $V^k$. We say the CSP instance $I$ is $\Delta$-dense if $\omega$ is $\Delta$-dense; say $I$ is $\Delta$-locally dense if $\omega$ is $\Delta$-locally dense. We also say an instance is dense or locally-dense if it is $O(1)$-dense or $O(1)$-locally dense. Here we define,

**Definition 2.1.8** (dense and locally dense distributions). Let $\omega$ be a probability distribution over a finite set $\Omega$. For $\Delta \in (0, 1]$, we say $\omega$ is $\Delta$-dense if for every $a \in \Omega$, it holds that $\Delta \cdot \omega(a) \leq \frac{1}{|\Omega|}$.

Let $\omega$ be a probability distribution over $V^k$. Let $d_i(v) = \Pr_{S \sim \omega}[S_i = v_i]$ be the probability that the $i$-th coordinate is $v$ under $\omega$. For $\Delta \in (0, 1]$, we say $\omega$ is $\Delta$-locally dense if for every $(v_1, \ldots, v_k) \in V^k$, it holds that

$$
\Delta \cdot \omega(v_1, \ldots, v_k) \leq \frac{1}{|V|^{k-1}} \sum_{1 \leq i \leq k} d_i(v_i).
$$

Since $d_i(v) = \sum_{S \in V^k : S_i = v} \omega(S)$, the RHS of the locally dense condition is equal to

$$
\sum_{1 \leq i \leq k} \mathbb{E}_{S \sim V^k}[\omega(S) \mid S_i = v_i].
$$

Thus the locally dense condition says that no tuple $(v_1, \ldots, v_k)$ is “wild” in that $\omega(v_1, \ldots, v_k)$ is at most constant times the sum over $i$ of the average probability mass of $S$ with $S_i = v_i$. 

17
The notion of local density is introduced in [75] to generalize the metric condition. To see this, suppose \( \omega : V^2 \to \mathbb{R} \) is a metric. Then, \( \omega \) is 1-locally dense since, for any \( u, v \in V \), we have
\[
\frac{1}{n}(d_1(u) + d_2(v)) = \frac{1}{n} \sum_w (\omega(u, w) + \omega(w, v)) \geq \frac{1}{n} \sum_w \omega(u, v) \geq \omega(u, v).
\]

2.2 Approximation and hardness of approximation

In this section, we define the notions of approximation algorithms and hardness of approximation.

Let us first fix a problem \( Q \) (which is usually computationally hard to calculate the exact optimal objective solution). We suppose that the problem \( Q \) is a maximization problem throughout this subsection, while the definitions given can be naturally adapted to minimization problems.

Fix an instance \( I \) from the problem \( Q \). We denote the optimal objective value of the problem to be \( \text{val}(I) \). Given an algorithm \( A \) for the problem \( G \), we use \( \text{val}_A(I) \) to denote the value of the solution output by \( A \) on input \( G \). Now we define the following measure of the quality of \( A \) based on the approximation ratio.

**Definition 2.2.1.** We say that an algorithm \( A \) is an \( \alpha \)-approximation algorithm \((0 \leq \alpha \leq 1)\) for the problem \( Q \) if for every instance \( I \) from \( Q \), we have
\[
\frac{\text{val}_A(I)}{\text{val}(I)} \geq \alpha.
\]

We also introduce the definition of polynomial-time approximation scheme (PTAS) where the algorithm gives arbitrarily (and constantly) good approximation to the problem in polynomial time.

**Definition 2.2.2.** Fix an optimization problem \( Q \), a PTAS is an algorithm which takes an instance \( I \) of \( Q \) of size \( n \) and a parameter \( \epsilon > 0 \), and in time \( T(\epsilon, n) \), outputs a solution that is \((1 - \epsilon)\)-approximation to the optimal solution of the instance, where for every \( \epsilon > 0 \) there exists a constant \( C = C(\epsilon) \) such that \( T(\epsilon, n) \leq O(n^C) \).

Similarly, a quasi-polynomial-time approximation scheme (quasi-PTAS) for \( Q \) is an algorithm which takes an instance \( I \) of \( Q \) of size \( n \) and a parameter \( \epsilon > 0 \), and in time \( T'(\epsilon, n) \), outputs a solution that is \((1 - \epsilon)\)-approximation to the optimal solution of the instance, where for every \( \epsilon > 0 \) there exists a constant \( C' = C'(\epsilon) \) such that \( T'(\epsilon, n) \leq 2^{C'(\log n)^{(1)}} \).
Sometimes we need a more refined definition than [Definition 2.2.1](#). Take the problem MAXCUT for example (please refer to [Section 2.1.1](#) for the precise definition of MAX-CUT). Currently the best known approximation algorithm for MAXCUT is by Goemans and Williamson [94]; and by [Definition 2.2.1](#), the Goemans-Williamson algorithm is an $\alpha_{GW}$-approximation algorithm (where $\alpha_{GW} \approx .878$). However, when the optimal solution in a MAXCUT instance cuts almost all the edges (say $(1 - \epsilon)$ of the edges), Goemans-Williamson algorithm guarantees to output a cut that cuts $(1 - O(\sqrt{\epsilon}))$ of the edges, while we expect a .878-approximation algorithm to output a cut only cutting .878$(1 - \epsilon)$ of the edges (which is much smaller than $(1 - O(\sqrt{\epsilon}))$ when $\epsilon$ is small). Therefore, we need to introduce the following refined notion of approximation to address this difference.

**Definition 2.2.3.** Fix $c \geq s \geq 0$, we say that an algorithm $A$ is a $(c, s)$-approximation algorithm for the problem $Q$ if for every instance $I$ from $Q$, when $\text{val}(I) \geq c$, we have $\text{val}_A(I) \geq s$.

By definition [Definition 2.2.3](#), an $\alpha$-approximation algorithm is a $(c, \alpha c)$-approximation algorithm for every $c > 0$. The Goemans-Williamson algorithm is an $\alpha_{GW}$-approximation algorithm for MAXCUT in general; but it is also a $(1 - \epsilon, 1 - O(\sqrt{\epsilon}))$-approximation algorithm.

[Definition 2.2.3](#) motivates us to define the following decision problem for every optimization problem $Q$.

**Definition 2.2.4.** Given an optimization problem $Q$, for every $c \geq s > 0$, let the problem $(c, s)$-gap-$Q$ be the problem that given an instance $I$, to

- output YES when $\text{val}(I) \geq c$;
- output NO when $\text{val}(I) < s$.

A simple observation is that fix the problem $Q$ and the parameters $c \geq s > 0$, if $(c, s)$-gap-$Q$ is NP-hard, then it is NP-hard to $\frac{c}{s}$-approximate the problem $Q$. Therefore, a usual strategy of proving hardness of approximation statement for an optimization problem $Q$ is to prove the hardness of $(c, s)$-gap-$Q$ problem.

### 2.2.1 Robust algorithms

Robust algorithms are approximation algorithms concerned with the case that the problem has an “almost perfect” solution (e.g. when all the constraints are satisfied, when two
graphs are isomorphic, etc.). At a high level, instead of giving approximation ratio guarantee for general inputs, robust algorithms extremely well-approximate an almost perfect solution when such solutions exist. Such algorithms can be viewed as a robust version of the algorithms designed to find (exact) perfect solutions. One motivation to design robust algorithms is that in practical situations (e.g. learning with noise), instances with perfect solutions might be corrupted by a small amount of noise; an robust algorithm becomes useful since it can still satisfy most of the constraints of the noisy instance.

The notion of robust algorithms was first explicitly introduced by Zwick [228] for constraint satisfiability problems (CSPs), where he showed robust algorithms for several CSPs including MAX2SAT and MAXHORN SAT. In this proposal, we will focus on the robust algorithms for general CSPs, MAXBISECTION, and GRAPH ISOMORPHISM. Now we give the explicit definitions for these robust algorithms.

**Definition 2.2.5 (RobustSatisfiability).** Fix a CSP, we say that an algorithm $A$ is a robust satisfiability algorithm for the CSP if there exists a function $r : [0, 1] \rightarrow [0, 1]$ satisfying $r(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0^+$ such that whenever $A$ is given an instance $\mathcal{I}$ with $\text{val}(\mathcal{I}) \geq 1 - \epsilon$, $A$ outputs a solution satisfying $(1 - r(\epsilon))$ of the constraints.

By definition, the famous Goemans-Williamson [94] algorithm for MAXCUT is an robust satisfiability algorithm with $r(\epsilon) = O(\sqrt{\epsilon})$. Zwick [228] gave an robust satisfiability algorithm for MAX2SAT with $r(\epsilon) = O(\sqrt{\epsilon})$, and an robust satisfiability algorithm for MAXHORN SAT with $r(\epsilon) = O\left(\frac{\log \log(1/\epsilon)}{\log(1/\epsilon)}\right)$.

We also define the notion of robust algorithms for the following two problems which will be studied in this thesis.

**Definition 2.2.6 (RobustMaxBisection).** We say that an algorithm $A$ solves the RobustMaxBisection problem if there exists a function $r : [0, 1] \rightarrow [0, 1]$ satisfying $r(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0^+$ such that whenever $A$ is given an undirected graph $G$ with MAXBISECTION optimum at least $1 - \epsilon$, $A$ outputs a bisection with $(1 - r(\epsilon))$ of the edges across the bisection.

**Definition 2.2.7 (RobustGiso).** We say that an algorithm $A$ solves the RobustGiso problem if there exists a function $r : [0, 1] \rightarrow [0, 1]$ satisfying $r(\epsilon) \rightarrow 0$ as $\epsilon \rightarrow 0^+$ such that whenever $A$ is given a pair of graphs which are $(1 - \epsilon)$-isomorphic, $A$ outputs a $(1 - r(\epsilon))$-isomorphism between them.
2.3 The Unique Games Conjecture

The Unique Games Conjecture states,

**Conjecture 1** (Unique Games Conjecture [136]). *For every constant $\epsilon > 0$, there exists a large enough $q$ so that the $(1 - \epsilon, \epsilon)$-gap-UG problem with domain size $q$ is NP-hard.*

The Unique Games Conjecture, soon after its initial proposal Khot [136], became a central problem in approximation algorithms research. If the conjecture is true, it implies optimal inapproximability results for many problems including the broad class of CSPs [189], covering and packing problems [152], ordering CSPs [106] and MULTIWAY CUT [168]. While numerous research has been conducted to investigate the correctness of the conjecture (e.g. [136] 218 144 105 59 19 16 193), the status of the conjecture remains a major open question in the field, and there is no compelling opinion about its truth. However, through these intensive studies of UG, many connections have emerged between the conjecture, analysis, geometry and mathematical programming, leading to many exciting advancements in both algorithm design and hardness of approximations. The readers are encouraged to refer to [138] for more on the conjecture.
Chapter 3

Summary of contributions and organization of this thesis

The technical part of this thesis consists of 4 parts.

Part I and Part II will discuss how to strengthen our current (possibly limited) understanding of convex relaxation hierarchies. Table 3.1 is a list of the relevant results.

Part III will be on the design (and impossibility of designing) robust algorithms. Part IV will consist of a few other inapproximability results. Table 3.2 is a list of the relevant results.

Finally, the thesis will be concluded by Part V with a few future directions.

In the following 4 sections, we will give overviews of the 4 technical parts respectively.

3.1 Overview of Part I: study of the LP/SDP relaxation hierarchies

As we mentioned before, the LP/SDP relaxation hierarchies are parameterized by an integer \( r \) – the level in the hierarchy. One challenge here is to understand the trade-off between the approximation guarantee and the number of levels. In this subsection, we describe the results related to this question. For some problems, we prove that a small number of levels in the hierarchies effectively approximates the optimal solution; in other cases, we show lower bounds for the hierarchies, i.e. a large number of levels is needed to obtain good approximation.
### 3.1.1 Algorithmic results

In Chapter 4, we study the algorithmic guarantee of the Sherali-Adams LP relaxation hierarchy for dense and locally dense CSPs and assignment problems (APs). Prior to our work, there was a long series of works (e.g., [74, 87, 18, 77, 17, 76, 75]), using various techniques, such as sampling, regularity lemma, and tensor decomposition, to design PTAS and quasi-PTAS for these instances.

We show that the Sherali-Adams LP relaxation hierarchy is a unified algorithmic framework to obtain all the previously known results. In particular, we first show that

**Theorem 3.1.1** (Pre-statement of Theorem 4.1.1). For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation hierarchy gives $(1 - \epsilon)$-approximation to dense or locally dense $\text{MAX}^k\text{CSP}$.

Then, we turn to dense and locally dense $\text{MAX}^k\text{CSP}$ with global cardinality constraints. For explanatory purposes, we only consider bisection constraint, i.e., the domain is $\{0, 1\}$ and the number of variables that are assigned to 0 should be equal to the number of variables that are assigned to 1. We show that

**Theorem 3.1.2** (Pre-statement of Theorem 4.8.2). For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation hierarchy gives $(1 - \epsilon)$-approximation to dense or locally dense $\text{MAX}^k\text{CSP}$.
Approximation algorithms | Hardness of approximation
---|---
Robust algorithm for width-1 constraint satisfiability problems (Theorem 3.3.1) | Hardness of ROBUSTGISO (Theorem 3.3.3)
An algorithm for ROBUSTMAXBISECTION (Theorem 3.3.2) | Hardness of MAXΓ-2-LIN and MAXΓ-3-LIN over integers (Theorem 3.4.3)
A robust isomorphism algorithm for trees (Theorem 3.3.4) | Hardness of approximating almost satisfiable MAXHORN3-SAT (Theorem 3.4.4)
Approximating the $2 \to 4$ norm of random linear operators (Theorem 3.4.1) |

Table 3.2: Table of contributions on other approximation algorithms and hardness of approximation

Finally, we consider the dense MAX$k$CSP.

*bisection MAX$k$CSP.*

In Chapter 5, Chapter 6, and Chapter 7, we study several important combinatorial optimization problems and show that the strongest known SDP hierarchy (i.e. the Parrilo–Lasserre hierarchy) does not give good approximation for them. Given that there is no concrete inapproximability result for these problems, and (as pointed out previously) that our results are among the few ones proving Parrilo–Lasserre lower bounds beating known NP-hardness results, our lower bound theorems can be viewed as strong evidence of the inapproximability of these fundamental combinatorial optimization problems.

**The Dense$k$Subgraph problem.** The Dense$k$Subgraph problem is believed to be very hard to approximate as the best known approximation algorithm due to [41]
gives \( O(n^{1/4+\epsilon}) \)-approximation in time \( n^{O(1/\epsilon)} \) for any constant \( \epsilon > 0 \). On the inapproximability side, \cite{88} initially showed that a small constant factor inapproximability for \textsc{DenseSubgraph} using the random 3-SAT assumption. \cite{137} used quasi-random PCPs to rule out a PTAS. More recently, \cite{193, 7} used more non-standard assumptions to rule out any constant factor approximation algorithms. In the work with Bhaskara et al. \cite{40}, we showed the following integrality gap theorem.

**Theorem 3.1.4** (Pre-statement of Theorem 5.1.1). For every \( \epsilon > 0 \), there is a lower bound of \( n^{2/53-\epsilon} \) on the integrality gap of level-\( n^{\Omega(\epsilon)} \) Parrilo–Lasserre SDP relaxation hierarchy for the \textsc{DenseSubgraph} problem; there is also a lower bound of \( n^\epsilon \) on the integrality gap of level-\( n^{1-O(\epsilon)} \) Parrilo–Lasserre SDP relaxation hierarchy.

**The BalancedSeparator and UniformSparsestCut problems.** For these two problems, the best algorithms, based on semidefinite relaxations (SDPs) with triangle inequalities, give \( O(\sqrt{\log n}) \)-approximation \cite{21}. On the inapproximability side, a Polynomial Time Approximation Scheme (PTAS) is ruled out for both problems assuming 3-SAT does not have randomized subexponential-time algorithms \cite{10}. In the work with Guruswami and Sinop \cite{112}, we showed the following integrality gaps for the two problems.

**Theorem 3.1.5** (Pre-statement of Theorem 6.1.2). For \( 0.45 < \tau < 0.5 \), there are linear-round Parrilo–Lasserre SDP gap instances for the \( \tau \) vs \( (1-\tau) \) BalancedSeparator problem, such that the integral optimal solution is at least \( (1+\epsilon(\tau)) \) times the SDP solution, where \( \epsilon(\tau) > 0 \) is a constant dependent on \( \tau \).

**Theorem 3.1.6** (Pre-statement of Theorem 6.1.3). There are linear-round Parrilo–Lasserre SDP gap instances for the UniformSparsestCut problem, such that the integral optimal solution is at least \( (1+\epsilon) \) times the SDP solution, for some constant \( \epsilon > 0 \).

**The GraphIsomorphism problem.** A recent work of Atserias and Maneva \cite{27} (see also \cite{103}) shows that the power of WL\(_k\) algorithm is precisely sandwiched between the \( k\)-th and \( (k+1)\)-st level of the canonical Sherali-Adams LP relaxation hierarchy of the GI\_SO problem. Given the power of WL\(_k\) algorithm, this connection shows that LP relaxation hierarchies are also useful for solving GI\_SO. On the other hand, by the work of \cite{55}, we also know that the Sherali-Adams LP relaxation hierarchy also needs linearly many levels to fully solve GI\_SO.

This raises the natural question whether stronger LP/SDP relaxation hierarchies might prove more powerful than WL\(_k\) in the context of GI\_SO. In the work with O’Donnell et
al. [182], we study the Parrilo–Lasserre SDP relaxation hierarchy (which is the strongest known hierarchy known in the literature as discussed before) for GI and show that

**Theorem 3.1.7** (Pre-statement of Theorem 7.1.2). For infinitely many \( n \), there exists pairs of \( n \)-vertex, \( O(n) \)-edge graphs \( G \) and \( H \) such that

- \( G \) and \( H \) are not \((1 - 10^{-14})\)-isomorphic;
- in order to tell that \( G \) and \( H \) are not 1-isomorphic (i.e. isomorphic), the Parrilo–Lasserre SDP relaxation hierarchy needs \( \Omega(n) \) levels.

This theorem says that the linear-level Parrilo–Lasserre hierarchy not only fails on distinguishing nonisomorphic pairs of graphs, but also fails spectacularly – the hierarchy cannot tell the two graphs are nonisomorphic even when they are different by a constant fraction of the edges.

### 3.2 Overview of Part II: using the Parrilo–Lasserre hierarchy to solve hard instances for weaker hierarchies

In Part II, we study the Parrilo–Lasserre SDP relaxation hierarchy when applied to the known integrality gap instances (for other relaxation hierarchies such as Sherali-Adams+SDP) in literature for several central combinatorial optimization problems, and will show that these instances are no longer integrality gap instances for constant-level Parrilo–Lasserre SDP relaxation hierarchy. In order to obtain such types of results, we will use a special and novel view of the Parrilo–Lasserre SDP relaxation hierarchy, i.e. to view the hierarchy as the so-called “sum-of-squares proof system” and to prove the success of the hierarchy (on given instances) via giving a proof that the given instance does not have great objective value in the sum-of-squares proof system. Using this connection, we hope to understand more about the the power of Parrilo–Lasserre SDP relaxation hierarchy, and proof techniques that might be helpful to construct integrality gaps for the Parrilo–Lasserre hierarchy. In particular, we will present the following results along this line.

**The UNIQUEGAMES problem.** We begin with showing that a very small constant level of the Parrilo–Lasserre SDP hierarchy suffices to solve the UNIQUEGAMES instances in the literature.
Theorem 3.2.1 (Pre-statement of Theorem 9.0.8). For sufficiently small $\epsilon$ and large $k$, and every $n \in \mathbb{N}$, let $W$ be an $n$-variable $k$-alphabet UNIQUEGAMES instance of the type considered in [192, 154, 142] obtained by composing the “quotient noisy cube” instance of [144] with the long-code alphabet reduction of [141] so that the best assignment to $W$’s variables satisfies at most an $\epsilon$ fraction of the constraints. Then, there is a degree-8 SOS refutation for the statement that the best assignment to $W$’s variables satisfy at least $1/100$ fraction of the constraints.

Thus just the level-4 Lasserre SDP hierarchy (essentially) solves the the UNIQUEGAMES instances.

The BalancedSeparator problem. Devanur et al [79] gave a family of $n$-vertex BalancedSeparator instances (which we will refer to as the DKSV instances) which are integrality gap instances with ratio $\Theta(\log \log n)$ for the natural SDP relaxation with triangle inequalities. Raghavendra and Steurer [188] showed that a factor-$(\log \log n)^\Omega(1)$ gap persists for these instances even for $(\log \log n)^\Omega(1)$ rounds of the “LH+SDP relaxation hierarchy”. In the work with O’Donnell [184], we show that

Theorem 3.2.2 (Corollary of Theorem 10.3.1 and Theorem 10.3.3). The level-2 Parrilo–Lasserre SDP relaxation hierarchy for the BalancedSeparator problem has integrality gap at most $O(1)$ for the DKSV instances.

The MaxCut problem. Assuming the Unique Games Conjecture, Khot et al. [141] showed that the Goemans-Williamson algorithm [94] for MaxCut achieves the best possible approximation factor, namely $\alpha_{GW} \approx .878$ approximation. Khot and Vishnoi [144] gave integrality gap instances of ratio $\alpha_{GW}$ for the MaxCut problem, by composing their UNIQUEGAMES instances with the MaxCut reduction in [141]. Khot and Saket [154] subsequently showed that this gap persists even for level-$(\log \log \log n)^\Omega(1)$ Sherali-Adams+SDP relaxation hierarchy. In the work with O’Donnell [184], we show that constant level of the Parrilo–Lasserre SDP relaxation hierarchy gives better than $\alpha_{GW}$-approximation to the integrality gap instances by Khot and Vishnoi. In particular, we prove that

Theorem 3.2.3 (Pre-statement of Theorem 8.1.3). There exists a universal integer constant $C$ such for the level-$C$ Parrilo–Lasserre SDP relaxation hierarchy, the Khot-Vishnoi MaxCut instance has integrality gap ratio at most $1/.952(1/\alpha_{GW})$. 

28
3.3 Overview of Part III: Robust Algorithms

In this subsection, we present several of our related results on robust algorithms.

Robust algorithms for satisfiability problems. As mentioned previously, Zwick [228] showed robust algorithms for MAX2SAT and MAXHORN SAT. In sharp contrast, for other problems such as MAX3LIN, although deciding whether an instance is satisfiable is in $\textsf{P}$, there is no efficient robust algorithm unless $\textsf{P} = \textsf{NP}$ [116]. A natural theoretical question arising at this point is to characterize the class of CSPs that admit efficient robust algorithms.

Towards answering this question, in a work with Guruswami [114], we conjectured that the CSPs which have efficient robust algorithms (assuming $\textsf{P} \neq \textsf{NP}$) are precisely those of "bounded width" – a notion frequently used in algebraic dichotomy theory where people study the characterization of CSPs with exact satisfiability algorithms (we will refer to this conjecture as Guruswami–Zhou Conjecture throughout this thesis). Roughly speaking, bounded-width CSPs are the ones do not encode linear equations over abelian groups; they also coincide with the CSPs solvable by the "$k$-consistency heuristic" algorithm in artificial intelligence. If our conjecture is true, the natural basic SDP relaxation would be the desired robust algorithm for every bounded-width CSP.

Towards proving the conjecture, in Chapter 12, we prove the following theorem.

Theorem 3.3.1 (Pre-statement of Theorem 12.1.1). If a CSP has "width-1", there is a polynomial-time robust satisfiability algorithm for the CSP; and the algorithm is based on the natural LP relaxation of the problem.

The Guruswami–Zhou Conjecture was later fully confirmed by Barto and Kozik [38].

The ROBUSTMAXBISECTION problem. In Chapter 13, we will prove the first polynomial-time algorithm for the ROBUSTMAXBISECTION problem. Our theorem is stated as follows.

Theorem 3.3.2 (Pre-statement of Theorem 13.1.3). There is a randomized polynomial-time algorithm such that for every $\epsilon > 0$, given an edge-weighted graph $G$ with a MAXBISECTION solution of value $\frac{1}{1-\epsilon}$, finds a MAXBISECTION of value $1 - O(\sqrt{\epsilon \log(1/\epsilon)})$.

The value of a cut in an edge-weighted graph is defined as the weight of the edges crossing the cut divided by the total weight of all edges.
Prior to our work ([107]), researchers from theoretical computer science and operations research designed various algorithms for this problem [93, 226, 115, 91], but none of them is guaranteed to find a bisection cutting most of the edges even when the graph has a near-perfect bisection cutting \((1 - \epsilon)\) of the edges (in fact, they may not even cut 75% of the edges).

**The ROBUSTGISO problem.** Although it is not known whether GISO has polynomial-time algorithm, we show that assuming the so-called Feige’s R3XOR hypothesis [88], there is no polynomial-time algorithm for ROBUSTGISO. In particular, we prove the following theorem.

**Theorem 3.3.3** (Pre-statement of Theorem 7.1.4). Assume Feige’s R3XOR Hypothesis [88]. Then there is no polynomial-time algorithm for ROBUSTGISO. More precisely, there exists \(\epsilon_0 > 0\), such that suppose there exists \(\epsilon > 0\) and a \(t(n)\)-time algorithm which can distinguish \((1 - \epsilon)\)-isomorphic \(n\)-vertex, \(m\)-edge graph pairs from pairs which are not even \((1 - \epsilon_0)\)-isomorphic (where \(m = O(n)\)). Then there is a universal constant \(\Delta \in \mathbb{Z}^+\) and a \(t(O(n))\)-time algorithm which outputs “typical” for almost all \(n\)-variable, \(\Delta n\)-constraint instances of the 3-XOR problem, yet which never outputs “typical” on instances which are \((1 - \Theta(\epsilon))\)-satisfiable.

Our proof uses a linear-size reduction from ROBUSTGISO to MAX3-LIN and uses a hardness of approximation result for MAX3-LIN by Håstad [116]. Therefore, by the efficient-construction of the PCP theorem [173], and assuming the ETH ([124], i.e. 3-SAT does not have subexponential-time algorithms), there is no subexponential-time algorithm for ROBUSTGISO. This is in contrast with the \(2^{O(\sqrt{n \log n})}\)-time algorithm for GISO due to [33].

Despite the impossibility of constructing efficient ROBUSTGISO algorithm for general graphs, we design a ROBUSTGISO algorithm for trees. In Chapter 14, we prove

**Theorem 3.3.4** (Pre-statement of Theorem 14.1.5). Given two \(n\)-vertex trees \(T_1\) and \(T_2\), there is a polynomial-time algorithm to find a \((1 - O(\epsilon^{1/4}))\)-isomorphism between them whenever they are \((1 - \epsilon)\)-isomorphic, for every \(\epsilon > 0\).
3.4 Overview of Part IV: other approximation and hardness of approximation results

In this subsection, we describe several related hardness of approximation results obtained by the author.

3.4.1 Approximating the $2 \to 4$ norm of random linear operators

For a function $f : \Omega \to \mathbb{R}$ on a (finite) probability space $\Omega$, the $p$-norm is defined as $\|f\|_p = (E_{\Omega} f^p)^{1/p}$. The $p \to q$ norm $\|A\|_{p \to q}$ of a linear operator $A$ between vector spaces of such functions is the smallest number $c \geq 0$ such that $\|Af\|_q \leq c \|f\|_p$ for all functions $f$ in the domain of $A$.

In Chapter 15, we are interested in approximating $\|A\|_{2 \to 4}$. We study a natural semidefinite programming (SDP) relaxation for computing the $2 \to 4$ norm of a given linear operator which we call TensorSDP. While TensorSDP is very unlikely to provide a polynomial-time constant-factor approximation for the $2 \to 4$ norm in general (as shown in [34]), we do show that it provides such approximation on random linear operators, as we describe below.

**Theorem 3.4.1** (Informal version of Theorem 15.3.1). TensorSDP certifies a constant upper bound on the ratio $\|A\|_{2 \to 4}/\|A\|_{2 \to 2}$ where $A : \mathbb{R}^n \to \mathbb{R}^m$ is a random linear operator (e.g., obtained by a matrix with entries chosen as i.i.d Bernoulli variables) and $m \geq \Omega(n^2)$.

In contrast, if $m = o(n^2)$ then this ratio is $\omega(1)$, and hence this result is almost tight in the sense of obtaining “good approximation” in the sense mentionen above.

3.4.2 Hardness of Max$\Gamma$-2-Lin and Max$\Gamma$-3-Lin over integers

To describe our contributions in Chapter 16, we first define the Max$\Gamma$-2-Lin and Max$\Gamma$-3-Lin problems over $\mathbb{Z}$ as follows.

**Definition 3.4.2.** In a Max$\Gamma$-2-Lin ($\mathbb{Z}$) instance, there are $n$ variables $x_1, x_2, \ldots, x_n \in \mathbb{Z}$, and $m$ equations where each equation is in the form $x_i - x_j = c$ where $c \in \mathbb{Z}$. The goal is to find an assignment of the variables such as the fraction of the satisfied equations is maximized.
In a $\text{MAX-3-LIN}(\mathbb{Z})$ instance, there are $n$ variables $x_1, x_2, \ldots, x_n \in \mathbb{Z}$, and $m$ equations where each equation is in the form $x_i + x_j - x_k = c$ where $c \in \mathbb{Z}$. The goal is to find an assignment of the variables such as the fraction of the satisfied equations is maximized.

Observe that by definition $\text{MAX-2-LIN}(\mathbb{Z})$ and $\text{MAX-3-LIN}(\mathbb{Z})$ are not ordinary CSPs since the domain size is not constant (even not finite).

In a seminal result by Håstad [116], it was proved that for every constant $\epsilon > 0$, the $(1 - \epsilon, \epsilon)$-gap-$\text{MAX-3-LIN}(\mathbb{Z}_q)$ problem is NP-hard. Khot et al. [141] proved a similar hardness of approximation result for the $\text{MAX-2-LIN}(\mathbb{Z}_q)$ problem. They proved that for every constant $\epsilon$, there exists $q > 0$ such that there is no polynomial-time algorithm for the $(1 - \epsilon, \epsilon)$-gap-$\text{MAX-2-LIN}(\mathbb{Z}_q)$ problem, assuming the Unique Games Conjecture.

Guruswami and Raghavendra [109] later extended Håstad’s theorem to the $\text{MAX-3-LIN}(\mathbb{Z})$ problem, proving that for every constant $\epsilon > 0$, the $(1 - \epsilon, \epsilon)$-gap-$\text{MAX-3-LIN}(\mathbb{Z})$ problem is NP-hard.

We will give a simplified proof of Guruswami and Raghavendra’s result, and prove an analogue of the theorem for $\text{MAX-3-LIN}(\mathbb{Z})$ assuming the Unique Games Conjecture. In particular, we prove the following theorem.

**Theorem 3.4.3.** For every constant $\epsilon > 0$, the $(1 - \epsilon, \epsilon)$-gap-$\text{MAX-3-LIN}(\mathbb{Z})$ problem is NP-hard. The proof of this theorem simplifies the proof of the similar result by Guruswami and Raghavendra.

Assuming the Unique Games Conjecture, for every constant $\epsilon > 0$, there is no polynomial-time algorithm for the $(1 - \epsilon, \epsilon)$-gap-$\text{MAX-2-LIN}(\mathbb{Z})$ problem.

3.4.3 Hardness of approximating almost satisfiable $\text{MAXHORN3-SAT}$

Zwick [228] gave an algorithm for the almost satisfiable $\text{MAXHORN3-SAT}$ problem. When the input instance is $(1 - \epsilon)$-satisfiable, Zwick’s algorithm finds an assignment satisfying a $\left(1 - O \left( \frac{1}{\log(1/\epsilon)} \right) \right)$-fraction of the constraints.

In Chapter 17, we will prove that Zwick’s algorithm is essentially optimal assuming the Unique Games Conjecture. In particular, we proved the following theorem.

**Theorem 3.4.4** (Pre-statement of Theorem 17.2.1). Assuming the Unique Games Conjecture [136], for some absolute constant $C$, for every $\epsilon > 0$, given a $(1 - \epsilon)$-satisfiable instance of $\text{MAXHORN3-SAT}$, there is no polynomial-time algorithm to find an assignment satisfying more than a fraction of $\left(1 - \frac{C}{\log(1/\epsilon)} \right)$ of the constraints.
Part I

Study of the LP/SDP relaxation hierarchies
Chapter 4

Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems

4.1 Introduction

Recall that in a maximum constraint satisfaction problem (MAX-CSP), given a variable set $V$ over the domain $D$ and a set of constraints $C$ over the variables in $V$, we want to find an assignment $\alpha : V \rightarrow D$ that maximizes the fraction of constraints satisfied by $\alpha$. MAX-CSP includes many fundamental problems such as MAX-CUT and MAX-SAT.

In general, MAX-CSP is NP-Hard, and it is even NP-Hard to approximate within a constant factor [20]. However, de la Vega [74] showed that there is a polynomial-time approximation scheme for MAX-CUT if the input graph is dense, i.e., it has $\Omega(n^2)$ edges. Here, a polynomial-time approximation scheme (PTAS) is an algorithm that, given $\epsilon > 0$ as a parameter, gives a $(1 - \epsilon)$-approximation to the optimal value, and runs in polynomial time for any constant $\epsilon$. MAX$k$CSP is a subproblem of MAX-CSP, in which each constraint involves at most $k$ variables, where $k$ is a constant. Arora et al. [18] and Frieze and Kannan [87] showed PTASs for dense MAX$k$CSP, i.e., the input instance has $\Omega(n^k)$ constraints. Now it is known that we can compute $(1 - \epsilon)$-approximation to the optimal value in time that depends only on $k$ and $\epsilon$ [8].

There are two directions to generalize PTASs for dense MAX$k$CSP. The first one is
to generalize the notion of the density condition. We say that an instance of Max-2-CSP is *metric* if the weights of the constraints form a metric. Max-Cut \[77\] and Max-Bisection \[76\] admit PTASs if the instance is metric. The notion of local density is introduced to generalize the notion of metric to constraints over more than two variables. If the instance is locally dense, Max\(k\)CSP admits PTASs \[75\].

The second direction is to handle the maximum assignment problems (Max-AP). In this problem, given a variable set \(V\) and a set of constraints, we want to find a permutation \(\pi\) of \(V\) to maximize the fraction of satisfied constraints. Max-AP includes many fundamental problems such as MaximumAcyclicSubgraph, Betweenness, MaximumGraph Isomorphism, Dense\(k\)Subgraph, and QuadraticAssignmentProblem. Max\(k\)AP is a special case of Max-AP, in which each constraint involves at most \(k\) variables (see Section 4.2 for the precise definition). We say that an instance of Max\(k\)AP is dense if it has \(\Omega(n^k)\) constraints. Arora et al. \[17\] showed a quasi-polynomial-time approximation scheme for dense Max\(k\)AP and PTASs for many special cases.

As we have seen, Max-CSP and Max\(k\)AP admit PTASs (or quasi-PTASs) in the dense case and the locally dense case. However, the techniques to obtain them vary a lot. For example, \[18\] is based on the idea of exhaustively trying all assignments for a small number of variables and then solving the rest using the partial assignment. On the other hand, \[87\] used a variant of Szemerédi’s regularity lemma \[216\]. To deal with the metric case, \[77\] used the method of copying important variables, and \[75\] considered a variant of singular value decomposition of tensors to deal with the locally dense case.

### 4.1.1 Linear Programming (LP) relaxation and LP relaxation hierarchies

Much about LP/SDP relaxations and their hierarchies are introduced at the beginning of this thesis. Here we would like to emphasize that LP relaxation and its hierarchies have found many connections to other known algorithmic frameworks, and to be a unified approach to solve several classes of problems. A few examples are listed as follows. Assuming the Unique Games Conjecture, a canonical LP relaxation (also referred to as the Basic LP) is shown to provide optimal approximation guarantee for CSPs with strict constraints \[152\]. It is known to the author that constant-round Sherali-Adams LP relaxation decides the satisfiability of bounded-width CSPs; Atserias and Maneva \[27\] recently showed that the Sherali-Adams LP relaxation hierarchy for graph isomorphism interleave with the levels of pebble-game equivalence with counting (i.e. higher-dimensional Weisfeiler-Lehman color refinement algorithm).
4.1.2 Our contributions

In this chapter, we present the Sherali-Adams LP relaxation hierarchy as a unified approach to dense and locally dense problems – we show that a small number of rounds of the Sherali-Adams LP relaxation gives an approximation scheme to dense $\text{MAX}_k\text{CSP}$ and all their variants studied in the previous works.

Our first main theorem deals with dense and locally dense $\text{MAX}_k\text{CSP}$.

**Theorem 4.1.1** (Informal version of Theorem 4.5.1). For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1 - \epsilon)$-approximation to dense or locally dense $\text{MAX}_k\text{CSP}$.

Then, we turn to dense and locally dense $\text{MAX}_k\text{CSP}$ with global cardinality constraints. For explanatory purposes, we only consider bisection constraint, i.e., the domain is $\{0, 1\}$ and the number of variables that are assigned to 0 should be equal to the number of variables that are assigned to 1. We show that

**Theorem 4.1.2** (Informal version of Theorem 4.8.2). For any $\epsilon > 0$, $O(\frac{1}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1 - \epsilon)$-approximation to dense or locally dense bisection $\text{MAX}_k\text{CSP}$.

Finally, we consider the dense $\text{MAX}_k\text{AP}$ problems, and show that

**Theorem 4.1.3** (Informal version of Theorem 4.5.2). For any $\epsilon > 0$, $O(\frac{\log n}{\epsilon^2})$-round Sherali-Adams LP relaxation gives $(1 - \epsilon)$-approximation to dense or locally dense $\text{MAX}_k\text{AP}$ problems with $n$ variables.

In all the precise theorem statements, we actually show additive approximation guarantee (i.e. the value of the rounded solution being at least the fractional optimal value minus a constant error) instead of multiplicative approximation guarantee. However, since we define the problems in a way that the optimal solution is $\Omega(1)$ (see Section 4.2 for the precise definition of the problems), an additive approximation scheme implies a multiplicative approximation scheme.

**New algorithmic guarantees.** Let us define the problem $\text{MAXIMUM}_k$-$\text{HYPERGRAPH ISOMORPHISM}$ as follows. Given two weighted $k$-uniform hypergraph $G = (V, \omega')$ and $H = (V, \omega'')$, where $\omega', \omega'' : V^k \to [0, 1]$ are the weight functions over all possible hyperedges. The goal is to find a permutation $\pi$ over $V$ so that $\sum_{e \in V^k} \omega'(e)\omega''(\pi(e))$ is maximized (where $\pi(e)$ is the edge obtained by applying $\pi$ on each incident vertex of $e$). It is easy to see **Theorem 4.1.3** implies that $O(\frac{\log n}{\epsilon^2})$-round
Sherali-Adams LP relaxation gives \((1 - \epsilon)\)-approximation to \textsc{MAXIMUM}\(k\)-\textsc{HYPERGRAPHISOMORPHISM} when both \(G\) and \(H\) are dense.

We are able to apply our analysis framework for the Sherali-Adams LP relaxation to another special case of \textsc{MAXIMUM}\(k\)-\textsc{HYPERGRAPHISOMORPHISM}, getting the following new algorithmic guarantee.

Theorem 4.1.4 (Informal version of Theorem 4.5.3). For any \(\epsilon > 0\), \(O\left(\frac{\log n}{\epsilon^2}\right)\)-round Sherali-Adams LP relaxation gives \((1 - \epsilon)\) approximation to the \textsc{MAXIMUM} \(k\)-\textsc{HYPERGRAPHISOMORPHISM} problem when one of the two graphs is locally dense and the other graph is dense, where \(n\) is the number of vertices in the hypergraphs. Therefore, this special case of the problem admits a \((1 - \epsilon)\)-approximation algorithm in time \(n^{O\left(\frac{\log n}{\epsilon^2}\right)}\).

### 4.1.3 Proof overview

The first step of our algorithms is to condition on a set of random variables in a solution to the Sherali-Adams LP relaxation. In the \(\ell\)-round \textsc{Sherali-Adams LP} relaxation (or the \textsc{SA} relaxation for short), for each set of variables \(S\) of size at most \(\ell\), we have a probability distribution \(\mu_S\) over assignments on \(S\). First we solve \((k + \ell)\)-round \textsc{SA} relaxation, where \(\ell\) is a parameter depending on the error parameter \(\epsilon\). Then, we randomly sample a set of variables \(u_1, \ldots, u_\ell\) and assign values to them by sampling values from \(\mu_{\{u_1\}}, \ldots, \mu_{\{u_\ell\}}\), respectively. By this conditioning, we obtain a solution to \(k\)-round Sherali-Adams relaxation \(\mu'\) with the same LP value in expectation. An important fact here is that variables become almost independent in the sense that, if we sample a \(k\)-tuple \((v_1, \ldots, v_k)\) according to a dense (or locally dense) distribution (this distribution corresponds to the weights of the constraints in \(k\)\textsc{CSP} and \(k\)\textsc{AP} instances), the distribution \(\mu_{\{v_1, \ldots, v_k\}}\) and the product distribution \(\mu_{\{v_1\}} \times \cdots \times \mu_{\{v_k\}}\) are close in expectation.

The second step of our algorithms is to round the solution to the \textsc{SA} relaxation where the variables are almost independent. For dense (or locally dense) \(k\)\textsc{CSP} and bisection \(k\)\textsc{CSP}, the rounding algorithm just samples a value from \(\mu_{\{v\}}\) and assigning it to \(v\) for each variable \(v\). It is relatively easy to show that the expected value of the sampled solution is close to the LP value, and therefore gives a \((1 - \epsilon)\)-approximation.

For \(k\)\textsc{AP} problems, however, such independent sampling method does not work – there might be more than one variables assigned to the same value and we do not get a permutation when this happens. Instead, we view the marginal probability distributions on single variables, \(\mu_{\{v\}}(w)\), as a doubly stochastic matrix. We view this doubly stochastic matrix as a probability distribution of permutations. We iteratively choose two permutations in
the support of the distribution and merge them into a new permutation, until there is only one permutation left in the support – which is the output of our rounding algorithm. The operation of merging two permutations is interestingly similar to the merging operation used in [17], although the purposes are different. See Section 4.4.2 for more details.

4.1.4 Comparison to previous works

We first compare the running time of our SA relaxation-based algorithms with the previously known counterparts. For MaxkCSP, the running time $n^{O(1/\epsilon^2)}$ of our method matches the one of the method by [18]. For MaxkAP the running time $n^{O(\log n/\epsilon^2)}$ of our method matches the one of the method by [17]. [17] improved the running time to $n^{O(1/\epsilon^2)}$ for various problems by reducing them to CSPs. We can use the same techniques to obtain the same running time for these problems.

The number of rounds ($O(\frac{1}{\epsilon^2})$) in Theorem 4.1.1 improves the corresponding theorem in [78] which showed that $\tilde{O}(\frac{1}{\epsilon^4})$-round SA relaxation gives $(1-\epsilon)$-approximation to dense Max-Cut.

The idea of conditioning variables of a solution to LP/SDP hierarchies is used in [195,36] to solve variants of Max2CSP. Let $G = (V,E)$ be the underlying graph of an instance of Max2CSP. Barak et al. [36] showed that (i) the covariance between $u$ and $v$ over $V^2$ gets close to zero by conditioning, and (ii) the covariance between $u$ and $v$ over $E$ gets close to the covariance between $u$ and $v$ over $V^2$ by conditioning if $G$ is expander-like. Combining these two results, they show a PTAS for Max2CSP when $G$ is expander-like. This method can be also applied to dense graphs, but it is not clear how to generalize it to metric graphs and kCSP.

Raghavendra and Tan [195] used mutual information instead of covariance to measure correlation between two variables and simplified the proof. They noticed that conditioning is useful to deal with global constraints such as cardinality constraints since after conditioning we can sample variables independently and the resulting solution will not break global constraints much. With this idea, they gave a 0.85-approximation algorithm for Max-Bisection. Though our method and analysis are similar to theirs, we use the independence for obtaining PTASs for the dense and locally dense case as well as supporting global constraints. Also, to handle constraints of larger arities, we use total correlation instead of mutual information to measure correlation among variables.

Coja-Oghlan et al. [70] showed that, even if the instance is sparse, if it satisfies a certain pseudo-random condition, then MaxkCSP admits PTASs. If $k = 2$, this results can be seen as a special case of [195] because the pseudo-random condition would imply that the
underlying graph is expander-like. Their result is incomparable to ours because it is not clear how the pseudo-random condition and the locally dense condition.

### 4.1.5 Organization

In Section 4.2, we introduce definitions and notions used in this paper. In Section 4.3, we show an algorithm that obtains an almost independent solution to the Sherali-Adams LP relaxation. Section 4.4 is devoted to show how to round the obtained solution to the Sherali-Adams LP relaxation. We combine the two steps together in Section 4.5. Section 4.6 and Section 4.7 are devoted to prove auxiliary lemmas. We consider CSPs with global cardinality constraints in Section 4.8.

### 4.2 Preliminaries

For an integer \( a \geq 1 \), \([a]\) denotes the set \( \{1, \ldots, a\} \). For a set \( Y \) and \( 0 \leq k \leq |Y| \), \( \binom{Y}{k} \) denotes the family of sets \( X \subseteq Y \) with \( |X| = k \). We usually use \( V \) to denote the set of variables in a problem, and use \( n = |V| \) to denote the number of variables. For an event \( A \), \( 1[A] \) denotes the corresponding indicator function.

**Probability theoretic notions:** We recall several notions from probability theory. For a probability distribution \( \mu \) on \( \Omega \), \( \text{supp}(\mu) \) denotes the support of \( \mu \), i.e., \( \text{supp}(\mu) = \{i \in \Omega \mid \mu(i) > 0\} \). For a set \( S \), \( i \sim S \) means that we sample \( i \) uniformly at random from \( S \).

Let \( \mu_1 \) and \( \mu_2 \) be two probability distributions on a finite set \( \Omega \). Then, the \( L_1 \) distance between them is defined as \( \|\mu_1 - \mu_2\|_1 = \sum_{i \in \Omega} |\mu_1(i) - \mu_2(i)| \). The Kullback-Leibler divergence between them is defined as \( d_{KL}(\mu_1 \| \mu_2) = \sum_{i \in \Omega} \mu_1(i) \log \frac{\mu_1(i)}{\mu_2(i)} \). and the Kullback-Leibler divergence \( d_{KL}(\mu_1 \| \mu_2) \) are defined as follows. We provide the following fact without proof.

**Lemma 4.2.1.** Let \( \mu_1 \) and \( \mu_2 \) be two probability distributions on a finite set \( \Omega \). Then, \( \|\mu_1 - \mu_2\|_1 \leq \sqrt{2d_{KL}(\mu_1 \| \mu_2)} \).

**Information theoretic notions:** We now recall some definitions from information theory. For a random variable \( x \), \( \mu_x \) denotes the corresponding probability distribution. That is, for any \( i \), we have \( \mu_x(i) = \Pr[x = i] \).
Let \( x \) be a random variable on a finite set \( \Omega \). The entropy of \( x \) is defined as 
\[
H(x) = - \sum_{i \in \Omega} \Pr[x = i] \log \Pr[x = i].
\]

Let \( x \) and \( y \) be jointly distributed variables on a finite set \( \Omega \). The entropy of \( x \) conditioned on \( y \) is defined as 
\[
H(x \mid y) = E_{i \sim \mu_y}[H(x \mid y = i)].
\]

The mutual information of \( x \) and \( y \) is defined as 
\[
I(x; y) = d_{KL}(\mu_{x,y} \parallel \mu_x \times \mu_y).
\]

Let \( x_1, \ldots, x_k \) \((k \geq 2)\) be jointly distributed variables on a finite set \( \Omega \). The mutual information of \( x_1, \ldots, x_k \) is defined as 
\[
I(x_1; \ldots ; x_k) = I(x_1; \ldots ; x_{k-1}) - I(x_1; \ldots ; x_{k-1} \mid x_k),
\]
where 
\[
I(x_1; \ldots ; x_{k-1} \mid x_k) = E_{i \sim \mu_y}[I(x_1; \ldots ; x_{k-1} \mid x_k = i)].
\]
The total correlation of \( x_1, \ldots, x_k \) is defined as 
\[
C(x_1, \ldots, x_k) = d_{KL}(\mu(x_1, \ldots, x_k) \parallel \mu_{x_1} \times \cdots \times \mu_{x_k}).
\]

We give two well-known facts in information theory below.

**Lemma 4.2.2.** Let \( x \) and \( y \) be two jointly distributed variables on a finite set \( \Omega \). Then 
\[
I(x; y) = H(x) - H(x \mid y).
\]

Let \( x_1, \ldots, x_k \) be jointly distributed variables on a finite set \( \Omega \). Then 
\[
I(x_1; \ldots ; x_k) = \sum_{(i_1, \ldots, i_t) \subseteq [k], t \geq 1} (-1)^{t-1} H(x_{i_1}, \ldots, x_{i_t}).
\]

**Lemma 4.2.3.** Let \( x_1, \ldots, x_k \) be jointly distributed variables on a finite set \( \Omega \). Then 
\[
C(x_1, \ldots, x_k) = \sum_{(i_1, \ldots, i_t) \subseteq [k], t \geq 2} I(x_{i_1}; \ldots ; x_{i_t}).
\]

**Constraint satisfaction problems:** Let \( D \) be a nonempty finite domain and \( k \geq 2 \) be an integer. An instance \( \mathcal{I} = (V, \omega, P) \) of \( k \) CSP consists of a set \( V \) of variables, a scope distribution \( \omega \) over \( V^k \), and a set of payoff functions \( P = \{ P_S : D^S \to [0, 1] \mid S \subseteq V^k \} \).

An assignment for an instance \( \mathcal{I} = (V, \omega, P) \) is a mapping \( \alpha : V \to D \). The value of the assignment, denoted \( \text{val}(\mathcal{I}, \alpha) \in [0, 1] \), is defined as 
\[
\text{val}(\mathcal{I}, \alpha) = \Pr_{S \sim \omega}[P_S(\alpha|S)],
\]
where \( \alpha|_S \) is the projection of \( \alpha \) to \( S \). We define the optimum value of the instance \( \mathcal{I} \) to be 
\[
\text{opt}(\mathcal{I}) = \max_{\alpha} \{ \text{val}(\mathcal{I}, \alpha) \}.
\]

Let \( \mathcal{I} = (V, \omega, P) \) be an instance of CSP. A solution to the \( \ell \)-round Sherali-Adams relaxation consists of a probability distribution \( \mu_S \) over \( D^S \) for each set \( S \subseteq V \) of size at most \( \ell \). The objective function is the probability that \( \alpha \) is in \( P_S \), where \( S \) is sampled from \( \omega \) and \( \alpha \) is sampled from \( \mu_S \). Strictly speaking, we sample a tuple \((v_1, \ldots, v_k)\) from \( \omega \), but we regard it as the set \( \{v_1, \ldots, v_k\} \) when we use it as a subscript of \( \mu \). In other words, \( \mu_S \)
and $\mu_T$ are the same distribution for two tuples $S$ and $T$ if they are the same as sets. Also, for every pair of sets $S$ and $T$ with $|S \cup T| \leq \ell$, the corresponding probability distributions $\mu_S$ and $\mu_T$ must be consistent on $S \cap T$. Formally, the $\ell$-round Sherali-Adams relaxation for a $k$CSP instance $I = (V, \omega, P)$ ($\ell \geq k$) is written as follows.

\[
\text{maximize } \mathbb{E}_{S \sim \omega} \mathbb{E}_{\alpha \sim \mu_S} [P_S(\alpha)]
\]

\[
\text{subject to } \Pr_{\alpha \sim \mu_S} [\alpha_{|S \cap T} = \beta] = \Pr_{\alpha \sim \mu_T} [\alpha_{|S \cap T} = \beta] \quad \forall S, T \subseteq V, |S \cup T| \leq \ell, \beta \in D^{|S \cap T|}.
\]

It is not hard to see that the relaxation above can be written as a linear programming (see, e.g., [191] for details). We define $x_v$ as the random variable sampled from the distribution $\mu_{\{v\}}$. We use $\text{val}_{\text{LP}}(I, \mu)$ to denote the objective value of the LP solution $\mu$. The same definition applies to the following subsections.

**Assignment problems:** The assignment problem differs from CSP in that we want to maximize the objective function over the set of permutations. Similarly to CSP, for an integer $k \geq 2$, an instance of the degree-$k$ assignment problem is given as $I = (V, \omega)$, where $V$ is the set of variables, $\omega$ is a distribution over $V^k \times V^k$. The scope distribution of $I$ is the marginal distribution of $\omega$ on the first $k$ elements. An assignment for an instance $I = (V, \omega)$ is a permutation $\pi$ of $V$. The value of the assignment $\pi$, denoted $\text{val}(I, \pi)$, is defined as

\[
\text{val}(I, \pi) = n^k \Pr_{(U,W) \sim \omega} [\forall i \in [k] : \pi(u_i) = w_i],
\]

where $U = (u_1, u_2, \ldots, u_k)$ and $W = (w_1, w_2, \ldots, w_k)$. We define the optimum value of $I$ to be $\text{opt}(I) = \max_{\pi} \{\text{val}(I, \pi)\}$.

Though the definition of $\text{val}(I, \pi)$ may look non-standard, it is just the objective function used in [17] with a normalization factor that is multiplied to make the optimum $\Omega(1)$ when $\omega$ is dense.

The $\ell$-round Sherali-Adams relaxation of an $k$AP instance $I = (V, \omega)$ ($\ell \geq k$) is as follows.

\[
\text{maximize } \mathbb{E}_{(U,W) \sim \omega} \mathbb{E}_{\alpha \sim \mu_U} [\forall i \in [k] : \alpha(u_i) = w_i]
\]

\[
\text{subject to } \Pr_{\beta \sim \mu_S} [\beta_{|S \cap T} = \alpha] = \Pr_{\beta \sim \mu_T} [\beta_{|S \cap T} = \alpha] \quad \forall S, T \subseteq V, |S \cup T| \leq \ell, \alpha \in V^{|S \cap T|}
\]

\[
\sum_{\alpha \in V^S} \sum_{w \in V \setminus S} \mu_{S \cup \{w\}}(\alpha \cup \{u \rightarrow w\}) = \sum_{\alpha \in V^S} \mu_S(\alpha) \quad \forall w \in V, S \subseteq V, |S| < \ell.
\]

The difference from the Sherali-Adams relaxation for CSP is that we have extra constraints in the last line whose intended meaning is that each value $i$ can be taken by at most one variable.
Density condition: We now introduce the notion of dense and locally dense distributions.

Let \( \omega \) be a probability distribution over a finite set \( \Omega \). For \( \Delta \in (0, 1] \), we say \( \omega \) is \( \Delta \)-dense if for every \( a \in \Omega \), it holds that \( \Delta \cdot \omega(a) \leq \frac{1}{|\Omega|} \).

Let \( \omega \) be a probability distribution over \( V^k \). Let \( d_i(v) = \Pr_{S \sim \omega} [S_i = v] \) be the probability that the \( i \)-th coordinate is \( v \) under \( \omega \). For \( \Delta \in (0, 1] \), we say \( \omega \) is \( \Delta \)-locally dense if for every \( (v_1, \ldots, v_k) \in V^k \), it holds that

\[
\Delta \cdot \omega(v_1, \ldots, v_k) \leq \frac{1}{|V|^k - 1} \sum_{1 \leq i \leq k} d_i(v_i).
\]

Since \( d_i(v) = \sum_{S \in V^k : S_i = v} \omega(S) \), the RHS of the locally dense condition is equal to

\[
\sum_{1 \leq i \leq k} \mathbb{E}_{S \sim V^k} [\omega(S) | S_i = v_i].
\]

Thus the locally dense condition says that no tuple \((v_1, \ldots, v_k)\) is "wild" in that \( \omega(v_1, \ldots, v_k) \) is at most constant times the sum over \( i \) of the average probability mass of \( S \) with \( S_i = v_i \).

The notion of local density is introduced in [75] to generalize the metric condition. To see this, suppose \( \omega : V^2 \to \mathbb{R} \) is a metric. Then, \( \omega \) is 1-locally dense since, for any \( u, v \in V \), we have

\[
\frac{1}{n}(d_1(u) + d_2(v)) = \frac{1}{n} \sum_w (\omega(u, w) + \omega(w, v)) \geq \frac{1}{n} \sum_w \omega(u, v) \geq \omega(u, v).
\]

It is immediate to verify the following lemma.

Lemma 4.2.4. Let \( \omega \) be a probability distribution over \( \Omega_1 \times \Omega_2 \). If \( \omega \) is \( \Delta \)-dense (resp., \( \Delta \)-locally dense), then the marginal distribution \( \omega_1 \) of \( \omega \) on \( \Omega_1 \) is also \( \Delta \)-dense (resp., \( \Delta \)-locally dense).

4.3 Conditioning operations for Sherali-Adams LP hierarchy

Recall that, a solution to the \( \ell \)-round SA relaxation consists of distributions over sets of \( \ell \) variables. In this section, we show that, if the scope distribution is dense or locally dense, then by conditioning a small number of variables, we can make variables almost independent in these distributions. Once variables become almost independent, we can round variables independently without losing the objective value much (see Section 4.4).

Let \( I \) be an instance of kCSP or kAP with a variable set \( V \). Fix \( \ell \) and let \( \mu \) be a solution to the \( \ell \)-round Sherali-Adams relaxation. For a variable set \( S = (v_1, \ldots, v_k) \),
$C_\mu(x_S)$ denotes the total correlation $C(x_{v_1}, \ldots, x_{v_k})$ under the probability distribution $\mu_S$. We use the following notion to measure independence of variables.

**Definition 4.3.1.** Let $\mathcal{I}$ be an instance of $k$CSP or $k$AP with a scope distribution $\omega$. A solution $\mu$ to the $\ell$-round SA relaxation for $\mathcal{I}$ with $\ell \geq k$ is $\kappa$-independent with respect to distribution $\omega'$ if

$$\mathbb{E}_{S \sim \omega'}[C_\mu(x_S)] \leq \kappa.$$ 

We say that $\mu$ is $\kappa$-independent if it is $\kappa$-independent with respect to $\omega$.

In [Section 4.3.1], we explain how to condition variables. In [Section 4.3.2] and [Section 4.3.3], we show that the conditioning operation outputs $\kappa$-independent LP solutions for the dense case and the locally dense case, respectively.

### 4.3.1 Conditioning operations

We first describe the operation of conditioning one variable. Given a solution $\mu$ to the $\ell$-round SA relaxation with $\ell \geq 2$, we sample a vertex $u$ uniformly at random and then set $x_u = i$, where $i$ is a value sampled from $\mu\{u\}$. This operation gives a solution $\mu'$ to the $(\ell - 1)$-round SA relaxation: For each tuple $(v_1, \ldots, v_{\ell - 1})$ of $\ell - 1$ variables, we define $\mu'_{\{v_1, \ldots, v_{\ell - 1}\}}(i_1, i_2, \ldots, i_{\ell - 1}) = \mu_{\{v_1, \ldots, v_{\ell - 1}, u\}}(i_1, i_2, \ldots, i_{\ell - 1}, i)$. It is not hard to check that $\mu'$ is indeed a solution to the $(\ell - 1)$-round SA relaxation.

Our algorithm is given in [Algorithm 1]. Given a solution $\mu$ to the $(\ell + \ell')$-round SA relaxation, it iteratively conditions variables. We will show in subsequent sections that, if $\omega$ is $\Delta$-dense or $\Delta$-locally dense, then [Algorithm 1] outputs a $\kappa$-independent LP solution in $\ell'$ steps on average, where $\kappa = \frac{k \Delta \log |D|}{\ell' \ell}$. (If $\omega$ is $\Delta$-dense, $\kappa$ can be slightly smaller.)

We mention here the following simple fact.

**Lemma 4.3.2.** Let $\mu'$ be the solution output by [Algorithm 1]. Then, $\mathbb{E}_{\mathcal{I}}[\text{val}^{\ell}((\mathcal{I}, \mu')) = \text{val}^{\ell'}((\mathcal{I}, \mu)).$

**Proof.** Notice that the algorithm respects the marginal distributions provided by the SA relaxation during sampling the values to variables. Thus, the expected objective value is preserved. \qed
Algorithm 1 Conditioning operation of Sherali-Adams solutions

Input: A feasible solution \( \mu \) to the \((\ell + \ell')\)-round SA relaxation for a CSP instance \( \mathcal{I} = (V, \omega) \).

Output: An \( \kappa \)-independent solution to the \( \ell \)-round SA relaxation, where \( \kappa = \frac{k^4 \log |D|}{\ell'} \)

Set \( t = 1 \).

while the current LP solution is not \( \kappa \)-independent do

Sample a variable \( u_t \in V \) uniformly at random.

Sample a value \( a \) from its marginal distribution \( \mu_{\{u_t\}} \) after the first \( t - 1 \) fixings, and

condition the LP solution by setting \( x_{u_t} = a \).

\( t = t + 1 \).

4.3.2 The dense case

We consider the dense case, that is, \( \omega \) is a uniform distribution.

Lemma 4.3.3. If \( \omega \) is uniform distribution over \( V^k \), there exists \( t \leq \ell' \) such that

\[
E_{U \sim V^t} E_{S \sim V^k} [C_{\mu}(x_S | x_U)] \leq \frac{3k^4 \log |D|}{\ell'}.
\]

Proof. We consider the value

\[
\sum_{1 \leq t \leq \ell'} E_{U \sim V^t} E_{S \sim V^k} [C_{\mu}(x_S | x_U)].
\]

From Lemma 4.2.3, this value can be decomposed as

\[
\sum_{1 \leq t \leq \ell'} E_{U \sim V^t} E_{S \sim V^k} \left[ \sum_{2 \leq r \leq k} \sum_{R \in \binom{S}{r}} I_{\mu}(x_R | x_U) \right] = \sum_{2 \leq r \leq k} \binom{k}{r} \sum_{1 \leq t \leq \ell'} E_{U \sim V^t} E_{R \sim V^r} [I_{\mu}(x_R | x_U)],
\]

where for a set \( R = (v_1, \ldots, v_r) \), \( I_{\mu}(x_R) \) denotes the mutual information \( I_{\mu}(v_1; \ldots; v_r) \).

To bound this value, we recall the definition of mutual information. For any \( t \leq \ell' \),

\[
E_{U \sim V^t \atop R \sim V^r} [I_{\mu}(x_R | x_U)] = E_{U \sim V^t \atop R \sim V^{r-1}} [I_{\mu}(x_R | x_U)] - E_{U \sim V^{t+1} \atop R \sim V^{r-1}} [I_{\mu}(x_R | x_U)].
\]
Adding the equalities from $t = 0$ to $\ell'$, we get
\[
\sum_{0 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t} \mathbb{E}_{R \sim V^{t+1}} [I(\mathbf{x}_R)] - \mathbb{E}_{U \sim V^t} \mathbb{E}_{R \sim V^{t+1}} [I(\mathbf{x}_R | \mathbf{x}_U)] \leq 2^{t'} \log |D|,
\]
where the last inequality holds from $I_{\mu}(\mathbf{x}_R) \leq 2^{|R|} \log |D|$ by Lemma 4.2.2. Thus, we have
\[
\sum_{0 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t} \mathbb{E}_{S \sim \omega} [C(\mathbf{x}_S | \mathbf{x}_U)] \leq 3k \log |D|,
\]
and the lemma follows. \qed

The following corollary is immediate.

**Corollary 4.3.4.** If $\omega$ is a $\Delta$-dense distribution over $V^k$. Then there exists $t \leq \ell'$ such that
\[
\mathbb{E}_{U \sim V^t} \mathbb{E}_{S \sim \omega} [C_{\mu}(\mathbf{x}_S | \mathbf{x}_U)] \leq \frac{3k \log |D|}{\Delta t'}.
\]

### 4.3.3 The locally dense case

We now consider the case that the scope distribution $\omega$ is 1-locally dense.

**Lemma 4.3.5.** If $\omega$ is a 1-locally dense distribution over $V^k$, then there exists $t \leq \ell'$ such that
\[
\mathbb{E}_{U \sim V^t} \mathbb{E}_{S \sim \omega} [C_{\mu}(\mathbf{x}_S | \mathbf{x}_U)] \leq \frac{k4 \log k |D|}{\ell'}.
\]

**Proof.** We consider the value
\[
\sum_{1 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t} \mathbb{E}_{S \sim \omega} [C_{\mu}(\mathbf{x}_S | \mathbf{x}_U)]. \quad (4.1)
\]
From Lemma 4.2.3, this value can be decomposed as
\[
\sum_{1 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t} \mathbb{E}_{S \sim \omega} \left[ \sum_{2 \leq r \leq k} \sum_{R \in \binom{\mathcal{S}}{r}} I_{\mu}(\mathbf{x}_R | \mathbf{x}_U) \right] = \sum_{J \subseteq [k] : 2 \leq |J| \leq k} \sum_{1 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t} \mathbb{E}_{R \sim \omega | J} [I_{\mu}(\mathbf{x}_R | \mathbf{x}_U)],
\]

46
where \( \omega|_J \) denotes the marginal distribution of \( \omega \) on \( J \).

Fix \( J \subseteq [k] \) with \( |J| = r \geq 2 \). Let \( \omega_i \) be the marginal distribution of \( \omega \) on the \( i \)-th coordinate. Let \( \Omega_i = \omega_i \times V^{r-1} \) and \( \Omega'_i = \omega_i \times V^{r-2} \). We first analyze \( I(x_R) \) under \( \Omega_i \) instead of \( \omega|_J \).

From the definition, for any \( i \) and \( t \leq \ell' \),

\[
\mathbb{E}_{U \sim V^t \atop R \sim \Omega_i} I(x_R \mid x_U) = \mathbb{E}_{R \sim \Omega_i} I(x_R \mid x_U) - \mathbb{E}_{U \sim V^{t+1} \atop R \sim \Omega'_i} I(x_R \mid x_U).
\]

Adding the equalities from \( t = 0 \) to \( t = \ell' \), we get

\[
\sum_{0 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t \atop R \sim \Omega_i} I(x_R \mid x_U) = \mathbb{E}_{R \sim \Omega_i} \sum_{0 \leq t \leq \ell'} I(x_R \mid x_U) - \mathbb{E}_{U \sim V^{t+1} \atop R \sim \Omega'_i} \sum_{0 \leq t \leq \ell'} I(x_R \mid x_U) \leq 2^r \log |D|.
\]

(4.2)

Now we turn to analyze \( I(x_{v_1}; \ldots; x_{v_r}) \) under \( \omega|_J \).

\[
\mathbb{E}_{U \sim V^t \atop R \sim \omega|_J} I(x_R \mid x_U) = \mathbb{E}_{U \sim V^t \atop R \sim \Omega_i} \sum_{0 \leq t \leq \ell'} I(x_R \mid x_U) \leq \mathbb{E}_{R \sim \Omega_i} \sum_{1 \leq i \leq r} d_i(v_i) I(x_{v_1}; \ldots; x_{v_r} \mid x_U)
\]

(by local density and Lemma 4.2.4)

\[
= \sum_{1 \leq i \leq r} \mathbb{E}_{U \sim V^t \atop (v_1, \ldots, v_r) \sim V^r} \frac{1}{n^{r-1}} \sum_{1 \leq i \leq r} \mathbb{E}_{S \sim \omega} \mathbb{E}_{R \sim V^r} \mathbb{E}_{S \sim \omega} [S_i = v_i] I(x_{v_1}; \ldots; x_{v_r} \mid x_U)
\]

\[
= \sum_{1 \leq i \leq r} \mathbb{E}_{U \sim V^t \atop R \sim \Omega_i} \sum_{1 \leq i \leq r} \mathbb{E}_{R \sim V^r} \Omega_i(R) I(x_{v_1}; \ldots; x_{v_r} \mid x_U)
\]

Thus from (4.2),

\[
\sum_{0 \leq t \leq \ell'} \mathbb{E}_{U \sim V^t \atop R \sim \omega|_J} I(x_R \mid x_U) \leq \sum_{1 \leq i \leq r} 2^r \log |D| = r 2^r \log |D|.
\]

It follows that (4.1) \( \leq k^k k \log |D| \) and the lemma holds. \( \square \)

The following corollary is immediate.
Corollary 4.3.6. If $\omega$ is a $\Delta$-locally dense distribution over $V^k$, then there exists $t \leq \ell'$ such that

$$\mathbb{E}_{U \sim V^t} \mathbb{E}_{S \sim \omega} [C_\mu(x_S | x_U)] \leq \frac{k4^k \log |D|}{\Delta \ell'}.$$ 

4.4 Rounding $\kappa$-independent solutions

4.4.1 Constraint satisfaction problems

Lemma 4.4.1. Let $\mathcal{I} = (V, \omega, P)$ be a $k$CSP instance over finite domain $D$. Let $\mu$ be a $\kappa$-independent solution to the $k$-round Sherali-Adams LP relaxation. There is a randomized polynomial time algorithm to find an assignment $\alpha : V \rightarrow D$ such that $\text{val}(\mathcal{I}, \alpha) \geq \text{val}^{\text{LP}}(\mathcal{I}, \mu) - 3\sqrt{\kappa}$.

Proof. For each $v \in V$, let $\alpha(v)$ be independently sampled from $\mu_{\{v\}}$. For each $S \in V^k$, by the definition of total correlation, Lemma 4.2.1, and the fact that $P_S(\beta) \in [0, 1]$ we have

$$\left| \mathbb{E}_{\beta \sim \mu_S} P_S(\beta) - \mathbb{E}_\alpha P_S(\alpha|S) \right| \leq 2\sqrt{C(x_S)}.$$ 

Therefore by $\kappa$-independence,

$$\left| \mathbb{E}_{S \sim \omega} \left( \mathbb{E}_{\beta \sim \mu_S} P_S(\beta) - \mathbb{E}_\alpha P_S(\alpha|S) \right) \right| \leq \mathbb{E}_{S \sim \omega} 2\sqrt{C(x_S)} \leq 2\sqrt{\mathbb{E}_{S \sim \omega} C(x_S)} \leq 2\sqrt{\kappa}.$$ 

We have proved that $\mathbb{E}_\alpha [\text{val}(\mathcal{I}, \alpha)] \geq \text{val}^{\text{LP}}(\mathcal{I}, \mu) - 2\sqrt{\kappa}$. Therefore we can sample an $\alpha$ in expected polynomial time such that $\text{val}(\mathcal{I}, \alpha) \geq \text{val}^{\text{LP}}(\mathcal{I}, \mu) - 3\sqrt{\kappa}$. $\square$
4.4.2 Assignment problems

Let \( I = (V, \omega) \) be a \( \Delta \)-dense \( k \)AP instance. We introduce the following relaxation \( H \), and let \( \text{val}_H(\mathcal{I}) \) be its optimal value.

\[
\begin{align*}
\text{maximize} & \quad n^k \mathbb{E}_{(U,W) \sim \omega} \prod_{i=1}^{k} x_{u_i,w_i} \\
\text{subject to} & \quad x_{u,w} \geq 0 \quad \forall u, w \in V^k \\
& \quad \sum_{u \in V} x_{u,w} = 1 \quad \forall w \in V \\
& \quad \sum_{w \in V} x_{u,w} = 1 \quad \forall u \in V.
\end{align*}
\]

4.4.2.1 From \( \kappa \)-independence to relaxation \( H \)

We first see that we can find a good solution to \( H \) using a solution to the Sherali-Adams LP relaxation of a dense instance \( \mathcal{I} \).

**Lemma 4.4.2.** Let \( I = (V, \omega) \) be a \( k \)AP instance such that \( \omega \) is \( \Delta \)-dense. Let \( \mu \) be a \( \kappa \)-independent solution (with respect to the uniform distribution rather than \( \omega \)) to the \( k \)-round Sherali-Adams LP relaxation of \( \mathcal{I} \). There is a polynomial-time algorithm, on input \( \mu \), to find a solution to \( H \) that certifies that \( \text{val}_H(\mathcal{I}) \geq \text{val}_{LP}(\mathcal{I}, \mu) - 2\sqrt{\kappa/\Delta} \).

**Proof.** Let \( x_{u,w} = \mu_{u}(w) \) for all \( u, w \in V \). For each \( S = (u_1, u_2, \ldots, u_k) \in V^k \), by the definition of total correlation and [Lemma 4.2.1](#) we have

\[
\sum_{T=(w_1, \ldots, w_k)} \left| \mu_{S}(T) - \prod_{i=1}^{k} x_{u_i,w_i} \right| \leq 2\sqrt{C(x_S)}. \tag{4.3}
\]

Therefore,

\[
\mathbb{E}_{(S,T)=(u_1, \ldots, u_k,v_1, \ldots, v_k) \sim \omega} \left( \mu_{S}(T) - \prod_{i=1}^{k} x_{u_i,w_i} \right) \leq \frac{1}{\Delta} \mathbb{E}_{(S,T)=(u_1, \ldots, u_k,v_1, \ldots, v_k) \sim V^{2k}} \left| \mu_{S}(T) - \prod_{i=1}^{k} x_{u_i,w_i} \right| \quad \text{(by density)}
\]

\[
\leq \frac{1}{\Delta} n^k \mathbb{E}_{S \sim V^k} 2\sqrt{C(x_S)} \quad \text{(by (4.3))}
\]

49
\[ \leq \frac{2}{\Delta}\sqrt{\frac{1}{n^k}} \sqrt{E_{S \sim \mathcal{V}^k} C(x_S)} \leq n^{-k} \cdot \frac{2\sqrt{\kappa}}{\Delta}. \] (by \( \kappa \)-independence)

The following variant of Lemma 4.4.2 is used when dealing with the locally dense case later.

**Lemma 4.4.3.** Let \( I = (V, \omega) \) be a \( k \)AP instance such that \( \omega(u_1, \ldots, u_k, w_1, \ldots, w_k) = \omega'(u_1, \ldots, u_k) \cdot \omega''(w_1, \ldots, w_k) \) where \( \omega'' \) is a \( \Delta \)-dense distribution over \( V^k \). Let \( \mu \) be a \( \kappa \)-independent solution to the \( k \)-round Sherali-Adams LP relaxation of \( I \). There is a polynomial-time algorithm, on input \( \mu \), to find a solution to \( H \) that certifies that \( \text{val}_H(I) \geq \text{val}_{\text{LP}}(I, \mu) - 2\sqrt{\kappa/\Delta} \).

**Proof.** Let \( x_u, w = \mu_u(w) \) for all \( u, w \in V \). Similar to the proof of Lemma 4.4.2, we have

\[ \left| \frac{1}{\Delta} \sum_{(S, T) = (u_1 \ldots u_k, v_1 \ldots v_k)} \omega' \left( \mu_S(T) - \prod_{i=1}^k x_{u_i, w_i} \right) \right| \leq \frac{1}{\Delta} \sum_{(S, T) = (u_1 \ldots u_k) \sim \omega'} \mu_S(T) - \prod_{i=1}^k x_{u_i, w_i} \right| \leq \frac{2}{\Delta n^k} \sqrt{\sum_{S \sim \omega'} E C(x_S)} \leq n^{-k} \cdot \frac{2\sqrt{\kappa}}{\Delta}. \] (by \( \kappa \)-independence)

---

**4.4.2.2 From relaxation \( \mathcal{H} \) to an integral solution**

At a first look, \( \mathcal{H} \) is very close to the \( k \)AP problem itself. However, we cannot independently sample \( \pi(v) \) from each \( x_v \) in \( \mathcal{H} \) to get a solution to \( k \)AP, since there is chance that \( \pi(v) = \pi(v') \), rendering \( \pi \) not a permutation. Indeed, we show in Section 4.9 that for some \( k \)AP instance \( I \), there is a gap between \( \text{val}_\mathcal{H}(I) \) and \( \text{val}(I) \). However, our following lemma shows that this gap cannot be very big for \( k \)AP instances \( I = (V, \omega) \) when \( \omega \) is \( \Delta \)-dense. The proof is given later in Section 4.6.
Lemma 4.4.4. Let $\mathcal{I} = (V, \omega)$ be a $k$AP instance such that $\omega$ is $\Delta$-dense. Given a solution $x$ to relaxation $\mathcal{H}$, let $val_{\mathcal{H}}(\mathcal{I}, x)$ be the value of the solution. There is a randomized polynomial-time algorithm to compute a permutation $\pi$ such that $val(\mathcal{I}, \pi) \geq val_{\mathcal{H}}(\mathcal{I}, x) - \frac{7k^2 \log n}{\Delta \sqrt{n}}$.

In Section 4.7, we prove the following variant of Lemma 4.4.4.

Lemma 4.4.5. Let $\mathcal{I} = (V, \omega)$ be a $k$AP instance such that $\omega$ is $\Delta$-locally dense. Given a solution $x$ to relaxation $\mathcal{H}$, let $val_{\mathcal{H}}(\mathcal{I}, x)$ be the value of the solution. There is a randomized polynomial-time algorithm to compute a permutation $\pi$ such that $val(\mathcal{I}, \pi) \geq val_{\mathcal{H}}(\mathcal{I}, x) - \frac{7k^2 \log n}{\Delta \Delta \sqrt{n}}$.

4.4.3 The rounding lemmas

Combining Lemma 4.4.2 and Lemma 4.4.4 and Lemma 4.4.3 and Lemma 4.4.5, we get the following main rounding lemmas for this subsection.

Lemma 4.4.6. Let $\mathcal{I} = (V, \omega)$ be a $k$AP instance such that $\omega$ is $\Delta$-dense. Let $\mu$ be a $\kappa$-independent solution (with respect to the uniform distribution rather than $\omega$) to the $k$-round Sherali-Adams LP relaxation for $\mathcal{I}$. There is a polynomial-time algorithm, on input $\mu$, to find a permutation $\pi$ such that $val(\mathcal{I}, \pi) \geq val_{LP}(\mathcal{I}, \mu) - \frac{2\sqrt{\kappa}}{\Delta} - \frac{7k^2 \log n}{\Delta \sqrt{n}}$.

Lemma 4.4.7. Let $\mathcal{I} = (V, \omega)$ be a $k$AP instance such that $\omega$ is $\Delta$-locally dense. Given a solution $x$ to the relaxation $\mathcal{H}$, let $val_{\mathcal{H}}(\mathcal{I}, x)$ be the value of the solution. There is a randomized polynomial-time algorithm to compute a permutation $\pi$ such that $val(\mathcal{I}, \pi) \geq val_{\mathcal{H}}(\mathcal{I}, x) - \frac{7k^2 \log n}{\Delta \Delta \sqrt{n}}$.

4.5 Putting things together

The following theorem gives PTASs for dense and locally dense MAX-CSP.

Theorem 4.5.1. Let $\mathcal{I} = (V, \omega, P)$ be a $k$CSP instance over finite domain $D$ such that $\omega$ is $\Delta$-dense or $\Delta$-locally dense. For any $\epsilon > 0$, let $\ell = \frac{9k^2 4^k \log |D|}{\epsilon^2 \Delta}$. The additive integrality gaps of the $(\ell + k)$-round Sherali-Adams LP relaxation is at most $\epsilon$; and there is a randomized rounding algorithm producing a solution whose value is at least $\text{opt}(\mathcal{I}) - \epsilon$, in expected $n^{O(\ell)}$ time.
Proof. Let \( \mu \) be a solution to the \((\ell + k)\)-round Sherali Adams LP relaxation. Let the random variable \((\mu|x_U)\) be the solution after conditioning on the variables in \(U\). By Corollary 4.3.4 and Corollary 4.3.6, we know that there exists \(t \leq \ell\) such that

\[
E_{U \sim V^t} \sqrt{E_{S \sim \omega} C_{I}(x_S|x_U)} \leq \sqrt{E_{U \sim V^t} E_{S \sim \omega} C_{I}(x_S|x_U)} \leq \sqrt{\frac{k2^{\ell + k} \log |D|}{\Delta^{\ell}} = \epsilon \frac{\Delta}{2}.
\]

Together with Lemma 4.3.2, we have

\[
E_{U \sim V^t} \left( \text{val}_{LP}(I, \mu|x_U) - 3 \sqrt{E_{S \sim \omega} C_{I}(x_S|x_U)} \right) \geq \text{val}_{LP}(I, \mu) - \epsilon.
\]

We enumerate all the possible ways of conditioning, and find out a solution \(\mu'\) to the \((k + \ell - t)\)-round Sherali-Adams LP relaxation such that \(\text{val}_{LP}(I, \mu') - 3 \sqrt{E_{S \sim \omega} C_{I}(x_S|x_U)} \geq \text{val}_{LP}(I, \mu) - \epsilon\). Since \(\mu'\) is always a \(E_{S \sim \omega} C_{I}(x_S|x_U)\)-independent solution, by Lemma 4.4.1 given \(\mu'\), we can find an assignment with value at least \(\text{val}_{LP}(I, \mu) - \epsilon\) in randomized polynomial time.

Now we prove that there is a quasi-polynomial-time approximation scheme for dense MAX-AP.

**Theorem 4.5.2.** Let \(I = (V, \omega)\) be a \(k\)AP instance such that \(\omega\) is \(\Delta\)-dense. For any \(\epsilon > 0\), let \(\ell = \frac{4k4^{k} \log |D|}{\epsilon^{2} \Delta^{2}}\). The additive integrality gaps of the \((\ell + k)\)-round Sherali-Adams LP relaxation is at most \(\epsilon + 7k^{2} \log \frac{n}{\Delta \sqrt{n}}\); and there is a randomized rounding algorithm producing a solution whose value is at least \(\text{opt}(I) - \epsilon - 7k^{2} \log \frac{n}{\Delta \sqrt{n}}\), in expected \(n^{O(\ell)}\) time.

**Proof.** Let \(\mu\) be a solution to the \((\ell + k)\)-round Sherali Adams LP relaxation. By Lemma 4.3.3 we know that there exists \(t \leq \ell\) such that

\[
E_{U \sim V^t} \sqrt{E_{S \sim V^k} C_{I}(x_S|x_U)} \leq \sqrt{E_{U \sim V^t} E_{S \sim V^k} C_{I}(x_S|x_U)} \leq \sqrt{\frac{k4^{k} \log n}{\ell}} = \epsilon \Delta
\]

Together with Lemma 4.3.2, we have

\[
E_{U \sim V^t} \left( \text{val}_{LP}(I, \mu|x_U) - 2 \sqrt{E_{S \sim V^k} C_{I}(x_S|x_U)} \right) \geq \text{val}_{LP}(I, \mu) - \epsilon.
\]

We enumerate all the possible ways of conditioning, and find out a solution \(\mu'\) to the \((k + \ell - t)\)-round Sherali-Adams LP relaxation such that \(\text{val}_{LP}(I, \mu') - 2 \sqrt{E_{S \sim V^k} C_{I}(x_S|x_U)} \geq \text{val}_{LP}(I, \mu) - \epsilon\). By Lemma 4.4.6 given \(\mu'\), we can find a permutation with value at least \(\text{val}_{LP}(I, \mu) - \epsilon - \frac{7k^{2} \log n}{\Delta \sqrt{n}}\). \(\square\)
Using Corollary 4.3.6 and Lemma 4.4.7 instead of Lemma 4.3.3 and Lemma 4.4.6 the same argument shows that there is a quasi-polynomial-time approximation scheme for locally dense MAX-AP.

**Theorem 4.5.3.** Let $\mathcal{I} = (V, \omega)$ be a $k$-AP instance such that $\omega(u_1, \ldots, u_k, w_1, \ldots, w_k) = \omega'((u_1, \ldots, u_k) \cdot \omega''((w_1, \ldots, w_k)$ where $\omega'$ is $\Delta'$-locally dense and $\omega''$ is $\Delta$-dense. For any $\epsilon > 0$, let $\ell = \frac{4k^24^\ell \log |D|}{\epsilon^2 \Delta' \Delta n}$. The additive integrality gaps of the $(\ell + k)$-round Sherali-Adams LP relaxation is at most $\epsilon + \frac{7k^2 \log n}{\Delta' \sqrt{n}}$; and there is a randomized rounding algorithm producing a solution whose value is at least $\text{opt}(\mathcal{I}) - \epsilon - \frac{7k^2 \log n}{\Delta \Delta' \sqrt{n}}$, in expected $n^{O(\ell)}$ time.

### 4.6 Proof of Lemma 4.4.4

Observe that a solution $x$ to the relaxation $\mathcal{H}$ corresponds to a doubly stochastic matrix. Now let us decompose a solution $x$ into a distribution of permutations $\mathcal{D} = \{\pi : V \rightarrow V\}$ such that for any $u, w \in V$, we have $\Pr_{\pi \sim \mathcal{D}}[\pi(u) = w] = x_{u,w}$. Let $\text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) = \text{val}_\mathcal{H}(\mathcal{I}, x)$ be the value of relaxation $\mathcal{H}$ on $x$ for instance $\mathcal{I}$. Our goal is to “merge” the permutations in $\mathcal{D}$ into one permutation while not losing much in the objective value. The following lemma proves this for the special case when $\mathcal{D}$ is supported on only two permutations.

**Lemma 4.6.1.** Let $\mathcal{D}$ be the distribution over $\pi_1$ and $\pi_2$ such that $\pi_1$ is chosen with probability $p$ and $\pi_2$ is chosen with probability $(1 - p)$. There exists a distribution $\mathcal{D}'$ over permutations such that for any $k \geq 2$ and any AP instance $\mathcal{I} = (V, \omega)$ such that $\omega$ is $\Delta$-dense, we have

$$
\mathbb{E}_{\pi \sim \mathcal{D}'}[\text{val}(\mathcal{I}, \pi)] \geq \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) - \frac{2k^2}{\Delta \sqrt{n}}.
$$

Moreover, $\mathcal{D}'$ can be sampled in polynomial time.

**Proof.** Let us assume w.l.o.g. that $V = [n]$, $\pi_1 = \text{id}$ (i.e. $\pi_1(i) = i$ for all $i \in [n]$). For any set $A = \{a_i : a_1 < a_2 < \cdots < a_{|A|} = n\} \subseteq [n]$, let us define $\pi_A$ be the permutation over $[n]$ so that $\pi_A(i) = a_{i-1} + 1$ if $i = a_t$ for some $t \in [|A|]$ and $\pi_A(i) = i + 1$ otherwise (assuming $a_0 = 0$). We can also assume w.l.o.g. that there exists $A \subseteq [n]$ such that $\pi_2 = \pi_A$. See Figure 4.1. We can add at most $\sqrt{n}$ elements into $A$ to get $A' \subseteq [n]$ such that there is no set of $\sqrt{n}$ consecutive integers that does not intersect $A'$. It is easy to show that $\pi_A$ and $\pi_{A'}$ differ at most $2\sqrt{n}$ places. Let $\mathcal{D}_{A'}$ be the probability distribution that chooses $\pi_1$ with probability $p$ and $\pi_{A'}$ with probability $(1 - p)$. For any $k$ and any $k$AP instance $\mathcal{I} = (V, \omega)$ such that $\omega$ is $\Delta$-dense, we have

$$
\text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) - \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}_{A'})
$$

53
Figure 4.1: Permutations $\pi_1$ and $\pi_2$ over $[5]$ shown as mappings from $[5]$ to $[5]$.

Solid arrow represents $\pi_1$ and dashed arrows represent $\pi_2$. For any permutation $\pi_1$, we can move vertices in the right side (and relabeling them accordingly) so that the resulting $\pi_1$ is the identity permutation. Then for any permutation $\pi_2$, we can move pairs of vertices (and relabeling them accordingly) so that $\pi_1$ remains the identity permutation whereas the cycles formed by $\pi_1$ and $\pi_2$ are drawn disjointly. In such a case, $\pi_2$ satisfies the condition in the body text.

$$n^k \mathbb{E}_{(U,W) \sim \omega} \left( \prod_{i=1}^{k} \Pr_{\pi \sim D} [\pi(u_i) = w_i] - \prod_{i=1}^{k} \Pr_{\pi \sim D_A'} [\pi(u_i) = w_i] \right)$$

$$\leq n^k \mathbb{E}_{(U,W) \sim \omega} 1 \left[ \exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i) \right] \cdot \prod_{i=1}^{k} \Pr_{\pi \sim D} [\pi(u_i) = w_i]$$

$$\leq \frac{n^k}{\Delta} \cdot \mathbb{E}_{(U,W) \sim V^{2k}} 1 \left[ \exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i) \right] \cdot \prod_{i=1}^{k} \Pr_{\pi \sim D} [\pi(u_i) = w_i], \quad (4.4)$$

where the last inequality is by the density of $\omega$.

Since

$$n^k \mathbb{E}_{W \sim V^k} \prod_{i=1}^{k} \Pr_{\pi \sim D} [\pi(u_i) = w_i] = \prod_{i=1}^{k} \sum_{w_i \sim V} \Pr_{\pi \sim D} [\pi(u_i) = w_i] = 1, \quad (4.5)$$
we have
\[
\Pr_{U \sim V^k} \left[ \exists i \in [k] : \pi_{A'}(u_i) \neq \pi_{A}(u_i) \right] \leq \frac{2k}{\Delta \sqrt{n}}. \tag{4.6}
\]

Now we define the distribution \(D'\). Let us assume that the elements in \(A'\) are \(a'_1 < a'_2 < \ldots < a'_{|A'|} = n\); let \(a'_0 = 0\) for convenience. To draw a permutation \(\pi \sim D'\), we sample \(|A'|\) i.i.d. 0/1 bits \(b_1, b_2, \ldots, b_{|A'|}\), each of which has mean \(p\). For each \(i\), we find out the unique \(t \in [|A'|]\) so that \(a'_{t-1} < i \leq a'_t\); let \(\pi(i) = \pi_1(i) = i\) if \(b_t = 0\); let \(\pi(i) = \pi_{A'}(i)\) otherwise.

For any \(k\) and any \(\Delta\)-dense \(k\)AP instance \(\mathcal{I} = (V, \omega)\), we have
\[
\val_{\mathcal{I}}(\mathcal{I}, \mathcal{D}_{A'}) - \mathbb{E}_{\pi \sim D'} \val(\mathcal{I}, \pi)
= n^k \mathbb{E}_{(U, W) \sim \omega} \left( \prod_{i=1}^{k} \Pr_{\pi \sim D_{A'}} [\pi(u_i) = w_i] - \Pr_{\pi \sim D_{A'}} [\forall i \in [k] : \pi(u_i) = w_i] \right)
= n^k \mathbb{E}_{(U, W) \sim \omega} \left( \prod_{i=1}^{k} \Pr_{\pi \sim D_{A'}} [\pi(u_i) = w_i] - \prod_{i=1}^{|A'|} \Pr_{\pi \sim D_{A'}} [\forall i \in [k], a'_{i-1} < u_i \leq a'_i : \pi(u_i) = w_i] \right)
\leq \frac{n^k}{\Delta} \mathbb{E}_{(U, W) \sim \omega} 1 \left[ \exists t \in [|A'|] : \exists i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right]
\cdot \prod_{i=1}^{k} \Pr_{\pi \sim D_{A'}} [\pi(u_i) = w_i]
\leq \frac{n^k}{\Delta} \mathbb{E}_{(U, W) \sim V^k} 1 \left[ \exists t \in [|A'|] : \exists i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right]
\cdot \prod_{i=1}^{k} \Pr_{\pi \sim D_{A'}} [\pi(u_i) = w_i] \quad \text{(by density)}
= \frac{1}{\Delta} \Pr_{U \sim V^k} [\exists t \in [|A'|] : \exists i \text{ s.t. } a'_{t-1} < u_i \leq a'_t] \quad \text{(by (4.5))}
\leq \frac{1}{\Delta} \cdot \binom{k}{2} \cdot \sum_{t \in [|A'|]} \left( \frac{a'_t - a'_{t-1}}{n} \right)^2 \leq \frac{1}{\Delta} \cdot \binom{k}{2} \cdot \sum_{t \in [|A'|]} \left( \frac{a'_t - a'_{t-1}}{n} \right) \cdot \frac{1}{\sqrt{n}} \leq \frac{k^2}{\sqrt{n}}. \tag{4.7}
\]

The lemma is proved by combining (4.6) and (4.7). \(\square\)
of Lemma 4.4.4] Let \( \mathcal{D} \) be supported on \( \pi_1, \pi_2, \ldots, \pi_m \), each \( \pi_i \) is chosen with probability \( p_i \). We can assume that \( m \leq n^2 \) by preserving only the \( n^2 \) permutations with the largest probabilities and proper normalization, which would cause a loss of at most \( n^{-1} \) in the objective value \( \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) \). Now we show that for any such distribution, we can find a distribution \( \mathcal{E} \) that is supported on \( (m - 1) \) permutations, such that

\[
\text{val}_\mathcal{H}(\mathcal{I}, \mathcal{E}) \geq \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{2k^2}{\Delta \sqrt{n}}. \tag{4.8}
\]

In other words, since \( \pi_1 \) and \( \pi_2 \) are arbitrary, we are able to “merge” any two permutations \( \pi_i \) and \( \pi_j \) in \( \mathcal{D} \) by paying a loss of \( (p_1 + p_2) \frac{2k^2}{\Delta \sqrt{n}} \) in the objective value. We repeatedly merge the two permutations with the smallest probability mass in the distribution until there is only one permutation left, during this process we lose at most \( \log m \frac{2k^2}{\Delta \sqrt{n}} \leq \frac{6k^2 \log n}{\Delta \sqrt{n}} \) in objective value. Together with the \( n^{-1} \) loss at the beginning of the proof, we lose at most \( \frac{7k^2 \log n}{\Delta \sqrt{n}} \) for sufficiently large \( n \).

In order to show (4.8), let us define a distribution \( \tilde{\mathcal{E}} \) of distributions of permutations as follows. Let \( \mathcal{F} \) be the distribution of permutations that chooses \( \pi_1 \) with probability \( \frac{p_1}{p_1 + p_2} \) and \( \pi_2 \) with the remaining probability. Apply Lemma 4.6.1 on \( \mathcal{F} \) to get \( \mathcal{F}' \). A distribution \( \mathcal{E} \) from \( \tilde{\mathcal{E}} \) is sampled by first sampling a permutation \( \pi \) from \( \mathcal{F}' \), and returning the distribution that puts probability mass \( (p_1 + p_2) \) on \( \pi \) and \( p_i \) on \( \pi_i \) for all \( i : 3 \leq i \leq m \). For every \( u, w \in V \), let \( \gamma_{u,w} = \sum_{i=3}^{m} p_i 1[\pi_i(u) = w] \leq 1 \). We have

\[
\mathbb{E}_{\mathcal{E} \sim \tilde{\mathcal{E}}} \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{E}) = \mathbb{E}_{\mathcal{E} \sim \tilde{\mathcal{E}}} \mathbb{E}_{(U,W) \sim \omega} \prod_{i=1}^{k} \Pr_\mathcal{E}[\pi(u_i) = w_i] = \mathbb{E}_{\pi \sim \mathcal{F}'} \mathbb{E}_{(U,W) \sim \omega} \prod_{i=1}^{k} \left( (p_1 + p_2) 1[\pi(u_i) = w_i] + \gamma_{u_i,w_i} \right)
\]

\[
= \sum_{Q \subseteq [k]} \mathbb{E}_{(U,W) \sim \omega} \left( \prod_{i \in Q} \mathbb{E}_{\pi \sim \mathcal{F}_i} \left( \prod_{i \in Q} (p_1 + p_2) 1[\pi(u_i) = w_i] \right) \right) \left( \prod_{i \notin Q} \gamma_{u_i,w_i} \right)
\]

\[
= \sum_{Q \subseteq [k]} \left( \prod_{i \in Q} \mathbb{E}_{(U,W) \sim \omega} \gamma_{u_i,w_i} \right) \left( p_1 + p_2 \right)^{|Q|} \mathbb{E}_{(U,W) \sim \omega} \left( \prod_{i \in Q} 1[\pi(u_i) = w_i] \right), \tag{4.9}
\]

where \( U_A \) is the restriction of vector \( U \) over coordinates in \( A \), \( \omega_{(A,B)} \) is the marginal distribution of \( \omega \) over \( A \) in the first \( k \) coordinates and \( B \) in the last \( k \) coordinates, and \( \omega|_{(U_A,W_B)} \) is the distribution \( \omega \) conditioned on that coordinates in \( A \) are assigned \( U \) and
coordinates in $B$ are assigned $W$. Let $\mathcal{I}_{U_{\overline{Q}}, W_{\overline{Q}}}$ be the $|Q|$-AP instance $(V, \omega((U_{\overline{Q}}, W_{\overline{Q}})))$. We know that $\omega((U_{\overline{Q}}, W_{\overline{Q}}))$ is $\Delta \cdot n^{2|Q|} \cdot \omega((\overline{Q}, Q))(U_{\overline{Q}}, W_{\overline{Q}})$-dense. Therefore for every $Q \neq \emptyset$, by Lemma 4.6.1 we have

\[
\mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(\prod_{i \in Q} \mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(1[\pi(u_i) = w_i]) = \mathbb{E}_{\pi \sim \mathcal{F}, \mathcal{I}_{U_{\overline{Q}}, W_{\overline{Q}}}, \pi}(\text{val}(\mathcal{I}_{U_{\overline{Q}}, W_{\overline{Q}}}, \pi))
\]

\[
\geq \text{val}_\mathcal{H}(\mathcal{I}_{U_{\overline{Q}}, W_{\overline{Q}}}, \mathcal{F}) - \frac{2k^2 n^{-2|Q|}}{\Delta \omega((\overline{Q}, Q))(U_{\overline{Q}}, W_{\overline{Q}})\sqrt{n}} = \mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(\prod_{i \in Q} p_1 1[\pi_1(u_i) = w_i] + p_2 1[\pi_2(u_i) = w_i]) - \frac{2k^2 n^{-2|Q|}}{\Delta \omega((\overline{Q}, Q))(U_{\overline{Q}}, W_{\overline{Q}})\sqrt{n}}.
\]

Therefore we have

\[
(4.9) \geq \sum_{Q \subseteq [k]} \mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(\prod_{i \in Q} \gamma_{u_i, w_i}) \mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(\prod_{i \in Q} (p_1 1[\pi_1(u_i) = w_i] + p_2 1[\pi_2(u_i) = w_i]))
\]

\[
- \sum_{\emptyset \neq Q \subseteq [k]} \mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(\prod_{i \in Q} \gamma_{u_i, w_i})(p_1 + p_2)^{|Q|} \cdot \frac{2k^2 n^{-2|Q|}}{\Delta \omega((\overline{Q}, Q))(U_{\overline{Q}}, W_{\overline{Q}})\sqrt{n}}
\]

\[
= \mathbb{E}_{(U, W) \sim \omega}(\sum_{t=1}^{m} p_t 1[\pi_t(u_i) = w_i])
\]

\[
- \sum_{\emptyset \neq Q \subseteq [k]} \mathbb{E}_{(U_{\overline{Q}}, W_{\overline{Q}}) \sim \omega((U_{\overline{Q}}, W_{\overline{Q}}))}(\prod_{i \in Q} \gamma_{u_i, w_i})(p_1 + p_2)^{|Q|} \cdot \frac{2k^2}{\Delta \sqrt{n}}
\]

\[
\geq \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{2k^2}{\Delta \sqrt{n}}.
\]

In all, we have proved that $\mathbb{E}_{\mathcal{E} \sim \tilde{\mathcal{E}}} \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{E}) \geq \text{val}_\mathcal{H}(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{2k^2}{\Delta \sqrt{n}}$. Since $\tilde{\mathcal{E}}$ can be sampled in polynomial time, there is a randomized polynomial-time algorithm to find out a $\mathcal{E}$ satisfying (4.8). \hfill \Box
4.7 Proof of Lemma 4.4.5

We say a distribution $\omega'$ over $V^k$ is $\Delta'$-well spread if for every $i, j \in [k]$ such that $i \neq j$, and for every disjoint partition $V = V_1 \cup V_2 \cup \cdots \cup V_t$, we have

\[
\Delta' \cdot \Pr_{(u_1, \ldots, u_k) \sim \omega'} \left[ \exists t' \in [t] : u_i \in V_{t'} \text{ and } u_j \in V_{t'} \right] \leq \frac{\max_{t' \in [t]} |V_{t'}|}{n}.
\]

Claim 4.7.1. A $\Delta'$-locally dense distribution $\omega'$ is $(\Delta'/k)$-well spread.

Proof. W.l.o.g. we assume that $i = 1$ and $j = 2$. For every $Z \subseteq V$, we have

\[
\Pr_{(u_1, \ldots, u_k) \sim \omega'} \left[ \exists t' \in [t] : u_i \in V_{t'} \text{ and } u_j \in V_{t'} \right]
\]

\[
= \sum_{t' \in [t]} \sum_{u_1, u_2, u_3, \ldots, u_k \in V_{t'}} \omega'(u_1, \ldots, u_k) \leq \sum_{t' \in [t]} \sum_{u_1, u_2, u_3, \ldots, u_k \in V_{t'}} \frac{\sum_{i=1}^k d_i(u_i)}{\Delta' n^{k-1}}
\]

\[
= \sum_{t' \in [t]} \left( \sum_{u_2 \in V_{t'}} \frac{\sum_{u_1, u_3, \ldots, u_k \in V} d_1(u_1)}{\Delta' n^{k-1}} + \sum_{u_4 \in V_{t'}} \frac{\sum_{u_3, u_5, \ldots, u_k \in V} d_2(u_2)}{\Delta' n^{k-1}} + \sum_{i=3}^k \sum_{u_1, u_2, u_3, \ldots, u_{i+1}, u_{i+2}, \ldots, u_k \in V} \frac{\sum_{u_i \in V} d_i(u_i)}{\Delta' n^{k-1}} \right)
\]

\[
\leq 2 \cdot \frac{\max_{t' \in [t]} |V_{t'}|}{\Delta' n} + (k - 2) \cdot \frac{\Delta' \max_{t' \in [t]} |V_{t'}|^2}{n^2} \leq \frac{k \max_{t' \in [t]} |V_{t'}|}{\Delta' n}.
\]

We will prove a slightly stronger statement than that of Lemma 4.4.5 in the sense that we prove the lemma for every $\omega$ such that $\omega = \omega' \cdot \omega''$ where $\omega'$ is $\Delta'$-well spread and $\omega''$ is $\Delta$-dense.

The proof goes along the lines of the proof of Lemma 4.4.4. We decompose $x$ into a distribution of permutations $\mathcal{D} = \{ \pi : V \rightarrow V \}$ such that for any $u, w \in V$, we have $\Pr_{\pi \sim \mathcal{D}} [\pi(u) = w] = x_{u,w}$. We first prove following lemma, which is an analogy of Lemma 4.6.1.

**Lemma 4.7.2.** Let $\mathcal{D}$ be the distribution over $\pi_1$ and $\pi_2$ such that $\pi_1$ is chosen with probability $p$ and $\pi_2$ is chosen with probability $(1 - p)$. There exists a distribution $\mathcal{D}'$ over
permutations and a distribution $\mathcal{V}$ over the disjoint partitions $\{(V_1 \cup \cdots \cup V_i)\}$ where each $V_i$ has at most $2\sqrt{n}$ elements, such that for any $k \geq 2$ and any $k$AP instance $I = (V, \omega)$ such that $\omega = \omega' \cdot \omega''$ where $\omega''$ is $\Delta$-dense, we have

$$\mathbb{E}_{\pi \sim D'}[\text{val}(I, \pi)] \geq \text{val}_H(I, D) - \frac{2k}{\Delta \sqrt{n}} - \frac{1}{\Delta} \sum_{1 \leq i < j < k} \mathbb{E} \sum_{\pi' \sim D'} \Pr_{(u_1, \ldots, u_k) \sim \mathcal{V}}[u_i \in V_t \text{ and } u_j \in V_r].$$

Moreover, $D'$ can be sampled in polynomial time.

**Proof.** Let us assume w.l.o.g. that $V = [n], \pi_1 = \text{id}$ (i.e. $\pi_1(i) = i$ for all $i \in [n]$). For any set $A = \{a_i : a_1 < a_2 < \cdots < a_{|A|} = n\} \subseteq [n]$, let us define $\pi_A$ be the permutation over $[n]$ so that $\pi_A(i) = a_{i-1} + 1$ if $i = a_t$ for some $t \in [|A|]$ and $\pi_A(i) = i + 1$ otherwise (assuming $a_0 = 0$). We can also assume w.l.o.g. that there exists $A \subseteq [n]$ such that $\pi_2 = \pi_A$.

Now we define the random set variable $A' : A \subseteq A \subseteq V$ as follows. We start from $A' = A$, and for each $i \in [\sqrt{n}]$, we uniformly sample an element $a$ from $((i - 1)\sqrt{n}, i\sqrt{n}]$ and let $A' \leftarrow A' \cup \{a\}$. In this way, we know that there is no set of $2\sqrt{n}$ consecutive integers that does not intersect $A'$. It is easy to show that for every $v \in V$, $\Pr_{A'}[\pi_A(v) \neq \pi_{A'}(v)] \leq \frac{2}{\sqrt{n}}$.

Let $D_{A'}$ be the probability distribution that chooses $\pi_1$ with probability $p$ and $\pi_{A'}$ with probability $(1 - p)$. For any $k$ and any $k$AP instance $I = (V, \omega)$ such that $\omega$ is $\Delta$-dense, we have

$$\text{val}_H(I, D) - \mathbb{E}_{A'} \text{val}_H(I, D_{A'})$$

$$= n^k \mathbb{E}_{(U, W) \sim \omega} \left( \prod_{i=1}^{k} \Pr_{\pi \sim D}[\pi(u_i) = w_i] - \prod_{i=1}^{k} \Pr_{\pi \sim D_{A'}}[\pi(u_i) = w_i] \right)$$

$$\leq n^k \mathbb{E}_{U \sim \omega', W \sim \omega''} \prod_{A'}[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)] \cdot \prod_{i=1}^{k} \Pr_{\pi \sim D}[\pi(u_i) = w_i]$$

$$\leq \frac{n^k}{\Delta} \cdot \mathbb{E}_{U \sim \omega', W \sim \omega''} \prod_{A'}[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)] \cdot \mathbb{E}_{W \sim V^k} \prod_{i=1}^{k} \Pr_{\pi \sim D}[\pi(u_i) = w_i],$$

where the last inequality is by the density of $\omega''$. By (4.5), we have

$$\frac{n^k}{\Delta} \cdot \mathbb{E}_{U \sim \omega', A'} \Pr[\exists i \in [k] : \pi_{A'}(u_i) \neq \pi_A(u_i)] \leq \frac{2k}{\Delta \sqrt{n}}.$$  

(4.11)
For every $A' \subseteq [n]$, we define the distribution $\mathcal{D}'_{A'}$. Let us assume that the elements in $A'$ are $a'_1 < a'_2 < \ldots < a'_{|A'|} = n$; let $a'_0 = 0$ for convenience. To draw a permutation $\pi \sim \mathcal{D}'_{A'}$, we sample $|A'|$ i.i.d. 0/1 bits $b_1, b_2, \ldots, b_{|A'|}$, each of which has mean $p$. For each $i$, we find out the unique $t \in [|A'|]$ so that $a'_{t-1} < i \leq a'_t$; let $\pi(i) = \pi_1(i) = i$ if $b_t = 0$; let $\pi(i) = \pi_{A'}(i)$ otherwise.

Now we define the distribution $\mathcal{D}'$. To draw a permutation $\pi \sim \mathcal{D}'$, we first sample a random set $A'$, and then draw a permutation from $\mathcal{D}'_{A'}$.

For any $k$ and any $k$AP instance $\mathcal{I} = (V, \omega)$ such that $\omega = \omega' \cdot \omega''$ where $\omega''$ is $\Delta$-dense, we have

$$
\mathbb{E}_{A'} \mathbb{E}_\mathcal{H}(\mathcal{I}, \mathcal{D}_{A'}) - \mathbb{E}_{\pi \sim \mathcal{D}'} \mathbb{E}_{\pi \sim \mathcal{D}_{A'}} \mathbb{E}_{\pi \sim \mathcal{D}_{A'}} \mathbb{E} \left( \prod_{i=1}^{\Delta} \Pr_{\pi \sim \mathcal{D}_{A'}}[\pi(u_i) = w_i] \right)
= n^k \mathbb{E}_{U \sim \omega} \mathbb{E}_{\mathcal{A}} \left( \prod_{i=1}^{\Delta} \Pr_{\pi \sim \mathcal{D}_{A'}}[\pi(u_i) = w_i] \right)
\leq n^k \mathbb{E}_{U \sim \omega} \mathbb{E}_{\mathcal{A}} \left( \prod_{i=1}^{\Delta} \Pr_{\pi \sim \mathcal{D}_{A'}}[\pi(u_i) = w_i] \right)
\leq n^k \mathbb{E}_{U \sim \omega} \mathbb{E}_{\mathcal{A}} \left[ \exists t \in [|A'|]: \exists i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right]
\leq n^k \mathbb{E}_{U \sim \omega} \mathbb{E}_{\mathcal{A}} \left[ \exists t \in [|A'|]: \exists i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right]
\leq \frac{1}{\Delta} \mathbb{E}_{U \sim \omega} \mathbb{E}_{\mathcal{A}} \left[ \exists t \in [|A'|]: \exists i \text{ s.t. } a'_{t-1} < u_i \leq a'_t \right]
\leq \frac{1}{\Delta} \sum_{1 \leq i \leq k} \mathbb{E}_{A'} \sum_{t \in [|A'|]} \Pr_{|A'|} \left[ a'_{t-1} < u_i, u_i \leq a'_t \right].
$$

The lemma is proved by combining (4.11) and (4.12).

Now we are ready to prove Lemma 4.4.5
of Lemma 4.4.5 Let $\mathcal{D}$ be supported on $\pi_1, \pi_2, \ldots, \pi_m$, each $\pi_i$ is chosen with probability $p_i$. We can assume that $m \leq n^2$ by preserving only the $n^2$ permutations with the largest probabilities and proper normalization, which would cause a loss of at most $n^{-1}$ in the objective value $\text{val}_H(\mathcal{I}, \mathcal{D})$. Now we show that for any such distribution, we can find a distribution $\mathcal{E}$ that is supported on $(m - 1)$ permutations, such that

$$\text{val}_H(\mathcal{I}, \mathcal{E}) \geq \text{val}_H(\mathcal{I}, \mathcal{D}) - (p_1 + p_2) \frac{2k^2}{\Delta' \sqrt{n}}. \quad (4.13)$$

In other words, since $\pi_1$ and $\pi_2$ are arbitrary, we are able to “merge” any two permutations $\pi_i$ and $\pi_j$ in $\mathcal{D}$ by paying a loss of $(p_1 + p_2) \frac{2k^2}{\Delta' \sqrt{n}}$ in the objective value. We repeatedly merge the two permutations with the smallest probability mass in the distribution until there is only one permutation left, during this process we lose at most $rac{6k^2 \log n}{\Delta' \sqrt{n}}$ in objective value. Together with the $n^{-1}$ loss at the beginning of the proof, we lose at most $\frac{7k^2 \log n}{\Delta' \sqrt{n}}$ for sufficiently large $n$.

In order to show (4.13), let us define a distribution $\tilde{\mathcal{E}}$ of distributions of permutations as follows. Let $\mathcal{F}$ be the distribution of permutations that chooses $\pi_1$ with probability $\frac{p_1}{p_1 + p_2}$ and $\pi_2$ with the remaining probability. Apply Lemma 4.7.2 on $\mathcal{F}$ to get $\mathcal{F}'$. A distribution $\mathcal{E}$ from $\tilde{\mathcal{E}}$ is sampled by first sampling a permutation $\pi$ from $\mathcal{F}'$, and returning the distribution that puts probability mass $(p_1 + p_2)$ on $\pi$ and $p_i$ on $\pi_i$ for all $i : 3 \leq i \leq m$. For every $u, w \in V$, let $\gamma_{u,w} = \sum_{t=3}^{m} p_t 1[\pi_t(u) = w] \leq 1$. We have

$$\mathbb{E}_{\mathcal{E} \sim \tilde{\mathcal{E}}} \text{val}_H(\mathcal{I}, \mathcal{E}) = \mathbb{E}_{\mathcal{E} \sim \tilde{\mathcal{E}}} \mathbb{E}_{(U,W) \sim \omega} \prod_{i=1}^{k} \mathbb{P}[\pi(u_i) = w_i]$$

$$= \mathbb{E}_{\pi \sim \mathcal{F}'} \mathbb{E}_{(U,W) \sim \omega} \prod_{i=1}^{k} \left( (p_1 + p_2) 1[\pi(u_i) = w_i] + \gamma_{u_i,w_i} \right)$$

$$\leq \sum_{Q \subseteq [k]} \mathbb{E}_{(U,W) \sim \omega} \left( \prod_{i \in Q} (p_1 + p_2) 1[\pi(u_i) = w_i] \right) \left( \prod_{i \in Q^c} \gamma_{u_i,w_i} \right)$$

$$\leq \sum_{Q \subseteq [k]} \mathbb{E}_{(U,W) \sim \omega} \left( \prod_{i \in Q} \gamma_{u_i,w_i} \right) \left( (p_1 + p_2)^{|Q|^2} \mathbb{E}_{(U,Q,W) \sim \omega|U \sim U_{\mathcal{I}}} \prod_{i \in Q} 1[\pi(u_i) = w_i] \right), \quad (4.14)$$

where $U_A$ is the restriction of vector $U$ over coordinates in $A$, $\omega''_A$ is the marginal distribution of $\omega''$ over the coordinates in $A$, and $\omega_{(A,B)}$ is the marginal distribution of $\omega$ over $A$ in the first $k$ coordinates and $B$ in the last $k$ coordinates. Let $\mathcal{I}_{U,\mathcal{I}}$ be the $|Q|\cdot\text{AP}$
instance \((V, \omega| (U_{T}, W_{T}))\). We know that \(\omega|(U_{T}, W_{T}) = (\omega'|U_{T}) \cdot (\omega''|W_{T})\), and \(\omega''|W_{T}\) is \(\Delta \cdot n^{\overline{\mathcal{Q}}} \cdot \omega''_{Q}(W_{T})\)-dense. Therefore for every \(Q \neq \emptyset\), by Lemma 4.7.2, we have

\[
\mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})}\left( \mathbb{E}_{\pi \sim \mathcal{F}} \prod_{i \in Q} 1[\pi(u_i) = w_i] \right) = \mathbb{E}_{\pi \sim \mathcal{F}} \text{val}(\mathcal{I}_{U_{T}, W_{T}}, \pi)
\]

\[
\geq \text{val}_{\mathcal{H}}(\mathcal{I}_{U_{T}, W_{T}}, \mathcal{F}) - \frac{2kn^{-|\overline{\mathcal{Q}}|}}{\Delta_{\omega''_{Q}(W_{T})} \sqrt{n}}
\]

\[
- \frac{n^{-|\overline{\mathcal{Q}}|}}{\Delta_{\omega''_{Q}(W_{T})}} \sum_{i,j \in Q \atop i \neq j} \sum_{\nu \sim \omega'|U_{T}} \sum_{t \in [t]} \Pr_{U_{Q} \sim \omega'|U_{T}}[u_i \in V_t \text{ and } u_j \in V_t]
\]

\[
= \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})} \prod_{i \in Q} \left( p_1 1[\pi_1(u_i) = w_i] + p_2 1[\pi_2(u_i) = w_i] \right) - \frac{2kn^{-|\overline{\mathcal{Q}}|}}{\Delta_{\omega''_{Q}(W_{T})} \sqrt{n}}
\]

\[
- \frac{n^{-|\overline{\mathcal{Q}}|}}{\Delta_{\omega''_{Q}(W_{T})}} \sum_{i,j \in Q \atop i \neq j} \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})} \prod_{i \in Q} \Pr_{U_{Q} \sim \omega'|U_{T}}[u_i \in V_t \text{ and } u_j \in V_t]
\]

Therefore we have

\[(4.14)\]

\[
\geq \sum_{Q \subseteq [k]} \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})} \left( \prod_{i \in Q} \gamma_{u_i, w_i} \right) \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})} \prod_{i \in Q} \left( p_1 1[\pi_1(u_i) = w_i] + p_2 1[\pi_2(u_i) = w_i] \right)
\]

\[
- \sum_{\emptyset \neq Q \subseteq [k]} \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})} \left( \prod_{i \in Q} \gamma_{u_i, w_i} \right) (p_1 + p_2)^{|Q|} \cdot \frac{2kn^{-|\overline{\mathcal{Q}}|}}{\Delta_{\omega''_{Q}(W_{T})} \sqrt{n}}
\]

\[
- \sum_{\emptyset \neq Q \subseteq [k]} \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega|(U_{T}, W_{T})} \left( \prod_{i \in Q} \gamma_{u_i, w_i} \right) (p_1 + p_2)^{|Q|}
\]

\[
\cdot \frac{n^{|\overline{\mathcal{Q}}|}}{\Delta_{\omega''_{Q}(W_{T})}} \sum_{i,j \in Q \atop i \neq j} \mathbb{E}_{(U_{Q}, W_{Q}) \sim \omega'|U_{T}} \sum_{t \in [t]} \Pr_{U_{Q} \sim \omega'|U_{T}}[u_i, u_j \in V_t]
\]

\[
= \mathbb{E}_{(U, W) \sim \omega} \left( \sum_{i=1}^{m} p_i 1[\pi_i(u_i) = w_i] \right)
\]
\[
\sum_{\emptyset \neq Q \subseteq [k]} \frac{E}{W} \prod_{i \in Q} \gamma_{u_i, w_i} (p_1 + p_2)^{|Q|} \cdot \frac{2k}{\Delta \sqrt{n}}
\]

\[
\sum_{\emptyset \neq Q \subseteq [k]} \frac{E}{W} \prod_{i \in Q} \gamma_{u_i, w_i} (p_1 + p_2)^{|Q|} \sum_{i \neq j} \frac{1}{\Delta} \sum_{i \in Q} \sum_{j \in Q} 1[u_i, u_j \in V']
\]

\[
\geq \text{val}_{\mathcal{H}}(\mathcal{I}, D) - (p_1 + p_2) \frac{2k^2}{\Delta \sqrt{n}} \geq \text{val}_{\mathcal{H}}(\mathcal{I}, D) - (p_1 + p_2) \frac{2k^2}{\Delta \sqrt{n}}.
\]

(by well-spreadness of \(\omega'\) and the maximum size of \(|V'_t|\))

In all, we have proved that \(E_{\mathcal{E} \sim \tilde{\mathcal{E}}} \text{val}_{\mathcal{H}}(\mathcal{I}, \mathcal{E}) \geq \text{val}_{\mathcal{H}}(\mathcal{I}, D) - (p_1 + p_2) \frac{2k^2}{\Delta \Delta' \sqrt{n}}\). Since \(\tilde{\mathcal{E}}\) can be sampled in polynomial time, there is a randomized polynomial-time algorithm to find out a \(\mathcal{E}\) satisfying (4.13).

4.8 Bisection MaxkCSP

In this section, we consider the bisection Max-CSP as a notable example of Max-CSP with globally cardinality constraints.

Fix a finite domain \(D\) and a \(k\)-CSP instance \(\mathcal{I}\) over \(D\). A global cardinality constraint is a linear constraint on the numbers of variables that are assigned to the values in \(D\).

For simplicity and illustration purpose, here we only consider the bisection constraint – i.e., assuming \(D = \{0, 1\}\), the number of variables that take value 1 is exactly \(n/2\) (for even integers \(n\)). For a bisection \(k\)-CSP instance \(\mathcal{I} = (V, \omega, P)\), we define its optimal value to be

\[
\text{opt}(\mathcal{I}) = \max_{\alpha:|\{v \in V: \alpha(v) = 1\}| = n/2} \{\text{val}(\mathcal{I}, \alpha)\},
\]

where the definition of \(\text{val}(\mathcal{I}, \alpha)\) remains the same as in the ordinary \(k\)-CSP case.

The \(\ell\)-round Sherali-Adams relaxation for a bisection \(k\)-CSP instance \(\mathcal{I} = (V, \omega, P)\)
\( (\ell \geq k) \) is written as follows.

\[
\begin{align*}
&\text{maximize } \mathbb{E}_{S \sim \omega} \mathbb{E}_{\alpha \sim \mu_S} [P_S(\alpha)] \\
&\text{subject to } \Pr_{\alpha \sim \mu_S} [\alpha|_{S \cap T} = \beta] = \Pr_{\alpha \sim \mu_T} [\alpha|_{S \cap T} = \beta] \\
&\quad \forall S, T \subseteq V, |S \cup T| \leq \ell, \beta \in D^{S \cap T} \\
&\quad \sum_{v \in V} \Pr_{\alpha \sim \mu_{S \cup \{v\}}} [\alpha = \beta \text{ and } \alpha(v) = 1] = \frac{n}{2} \cdot \mu_S(\beta) \\
&\quad \forall S \subseteq V, S < \ell, \beta \in \{0, 1\}^S,
\end{align*}
\]

where the last constraint corresponds to the bisection constraint.

We now turn to how to round \( \kappa \)-independent solutions. The following lemma is similar to Lemma 4.4.1.

**Lemma 4.8.1.** Let \( I = (V, \omega, P) \) be a bisection \( k \)CSP instance. Let \( \mu \) be an \( \kappa \)-independent solution (with respect to both uniform distribution and \( \omega \), \( 0 \leq \kappa \leq 1 \)) to the \( k \)-round Sherali-Adams LP relaxation. There is a randomized polynomial time algorithm to find an assignment \( \alpha : V \rightarrow \{0, 1\} \) such that \( \text{val}(I, \alpha) \geq \text{val}_{\text{LP}}(I, \mu) - 3k\kappa^{1/4} \) and \( |\{v \in V : \alpha(v) = 1\}| = n/2 \).

**Proof.** We sample \( \alpha \) in the same way as we did in the proof of Lemma 4.4.1 and we see that \( E_{\alpha}[\text{val}(I, \alpha)] \geq \text{val}_{\text{LP}}(I, \mu) - 2\sqrt{\kappa} \). Also observe that

\[
\mathbb{E}_{\alpha} \left[ \sum_{v \in V} \alpha(v) - \frac{n}{2} \right] \leq \sqrt{\mathbb{E}_{\alpha} \left( \sum_{v \in V} \alpha(v) - \frac{n}{2} \right)^2} = \sqrt{\sum_{v_1, v_2 \in V} \mathbb{E}_{\alpha} \left[ \alpha(v_1) \alpha(v_2) \right] - n \sum_{v \in V} \mathbb{E}_{\alpha} \left[ \alpha(v) \right] + \frac{n^2}{4} = \sqrt{\sum_{v_1, v_2 \in V} \mathbb{E}_{\alpha} \left[ \alpha(v_1) \alpha(v_2) \right] - \frac{n^2}{4}} \leq \sqrt{\sum_{v_1, v_2 \in V} \Pr_{\beta \sim \mu_{\{v_1, v_2\}}} [\beta(v_1) = \beta(v_2) = 1] + \sqrt{\kappa - \frac{n^2}{4}}} = \kappa^{1/4},
\]

where the last inequality is because of \( \kappa \)-independence with respect to uniform distribution, the definition of total correlation, and Lemma 4.2.1; the last equality is because of Sherali-Adams constraints.

In all, we have

\[
\mathbb{E}_{\alpha} \left( \text{val}(I, \alpha) - k \left| \sum_{v \in V} \alpha(v) - \frac{n}{2} \right| \right) \geq \text{val}_{\text{LP}}(I, \mu) - 2\sqrt{\kappa} - k\kappa^{1/4}
\]

64
We can sample an \( \alpha \) in expected polynomial time so that

\[
\mathbb{E}_\alpha \left[ \text{val}(\mathcal{I}, \alpha) - \sum_{v \in V} \alpha(v) - \frac{n}{2} \right] \geq \text{val}_{LP}(\mathcal{I}) - 3k\kappa^{1/4}.
\]

By greedily rearranging \( \left| \mathbb{E}_{v \in V} [\alpha(v)] - \frac{1}{2} \right| \)-fraction of the entries in \( \alpha \), we get a bisection assignment \( \alpha' \) such that \( \text{val}(\mathcal{I}, \alpha') \geq \text{val}_{LP}(\mathcal{I}, \mu) - 3k\kappa^{1/4} \).

Finally as a counterpart to Theorem 4.5.1, we show the following.

**Theorem 4.8.2.** Let \( \mathcal{I} = (V, \omega, P) \) be a bisection \( k \)CSP instance over domain \( \{0, 1\} \) such that \( \omega \) is \( \Delta \)-dense or \( \Delta \)-locally dense. For any \( \epsilon > 0 \), let \( \ell = \frac{2^{3k64^k \log |D|}}{\epsilon^3 k} \). The additive integrality gaps of the \( (\ell + k) \)-round Sherali-Adams LP relaxation is at most \( \epsilon \); and there is a randomized rounding algorithm producing a solution whose value is at least \( \text{opt}(\mathcal{I}) - \epsilon \), in expected \( n^{O(\ell)} \) time.

**Proof.** Let \( \mu \) be a solution to the \( (\ell + k) \)-round Sherali-Adams LP relaxation. Similar as in the proof of Theorem 4.5.1, Corollary 4.3.4 and Corollary 4.3.6, we know that there exists \( t \leq \ell \) such that

\[
\mathbb{E}_{U \sim V^t} \left[ \mathbb{E}_{S \sim V^k} C_\mu(x_S|x_U) + \mathbb{E}_{S \sim \omega} C_\mu(x_S|x_U) \right] \leq \frac{\epsilon}{3k^2}.
\]

Therefore, together with Lemma 4.3.2, we have

\[
\mathbb{E}_{U \sim V^t} \left( \text{val}_{LP}(\mathcal{I}, \mu|x_U) - 3k \left( \sqrt{\frac{1}{\mathbb{E}_{S \sim V^k} C_\mu(x_S|x_U)} + \sqrt{\frac{1}{\mathbb{E}_{S \sim \omega} C_\mu(x_S|x_U)}}} \right) \right) \geq \text{val}_{LP}(\mathcal{I}, \mu) - \epsilon.
\]

We enumerate all the possible ways of conditioning, and find out a solution \( \mu' \) to the \( (k + \ell - t) \)-round Sherali-Adams LP relaxation such that

\[
\text{val}_{LP}(\mathcal{I}, \mu') - 3k \left( \sqrt{\frac{1}{\mathbb{E}_{S \sim V^k} C_{\mu'}(x_S)} + \sqrt{\frac{1}{\mathbb{E}_{S \sim \omega} C_{\mu'}(x_S)}}} \right) \geq \text{val}_{LP}(\mathcal{I}, \mu) - \epsilon.
\]
Since $\mu'$ is always $\kappa$-independent with respect to both uniform distribution and $\omega$ for $\kappa = E_{S \sim \mathcal{V}^k} C_{\mu'}(x_S) + E_{S \sim \omega} C_{\mu'}(x_S)$, by Lemma 4.8.1 given $\mu'$, we can find an assignment with value at least $\text{val}_{LP}(\mathcal{I}, \mu) - \epsilon$ in randomized polynomial time.

### 4.9 A gap instance for relaxation $\mathcal{H}$

In this section, we show a gap instance for the relaxation $\mathcal{H}$. Consider the following 2-AP instance $\mathcal{I}([5], \omega)$. Let us define $\omega_{i,j,p,q} = \frac{1}{64} A_{i,j} B_{p,q}$, where

$$
A = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0
\end{bmatrix}, \quad \text{and} \quad
B = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}.
$$

If we view $A$ and $B$ as the adjacency matrices of two 5-vertex graphs, $\text{val}(\mathcal{I}, \pi)$ is the number of edges in $A$ that are mapped to an edge in $B$ by $\pi$, multiplied by $\frac{25}{32}$. Since $A$ is a 4-cycle with one isolated vertex, and $B$ is a 3-cycle plus an edge, at most 2 edges in $A$ can be mapped to $B$. Therefore, $\text{opt}(\mathcal{I}) = \frac{25}{16}$.

On the other hand, let us consider the following distribution $\mathcal{D}$ of permutations, where $\mathcal{D}$ is supported on $\pi_1$ and $\pi_2$ with equal probability $(1/2)$. $\pi_1$ is the identity permutation; $\pi_2(i) = (i \mod 5) + 1$ for all $i \in [5]$. We have

$$
\text{val}(\mathcal{I}, \mathcal{D}) = \frac{25}{64} \cdot \sum_{i,j} \sum_{p,q} A_{i,j} B_{p,q} \frac{1}{2} (1[\pi_1(i) = p \lor \pi_2(i) = p]) \cdot \frac{1}{2} (1[\pi_1(j) = q \lor \pi_2(j) = q])
$$

$$
= \frac{225}{128} > \text{opt}(\mathcal{I}).
$$
Chapter 5

Lasserre integrality gaps for \textsc{DensekSubgraph}

5.1 Introduction

As we defined at the beginning of this thesis, the \textsc{DensekSubgraph} problem takes as input a graph $G(V, E)$ on $n$ vertices and a parameter $k$, and asks for a subgraph of $G$ on at most $k$ vertices having the maximum number of edges.

While it is a fundamental graph optimization problem and arises in several applications (community detection in social networks, identifying protein families and molecular complexes in protein-protein interaction networks, etc), there is a huge gap between the best approximation algorithm and the known inapproximability results. The current best approximation algorithm due to [41] gives $O(n^{1/4+\epsilon})$-factor approximation algorithm which runs in time $n^{O(1/\epsilon)}$ for any constant $\epsilon > 0$. On the inapproximability side, [88] initially showed a small constant factor inapproximability for \textsc{DensekSubgraph} using the random 3-SAT assumption. [137] used quasi-random PCPs to rule out a PTAS. More recently, [193, 7] used more non-standard assumptions to rule out any constant factor approximation algorithms.

While only constant factor approximations have been ruled out, it is commonly believed that \textsc{DensekSubgraph} is much harder to approximate even on average (for a natural distribution on hard instances). Recently, average-case hardness assumptions based on the hardness of “planted” versions of \textsc{DensekSubgraph} were used for public key cryptography [12] and in showing that financial derivate can be fraudulently priced without detection [15]. Given the interest in \textsc{DensekSubgraph} from both the algorithms...
and the complexity point of view, developing a better understanding of the problem is an important challenge for the field.

### 5.1.1 Our contributions

In this chapter, we study the limitation of the most powerful Lasserre SDP relaxation hierarchy for \textsc{DensekSubgraph}. We show an integrality gap of polynomial ratio \(n^\epsilon\), for small enough constant \(\epsilon\) for almost linear \(O(\epsilon)\) levels of the Lasserre relaxation. If we only aim at an integrality gap for polynomial \(n^\epsilon\) levels of the Lasserre relaxation, the ratio of the gap can be as large as \(n^{2/53-O(\epsilon)}\). Informally, we prove

**Theorem 5.1.1.** [Informal version of Theorem 5.3.6 and Theorem 5.3.7] For every \(\epsilon > 0\), there is a lower bound of \(n^{2/53-\epsilon}\) on the integrality gap of level-\(n^{3\epsilon}\) Parrilo-Lasserre SDP relaxation hierarchy for the \textsc{DensekSubgraph} problem; there is also a lower bound of \(n^{\epsilon}\) on the integrality gap of level-\(n^{1-O(\epsilon)}\) Parrilo-Lasserre SDP relaxation hierarchy.

As we mentioned in the introduction part of this thesis, our integrality gaps are among the few known cases where the integrality gap ratio is much bigger than the best known \(\mathsf{NP}\)-hardness inapproximability bounds. In the absence of inapproximability results for \textsc{DensekSubgraph}, our results show that beating a factor of \(n^{\Omega(1)}\) is a barrier for even the most powerful SDPs, and in fact even beating the best known \(n^{1/4}\) factor is a barrier for current techniques.

### 5.2 Preliminaries

#### 5.2.1 Notations

We introduce some notation which will be used throughout this chapter. \(G = (V,E)\) refers to a graph which is an instance of the \textsc{DensekSubgraph} problem on \(n\) vertices, and \(k\) refers to the size of the subgraph we are required to output. For an induced subgraph \(H \subseteq G\), we denote by \(d(H)\) the average degree (or density of \(H\)). For a vertex \(v\) in subgraph \(H\), we will denote by \(\Gamma_H(v)\) the set of neighbors of \(v\) in \(H\) (the suffix will be dropped when \(H = G\)).

The phrase “with high probability” will mean: with probability \(1 - \frac{1}{p(n)}\), for any polynomial \(p(n)\). It will be clear from the context that there are constants which depend on the degree of \(p\).
5.2.2 The natural and min degree integer programmings for Dense$k$Subgraph

The natural integer programming for Dense$k$Subgraph has variables \( \{x_i\} \) to denote if vertex \( i \) belongs to the solution, and edge variables \( \{x_{ij}\}_{(i,j) \in E(G)} \) to denote if both \( i, j \) are in the subgraph.

For our integrality gaps, we will also consider to start with a different integer programming (D$k$S-IP2) which is equivalent upto a factor of 2 (see [41]). Intuitively, it tries to find a \( k \)-subgraph \( H \) such that the minimum degree \( d_H \) is maximized. An LP hierarchy obtained from this min. degree IP (D$k$S-IP2) was in fact used by [41] to obtain their approximation algorithm. (While the program as stated is not linear, we guess the degree \( d \) and consider the feasibility linear program that is obtained.)

<table>
<thead>
<tr>
<th>Natural IP (D$k$S-IP1)</th>
<th>Min degree IP (D$k$S-IP2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximize ( \sum_{(i,j) \in E(G)} x_i x_j )</td>
<td>Maximize ( d )</td>
</tr>
<tr>
<td>s.t. ( \sum_{i \in V} x_i \leq k )</td>
<td>s.t. ( \sum_{i \in V} x_i \leq k )</td>
</tr>
<tr>
<td>( \forall i \in V \ x_i \in {0, 1} )</td>
<td>( \forall i \in V \ \sum_{j \in \Gamma(i)} x_i x_j \geq d x_i )</td>
</tr>
<tr>
<td></td>
<td>( \forall i \in V \ x_{ij} = x_{ji} )</td>
</tr>
<tr>
<td></td>
<td>( \forall i \in V \ x_i \in {0, 1} )</td>
</tr>
</tbody>
</table>

5.2.3 The Lasserre hierarchy for Dense$k$Subgraph

As in [66], the \( r \)-level Lasserre SDP for Dense$k$Subgraph for the natural IP (D$k$S-IP1) introduces a vector \( U_S \) for each subset \( S \subseteq V \) with \( |S| \leq r \).
Lasserre hierarchy \((r \text{ levels})\) for \textsc{DensekSubgraph}:

\[
\sum_{(u,v) \in E} \| U_{\{u,v\}} \|^2
\]

such that
\[
\langle U_{S_1}, U_{S_2} \rangle \geq 0 \text{ for all } S_1, S_2
\]
\[
\langle U_{S_1}, U_{S_2} \rangle = \langle U_{S_3}, U_{S_4} \rangle \text{ when } S_1 \cup S_2 = S_3 \cup S_4
\]
\[
\sum_{v \in V} \| U_{\{v\}} \|^2 \leq k
\]
\[
\| U_{\emptyset} \|^2 = 1
\]

The intended solution sets \(U_S = U_{\emptyset}\) if every vertex in \(S\) belongs to the densest \(k\)-subgraph, and \(U_S = 0\) otherwise. The vector lengths \(\| U_S \|^2\) correspond to valid LP values \(x_S\) for the Sherali-Adams relaxation presented above.

The Lasserre SDP for the min degree IP \((\text{DkS-IP2})\) tries to find the \(k\)-subgraph of largest induced minimum degree \(d\). This can be captured by the SDP constraint

\[
\forall u \in V, \sum_{v \in \Gamma(u)} \| U_{u,v} \|^2 \geq d \cdot \| U_u \|^2 \tag{5.1}
\]

We will show in Section 5.3.3.1 that our integrality gaps also hold for the Lasserre hierarchy defined by this SDP. We refer to the SDP with constraint \((5.1)\) as the \textit{Min degree Lasserre SDP}.

5.3 The integrality gap

In this section, we show a gap instance with arbitrary large constant ratio for linear-round Lasserre relaxation, and a gap instance with \(n^\epsilon\) ratio for \(n^{1 - O(\epsilon)}\)-round Lasserre relaxation (Theorem 5.3.6). We also aim at maximizing the ratio of a polynomial-round Lasserre gap instance, getting a ratio of \(\Omega(\frac{n^2}{53} - \epsilon)\) (Theorem 5.3.7).

Our construction is based on a variant of Tulsiani’s gap instance for \(k\text{CSP}\) [220] – we extend the parameter range of Tulsiani’s instance. Then we convert the \(k\text{CSP}\) instance to a constraint-variable graph and duplicate the variable vertices, which is our gap instance for \textsc{DensekSubgraph}. Note that the gap for \(k\text{CSP}\) problem is indeed a set of random instances. The vector solution from Lasserre gap for \(k\text{CSP}\) will help us exhibit a good
Lasserre vector solution for DENSE-$k$SUBGRAPH. We finally use the structure of random instances of $k$CSP to show the soundness holds with high probability.

Now, let us proceed to the first step, the gap instance for $k$CSP.

### 5.3.1 Lasserre gap for $k$CSP from Tulsiani

We start by defining the $k$CSP problem.

**Definition 5.3.1.** Let $C \subseteq \mathbb{F}_q^K$ be a $q$-ary linear code of block length $K$.

1. An instance $\Phi$ of $k$CSP($C$) is a set of constraints $C_1, C_2, \ldots, C_m$ where each constraint $C_i$ is over a $K$-tuple $T_i = (x_{i1}, x_{i2}, \ldots, x_{iK})$, and is of the form $(x_{i1} + b^{(i)}_1, x_{i2} + b^{(i)}_2, \ldots, x_{iK} + b^{(i)}_K) \in C$ for some $b^{(i)} \in \mathbb{F}_q^K$.

2. A random instance of $k$CSP($C$) is sampled by choosing each constraint $C_i$ independently, where we sample $K$ variables without replacement from $[n]$ to get $T_i = (x_{i1}, x_{i2}, \ldots, x_{iK})$ and $b^{(i)}$ is chosen from $\mathbb{F}_q^K$ uniformly.

The following theorem is an extension of the main theorem in [220], showing that polynomial-round Lasserre relaxation cannot refute random $k$CSP with high probability.

**Theorem 5.3.2.** If $C$ is the dual code of a distance $2\delta \geq 3$ code (in terms of number of coordinates, not fractional distance), for every $10 \leq K < n^{1/2}$, if $n^{\kappa-1} \leq \eta \leq 1/(10^8 \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)})$ for some $\kappa > 0$, then for large enough $n$, a random instance $\Phi$ of $k$CSP($C$) over $m = \beta n$ constraints and $n$ variables, with probability $1 - o(1)$, admits perfect solution for the SDP relaxation obtained by $\eta n/16$ rounds of the Lasserre hierarchy, i.e. there are vectors $V_{(S, \alpha)}$ for all $S \subseteq [n]$ with $|S| \leq \eta n/16$ and all $\alpha : S \rightarrow \mathbb{F}_q$, such that

- the value of the solution is perfect: $\sum_{i=1}^m \sum_{\alpha : T_i \rightarrow \mathbb{F}_q} C_i(\alpha) \|V_{T_i, \alpha}\|^2 = m$;
- $\langle V_{(S_1, \alpha_1)}, V_{(S_2, \alpha_2)} \rangle \geq 0$ for all $S_1, S_2, \alpha_1, \alpha_2$;
- $\langle V_{(S_1, \alpha_1)}, V_{(S_2, \alpha_2)} \rangle = 0$ if $\alpha_1(S_1 \cap S_2) \neq \alpha_2(S_1 \cap S_2)$;
- $\langle V_{(S_1, \alpha_1)}, V_{(S_2, \alpha_2)} \rangle = \langle V_{(S_3, \alpha_3)}, V_{(S_4, \alpha_4)} \rangle$ for all $S_1 \cup S_2 = S_3 \cup S_4$ and $\alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4$;
- $\|V_{(\emptyset, \emptyset)}\|^2 = 1$ and $\sum_{j \in \mathbb{F}_q} \|V_{(\{i\}, \{i-j\})}\|^2 = 1$ for all $i \in [n]$. 


Note that Theorem 5.3.2 extends the original theorem of [220] to the regime where $K$ might be superconstant (even poly($n$)). The proof of Theorem 5.3.2 follows the proof in Tulsiani’s paper, with the following changes.

Recall that Tulsiani showed that, if the constraint-variable graph of a $k$CSP($C$) instance has very high left-expansion, then the Lasserre SDP admits perfect solution for it. Formally, the following lemma is (implicitly) shown in [220].

**Lemma 5.3.3 ([220]).** Given a $k$CSP($C$) instance, if every set of constraints of cardinality $s \leq r$ involves more than $(K - \delta)s$ variables (where $2\delta$ is the distance of the dual code of $C$), and if $4\delta \leq K$, then there is a perfect solution for the SDP relaxation obtained by $r/16$ rounds of the Lasserre hierarchy.

Hence, we only need to prove the following lemma which shows that the constraint-variable graph still has very high left-expansion, even when a constraint might involve superconstant many variables (i.e. the left degree might be superconstant).

**Lemma 5.3.4.** Given $\beta, \eta, K$ as in Theorem 5.3.2 with probability $1 - o(1)$, for all $2 \leq s \leq \eta n$, every set of $s$ constraints involves more than $(K - \delta)s$ variables.

A similar lemma can be found in [220] (Lemma A.1), which only deals with constant $K$. We need a more refined argument for superconstant $K$, which is in Section 5.3.4.

### 5.3.2 The gap instance for Dense$k$Subgraph

The gap instance is reduced from the gap instance for $k$CSP in Theorem 5.3.2. Let $C$ be the dual code of a $[K, K - t, 2\delta]_q$ code as used in Theorem 5.3.2, where $K$ is the block length, $(K - t)$ is the dimension, and $2\delta \geq 3$ is the distance of the code. Such a code has size $|C| = q^t$, and is very sparse for small enough $t$. For $1000 < q$ and $K > q^2$, we let $\beta = (40q^{t+2} \ln q)/K$, and do the following reduction.

Given a $k$CSP($C$) instance $\Phi$ with $m = \beta n$ constraints and $n$ variables. Let $G_\Phi = (L_\Phi, R_\Phi, E_\Phi)$ be the bipartite graph with $m|C|$ left vertices and $nq$ right vertices. For every constraint $C_i$ and every partial assignment to variables in the corresponding tuple $T_i$ which satisfies the constraint $C_i$, we introduce a left vertex. For every variable $x_j$ and its corresponding assignment, we introduce a right vertex. Formally,

\[
L_\Phi = \{(C_i, \alpha) | i \in [m], \alpha : T_i \rightarrow \mathbb{F}_q, C_i(\alpha) = 1\},
\]

\[
R_\Phi = \{(x_j, \alpha) | j \in [n], \alpha : \{x_j\} \rightarrow \mathbb{F}_q\}.
\]
We connect a left vertex \((C_i, \alpha)\) and right vertex \((x_j, \alpha')\) when \(x_j \in T_i\) and \(\alpha'\) is consistent with \(\alpha\), i.e.

\[
E_\Phi = \{((C_i, \alpha), (x_j, \alpha')) | (C_i, \alpha) \in L_\Phi, x_j \in T_i, \alpha'(x_j) = \alpha(x_j)\}.
\]

Now we define the final graph \(G'_\Phi = (L_\Phi, R'_\Phi, E'_\Phi)\) in which we want to find a dense \(k\)-subgraph where \(k = 2m\). We take \(\beta\) copies of the right vertices in \(R_\Phi\) to get \(R'_\Phi\). To get \(E'_\Phi\), we connect a left vertex \(u \in L_\Phi\) and a right vertex \(v \in R'_\Phi\) if \(u\) is connected to \(v\)'s corresponding vertex in \(R_\Phi\) in \(E_\Phi\). The graph \(G'_\Phi\) has \(N = m|C| + \beta nq = O(nq^{2t+2} \ln q/K)\) vertices.

In our analysis of the reduction, we need a \(q\)-ary linear code \(C\) that has a small constant distance (but no less than 3), small block length (but more than \(q\)), and very high dimension. Thus, we instantiate the code \(C\) with Generalized BCH codes given by the following.

**Lemma 5.3.5 (Generalized BCH Codes).** For every prime tower \(q\), and integer \(2\delta \geq 3\), there are \(q\)-ary linear codes of block length \(K = q^2 - 1\), dimension \((K - 4\delta + 3)\), and distance at least \(2\delta\).

We include a simple proof of Lemma 5.3.5 as follows.

**Proof.** Let \(\gamma\) be a primitive element of \(\mathbb{F}_{q^2}\). Let \(D = 2\delta\) for notational ease. We construct the following code

\[
\tilde{C} = \{(c_1, c_2, \ldots, c_{q^2 - 1}) \in \mathbb{F}_{q^2}^{q-1} | c(1) = c(\gamma) = c(\gamma^2) = \cdots = c(\gamma^{D-2}) = 0, \text{ where } c(X) = c_1X + c_2X^2 + c_3X^3 + \cdots + c_{q^2 - 1}X^{q^2 - 1}\}.
\]

We first show that the distance of \(\tilde{C}\) is at least \(D\). Since \(\tilde{C}\) is a linear code, we only need to show that every non-zero codeword has weight at least \(D\).

We show the contrapositive statement: the only codeword of weight at most \(D - 1\) is 0. For every codeword of weight at most \(D - 1\), suppose the non-zero entries are in the set \(\{c_{i_1}, c_{i_2}, c_{i_3}, \ldots, c_{i_{D-1}}\}\). We have

\[
\begin{align*}
  c_{i_1} + & \cdots + c_{i_{D-1}} = 0\\
  \gamma^{i_1}c_{i_1} + & \cdots + \gamma^{i_{D-1}}c_{i_{D-1}} = 0\\
  \gamma^{2i_1}c_{i_1} + & \cdots + \gamma^{2i_{D-1}}c_{i_{D-1}} = 0\\
  & \vdots
\end{align*}
\]

73
\[ \gamma^{(D-2)i_1}c_{i_1} + \gamma^{(D-2)i_2}c_{i_2} + \gamma^{(D-2)i_3}c_{i_3} + \cdots + \gamma^{(D-2)i_{D-1}}c_{i_{D-1}} = 0 \]

Note that the coefficients form a Vandermonde matrix (which has full rank). Therefore we have \( c_{i_1} = c_{i_2} = c_{i_3} = \cdots = c_{i_{D-1}} = 0 \), i.e. the codeword is 0.

Now we show that the dimension of \( \tilde{C} \) is at least \((K - 2D + 3)\). Note that each constraint \( c(\gamma^i) = 0 \) (\( 1 \leq i \leq D - 2 \)) can be implemented by 2 linear constraints in \( \mathbb{F}_q \) (since \( \gamma^i \in \mathbb{F}_q^2 \)), while the constraint \( c(1) = 0 \) is indeed a linear constraint in \( \mathbb{F}_q \). Therefore, we need at most \( 2(D - 2) + 1 = 2D - 3 \) linear constraints for \( \tilde{C} \), i.e. the dimension of \( \tilde{C} \) is at least \((K - 2D + 3)\).

Finally, if the dimension of \( \tilde{C} \) is more than \((K - 2D + 3)\), we can take a linear subspace of \( \tilde{C} \) of dimension \((K - 2D + 3)\), while the distance of the subspace code is no less than the distance of \( \tilde{C} \).

5.3.3 Analysis

We get a family of gap instances \( G'_\Phi \) parameterized by \( q > 1000 \) and \( 2\delta \geq 3 \) (using Lemma 5.3.5). We obtain our two main results of this section by picking appropriate parameters for code \( C \) as follows. To get lasserre integrality gaps for \( N^{1-O(\epsilon)} \) levels , we show the following by setting the distance \( 2\delta = 3 \).

**Theorem 5.3.6.** For every \( 1000 < q < N^{\epsilon} \) (where \( \epsilon \) is an absolute small constant), there is a gap instance of ratio \( \Omega(q) \) for \( N/q^{O(1)} \)-level Lasserre SDP. The same construction also works for the min degree Lasserre SDP, when \( q = \Omega(\log n) \) and \( q < N^{\epsilon} \).

We now aim at getting a gap instance of ratio \( N^{\epsilon} \) for polynomial-round Lasserre SDP, where \( \epsilon \) is maximized. By setting \( q = n^\gamma \) for some small constant \( \gamma > 0 \), the distance \( 2\delta = 4 \), and optimizing the other parameters, we obtain the following (refer to section 5.3.3.4 for details)

**Theorem 5.3.7.** For small enough \( \kappa > 0 \), there is a gap instance of ratio \( N^{2/53-O(\kappa)} \) for the \( N^\kappa \)-round Min degree Lasserre SDP.

The two theorems follow because of [Theorem 5.3.2], [Lemma 5.3.8], [Lemma 5.3.9] (completeness) and [Lemma 5.3.10] (soundness). In the completeness case, we will use our \( r \)-level Lasserre solution for \( k \)CSP to show that the Lasserre SDP after \( R = r/K \) levels of the hierarchy has value at least \( \beta mK \). In the soundness case, we show that with probability \( 1 - o(1) \), the graph \( G'_\Phi \) does not have any \( 2m \)-subgraph of value more than \( 17/q \) times
the SDP value (Lemma 5.3.10). Therefore, the graph $G'_\Phi$ is a gap instance of ratio $\Omega(q)$ for $R$-round Lasserre SDP. We proceed by first proving these lemmas.

### 5.3.3.1 Completeness

**Lemma 5.3.8.** If the $k\text{CSP}(C)$ instance $\Phi$ admits perfect solution for $r$-round Lasserre SDP relaxation, then the $r/K$-round Lasserre SDP relaxation for the DensekSubgraph instance $G'_\Phi$ has a solution of value $\beta m K$.

**Proof.** For any set $S = L_\Phi \cup R'_\Phi$, suppose the left vertices included in $S$ are

$$(C_{i_1}, \alpha_{i_1}), (C_{i_2}, \alpha_{i_2}), \ldots, (C_{i_{r_1}}, \alpha_{r_1}),$$

and the right vertices included in $S$ are

$$(x_{j_1}, \alpha'_{j_1}), (x_{j_2}, \alpha'_{j_2}), \ldots, (x_{j_{r_2}}, \alpha'_{r_2}),$$

where $r_1 + r_2 \leq r/K$. Let

$$S' = T_{i_1} \cup T_{i_2} \cup \cdots \cup T_{i_{r_1}} \cup \{x_{j_1}\} \cup \{x_{j_2}\} \cup \cdots \cup \{x_{j_{r_2}}\}.$$ 

We have $|S'| \leq K r_1 + r_2 \leq r$. If all the partial assignments $\alpha_i$’s and $\alpha'_i$’s are consistent to each other (i.e. there are not two of them assigning the same variable to different values), we can define

$$\alpha = \alpha_1 \circ \alpha_2 \circ \cdots \circ \alpha_{r_1} \circ \alpha'_1 \circ \alpha'_2 \circ \cdots \circ \alpha'_{r_2},$$

and let $U_S = V_{S', \alpha}$, or we let $U_S = 0$.

We can check that all the Lasserre constraints are satisfied.

- For two sets $S_1, S_2$, either at least one of the vectors $U_{S_1}, U_{S_2}$ is 0 (therefore their inner-product is 0), or $U_{S_1} = V_{S'_1, \alpha_1}, U_{S_2} = V_{S'_2, \alpha_2}$ for some $S'_1, S'_2, \alpha_1, \alpha_2$ and \langle $U_{S_1}, U_{S_2}$ \rangle = \langle $V_{S'_1, \alpha_1}, V_{S'_2, \alpha_2}$ \rangle \geq 0.

- For any $S_1, S_2, S_3, S_4$ such that $S_1 \cup S_2 = S_3 \cup S_4$, either the set of partial assignments in $S_1 \cup S_2 = S_3 \cup S_4$ are consistent to each other, in which case we have $U_{S_1 \cup S_2} = U_{S_3 \cup S_4} = V_{S, \alpha}$ where $S$ is the union of all the variables included in $S_1 \cup S_2$ and $\alpha$ is the concatenation of the partial assignments in $S_1 \cup S_2$; or we have $U_{S_1 \cup S_2} = U_{S_3 \cup S_4} = 0$. 

75
The “(fractional) number of chosen vertices”

\[
\sum_{v \in L_\Phi \cup K_\Phi} \left\| U_{\{v\}} \right\|^2 = \sum_{(C_i, \alpha) \in L_\Phi} \left\| U_{\{(C_i, \alpha)\}} \right\|^2 + \sum_{(x_j, \alpha) \in K_\Phi} \left\| U_{\{(x_j, \alpha)\}} \right\|^2
\]

\[
= \sum_{i=1}^{m} \sum_{\alpha : T_i \to \mathbb{F}_q} C_i(\alpha) \left\| U_{\{(C_i, \alpha)\}} \right\|^2 + \beta \sum_{j=1}^{n} \sum_{\alpha : \{x_j\} \to \mathbb{F}_q} \left\| U_{\{(x_j, \alpha)\}} \right\|^2
\]

\[
= \sum_{i=1}^{m} \sum_{\alpha : T_i \to \mathbb{F}_q} C_i(\alpha) \left\| V_{\{T_i, \alpha\}} \right\|^2 + \beta \sum_{j=1}^{n} \sum_{\alpha : \{x_j\} \to \mathbb{F}_q} \left\| V_{\{(x_j, \alpha)\}} \right\|^2
\]

\[
= m + \beta n = 2m,
\]

and \( \left\| U_0 \right\|^2 = \left\| V_{\emptyset, \emptyset} \right\|^2 = 1. \)

Now, we calculate the value of the solution

\[
\sum_{(u,v) \in E_\Phi} \left\| U_{\{u,v\}} \right\|^2 = \beta \sum_{(u,v) \in E_\Phi} \left\| U_{\{u,v\}} \right\|^2 = \beta \sum_{i=1}^{m} \sum_{\alpha : T_i \to \mathbb{F}_q, C_i(\alpha) = 1} \sum_{x_j \in T_i} \left\| U_{\{(C_i, \alpha), (x_j, \alpha \cdot (x_j))\}} \right\|^2
\]

\[
= \beta \sum_{i=1}^{m} \sum_{\alpha : T_i \to \mathbb{F}_q, C_i(\alpha) = 1} K \left\| V_{\{T_i, \alpha\}} \right\|^2 = \beta \sum_{i=1}^{m} K = \beta mK.
\]

If we add the constraint \((5.1)\), we can still get a good SDP solution for the Min degree Lasserre SDP with high probability, as long as \(q\) is superconstant.

**Lemma 5.3.9.** For \( q = \Omega(\log n) \), with probability \( 1 - o(1) \), this vector solution also satisfies the added constraint \((5.1)\) with \( d = \beta K/2 \), i.e., for each vertex \( u \), we have

\[
\sum_{v \in \Gamma(u)} \left\| U_{\{u,v\}} \right\|^2 \geq \beta K/2 \cdot \left\| U_u \right\|^2.
\]

**Proof.** For each left vertex \((C_i, \alpha)\), we have

\[
\sum_{v \in \Gamma((C_i, \alpha))} \left\| U_{\{(C_i, \alpha), v\}} \right\|^2 = \beta \sum_{x_j \in T_i} \left\| U_{\{(C_i, \alpha), (x_j, \alpha \cdot (x_j))\}} \right\|^2 = \beta \sum_{x_j \in T_i} \left\| U_{\{(C_i, \alpha)\}} \right\|^2 = \beta K \left\| U_{\{(C_i, \alpha)\}} \right\|^2.
\]

For each right vertex \((x_j, \alpha')\), we have

\[
\sum_{v \in \Gamma((x_j, \alpha'))} \left\| U_{\{(x_j, \alpha'), v\}} \right\|^2 = \sum_{i : T_i \ni x_j, \alpha : T_i \to \mathbb{F}_q, C_i(\alpha) = 1, \alpha(x_j) = \alpha'(x_j)} \sum_{\alpha : \{x_j\} \to \mathbb{F}_q} \left\| U_{\{(x_j, \alpha'), (C_i, \alpha)\}} \right\|^2
\]

76
where the last equality is because we know that \( U\{C_i, \alpha\} = 0 \) when \( C_i(\alpha) \neq 1 \). By the property of Lasserre vectors, we know that for each \( i \in \left[ m \right] \),

\[
\sum_{\alpha : T_i \rightarrow F_q, \alpha(x_j) = \alpha'(x_j)} \| U\{C_i, \alpha\} \|^2 = \sum_{i : T_i \ni x_j} \sum_{\alpha : T_i \rightarrow F_q, \alpha(x_j) = \alpha'(x_j)} \| U\{C_i, \alpha\} \|^2 ,
\]

therefore

\[
\sum_{v \in \Gamma((x_j, \alpha'))} \| U\{x_j, \alpha', v\} \|^2 = \sum_{i : T_i \ni x_j} \| U\{x_j, \alpha'\} \|^2 .
\]

For \( q = \Omega(\log n) \), the expected number of constraints containing \( x_j \) is \( \beta K = \Omega((\log n)^{t+2}) = \Omega(\log n) \), by our choice of \( \beta \). Therefore, by Chernoff bound and union bound, with probability \( 1 - o(1) \), for all \( x_j \), there are at least \( \beta K / 2 \) constraints containing \( x_j \), and we have

\[
\sum_{v \in \Gamma((x_j, \alpha'))} \| U\{x_j, \alpha', v\} \|^2 \geq \beta K / 2 \cdot \| U\{x_j, \alpha'\} \|^2 ,
\]

for every \( x_j \) and \( \alpha' \).

### 5.3.3.2 Soundness

Now, we show that random instances of \( k \text{CSP} \) give rise to graphs \( G'_\Phi \) whose \( 2m \)-sized subgraphs have density \( O(\beta K / q) \). Note that the large alphabet size \( q \) allows us to get a much larger gap than we would starting from random AND instances \[88\]. This allows us some slack in the size of the subgraphs we need to argue about.

For \( C \) the dual of a \([K, K - t, 2\delta]_q \) code, we prove the following soundness lemma.

**Lemma 5.3.10.** When \( \beta \geq (40q^{t+2} \ln q) / K \), for a random \( k \text{CSP}(C) \) instance \( \Phi \), with probability \( 1 - o(1) \), any subgraph of \( G'_\Phi \) obtained by choosing \( 2m \) left vertices and \( 2m \) right vertices contains at most \( 17\beta mK / q \) edges, and therefore any \( 2m \)-subgraph of \( G'_\Phi \) contains at most \( 17\beta mK / q \) edges.

Note that \( G'_\Phi \) was constructed by taking \( \beta \) copies of the right bipartition and replicating the edges. To prove [Lemma 5.3.10], we only need to prove the following lemma.
Lemma 5.3.11. Suppose that $q > 1000, K > q^2/2, t \leq 10$. When $\beta \geq (40q^{t+2}\ln q)/K$, for a random $k$CSP$(C)$ instance $\Phi$, with probability $1-o(1)$, any subgraph of $G_\Phi$ obtained by choosing $2m$ left vertices and $2n$ right vertices contains at most $17mK/q$ edges.

Proof of Lemma 5.3.10 from Lemma 5.3.11. We only need to prove once there is a $2m \times 2m$ subgraph of $G'_\Phi$ with $t$ edges, there is a $2m \times 2n$ subgraph of $G_\Phi$ with at least $t/\beta$ edges. Fix $2m$ left vertices in $G'_\Phi$, to maximize the number of edges in the subgraph, we need to select the $2m$ right vertices with most edges connected to the chosen $2m$ left vertices. Since any two right vertices $G'_\Phi$ corresponding to the same right vertex in $G_\Phi$ have the same set of neighbors, there is a densest $2m \times 2m$ subgraph $H'$ of $G'_\Phi$ that, for any two such vertices, chooses either both or neither of them. Now we define an subgraph $H$ of $G_\Phi$ that contains the same $2m$ left vertices. It contains a right vertex if any copy of the vertex is contained in $H'$. $H$ contains $2m/\beta = 2n$ vertices, and it is easy to see that there are (at least) $t/\beta$ edges in $H$.

We proceed by fixing a set of $2n$ vertices $R$ on the right. Lemma 5.3.11 follows from the following claim by a standard union bound over all possible choices of $R$.

Claim 5.3.12. Recall that $G_\Phi = (L_\Phi, R_\Phi, E_\Phi)$. Suppose that $q > 1000, K > q^2/2, t \leq 10$. Fix a subset $R \subseteq R_\Phi$ (note that $R_\Phi$ is the same for all the instances $\Phi$ of $n$ variables), $|R| = 2n$, the probability (over choice of $\Phi$) that there does not exist a subset $L \subseteq L_\Phi$ of size $2m$ such that the number of edges in the induced subgraph by $L \cup R$ is more than $17mK/q$, is at least $1 - \exp(-mK/(10q^{t+2}))$.

Proof of Lemma 5.3.11 from Claim 5.3.12. Since there are only $\binom{q^n}{2n} \leq \exp(2n(\ln q + 1))$ choices of $R$, by a union bound, with probability at least

$$1 - \exp(2n(\ln q + 1)) \cdot \exp(-mK/(10q^{t+2})) = 1 - \exp(2n(\ln q + 1) - \beta nK/(10q^{t+2})),$$

there is no $2m \times 2n$ subgraph of $G_\Phi$ containing more than $17mK/q$ edges. The probability becomes $1 - o(1)$ when $\beta = (40q^{t+2}\ln q)/K$.

Proof of Claim 5.3.12. First, we show that with high probability, a constraint $C_i$ is “poorly satisfied”. That is, none of the left vertices corresponding to a constraint $C_i$ has more than $\Omega(K/q)$ neighbors in $R$ – this number is roughly $1/q$ times the corresponding value in completeness case. We prove this in the following two steps.

**Step 1.** Fix a subset $R \subseteq R_\Phi$, $|R| = 2n$, for each variable $x_j$, let $\deg(x_j)$ be the number of vertices in $R$ that corresponding to $x_j$, i.e. let $\deg(x_j) = |R \cap \{(x_j, \alpha) | \alpha : \{x_j\} \to \mathbb{F}_q\}|.$
For a subset of variables $T \subseteq \{x_1, x_2, \ldots, x_n\}$, let $\deg(T) = \sum_{x_j \in T} \deg(x_j)$. We call $T$ good if the average degree of variables in $T$ is not more than 4, i.e. $\deg(T) \leq 4|T|$.

For a random $T$ with $|T| = K$, note that the expected degree $E[\deg(T)] = 2K$. Therefore, by Hoeffding’s inequalities for sampling without replacement (Theorem 1 and Theorem 4 in [119]), we have

$$\Pr[T \text{ is not good}] = \Pr[\deg(T) > 4K] < \exp(-\ln(4/e) \cdot 2K/q) < \exp(-K/(2q)).$$

**Step 2.** Again, fix $R \subseteq R_T$, $T \subseteq \{x_1, x_2, \ldots, x_n\}$, for a codeword $\alpha$ on coordinates in $T$, i.e. $\alpha : T \rightarrow \mathbb{F}_q$, let $\agr_T(\alpha, R) = |\{(x_j, \alpha_{|x_j}|) | x_j \in T\} \cap R|$. For a constraint $C_i$, say it is poorly satisfied if for all $\alpha : T \rightarrow \mathbb{F}_q$ such that $C_i(\alpha) = 1$, we have $\agr_{T_i}(\alpha, R) \leq 8K/q$.

Recall that to sample a random constraint $C_i$, we first sample a random $K$-tuple $T_i$, and a random shifting function $b^{(i)}$. Note that for a fixed $\alpha : T \rightarrow \mathbb{F}_q$, and a fixed $T_i$ that is good, when we take a random shifting function $b^{(i)} : T_i \rightarrow \mathbb{F}_q$, we have $\mathbb{E}_{b^{(i)}}[\agr_{T_i}(\alpha - b^{(i)}, R)] = \deg(T_i)/q = 4K/q$, therefore, by standard Chernoff bound, for a fixed codeword $\alpha \in C$, the probability that $\alpha$ makes $C_i$ not poorly satisfied is bounded from above by

$$\Pr[\agr_{T_i}(\alpha - b^{(i)}, R) > 8K/q] < \exp(-\ln(4/e) \cdot 4K/q) < \exp(-K/q).$$

Since there are $|C| = q^t \leq q^{10}$ codewords, by a union bound, for $K > q^2/2$ and $q > 1000$, we have

$$\Pr[C_i \text{ is not poorly satisfied}|T_i \text{ is good}] < q^{10} \cdot \exp(-K/q) < \exp(-K/(2q)).$$

In all, we have

$$\Pr[C_i \text{ is poorly satisfied}] \geq \Pr[C_i \text{ is poorly satisfied}|T_i \text{ is good}] \cdot \Pr[T \text{ is good}]$$

$$> (1 - \exp(-K/q))(1 - \exp(-K/(2q))) > 1 - \exp(-K/(3q)).$$

Now, again, by standard Chernoff bound, we have

$$\Pr||\{C_i|C_i \text{ is not poorly satisfied}\}| > m/(q \cdot |C|) < (e \cdot |C| \cdot q \cdot \exp(-K/(3q)))^{m/(q|C|)}$$

$$\leq (e \cdot q^{10} \cdot q \cdot \exp(-K/(3q)))^{m/(q|C|)} < \exp(-K/(10q))^{m/(q|C|)} = \exp(-mK/(10q^{t+2})).$$

By the calculation above we know that with probability at least $1 - \exp(-mK/(10q^{t+2}))$, there are at most $m/(q \cdot |C|)$ constraints that are not poorly satisfied.
For each left vertex \((C_i, \alpha) \in L_{\Phi}\), if \(C_i\) is poorly satisfied, we know there are at most \(8K/q\) edges from \((C_i, \alpha)\) to \(R\). If \(C_i\) is not poorly satisfied, there are at most \(K\) edges to \(R_{\Phi}\) – this upper bound also applies to \(R\).

Therefore, with probability at least \(1 - \exp(-mK/(10q^{t+2}))\), any set of \(2m\) left vertices has at most \(2m \cdot 8K/q + m/|C| \cdot |C| \cdot K \leq 17mK/q\) edges connected to \(R\).

We now complete the proofs of the main theorems in this section.

### 5.3.3.3 Proof of Theorem 5.3.6

By combining Theorem 5.3.2, Lemma 5.3.8, Lemma 5.3.9 (completeness), and Lemma 5.3.10 (soundness) we see that with probability \(1 - o(1)\), the graph \(G'_\Phi\) provides a \(\Omega(q)\) integrality for the number of levels \(R\) given by

\[
R = \Omega\left(\frac{n}{K(\beta K^{2\delta+0.75})^{1/(\delta-1)}}\right) = \Omega\left(\frac{N}{q^{2t+2} \ln q \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)}}\right)
\]

\[
= \Omega\left(\frac{N}{K^{(2\delta-0.25)/(\delta-1)}q^{2t+2} + (t+2)/(\delta-1)\text{poly log } q}\right). \quad \text{(recall that } \beta = (40q^{t+2} \ln q)/K)\]

Recall that \(K = q^2 - 1\). By setting \(K = q^2 - 1\) and \(2\delta = 3\), we verify that the theorem holds.

### 5.3.3.4 Proof of Theorem 5.3.7

Let \(q = n^\gamma\), since \(N = O(nq^{2t+2} \ln q/K) = O(nq^{2t} \ln q)\), ratio of the gap due to Lemma 5.3.8 and Lemma 5.3.10 is

\[
\Omega(q) = \Omega(N^{\gamma/(1+2\gamma + o(1))}) = \Omega(N^{\gamma/(1+(8\delta-6)\gamma + o(1))}). \quad \text{(note that } t = 4\delta - 3)\]

This means that

\[
\epsilon = \frac{\gamma}{1 + (8\delta - 6)\gamma + o(1)} = \frac{1}{8\delta - 6} - \frac{1 + o(1)}{(1 + (8\delta - 6)\gamma + o(1))(8\delta - 6)}.
\]

Note that when \(2\delta \geq 3\) is fixed, \(\epsilon\) is maximized when \(\gamma\) is maximized.

The number of rounds (due to Theorem 5.3.2 and Lemma 5.3.8) is

\[
R = \Omega\left(\frac{N}{K^{(2\delta-0.25)/(\delta-1)}q^{2t+2} + (t+2)/(\delta-1)\text{poly log } q}\right)
\]
\[
\begin{align*}
\Omega \left( \frac{N}{n^{2(2\delta-0.25)/(\delta-1) + (8\delta-4)/(\delta-1) + o(1)}} \right) \\
\Omega \left( \frac{N}{n^{2(8\delta-4)/(\delta-1) + o(1)}} \right) \\
\Omega \left( N^{1-\frac{\gamma(8\delta+4+6.5/(\delta-1) + o(1))}{1+o(1)+\gamma(8\delta-6)}} \right)
\end{align*}
\]

For very small \( \kappa > 0 \), to get a gap instance for \( N^{\Omega(\kappa)} \)-round Lasserre, we need

\[
1 - \frac{\gamma(8\delta + 4 + 6.5/(\delta - 1)) + o(1)}{1 + o(1) + \gamma(8\delta - 6)} \geq \Omega(\kappa)
\]

\[
\Rightarrow 1 + o(1) + \gamma(8\delta - 6) - (\gamma(8\delta + 4 + 6.5/(\delta - 1)) + o(1)) \geq \Omega(\kappa)
\]

\[
\Rightarrow 1 - \gamma(10 - 6.5(\delta - 1)) \geq \Omega(\kappa)
\]

\[
\Rightarrow \gamma \leq \frac{1}{\Omega(\kappa)}
\]

Let \( \gamma = \frac{1 - O(\kappa)}{10 + 6.5/(\delta - 1)} \), we have

\[
\epsilon = \frac{1}{8\delta - 6} - \frac{1 + o(1)}{(1 + (8\delta - 6)\gamma + o(1))(8\delta - 6)}
\]

\[
= \frac{1}{8\delta - 6} - \frac{1 + o(1)}{(1 + (8\delta - 6)\frac{1 - O(\kappa)}{10 + 6.5/(\delta - 1)} + o(1))(8\delta - 6)}
\]

\[
= \frac{1}{8\delta - 6} - \frac{1}{(1 + \frac{(8\delta - 6)}{10 + 6.5/(\delta - 1)})(8\delta - 6)} - O(\kappa).
\]

When \( 2\delta = 4 \), we get the maximized value \( \epsilon = 2/53 - O(\kappa) \).

5.3.4 Expansion for random \( k \)-CSP instances

In this section, we prove Lemma 5.3.4 restated as follows.

**Lemma 5.3.4 (restitated).** Given \( \beta, \eta, K \) as in Theorem 5.3.2 with probability \( 1 - o(1) \), for all \( 2 \leq s \leq \eta n \), every set of \( s \) constraints involves more than \( (K - \delta) s \) variables.

**Proof.** Fix \( 2 \leq s \leq \eta n \), let us upperbound the probability that there is a set of \( s \) constraints containing at most \( (K - \delta)s \) variables. Since there are \( \binom{\beta n}{s} \) such sets, the probability is at
most
\[ \binom{\beta n}{s} \Pr[\text{the first } s \text{ constraints contain at most } (K - \delta)s \text{ variables}] \]
\[ = \binom{\beta n}{s} \sum_{i=1}^{(K-\delta)s} \Pr[\text{the first } s \text{ constraints contain exactly } i \text{ variables}] \]

Fix a set \( T \) of \( i \) variables, let \( p(s, i) \) be the number of \( s \)-tuples \((T_1, T_2, \ldots, T_s)\) where for each \( 1 \leq j \leq s \), \( T_j \) is a set of \( K \) variables, such that \( \cup_{1 \leq j \leq s} T_j = T \). We have
\[ \Pr[\text{the first } s \text{ constraints contain exactly } i \text{ variables}] = \binom{n}{i} \cdot p(i, s) / \binom{n}{K}^s. \]

To upperbound \( p(i, s) \), we view the way to enumerating valid \((T_1, T_2, \ldots, T_s)\) as, to choose a multiset of \( Ks \) variables (each one from \( T \)) so that each element in \( T \) appears at least once in the multiset, then view each element in the multiset as a distinct element, and distribute these \( Ks \) elements to \( s \) sets, in a balanced way. Note that in this way, we are able to enumerate all the valid \( s \)-tuples (although some of them might be enumerated more than once). Since there are at most \( \binom{Ks-1}{s-1} < \binom{Ks}{s} \) valid multisets, we have
\[ p(i, s) \leq \binom{Ks}{i} / (K!)^s. \]

Therefore, we have
\[ \binom{\beta n}{s} \Pr[\text{the first } s \text{ constraints contain at most } (K - \delta)s \text{ variables}] \]
\[ = \binom{\beta n}{s} (Ks)! \cdot (K!)^{-s} \binom{n}{K}^{-s} \sum_{i=1}^{(K-\delta)s} \binom{n}{i} \binom{Ks}{i}. \]

Note that when \( K^2s < \delta n \) and \( i \leq (K - \delta)s \), we have \( i < nKs/(n + Ks) \) (since \( i \leq Ks(1 - \delta/K) \leq Ks/(1 + \delta/K) = nKs/(n + \delta n/K) < nKs/(n + Ks) \)), and therefore
\[ \frac{\binom{n}{i} \binom{Ks}{i}}{\binom{n}{i-1} \binom{Ks}{i-1}} = \frac{(n-i)(Ks-i)}{i^2} > 1 \quad (\iff (n-i)(Ks-i) > i^2 \iff nKs > (n + Ks)i), \]

therefore the function \( \binom{n}{i} \binom{Ks}{i} \) is increasing when \( i \leq (K - \delta)s \), therefore
\[ \binom{\beta n}{s} (Ks)! \cdot (K!)^{-s} \binom{n}{K}^{-s} \sum_{i=1}^{(K-\delta)s} \binom{n}{i} \binom{Ks}{i}. \]

82
\[
\leq \left( \frac{\beta n}{s} \right) (Ks)! \cdot (K!)^{-s} \left( \frac{n}{K} \right)^{-s} \cdot Ks \cdot \left( \frac{n}{(K-\delta)s} \right) \left( \frac{Ks}{(K-\delta)s} \right)
\]

\[
= \left( \frac{\beta n}{s} \right) (Ks)! \cdot (K!)^{-s} \left( \frac{n}{K} \right)^{-s} \cdot Ks \cdot \left( \frac{n}{(K-\delta)s} \right) \left( \frac{Ks}{(K-\delta)s} \right)
\]

for \( K \leq n^{1/2} \), we use the fact that \( \left( \frac{n}{K} \right) \geq (n - K)/K! \geq n^K/3/((K/e)^K \cdot (5\sqrt{K})) = (en/K)^K/(15\sqrt{K}) \) (since by Stirling’s formula, we have \( K! \leq 5\sqrt{K}(K/e)^K \)), and again use the fact that \( \sqrt{2\pi K}(K/e)^K \leq K! \leq 5\sqrt{K}(K/e)^K \), we bound the expression above by

\[
\left( \frac{e\beta n}{s} \right)^s \frac{5\sqrt{K}s(Ks/e)Ks}{(2\pi/\sqrt{K}(K/e)K)^s} \cdot \left( 15\sqrt{K} \left( \frac{K}{en} \right) \right)^s \cdot Ks \cdot \left( \frac{en}{(K-\delta)s} \right) \left( \frac{eK}{\delta} \right)^{(K-\delta)s}
\]

\[
\leq 5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta}K_{s-1}K_{s+\delta}^{-1}}{(2\pi n^{\delta-1}(K-\delta)^K-s\delta)} \right)^x
\]

\[
\leq 5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta}K_{s-1}K_{s+\delta}^{-1}}{(2\pi n^{\delta-1}\delta)} \right)^x
\]

For \( 2 \leq s \leq \ln^2 n \), since \( n^{\alpha-1} \leq 1/(10^8 \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)}) \), we have \( \beta^2 K^{4\delta+1.5} / n^{2(\delta-1)} \leq n^{-(2\delta-1)\alpha} \), we have

\[
5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta}K_{s-1}K_{s+\delta}^{-1}}{(2\pi n^{\delta-1}\delta)} \right)^x \leq 5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta}K_{s-1}K_{s+\delta}^{-1}}{(2\pi n^{\delta-1}\delta)} \right)^2
\]

\[
\leq \frac{5 \cdot 15e^{1+\delta}K_{s-1}K_{s+\delta}^{-1}}{(2\pi n^{\delta-1}\delta)} \leq O(n^{-(\delta-1)\alpha}).
\]

For \( \ln^2 n \leq s \leq \eta n \), since \( \eta \leq 1/(10^8 \cdot (\beta K^{2\delta+0.75})^{1/(\delta-1)}) \), we get \( \eta \leq 1/(10^8 \cdot (\beta K^{2\delta})^{1/(\delta-1)}) \), and further we have \( \beta K^{2\delta} \eta^{\delta-1} \leq \delta^4/(100 \cdot 15e^{1+\delta}/\sqrt{2\pi}) \) for all \( \delta > 5/4 \). Therefore,

\[
5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta}K_{s-1}K_{s+\delta}^{-1}}{(2\pi n^{\delta-1}\delta)} \right)^x \leq 5(Ks)^{1.5} \cdot \left( \frac{s^{\delta-1}}{100(\eta)^{\delta-1}} \right)^x \leq 5 \cdot \left( \frac{s^{\delta-1}(Ks)^{1.5/\ln^2 n}}{100(\eta)^{\delta-1}} \right)^x
\]

\[
\leq 5 \cdot \left( \frac{2}{100} \right)^x \quad \text{(by } s \leq \eta n \text{ and } Ks \leq n^2 \text{)}
\]

Now, we upperbound probability that there exists a set of constraints of size \( s \leq \eta n \)
involving at most \((K - \delta)s\) variables by

\[
\sum_{s=2}^{\eta n} 5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta} \beta s^{\delta-1} K^{2\delta+0.5}}{n^{\delta-1} \delta} \right)^s
= \sum_{s=2}^{\ln^2 n} 5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta} \beta s^{\delta-1} K^{2\delta+0.5}}{n^{\delta-1} \delta} \right)^s + \sum_{s=\ln^2 n + 1}^{\eta n} 5(Ks)^{1.5} \cdot \left( \frac{15e^{1+\delta} \beta s^{\delta-1} K^{2\delta+0.5}}{n^{\delta-1} \delta} \right)^s
\leq \sum_{s=2}^{\ln^2 n} O(n^{-\kappa/2}) + \sum_{s=\ln^2 n + 1}^{\eta n} 5 \cdot (1/50)^s = o(1).
\]
Chapter 6

Lasserre integrality gaps for \textsc{BalancedSeparator} and \textsc{UniformSparsestCut}

6.1 Introduction

Recall the problems \textsc{BalancedSeparator} and \textsc{UniformSparsestCut} are defined previously by \underline{Definition 2.1.1} and \underline{Definition 2.1.2} For readers’ convenience, we restate the definition as follows.

\textbf{Definition 6.1.1.} Given an undirected graph $G = (V, E)$ and $0 < \tau < 0.5$, the goal of the $\tau$ vs $1 - \tau$ \textsc{BalancedSeparator} problem is to find a set $A \subseteq V$ such that $\tau |V| \leq |A| \leq (1 - \tau)|V|$, while $\text{edges}(A, V \setminus A)$ is minimized. Here $\text{edges}(A, B)$ is the number of edges in $E$ that cross the cut $(A, B)$.

The goal of the \textsc{UniformSparsestCut} problem is to find a set $\emptyset \subseteq A \subseteq V$ such that the sparsity

$$\frac{\text{edges}(A, V \setminus A)}{|A||V \setminus A|}$$

is minimized.

Despite extensive research, there are still huge gaps between the known approximation algorithms and inapproximability results for these problems. The best algorithms,
based on semidefinite relaxations (SDPs) with triangle inequalities, give a $O(\sqrt{\log n})$ approximation to both problems \cite{21}. On the inapproximability side, a Polynomial Time Approximation Scheme (PTAS) is ruled out for both problems assuming 3-SAT does not have randomized subexponential-time algorithms \cite{11}. In this chapter, our focus is on the UNIFORMSPARSESTCUT problem; the general SPARSESTCUT problem has been shown to not admit a constant-factor approximation algorithm under the Unique Games Conjecture \cite{64, 144, 136}.

It is known that the SDP used in \cite{21} cannot give a constant factor approximation for UNIFORMSPARSESTCUT \cite{79}. Integrality gaps are also known for stronger SDPs: super-constant factor integrality gaps for both BALANCEDSEPARATOR and UNIFORMSPARSESTCUT are known for the Sherali-Adams+SDP hierarchy for a super-constant number of rounds \cite{192}.

However, if we turn to the stronger Lasserre SDP hierarchy, for both BALANCEDSEPARATOR and UNIFORMSPARSESTCUT, integrality gaps were not known even for a small constant number of rounds (before this thesis). It was not (unconditionally) ruled out, for example, that $1/\epsilon^{O(1)}$ rounds of the hierarchy could give a $(1 + \epsilon)$-approximation algorithm, thereby giving a PTAS. On the algorithmic side, \cite{110} recently showed that for these problems, SDPs using $O(r/\epsilon^2)$ rounds of the Lasserre hierarchy have an integrality gap at most $(1 + \epsilon)/\min\{1, \lambda_r\}$. Here $\lambda_r$ is the $r$-th smallest eigenvalue of the normalized Laplacian of the graph. This result implies an approximation scheme for these problems with runtime parameterized by the graph spectrum.

### 6.1.1 Our contributions

In this chapter, we study integrality gaps for the Lasserre SDP relaxations for BALANCEDSEPARATOR and UNIFORMSPARSESTCUT. As mentioned before, APX-hardness is not known for these two problems, even assuming the Unique Games Conjecture. (Superconstant hardness results are known based on a strong intractability assumption concerning the Small Set Expansion problem \cite{194}.) In contrast, we show that linear-round Lasserre SDP has an integrality gap bounded away from 1, and thus fails to give a factor $\alpha$-approximation for some absolute constant $\alpha > 1$. Specifically, we prove the following two theorems.

**Theorem 6.1.2 (Informal version of Theorem 6.3.1).** For $0.45 < \tau < 0.5$, there are linear-round Lasserre gap instances for the $\tau$ vs $(1 - \tau)$ BALANCEDSEPARATOR problem, such that the integral optimal solution is at least $(1 + \epsilon(\tau))$ times the SDP solution, where $\epsilon(\tau) > 0$ is a constant dependent on $\tau$. 

86
Theorem 6.1.3 (Informal version of Theorem 6.4.1). There are linear-round Lasserre gap instances for the UNIFORMSPARSESTCUT problem, such that the integral optimal solution is at least $(1 + \epsilon)$ times the SDP solution, for some constant $\epsilon > 0$.

6.1.2 Our techniques

All of our gap results are based on Schoenebeck’s ingenious Lasserre integrality gap for 3-XOR \cite{204}. For BALANCEDSEPARATOR and UNIFORMSPARSESTCUT, we use the ideas in \cite{11} to build gadget reductions and combine them with Schoenebeck’s gap instance. \cite{11} designed gadget reductions from Khot’s quasi-random PCP \cite{137} in order to show APX-Hardness of the two problems. If we view the Lasserre hierarchy as a computational model (as suggested in \cite{220}), we can view Schoenebeck’s construction as playing the role of a quasi-random PCP in the Lasserre model. Our gadget reductions, therefore, bear some resemblance to the ones in \cite{11}, though the analysis is different due to different random structures of the PCPs. We feel our reductions are slightly simpler than the ones in \cite{11}, although we need some additional tricks to make the reductions have only linear blowup. This latter feature is needed in order to get Lasserre SDP gaps for a linear number of rounds. We are also able to make the gap instance graphs have only constant degree, while the reductions in \cite{11} give graphs with unbounded degree.

Also, unlike 3-XOR, for balanced separator there is a global linear constraint (stipulating the balance of the cut), and our Lasserre solution must also satisfy a lifted form of this constraint \cite{158}. We make a general observation that such constraints can be easily lifted to the Lasserre hierarchy when the vectors in our construction satisfy a related linear constraint. This observation applies to constraints given by any polynomials, and to our knowledge, was not made before. It simplifies the task of constructing legal Lasserre vectors in such cases.

6.2 Preliminaries on Lasserre SDPs for BALANCEDSEPARATOR and UNIFORMSPARSESTCUT

In this section, we begin with a general description of semidefinite programming relaxations from the Lasserre hierarchy, followed by a useful observation about constructing feasible solutions for such a SDP. We then discuss the specific SDP relaxations for our problems of interest. Finally, we recall Schoenebeck’s Lasserre integrality gaps \cite{204} in a form convenient for our later use.
6.2.1 Useful theorems about the Lasserre hierarchy

Consider a binary programming problem with polynomial objective function \( P \) and a single constraint expressed as a polynomial \( Q \):

\[
\begin{align*}
\text{Minimize/Maximize} & \quad \sum_{T \in \binom{[n]}{\leq d}} P(T) \prod_{j \in T} x_j \\
\text{subject to} & \quad \sum_{T \in \binom{[n]}{\leq d}} Q(T) \prod_{j \in T} x_j \geq 0, \\
& \quad x_i \in \{0, 1\} \quad \text{for all } i \in [n].
\end{align*}
\]  

(6.1)

It is easy to see that this captures all problems we consider in this chapter: BALANCED-SEP\(\text{A}\)\(\text{R}\)ATOR (Section 6.2.2.1) and UNIFORM-\(\text{S}\)PARSEST\(\text{C}\)UT (Section 6.2.2.2).

We now define the Lasserre hierarchy semidefinite program relaxation for the above integer program. It is easily seen that the below is a relaxation by taking \( U_A = x_A I \) and \( Y_A = \sqrt{Q(x)} U_A \) where \( x \in \{0, 1\}^n \) is a feasible solution to (6.1), \( x_A = \prod_{i \in A} x_i \), and \( I \) is any fixed unit vector.

**Proposition 6.2.1.** For any positive integer \( r \geq d \), \( r \) rounds of Lasserre Hierarchy relaxation \([158]\) of (6.1) is given by the following semidefinite programming formulation:

\[
\begin{align*}
\text{Minimize/Maximize} & \quad \sum_T P(T) \|U_T\|^2 \\
\text{subject to} & \quad \|U_\emptyset\|^2 = 1, \\
& \quad \langle U_A, U_B \rangle = \|U_{A\cup B}\|^2 \quad \text{for all } A, B \text{ with } |A \cup B| \leq 2r, \\
& \quad \sum_{S \in \binom{[n]}{\leq d}} Q(S) \langle U_S, U_{A\cup B} \rangle = \langle Y_A, Y_B \rangle, \\
& \quad \langle Y_A, Y_B \rangle = \|Y_{A\cup B}\|^2 \quad \text{for all } A, B \text{ with } |A \cup B| \leq 2(r - d). \\
U_A, Y_B \in \mathbb{R}^r.
\end{align*}
\]

(6.2)

**Proof.** Given \( y \in \mathbb{R}^{\binom{[n]}{\leq 2r}} \), let \( \mathbb{M}(y) \in \text{Sym}(\binom{[n]}{\leq r}) \) be the moment matrix whose rows and columns correspond to subsets of size \( \leq r \). The entry at row \( S \) and column \( T \) of \( \mathbb{M}(y) \) is given by \( y_{S \cup T} \). For any multilinear polynomial \( P \) of degree-\( d \), let \( P \ast y \in \mathbb{R}^{\binom{[n]}{\leq 2r-d}} \) be the vector whose entry corresponding to subset \( S \) is given by \( \sum_T P_T y_{S \cup T} \). The Lasserre Hierarchy relaxation \([158]\) of (6.1) is given by:

\[
\begin{align*}
\text{Minimize/Maximize} & \quad \sum_T P(T) y_T \\
\text{subject to} & \quad y_\emptyset = 1, \\
& \quad \mathbb{M}(y) \succeq 0, \\
& \quad \mathbb{M}(Q \ast y) \succeq 0.
\end{align*}
\]

(6.3)
Proof of \((6.2) \implies (6.3)\) Given feasible solution for \((6.2)\), let \(y_S \triangleq \|U_S\|^2\) and \(z_S \triangleq \|Y_S\|^2\). We have \(y_0 = 1\) and \(\sum_T P(T)y_T = \sum_T P(T)\|U_S\|^2\). Observe that \(y_{S \cup T} = \|U_{S \cup T}\|^2 = \langle U_S, U_T \rangle\) therefore \(M(y) \succeq 0\). With a similar reasoning, we also have \(M(z) \succeq 0\). Finally, for any \(S\):

\[
(Q \ast y)_S = \sum_T Q(T)y_{S \cup T} = \sum_T Q(T)\langle U_T, U_S \rangle = \|Y_S\|^2 = z_S,
\]

which implies \(z = Q \ast y\). Hence \(y\) is a feasible solution for \((6.3)\).

Proof of \((6.3) \implies (6.2)\) Let \(y\) be a feasible solution for \((6.3)\). Define \(z \triangleq Q \ast y\). Since \(M(y) \succeq 0\) (resp. \(M(z) \succeq 0\)), there exists a matrix \(U = [U_S]_S\) (resp. \(Y = [Y_S]_S\)) such that \(M(y) = U^T U\) (resp. \(M(z) = Y^T Y\)). It is easy to see that \(\langle U_S, U_T \rangle = y_{S \cup T}\) and \(\langle Y_S, Y_T \rangle = z_{S \cup T}\). Therefore:

- \(\sum_T P(T)\|U_T\|^2 = \sum_T P(T)y_T\).
- \(\|U_\emptyset\|^2 = y_0 = 1\).
- \(\langle U_S, U_T \rangle = y_{S \cup T} = \|U_{S \cup T}\|^2\) (similar for \(Y\)).
- For any \(S\), \(\sum_T Q(T)\langle U_T, U_S \rangle = \sum_T Q(T)y_{S \cup T} = (Q \ast y)_S = z_S = \|Y_S\|^2\).

Therefore \((U, Y)\) is a feasible solution for \((6.2)\) with same objective value, completing our proof.

Note that a straightforward verification of last two constraints requires the construction of vectors \(Y_A\) in addition to \(U_A\). Below we give an easier way to verify these last two constraints without having to construct \(Y_A\)‘s. This greatly simplifies our task of constructing Lasserre vectors for the lifting of global balance constraints.

Theorem 6.2.2. Given vectors \(U_T\) for all \(T \in \binom{[n]}{\leq 2r}\) satisfying the first two constraints of \((6.2)\) if there exists a non-negative real \(\delta > 0\) such that

\[
\sum_{S \in \binom{[n]}{\leq 4}} Q(S)U_S = \delta \cdot U_\emptyset 
\]

then these vectors form (part of) a feasible solution to \((6.2)\).
Proof. Consider the following vectors. For each $A$ with $|A| \leq r$, let $Y_A = \sqrt{\delta} \cdot U_A$. By construction, these vectors satisfy the $\langle Y_A, Y_B \rangle = \|Y_{A \cup B}\|^2$ constraints since $\langle U_A, U_B \rangle = \|U_{A \cup B}\|^2$. Now we verify the other constraint:

$$\sum_{S \in \binom{[n]}{\leq d}} Q(S) \langle U_S, U_{A \cup B} \rangle = \left\langle \sum_{S \in \binom{[n]}{\leq d}} Q(S) U_S, U_{A \cup B} \right\rangle = \langle \delta U_{\emptyset}, U_{A \cup B} \rangle = \delta \langle U_A, U_B \rangle = \langle Y_A, Y_B \rangle.$$

\[ \square \]

6.2.2 Lasserre SDP for graph partitioning problems

In light of Theorem 6.2.2, to show good solutions for the Lasserre SDP for our problems of interest, we only need to show good solutions for the following SDPs.

6.2.2.1 BALANCEDSEPADOR

The standard integer programming formulation of BALANCEDSEPADOR is shown in the left part of Figure 6.1. The $r$ round SDP relaxation $\Psi_1$ (shown in the right part of Figure 6.1) has a vector $U_S$ for each subset $S \subseteq V$ with $|S| \leq r$. In an integral solution, the intended value of $U_{\{u\}}$ is $x_u U_{\emptyset}$ for some fixed unit vector $U_{\emptyset}$, and that of $U_S$ is $\left(\prod_{u \in S} x_u\right) U_{\emptyset}$.

6.2.2.2 UNIFORMSPARSESTCUT

The UNIFORMSPARSESTCUT problem asks to minimize the value of the quadratic integer program shown in the left part of Figure 6.2 over all $\tau \in \{1/n, 2/n, \ldots, [n/2]\}$. The corresponding SDP relaxation $\Psi_2$ is to minimize the value of the SDP shown in the right part of Figure 6.2 over all $\tau \in \{1/n, 2/n, \ldots, [n/2]\}$.

Remark Prior to this thesis, known lower bounds [79, 130] on the integrality gap of UNIFORMSPARSESTCUT problem used a weaker relaxation, where the last two equality constraints in $\Psi_2$ of Figure 6.2 are replaced by the following instead:

$$\sum_{u < v} \|U_{\{u\}} - U_{\{v\}}\|^2 = 1$$

90
### IP

<table>
<thead>
<tr>
<th>Minimize $\sum_{(u,v) \in E} (x_u - x_v)^2$</th>
<th>SDP Relaxation $\Psi_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>s.t. $\tau</td>
<td>V</td>
</tr>
<tr>
<td>$x_u \in {0, 1} \ \forall u \in V$</td>
<td>s.t. $\langle U_{S_1}, U_{S_2} \rangle \geq 0$ for all $S_1, S_2$</td>
</tr>
<tr>
<td></td>
<td>$\langle U_{S_1}, U_{S_2} \rangle = \langle U_{S_3}, U_{S_4} \rangle$ for all $S_1 \cup S_2 = S_3 \cup S_4$</td>
</tr>
<tr>
<td></td>
<td>$|U_\emptyset|^2 = 1$</td>
</tr>
<tr>
<td></td>
<td>$\sum_{v} U_{{v}} = \tau'</td>
</tr>
</tbody>
</table>

**Figure 6.1**: IP and SDP relaxations for BALANCEDSEPARATOR.

We can solve $\Psi_1$ by first enumerating over all $\tau' \in \{1/n, 2/n, \ldots, 1\} \cap [\tau, 1 - \tau]$ and then choosing $\tau$ which minimizes the objective function. Note that the resulting relaxation is **stronger** than usual Lasserre Hierarchy relaxation.

with the objective function being simply $\sum_{(u,v) \in E} \|U_{\{u\}} - U_{\{v\}}\|^2$.

### 6.2.3 Lasserre Gaps for 3-XOR from Schoenebeck

We start by defining the 3-XOR problem.

**Definition 6.2.3.** An instance $\Phi$ of 3-XOR is a set of constraints $C_1, C_2, \cdots, C_m$ where each constraint $C_i$ is over 3 distinct variables $x_{i1}, x_{i2},$ and $x_{i3},$ and is of the form $x_{i1} \oplus x_{i2} \oplus x_{i3} = b_i$ for some $b_i \in \{0, 1\}$.

A random instance of 3-XOR is sampled by choosing each constraint $C_i$ uniform independently from the set of possible constraints.

We will make use of the following fundamental result of Schoenebeck.

**Theorem 6.2.4 ([204]).** For every large enough constant $\beta > 1$, there exists $\eta > 0$, such that with probability $1 - o(1)$, a random 3-XOR instance $\Phi$ over $m = \beta n$ constraints and $n$ variables cannot be refuted by the SDP relaxation obtained by $\eta m$ rounds of the Lasserre hierarchy, i.e. there are vectors $W_{(S,\alpha)}$ for all $|S| \leq \eta n$ and all $\alpha : S \to \{0, 1\}$, such that
Stronger than usual Lasserre Hierarchy relaxation.

\begin{align*}
\text{Minimize} & \ \frac{1}{|V|^2\tau(1-\tau)} \sum_{(u,v) \in E} (x_u - x_v)^2 \\
\text{s.t.} & \ \sum_u x_u = \tau|V| \\
& \ x_u \in \{0, 1\} \ \forall u \in V
\end{align*}

\begin{align*}
\text{Minimize} & \ \frac{1}{|V|^2\tau(1-\tau)} \|\vec{U}_{\{u\}} - \vec{U}_{\{v\}}\|^2 \\
\text{s.t.} & \ \langle \vec{U}_{S_1}, \vec{U}_{S_2} \rangle \geq 0 \text{ for all } S_1, S_2 \\
& \ \langle \vec{U}_{S_1}, \vec{U}_{S_2} \rangle = \langle \vec{U}_{S_3}, \vec{U}_{S_4} \rangle \text{ for all } S_1 \cup S_2 = S_3 \cup S_4 \\
& \ \|\vec{U}_\emptyset\|^2 = 1 \\
& \ \sum_v \vec{U}_{\{v\}} = \tau|V|\vec{U}_\emptyset
\end{align*}

Figure 6.2: IP and SDP relaxations for UNIFORMSPARSESTCUT.

We can solve $\Psi_2$ by first enumerating over all $\tau \in \{1/n, 2/n, \ldots, (n-1)/n\}$ and then choosing $\tau$ which minimizes the objective function. Note that the resulting relaxation is stronger than usual Lasserre Hierarchy relaxation.

(i) the value of the solution is perfect: $\sum_{i=1}^m \sum_{\alpha: \alpha(x_{i_1}) \oplus \alpha(x_{i_2}) \oplus \alpha(x_{i_3}) = b_i} \|W_{(x_{i_1}, x_{i_2}, x_{i_3}, \alpha)}\|^2 = m$;

(ii) $\langle W_{(S_1, \alpha_1)}, W_{(S_2, \alpha_2)} \rangle \geq 0$ for all $S_1, S_2, \alpha_1, \alpha_2$;

(iii) $\langle W_{(S_1, \alpha_1)}, W_{(S_2, \alpha_2)} \rangle = 0$ if $\alpha_1(S_1 \cap S_2) \neq \alpha_2(S_1 \cap S_2)$;

(iv) $\langle W_{(S_1, \alpha_1)}, W_{(S_2, \alpha_2)} \rangle = \langle W_{(S_3, \alpha_3)}, W_{(S_4, \alpha_4)} \rangle$ for all $S_1 \cup S_2 = S_3 \cup S_4$ and $\alpha_1 \circ \alpha_2 = \alpha_3 \circ \alpha_4$. Here, when $\alpha_1(S_1 \cap S_2) = \alpha_2(S_1 \cap S_2)$, $\alpha_1 \circ \alpha_2$ is naturally defined as the mapping from $S_1 \cap S_2$ to $\{0, 1\}$ such that its restriction to $S_1$ equals $\alpha_1$ and its restriction to $S_2$ equals $\alpha_2$. We make similar definition for $\alpha_3 \circ \alpha_4$.

(v) $\sum_{\alpha: S \rightarrow \{0, 1\}} \|W_{(S, \alpha)}\|^2 = 1$ for all $S$.

Note that indeed we have for every $S$, $\sum_{\alpha: S \rightarrow \{0, 1\}} W_{(S, \alpha)} = W_{(\emptyset, \emptyset)}$. This is because $\|W_{(\emptyset, \emptyset)}\|^2 = 1$ and

$$\left\langle \left( \sum_{\alpha: S \rightarrow \{0, 1\}} W_{(S, \alpha)} \right), W_{(\emptyset, \emptyset)} \right\rangle = \sum_{\alpha: S \rightarrow \{0, 1\}} \left\langle W_{(S, \alpha)}, W_{(\emptyset, \emptyset)} \right\rangle = \sum_{\alpha: S \rightarrow \{0, 1\}} \|W_{(S, \alpha)}\|^2 = 1.$$

**Observation 6.2.5.** In the construction of Theorem 6.2.4, the vectors $W$ satisfy the following property. For any constraint $C_i$ over set of variables $S_i$, the vectors corresponding
to all satisfying partial assignments of $S_i$ sums up to $W_\emptyset$:

$$\sum_{\alpha: S_i \to \{0, 1\} \land C_i(\alpha) = 1} W_{(S_i, \alpha)} = W_\emptyset.$$

### 6.3 Gaps for BALANCEDSEPARATOR

In this section, we prove Theorem 6.1.2. We state the theorem in detail as follows.

**Theorem 6.3.1.** For large enough constants $\beta, M$, for all $0.45 < \tau < 0.5$, and for infinitely many positive integer N’s, there is an $N$-vertex instance $\mathcal{H}_\Phi$ for the $\tau$ vs. $(1 - \tau)$ BALANCEDSEPARATOR problem, such that the optimal solution is at least $4(3\tau - \tau^3)/5 - O(1/\sqrt{\beta} + 1/M)$ times the best solution of the $\Omega(N)$-round Lasserre SDP relaxation. Moreover, the solution for Lasserre SDP relaxation is a fractional $(0.5 - O(1/M))$ vs. $(0.5 + O(1/M))$ balanced separator.

The rest of this section is dedicated to the proof of Theorem 6.3.1. In Section 6.3.1, we will describe how to get a BALANCEDSEPARATOR instance from a 3-XOR instance. Then, we will show that when the 3-XOR instance is random, the corresponding BALANCEDSEPARATOR instance is a desired gap instance. This is done by showing there is an SDP solution with good objective value (completeness part, Lemma 6.3.2 in Section 6.3.2) while the instance in fact has not great integral solution (soundness part, Lemma 6.3.4 in Section 6.3.3). The completeness part relies Theorem 6.2.4 – we use the 3-XOR vectors (which exist for random instances by the theorem) to construct BALANCEDSEPARATOR vectors. In the soundness part, we first prove two pseudorandom structural properties exhibited in the random 3-XOR instances (Lemma 6.3.3), and then prove that any 3-XOR with these two properties leads to a BALANCEDSEPARATOR instance with bad integral optimum by our construction. Finally, in Section 6.3.4, we slightly twist our gap instance in order to make its vertex degree bounded.

#### 6.3.1 Reduction

Given a 3-XOR instance $\Phi$ with $m = \beta n$ constraints and $n$ variables, we build a graph $\mathcal{H}_\Phi = (V_\Phi, E_\Phi)$ for BALANCEDSEPARATOR as follows.

$\mathcal{H}_\Phi$ consists of an almost bipartite graph $H_\Phi = (L_\Phi, R_\Phi, E_\Phi)$ (obtained by replacing each right vertex of a bipartite graph by a clique), an expander $Z_r$, and edges between $L_\Phi$ and $Z_r$. 

93
The left side $L_\Phi$ of $H_\Phi$ contains $4m = 4\beta n$ vertices, each corresponds to a pair of a constraint and a satisfying partial assignment for the constraint, i.e.

$$L_\Phi = \{(C_i, \alpha) | \alpha : \{x_{i_1}, x_{i_2}, x_{i_3}\} \to \{0, 1\}, C_i(\alpha) = 1\}.$$

The right side $R_\Phi$ of $H_\Phi$ contains $2n$ cliques, each is of size $M\beta$, and corresponds to one of the $2n$ literals, i.e.

$$R_\Phi = \bigcup_{j, \alpha : \{x_j\} \to \{0, 1\}} C(x_j, \alpha),$$

where

$$C(x_j, \alpha) = \{(x_j, \alpha, t) | 1 \leq t \leq M\beta\}.$$

Call $(x_j, \alpha, 1)$ the representative vertex of $C(x_j, \alpha)$. Besides the clique edges, we connect a left vertex $(C_i, \alpha)$ and a right representative vertex $(x_j, \alpha', 1)$ if $x_j$ is accessed by $C_i$ and $\alpha'$ is consistent with $\alpha$, i.e.

$$E_\Phi = \{\text{clique edges}\} \cup \{(C_i, \alpha), (x_j, \alpha', 1) \mid x_j \in \{x_{i_1}, x_{i_2}, x_{i_3}\}, \alpha(x_j) = \alpha'(x_j)\}.$$

Now we have finished the definition of $H_\Phi$. To get $H_\Phi$, we add an $O(M)$-regular expander $Z_r$ of size $m = \beta n$ and edge expansion $M$. (I.e. the degree of each vertex in $Z_r$ is $O(M)$, and each subset $T \subseteq Z_r (|T| \leq |Z_r|/2)$ has at least $|T| \cdot M$ edges connecting to $Z_r \setminus T$. For more discuss on the definitions and applications of expander graphs, please refer to, e.g., [121].) We connect each vertex in $L_\Phi$ to two different vertices in $Z_r$, so that each vertex in $Z_r$ has the same number of neighbors in $L_\Phi$ (this number should be $4\beta n \cdot 2/(\beta n) = 8$). In other words, if we view each vertex in $L_\Phi$ as an undirected edge between its two neighbors in $Z_r$, the graph should be a regular graph.

The whole construction is shown in Figure 6.3. Our construction is very similar to the one in [11], but there are some technical differences. Instead of having cliques in $R_\Phi$, [11] has clusters of vertices with no edges connecting them. Also, in our construction, the vertices in $L_\Phi$ are connected to the representative vertices in $R_\Phi$ only, while in [11], all the vertices in the right clusters could be connected to the left side. The most important difference is that in our way, the cliques are of constant size, while the clusters in [11] has superconstantly many vertices. This means that our reduction blows up the instance size only by a constant factor, therefore we are able to get linear round Lasserre gap.

Observe that there are $|L_\Phi| + |R_\Phi| + |Z_r| = 4m + 2Mm + m = (2M + 5)m$ vertices in $H_\Phi$.

In the following two subsections, we will prove the completeness lemma (Lemma 6.3.2, which states that there is an SDP solution with a good objective value) and the soundness lemma (Lemma 6.3.4, which states that every integral solution has a bad objective value). Combining the two lemmas, we prove our main integrality gap theorem for BALANCED-SEPARATOR as follows.
Note that the incident edges are drawn for only one of the vertices in $L_\Phi$, while others can be drawn similarly.

**Lemma 6.3.2** Let $\Phi$ be a random 3-XOR instance over $m = \beta n$ constraints and $n$ variables. By Theorem 6.2.4 we know that, with probability $1 - o(1)$, $\Phi$ admits a perfect solution for $\Omega(n)$-round Lasserre SDP relaxation. Therefore, by Lemma 6.3.2 with probability $1 - o(1)$, $\Omega(n)$-round SDP relaxation $\Psi_1$ with parameter $\tau = 0.5 - O(1/M)$ for the BALANCEDSEPARATOR instance $\mathcal{H}_\Phi$ has a solution of value $5m$. On the other hand, by Lemma 6.3.4 with probability $1 - o(1)$, for $\tau > 1/3$, every $\tau \text{ vs. } (1-\tau)$ balanced separator has at least $4m(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))$ edges in the cut.

Therefore, with probability $1 - o(1)$, when $\tau > 1/3$, the ratio between the optimal integral solution (to $\mathcal{H}_\Phi$) and the optimal $\Omega(n)$-round $\Psi_1$ solution is at least $4(3\tau - \tau^3)/5$
This ratio is greater than 1.007 when $\tau > 0.45$ and $\beta$ and $M$ are large enough. By our observation in Section 6.2.2.1 this gap also holds for the Lasserre SDP relaxation.

Let $\Delta$ be the maximum number of occurrences of any variable in $\Phi$. By our construction, the graph has degree $\Theta(M + \Delta)$. When $\beta = O(1)$, we have $\Delta = \Theta(\log n / \log \log n)$ with probability $1 - o(1)$ (see, e.g. [96]). This means that our graph does not have the desired constant-degree property. However, since there are few edges incident to vertices with superconstant degree, we can simply remove all these edges to get a constant-degree graph, while the completeness and soundness are still preserved. We will discuss this in more details in Section 6.3.4.

6.3.2 Completeness: good SDP solution

Lemma 6.3.2 (Completeness). If the 3-XOR instance $\Phi$ admits perfect solution for $r$-round Lasserre SDP relaxation, then the $r/3$-round SDP relaxation $\Psi_1$ (in Figure 6.1) with parameter $\tau = 0.5 - O(1/M)$ for the BALANCEDSEPARATOR instance $H_\Phi$ has a solution of value $5m$.

Proof. We define a set of vectors (i.e. a solution to $\Psi_1$) using the vectors given in Theorem 6.2.4, as follows.

For each set $S \subseteq L_\Phi \cup R_\Phi \cup Z_r$ with $|S| \leq r/3$, we define the vector $\overline{U}_S$ as follows. If $S \cap Z_r \neq \emptyset$, let $\overline{U}_S = 0$. If $S \cap Z_r = \emptyset$, suppose that $S \cap L_\Phi$ contains

$$(C_{i_1}, \alpha_1), (C_{i_2}, \alpha_2), \ldots, (C_{i_{r_1}}, \alpha_{r_1}),$$

$S \cap R_\Phi$ contains

$$(x_{j_1}, \alpha'_{1}, t_1), (x_{j_2}, \alpha'_{2}, t_2), \ldots, (x_{j_{r_2}}, \alpha'_{r_2}, t_{r_2}),$$

we have $r_1 + r_2 = |S|$. Let $S'$ be the set of variables accessed by $C_{i_1}, \ldots, C_{i_{r_2}}$ together with $x_{j_1}, \ldots, x_{j_{r_2}}$. Note that $|S'| \leq 3r_1 + r_2 \leq 3|S| \leq r$. If there is no contradiction among the partial assignments $\alpha_i$’s and $\alpha_i'$’s (i.e. there are not two of them assigning the same variable to different values), we can define

$$\alpha = \alpha_1 \circ \alpha_2 \circ \cdots \circ \alpha_{r_1} \circ \alpha'_{1} \circ \alpha'_{2} \circ \cdots \circ \alpha'_{r_2}.$$

and let $\overline{U}_S = W_{(S', \alpha)}$, otherwise we let $\overline{U}_S = 0$.

We first check that the first 3 constraints in relaxation $\Psi_1$ are satisfied.
• For two sets \( S_1, S_2 \), either at least one of the vectors \( \mathbf{U}_{S_1}, \mathbf{U}_{S_2} \) is 0 (therefore their inner-product is 0), or \( \mathbf{U}_{S_1} = W_{S_1, \alpha_1}, \mathbf{U}_{S_2} = W_{S_2, \alpha_2} \) for some \( S'_1, S'_2, \alpha_1, \alpha_2 \) and \( \langle \mathbf{U}_{S_1}, \mathbf{U}_{S_2} \rangle = \langle W_{S'_1, \alpha_1}, W_{S'_2, \alpha_2} \rangle \geq 0 \).

• For any \( S_1, S_2, S_3, S_4 \) such that \( S_1 \cup S_2 = S_3 \cup S_4 \), either the set of partial assignments in \( S_1 \cup S_2 = S_3 \cup S_4 \) are consistent with each other, in which case we have \( \mathbf{U}_{S_1 \cup S_2} = \mathbf{U}_{S_3 \cup S_4} = W_{S, \alpha} \) where \( S \) is the union of all the variables included in \( S_1 \cup S_2 \) and \( \alpha \) is the concatenation of the partial assignments in \( S_1 \cup S_2 \); or we have \( \mathbf{U}_{S_1 \cup S_2} = \mathbf{U}_{S_3 \cup S_4} = 0 \).

• \( \| \mathbf{U}_0 \|^2 = \| W_{(0,0)} \|^2 = 1 \).

Now we check that the balance condition (the last constraint in relaxation \( \Psi_1 \)) is satisfied. We will prove that

\[
\sum_{v} \mathbf{U}_{\{v\}} = (M + 1)m \mathbf{U}_0.
\]

Since there are \((2M + 5)m\) vertices in \( \mathcal{H}_\Phi \), this shows that the solution is feasible for \( \Phi_1 \) with \( \tau = 0.5 - O(1/M) \). Using 6.2.5 we see that \( \sum_{(C_i, \alpha) \in L_\Phi} \mathbf{U}_{\{C_i, \alpha\}} = \sum_{C_i} \mathbf{U}_0 = m \mathbf{U}_0 \). Similarly

\[
\sum_{(x_j, \alpha, t) \in R_\Phi} \mathbf{U}_{\{x_j, \alpha, t\}} = \sum_{j=1}^{n} \sum_{\alpha : \{x_j\} \to \{0,1\}} \sum_{t=1}^{\beta M} \mathbf{U}_{\{x_j, \alpha, t\}} = \beta M n \cdot \mathbf{U}_0 = M m \mathbf{U}_0.
\]

Thus

\[
\sum_{v \in V} \mathbf{U}_{\{v\}} = \sum_{v \in L_\Phi \cup R_\Phi \cup \mathcal{Z}_\tau} \mathbf{U}_{\{v\}} = \sum_{(C_i, \alpha) \in L_\Phi} \mathbf{U}_{\{C_i, \alpha\}} + \sum_{(x_j, \alpha, t) \in R_\Phi} \mathbf{U}_{\{x_j, \alpha, t\}} = (M + 1)m \mathbf{U}_0.
\]

Now, we calculate the value of the solution

\[
\sum_{(u, v) \in E_\Phi} \left\| \mathbf{U}_{\{u\}} - \mathbf{U}_{\{v\}} \right\|^2
\]

\[
= \sum_{i=1}^{m} \sum_{\alpha : \{x_{i_1}, x_{i_2}, x_{i_3}\} \to \{0,1\}, C_i(\alpha) = 1} \sum_{z=1}^{3} \left\| \mathbf{U}_{\{C_i, \alpha\}} - \mathbf{U}_{\{x_{i_z}, \alpha_{\{x_{i_z}\}}, 1\}} \right\|^2
\]

\[
+ \sum_{i=1}^{m} \sum_{\alpha : \{x_{i_1}, x_{i_2}, x_{i_3}\} \to \{0,1\}, C_i(\alpha) = 1} \sum_{v \in \mathcal{Z}_\tau, \{((C_i, \alpha), v) \in E_\Phi}} \left\| \mathbf{U}_{\{C_i, \alpha\}} - \mathbf{U}_{\{v\}} \right\|^2
\]

97
+ \sum_{j=1}^{n} \sum_{\alpha \{x_j\} \to \{0,1\}} \sum_{z_1, z_2 \in [M \beta]} \| \mathcal{U}_{\{x_j, \alpha, z_1\}} - \mathcal{U}_{\{x_j, \alpha, z_2\}} \|^2 + \sum_{v_1, v_2 \in Z_r} \| \mathcal{U}_{\{v_1\}} - \mathcal{U}_{\{v_2\}} \|^2 \\
= \sum_{i=1}^{m} \sum_{\alpha \{x_i, x_{i_2}, x_{i_3}\} \to \{0,1\}, C_i(\alpha) = 1} \left( \sum_{z = 1}^{3} \| \mathcal{U}_{\{C_i, \alpha\}} - \mathcal{U}_{\{x_{i_2}, \alpha_{\{x_{i_2}\}}\}} \|^2 + 2 \| \mathcal{U}_{\{C_i, \alpha\}} \|^2 \right) \\
= \sum_{i=1}^{m} \sum_{\alpha \{x_i, x_{i_2}, x_{i_3}\} \to \{0,1\}, C_i(\alpha) = 1} \left( \sum_{z = 1}^{3} \| W_{\{x_{i_2}, \alpha_{\{x_{i_2}\}}\}} - W_{\{x_{i_1}, x_{i_2}, x_{i_3}\}} \|^2 \right) \\
= \sum_{i=1}^{m} \sum_{\alpha \{x_i, x_{i_2}, x_{i_3}\} \to \{0,1\}, C_i(\alpha) = 1} \sum_{z = 1}^{3} \| W_{\{x_{i_2}, \alpha_{\{x_{i_2}\}}\}} \|^2 - \sum_{i=1}^{m} \sum_{\alpha \{x_i, x_{i_2}, x_{i_3}\} \to \{0,1\}, C_i(\alpha) = 1} \sum_{z = 1}^{3} \| W_{\{x_{i_1}, x_{i_2}, x_{i_3}\}} \|^2 \\
= \sum_{i=1}^{m} \sum_{z = 1}^{3} \left( \| W_{\{x_{i_2}, \{x_i \to 0\}\}} \|^2 + \| W_{\{x_{i_2}, \{x_i \to 1\}\}} \|^2 \right) - m \\
= 6m - m = 5m. \\
\square

6.3.3 Soundness : bound for integral solutions

Let \( \mathcal{L} = \{(x_j, \alpha) \mid \alpha : \{x_j\} \to \{0, 1\}\} \) be the set of \( 2n \) literals. For each literal \((x_j, \alpha) \in \mathcal{L}\), let \( \text{deg}((x_j, \alpha)) \) be the number of left vertices that connect to the literal’s representative vertex \((x_j, \alpha, 1)\). For a set of literals \( \mathcal{L}' \subseteq \mathcal{L} \), let \( \text{deg}(\mathcal{L}') = \sum_{(x_j, \alpha) \in \mathcal{L}'} \text{deg}((x_j, \alpha)) \). Also, given a subset \( \mathcal{L}' \subseteq \mathcal{L} \), for left vertex \((C_i, \alpha)\), say \((C_i, \alpha)\) is contained in \( \mathcal{L}' \) if all the three
literals corresponding to the three neighbors of \((C_i, \alpha)\) in \(H_\Phi\) are contained in \(L'\), i.e.
\[
\{(x_{i_1}, \alpha|_{x_{i_1}}), (x_{i_2}, \alpha|_{x_{i_2}}), (x_{i_3}, \alpha|_{x_{i_3}})\} \subseteq L'.
\]

We first prove the following lemma regarding the structure of \(H_\Phi\), defined by a random 3-XOR instance \(\Phi\).

**Lemma 6.3.3.** Over the choice of random 3-XOR instance \(\Phi\), with probability \(1 - o(1)\), the following statements hold.

- For each \(L' \subseteq L\), if \(|L'| \geq n/3\), we have \(\deg(L') \geq 6m \cdot |L'|/n(1 - 20/\sqrt{\beta})\).
- For each \(L' \subseteq L\), if \(|L'| \geq n/3\), the number of left vertices in \(L_\Phi\) contained in \(L'\) is at most \(m \cdot |L'|^3/(2n^3) \cdot (1 + 100/\sqrt{\beta})\).

**Proof.** Fix a literal \((x_j, \alpha)\), a random constraint \(C_i\) accesses \(x_j\) with probability \(3/n\). Once \(C_i\) accesses \(x_j\), there are 2 vertices out of the 4 left vertices corresponding to \(C_i\) adjacent to \((x_j, \alpha)\). Therefore, in expectation, there are \(6/n\) edges from the left vertices corresponding to \(C_i\) to \((x_j, \alpha)\). By linearity of expectation, for fixed \(L' \subseteq L\), there are \(6|L'|/n\) edges from the left vertices corresponding to a random constraint \(C_i\) to \(L'\) in expectation.

Now for each \(C_i\), let the random variable \(X_i\) be the number of representative vertices in \(L'\) that is connected to left vertices corresponding to \(C_i\). By definition we have \(\deg(L') = \sum_{i=1}^{m} X_i\). Since each left vertex corresponding to \(C_i\) has 3 neighbors on the right side, and there are 4 of such left vertices, we know that \(X_i \in [0, 12]\). In the previous paragraph we have concluded that \(\mathbb{E}[X_i] = 6|L'|/n\) for all \(i = 1, 2, \ldots, m\). It is also easy to see that \(X_1, X_2, \ldots, X_m\) are independent random variables.

Now assuming that \(|L'| \geq n/3\), we use Hoeffding’s inequality for the random variables \(X_1, X_2, \ldots, X_m\), and get
\[
\Pr[\deg(L') < 6m \cdot |L'|/n(1 - 20/\sqrt{\beta})] = \Pr \left[ \sum_{i=1}^{n} X_i < 6m \cdot |L'|/n(1 - 20/\sqrt{\beta}) \right]
\leq \exp \left( - \frac{2 \cdot \left( \frac{20}{\sqrt{\beta}} \cdot \frac{6m \cdot |L'|}{n} \right)^2}{m \cdot 12^2} \right) = \exp \left( -200 \cdot \left( \frac{|L'|}{n} \right)^2 \cdot n \right) \leq \exp (-22n) \leq 2^{-4n}.
\]

Since there are at most \(2^{2n}\) such \(L'\)'s, by a union bound, with probability at least \(1 - 2^{-2n}\), the first statement holds.
For the second statement, fix an $L' \subseteq L$, let $a_0, a_1, a_2$ be the number of variables that have 0, 1, 2 corresponding literals in $L'$, respectively. Note that $a_0 + a_1 + a_2 = n$ and $a_1 + 2a_2 = |L'|$ Now, for a random constraint $C_i$, we are interested in the expected number of the four corresponding left vertices ($C_i, \alpha$) that are contained in $L'$. Note that once $C_i$ accesses a variable that corresponds to $a_0$, none of the four corresponding left vertices are contained in $L'$. Now let us condition on the case that, out of the 3 variables accessed by $C_i$, $t$ variables have two literals in $L'$ and the other $(3 - t)$ variables have one literal in $L'$. Observe that in expectation (which is over the random choice of $C_i$ while conditioned on $t$), there are $2^{t-1}$ left vertices corresponding to $C_i$ contained in $L'$.

In all, the expected number of the left vertices corresponding to $C_i$ that are contained in $L'$ is

$$\sum_{t=0}^{3} \left( \frac{a_1}{3-t} \right) \left( \frac{a_2}{t} \right) \cdot 2^{t-1} < (1 + \frac{10}{n}) \sum_{t=0}^{3} \left( \frac{3}{t} \right) (a_1/n)^{3-t} (a_2/n)^t \cdot 2^{t-1}$$

(for $n > 3$)

$$= (1 + \frac{10}{n})(a_1 + 2a_2)^3/(2n^3) = (1 + \frac{10}{n}) \cdot |L'|^3/(2n^3).$$

For each $C_i$, let the random variable $X_i$ be the number of left vertices corresponding to $C_i$ that are contained in $L'$. By the discuss above, we know that $\mathbb{E}[X_i] \leq (1 + \frac{10}{n}) \cdot |L'|^3/(2n^3)$. Now we are interested in the probability that the total number of left vertices contained in $L'$ (i.e. $\sum_{i=1}^{m} X_i$) is big. Since $X_i$'s are always bounded by $[0, 4]$, by standard Chernoff bound, we have

$$\Pr \left[ \sum_{i=1}^{m} X_i > m \cdot |L'|^3/(2n^3) \cdot (1 + 100/\sqrt{\beta}) \right]$$

$$= \Pr \left[ \sum_{i=1}^{m} X_i > m \cdot \left( 1 + \frac{10}{n} \right) \cdot |L'|^3/(2n^3) \cdot \frac{1 + 100/\sqrt{\beta}}{1 + 10/n} \right]$$

$$= \Pr \left[ \sum_{i=1}^{m} X_i > m \cdot \left( 1 + \frac{10}{n} \right) \cdot |L'|^3/(2n^3) \cdot \left( 1 + \frac{100/\sqrt{\beta} - 10/n}{1 + 10/n} \right) \right]$$

$$\leq \exp \left( -\frac{1}{4} \cdot m \cdot \left( 1 + \frac{10}{n} \right) \cdot |L'|^3/(2n^3) \cdot \frac{(100/\sqrt{\beta} - 10/n)^2}{3(1 + 10/n)^2} \right)$$

(for large enough $\beta$)

$$\leq \exp \left( -\frac{1}{4} \cdot m \cdot |L'|^3/(2n^3) \cdot \frac{(80/\sqrt{\beta})^2}{3} \right)$$

(for $n \gg \sqrt{\beta} \gg 1$)

$$= \exp \left( -\beta n \cdot \frac{|L'|^3}{n^3} \cdot \frac{1}{\beta} \cdot \frac{800}{3} \right)$$

$$\leq \exp \left( -n \cdot \frac{800}{3^4} \right)$$

(since $|L'| \geq n/3$)
\[ \leq 2^{-4n}. \]

Since there are at most \(2^{2n}\) such \(L'\)'s, by a union bound, with probability at least \(1 - 2^{-2n}\), the second statement holds. \qed

Now, we are ready to prove the soundness lemma.

**Lemma 6.3.4 (Soundness).** For \(\tau > 1/3\), with probability \(1 - o(1)\), the \(\tau\) vs. \((1 - \tau)\) balanced separator has at least \(4m(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))\) edges in the cut.

**Proof.** We are going to prove that, once the two conditions in Lemma 6.3.3 hold, we have the desired upper bound for \(\tau\) vs. \((1 - \tau)\) balanced separator. Let us assume that there is a balanced separator \((A', B')\) such that \(\text{edges}(A', B') \leq 4m(3\tau - \tau^3) \leq 12m\), we will show that \(\text{edges}(A', B') \geq 4m(3\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))\).

Based on \((A', B')\) we build another cut \((A, B)\) such that \(A \cap Z_r = A' \cap Z_r\) and \(A \cap R_\Phi = A' \cap R_\Phi\). For each left vertex in \(L_\Phi\), it has 5 edges going to \(Z_r\) and \(R_\Phi\). We assign the vertex to \(A\) if it has less than 3 edges going to \(B' \cap (Z_r \cup R_\Phi)\), and assign it to \(B\) otherwise. Note that edges \((A, B) \leq \text{edges}(A', B')\), therefore we only need to show that edges \((A, B) \geq m(12\tau - \tau^3 - O(1/\sqrt{\beta}) - O(1/M))\). Since \(L_\Phi\) contains only \(O(1/M)\) fraction of the total vertices, \((A, B)\) is still \((\tau - O(1/M))\) vs. \((1 - \tau + O(1/M))\) balanced.

Since edges \((A, B) \leq 12m\), for large enough constant \(M\), we have the following two statements.

1) One of \(A \cap Z_r\) and \(B \cap Z_r\) has at most \(100/M \cdot |Z_r| = 100m/M\) vertices.

2) Let \(C_{\text{bad}} = \{(x_j, \alpha) : \text{the clique } C_{(x_j, \alpha)} \text{ is broken by } (A, B)\}\), then \(|C_{\text{bad}}| \leq 20n/M\).

If 1) does not hold, then we see there are at least \((100/M) \cdot |Z_r| \cdot M = 100m\) edges in \(Z_r\) cut by \((A, B)\), by the expansion property. If 2) does not hold, for each clique \(C_{(x_j, \alpha)}\) that is broken by \((A, B)\), at least \((\beta M - 1)\) edges of the clique are in the cut. In all, there are at least \((\beta M - 1) \cdot 20n/M > 12\beta n = 12m\) edges in the cut.

Now, by 1), assume w.l.o.g. that \(A \cap Z_r\) is the smaller side – having at most \(100/M \cdot |Z_r|\) vertices, and let \(L'\) be the set of literals \((x_j, \alpha)\) such that its representative vertex \((x_j, \alpha, 1)\) is in \(A\).

To get a lower bound for \(|L'\|\), note that

\[ |A| \leq (|L'| + |C_{\text{bad}}|) \cdot M\beta + |Z_r| + |L_\Phi| = |L'| \cdot M\beta + O(1)m. \quad (6.5) \]
Also, since \((A, B)\) is a balanced separator, we have \(|A| \geq (\tau - O(1/M)) \cdot 2Mm\). Hence, by (6.5), we have \(|L'| \geq \tau - O(1/M) \cdot 2n\).

Let \(L_{\text{bad}} \subseteq L_\phi\) be the set of left vertices such that at least one of the two neighbors in \(Z_r\) falls into \(A \cap Z_r\). By the regularity of the graph where \(Z_r\) is the set of vertices and \(L_\phi\) is the set of edges, we know that \(|L_{\text{bad}}| \leq 8 \cdot 100/M \cdot |Z_r| \leq O(m/M)\).

Now let us get a lower bound on edges\((A, B)\). First, we have edges\((A, B) \geq \text{edges}(A \backslash L_{\text{bad}}, B \backslash L_{\text{bad}})\). Let \(L'_\phi = L_\phi \backslash L_{\text{bad}}\), we have

\[
\text{edges}(A \backslash L_{\text{bad}}, B \backslash L_{\text{bad}}) = \text{edges}(A \cap (L'_\phi \cup R_\phi \cup Z_r), B \cap (L'_\phi \cup R_\phi \cup Z_r)) \\
\geq \text{edges}(A \cap L'_\phi, B \cap Z_r) + \text{edges}(A \cap R_\phi, B \cap L'_\phi) \\
= \text{edges}(A \cap L'_\phi, B \cap Z_r) + \text{edges}(A \cap R_\phi, L'_\phi) - \text{edges}(A \cap R_\phi, A \cap L'_\phi) \\
\geq \text{edges}(A \cap L'_\phi, B \cap Z_r) + \text{edges}(A \cap R_\phi, L'_\phi) - |L_{\text{bad}}| \cdot 3 - \text{edges}(A \cap R_\phi, A \cap L'_\phi).
\]

Consider a left vertex \((C_i, \alpha) \in L'_\phi\). We claim that it is contained in \(L'\) if and only if \((C_i, \alpha) \in A\). This is because if it is contained in \(L'\), then we have \((C_i, \alpha) \in A\) because 3 out of 5 edges incident to \((C_i, \alpha)\) go to \(A\) side (the three variable representative vertices). If \((C_i, \alpha)\) is not contained in \(L'\), we have at least 3 out of the 5 edges going to \(B\) side (the two edges to \(B \cap Z_r\) and at least one of the variable representative vertices), and therefore we have \((C_i, \alpha) \in B\). By this claim, we know the following two facts.

- \(|A \cap L'_\phi|\) is small. Since \(\tau > 1/3\), we have \(|L'| \geq (2/3 - O(1/M))n > n/3\), and by the second property of Lemma 6.3.3, we have \(|A \cap L'_\phi| \leq m \cdot |L'|^3/(2n^3) \cdot (1 + 100/\sqrt{\beta})\).
- We have \(\text{edges}(A \cap L'_\phi, B \cap Z_r) = 2|A \cap L'_\phi|\) and \(\text{edges}(A \cap L'_\phi, A \cap R_\phi) = 3|A \cap L'_\phi|\).

For \(\text{edges}(A \cap R_\phi, L'_\phi)\), we know that this is exactly \(\deg(L')\). Again, since \(\tau > 1/3\), by the first property of Lemma 6.3.3, we know this value is lower-bounded by \(6m \cdot |L'|/n(1 - 20/\sqrt{\beta})\).

In all, we have

\[
\begin{align*}
\text{edges}(A, B) & \geq \text{edges}(A \cap L'_\phi, B \cap Z_r) + \text{edges}(A \cap R_\phi, L'_\phi) - |L_{\text{bad}}| \cdot 3 - \text{edges}(A \cap R_\phi, A \cap L'_\phi) \\
& = 2|A \cap L'_\phi| + \deg(L') - |L_{\text{bad}}| \cdot 3 - 3|A \cap L'_\phi| \\
& \geq \deg(L') - |A \cap L'_\phi| - O(m/M)
\end{align*}
\]
\[ \geq 6m \cdot |L'| / n (1 - 20 / \sqrt{\beta}) - m \cdot |L'|^3 / (2n^3) \cdot (1 + 100 / \sqrt{\beta}) - O(m/M) \]
\[ = m \left( 12\gamma - 4\gamma^3 - (240\gamma + 400\gamma^3) / \sqrt{\beta} - O(1/M) \right) \quad \text{(let } \gamma = |L'| / (2n)) \]
\[ \geq 4m \left( 3\tau - \tau^3 - O(1 / \sqrt{\beta}) - O(1/M) \right). \]

The last step follows because (i) \( 3\gamma - \gamma^3 \) monotonically increases when \( \gamma \in [0, 1] \), and (ii) \( \gamma \geq (\tau - O(1/M)) \).

6.3.4 Constant-degree integrality gap instance

In this subsection, we slightly modify the graph \( \mathcal{H}_\Phi \) obtained in the previous subsections to get an integrality gap instance with constant degree.

Observe that in \( \mathcal{H}_\Phi \), when \( M \) and \( \beta \) are constants, the only vertices whose degree might be superconstant are the representative vertices in \( R_\Phi \). Now consider the edges connecting vertices in \( L_\Phi \) and representative vertices: there are \( 12m \) of them, each of them corresponds to a combination of constraint \( C_i \), satisfying assignment \( \alpha \), and one of the variables in the constraint. Let \( E_b \) be the set of these edges.

For two edges \( e_1, e_2 \in E_b \), let the random variable \( Y_{\{e_1, e_2\}} = 1 \) if they share the same representative vertex, and let \( Y_{\{e_1, e_2\}} = 0 \) otherwise. Finally let \( Y = \sum_{e_1, e_2 \in E_b} Y_{\{e_1, e_2\}} \).

By the simple second moment method, we know that with probability \( 1 - o(1) \), we have \( Y \leq 1000m^2 / n = 1000\beta^2 n \).

For every edge \( e \in E_b \), if \( \sum_{e' \in E_b \setminus \{e\}} Y_{\{e, e'\}} > \beta M \), we remove \( e \) from the graph. In this way, we get a new graph, namely \( \mathcal{H}'_\Phi \). We claim the following properties about \( \mathcal{H}'_\Phi \).

1. The maximum degree of \( \mathcal{H}'_\Phi \) is \( O(\beta M) \). This is because the maximum degree of vertices other than representative vertices in \( \mathcal{H}_\Phi \) is \( O(\beta M) \), and after the edge removal process described above, the representative vertices have degree \( O(\beta M) \).
2. The number of edges removed is at most \( 2Y / (\beta M) \), and therefore \( 2000m / M \) with probability \( 1 - o(1) \). This is because whenever an edge is removed, we charge \( \beta M \) to \( Y \). Since each edge in \( Y \) can be charge at most twice, there are at most \( 2Y / (\beta M) \) edges to be removed.
3. The SDP solution in Lemma 6.3.2 is still feasible and has objective value at most \( 5m \) (since we removed edges) with probability \( 1 - o(1) \).
4. The soundness lemma Lemma 6.3.4 still holds since we removed only \( O(m/M) \) edges.
Therefore, we claim that $\mathcal{H}'_\Phi$ is an integrality gap instance for Theorem 6.3.1 with constant degree.

6.4 Gaps for $\textsc{UniformSparsestCut}$

In this section, we provide the full analysis of the gap instance for $\textsc{UniformSparsestCut}$. We first describe our construction of the gap instance for $\textsc{UniformSparsestCut}$ as follows.

We modify the gap instance we got for $\textsc{BalancedSeparator}$ to get an instance for the linear round Lasserre relaxation of $\textsc{UniformSparsestCut}$ in an almost black box style. In the $\textsc{BalancedSeparator}$ problem, we have the hard constraint that the cut is $\tau$-balanced. In the reduction from $\textsc{BalancedSeparator}$ to $\textsc{UniformSparsestCut}$, we need to use the sparsity objective to enforce this constraint. We do it as follows. Recall that given a 3-XOR instance $\Phi$, the corresponding gap instance for $\textsc{BalancedSeparator}$ consists of vertex set $L_\Phi \cup R_\Phi \cup Z_r$ and edge set $E_\Phi$. To get a gap instance for $\textsc{UniformSparsestCut}$, we add two more $O(M)$-regular expanders (with edge expansion $10^4 \cdot M$) $D_l$ and $D_r$ of size $1000 M m$ (where $M$ is the same parameter defined in the previous sections). Now, let the edge set $E'_\Phi$ contain the edges in $E_\Phi$, in the expanders $D_l$ and $D_r$, and the following edges: for each vertex $v \in L_\Phi \cup R_\Phi \cup Z_r$, introduce 2 new edges incident to it, one to a vertex in $D_l$ (say, $v_l$) and the other one to a vertex in $D_r$ (say, $v_r$). We arrange these edges (between $L_\Phi \cup R_\Phi \cup Z_r$ and $D_l$, $D_r$) in a way so that each vertex in $D_l$ (or $D_r$) has at most one neighbor in $L_\Phi \cup R_\Phi \cup Z_r$ – this can be done because $|L_\Phi| + |R_\Phi| + |Z_r| = (2M + 5)m < 1000 M m = |D_l| = |D_r|$.

Using the instance described above, we will prove our main integrality gap theorem (Theorem 6.1.3) for $\textsc{UniformSparsestCut}$. We state the full theorem as follows.

Theorem 6.4.1. For large enough constant $\beta, M$ (where $\beta$ is the same parameter as in previous sections), and infinitely many positive integer $N$'s, there is an $N$-vertex instance for $\textsc{UniformSparsestCut}$ problem, such that the optimal solution is at least $(1 + 1/(100 M))$ times worse than the optimal solution of the $\Omega(N)$-round Lasserre SDP.

Theorem 6.4.1 is directly implied by the following completeness lemma (Lemma 6.4.2) and soundness lemma (Lemma 6.4.3).

Lemma 6.4.2 (Completeness). The value of relaxation $\Psi_2$ (in Figure 6.2) is at most

$$(2M + 10)m/((1001 M + 1)m)^2$$
for $\tau = (1001M + 1)/(2002M + 5)$.

**Proof.** Given the SDP solution $\{U_{S'}\} S' \subseteq L_\Phi \cup R_\Phi \cup Z_\tau, |S'| \leq r/3$ in the completeness case of **BALANCEDSEPARATOR**, we extend it to the SDP solution $\{U_S\} S \subseteq L_\Phi \cup R_\Phi \cup Z_\tau \cup S \cup D_l \cup D_r, |S| \leq r/3$ for **UNIFORMSPARSESTCUT** by “putting $D_l$ and $D_r$ one per side”. That is, for each $S \subseteq L_\Phi \cup R_\Phi \cup Z_\tau \cup D_l \cup D_r$ with $|S| \leq r/3$, let $S' = S \cap (L_\Phi \cup R_\Phi \cup Z_\tau)$. Now we let $U_S = 0$ if $S \cap D_r \neq \emptyset$, and let $U_S = U_{S'}$ otherwise.

We first check that $\{U_S\} S \subseteq L_\Phi \cup R_\Phi \cup Z_\tau \cup D_l \cup D_r, |S| \leq r/3$ is a feasible SDP solution. We only check that the balance constraint (the last constraint in relaxation $\Phi_2$) is met.

We are going to prove prove that

$$\sum_{u \in L_\Phi \cup R_\Phi \cup Z_\tau} U_{\{u\}} = (1001M + 1)mU_\emptyset.$$ 

From the proof of **Lemma 6.3.2** we know that

$$\sum_{u \in L_\Phi \cup R_\Phi \cup Z_\tau} U_{\{u\}} = (M + 1)mU_\emptyset,$$

together with the fact that

$$\forall u \in D_l, U_{\{u\}} = U_\emptyset, \quad \forall u \in D_r, U_{\{u\}} = 0,$$

we get the desired equality.

Now we calculate the value of the solution. First, we calculate the following value.

$$\sum_{(u, v) \in E_\Phi} \|U_{\{u\}} - U_{\{v\}}\|^2 = \sum_{(u, v) \in E_\Phi} \|U_{\{u\}} - U_{\{v\}}\|^2 + \sum_{(u, v) \in E_\Phi \setminus E_\Phi} \|U_{\{u\}} - U_{\{v\}}\|^2$$

$$= 5m + \sum_{u, v \in D_l} \|U_{\{u\}} - U_{\{v\}}\|^2 + \sum_{u, v \in D_r} \|U_{\{u\}} - U_{\{v\}}\|^2$$

$$+ \sum_{u \in L_\Phi \cup R_\Phi \cup Z_\tau} \left(\|U_{\{u\}} - U_{\{v_l\}}\|^2 + \|U_{\{u\}} - U_{\{v_r\}}\|^2\right),$$

Note that $\sum_{u, v \in D_l} \|U_{\{u\}} - U_{\{v\}}\|^2 + \sum_{u, v \in D_r} \|U_{\{u\}} - U_{\{v\}}\|^2 = 0$, and

$$\sum_{u \in L_\Phi \cup R_\Phi \cup Z_\tau} \left(\|U_{\{u\}} - U_{\{v_l\}}\|^2 + \|U_{\{u\}} - U_{\{v_r\}}\|^2\right)$$

105
\[
\sum_{u \in L_{\Phi} \cup R_{\Phi} \cup Z_{r}} \left( 2 \left\| U_{\{u\}} \right\|^2 + \left\| U_{\{v_{l}\}} \right\|^2 + \left\| U_{\{v_{r}\}} \right\|^2 - 2 \left\langle U_{\{u\}}, U_{\{v_{l}\}} \right\rangle - 2 \left\langle U_{\{u\}}, U_{\{v_{r}\}} \right\rangle \right) \\
= \sum_{u \in L_{\Phi} \cup R_{\Phi} \cup Z_{r}} \left( 2 \left\| U_{\{u\}} \right\|^2 + 1 + 0 - 2 \left\| U_{\{u,v_{l}\}} \right\|^2 - 2 \left\| U_{\{u,v_{r}\}} \right\|^2 \right) \\
\sum_{u \in L_{\Phi} \cup R_{\Phi} \cup Z_{r}} 1 = \left| L_{\Phi} \right| + \left| R_{\Phi} \right| + \left| Z_{r} \right| = (2M + 5)m.
\]
Thus, we have
\[
\sum_{(u,v) \in E_{\Phi}} \left\| U_{\{u\}} - U_{\{v\}} \right\|^2 = (2M + 10)m.
\]
Since \( \tau < \frac{1}{2} \), the value of the solution is at most
\[
\frac{1}{\left| L_{\Phi} \cup R_{\Phi} \cup Z_{r} \cup D_{l} \cup D_{r} \right|^{\frac{1}{2}} \sum_{(u,v) \in E_{\Phi}} \left\| U_{\{u\}} - U_{\{v\}} \right\|^2 = (2M + 10)m/((1001M + 1)m)^{2}.
\]

**Lemma 6.4.3 (Soundness).** For large enough \( M \), the sparsity of the sparsest cut is at least
\[
\gamma = (1 + \frac{1}{(100M)}) \cdot (2M + 10)m/(1001Mm)^{2}.
\]

**Proof.** Let \( D_{l}' \) be the smaller part among \( D_{l} \cap S \) and \( D_{l} \cap \bar{S} \), and \( D_{r}' \) be the larger part. Also, let \( D_{r}' \) be the smaller part among \( D_{r} \cap S \) and \( D_{r} \cap \bar{S} \) and \( D_{r}'' \) be the larger part. Let \( (T, \bar{T}) \) be the cut restricted to \( L_{\Phi} \cup R_{\Phi} \cup Z_{r} \) (the BALANCEDSEPARATOR instance), i.e. let \( T = S \cap (L_{\Phi} \cup R_{\Phi} \cup Z_{r}) \) and \( \bar{T} = \bar{S} \cap (L_{\Phi} \cup R_{\Phi} \cup Z_{r}) \).

First, we show that to get a cut of sparsity better than \( \gamma \), \( |D_{l}'| \leq \frac{1}{10^{4}M} \cdot |D_{l}| \), and the same is true for \( D_{r} \) (by the same argument). This is because if \( |D_{l}'| > \frac{1}{10^{4}M} \cdot |D_{l}| \), by the expansion property, there are at least \( 10^{4}M \cdot |D_{l}'| > 1000Mm \) edges in the cut. Since the graph has \( |L_{\Phi}| + |R_{\Phi}| + |Z_{r}| + |D_{l}| + |D_{r}| = (2002M + 5)m \) vertices, therefore the sparsity of the cut is at least
\[
\frac{1000Mm}{\frac{1}{4} \cdot ((2002M + 5)m)^{2}} > \frac{500Mm}{(1001Mm)^{2}} > \gamma,
\]
for \( M > 1/25 \).

106
Second, we show that $D'_l$ and $D'_r$ should be on opposite sides of any cut of sparsity better than $\gamma$. Suppose not, let $S$ be the side of the cut which $D'_l$ and $D'_r$ are on. Recall that $T = S \cap (L_{\Phi} \cup R_{\Phi} \cup Z_r)$. We have

$$\text{edges}(S, \bar{S}) \geq \text{edges}(T, D''_l \cup D''_r) + \text{edges}(D'_l, D''_r) + \text{edges}(D'_r, D''_l).$$

Note that $\text{edges}(T, D''_l \cup D''_r) \geq 2|T| - |D'_l| - |D'_r|$ as each vertex in $D_l, D_r$ is connected to at most one vertex in $T$. Also, by the expansion property, $\text{edges}(D'_l, D''_r) + \text{edges}(D'_r, D''_l) \geq 1000M(|D'_l| + |D'_r|)$. Now, we have

$$\text{edges}(S, \bar{S}) \geq 2|T| - (|D'_l| + |D'_r|) + 1000M(|D'_l| + |D'_r|)$$

$$= 2(|T| + |D'_l| + |D'_r|) + (1000M - 3)(|D'_l| + |D'_r|) \geq 2(|T| + |D'_l| + |D'_r|) = 2|S|.$$

Therefore, the sparsity of the cut

$$\frac{\text{edges}(S, \bar{S})}{|S||\bar{S}|} \geq \frac{2|S|}{|S||\bar{S}|} = \frac{2}{|S|} \geq \frac{2}{(2002M + 5)m} > \gamma.$$

Third, we show that if the cut $(S, \bar{S})$ has sparsity better than $\gamma$, then the cut $(T, \bar{T})$ defined above is a 0.49 vs 0.51 balanced cut, i.e. $|T|/(|L_{\Phi}| + |R_{\Phi}| + |Z_r|) \in [0.49, 0.51]$. Supposing $(T, \bar{T})$ is not 0.49 vs 0.51 balanced, i.e. $||T| - |\bar{T}|| > 0.02 \cdot (2M + 5)m$, we have

$$||S| - |\bar{S}|| \geq ||T| - |\bar{T}|| - |D'_l| - |D'_r| \geq 0.02 \cdot (2M + 5)m - \frac{2}{1000M} \cdot 1000Mm$$

$$\geq (0.04M - 2)m \geq 0.01Mm,$$

for large enough $M$. Therefore, $(S, \bar{S})$ is not $0.5 - 10^{-6}$ vs $0.5 + 10^{-6}$ balanced. Thus,

$$|S||\bar{S}| < ((2002M + 5)m)^2 \cdot (0.5 - 10^{-6})(0.5 + 10^{-6}) < (1001Mm)^2 \cdot (1 - 10^{-12}).$$

Since $D''_l$ and $D''_r$ are on opposite sides of $(S, \bar{S})$, we know that $\text{edges}(S, \bar{S}) \geq (2M + 5)m - |D'_l| - |D'_r| \geq (2M + 5)m \cdot (1 - 1/M)$, and therefore the sparsity of the cut

$$\frac{\text{edges}(S, \bar{S})}{|S||\bar{S}|} \geq \frac{(2M + 5)m}{(1001Mm)^2} \cdot (1 - 1/M)(1 + 10^{-12}).$$

This value is greater than $\gamma$ when $M > 10^{20}$.

Finally, since $(T, \bar{T})$ is a 0.49 vs 0.51 balanced cut, by Lemma 6.3.4, we know that with probability $1 - o(1)$, $\text{edges}(T, \bar{T}) > (5.4 - O(1/\sqrt{3}) - O(1/M))m$. Therefore

$$\frac{\text{edges}(S, \bar{S})}{|S||\bar{S}|} \geq \frac{(2M + 5)m}{(1001Mm)^2} \cdot (1 - 1/M)(1 + 10^{-12}).$$

107
\[
\begin{align*}
\geq & \frac{\text{edges}(T, \bar{T}) + (2M + 5)m - |D'_l| - |D'_r|}{\frac{1}{4} \cdot ((2002M + 5)m)^2} \\
\geq & \frac{(5.4 - O(1/\sqrt{\beta}) - O(1/M))m + (2M + 5)m - \frac{1000Mm}{10^4M} - \frac{1000Mm}{10^4M}}{\frac{1}{4} \cdot ((2002M + 5)m)^2} \\
= & \frac{(2M + 10.2 - O(1/\sqrt{\beta}) - O(1/M))m}{\frac{1}{4} \cdot ((2002M + 5)m)^2} \\
\geq & \frac{(2M + 10.2 - O(1/\sqrt{\beta}) - O(1/M))m}{(1001Mm)^2} \cdot (1 - 1/(200M)) \\
\geq & \frac{(2M + 10.1)m}{(1001Mm)^2} \cdot (1 - 1/(200M)) \quad \text{(for large enough } \beta \text{ and } M) \\
\geq & \frac{(2M + 10)m}{(1001Mm)^2} \cdot (1 + 1/(30M))(1 - 1/(200M)) \quad \text{(for large enough } M) \\
\geq & \frac{(2M + 10)m}{(1001Mm)^2} \cdot (1 + 1/(100M)) = \gamma.
\end{align*}
\]
Chapter 7

Lasserre integrality gaps for ROBUSTGISO

7.1 Introduction

The GRAPHISOMORPHISM problem is one of the most intriguing and notorious problems in computational complexity theory (we will also refer to it as GISO for short); we refer to [149, 30, 24, 148, 69] for surveys. It was famously referred to as a “disease” over 35 years ago [199] and maintains its infectious status to this day. Together with FACTORING, it is one of the very rare problems in NP which is not known to be in P but which is believed to not be NP-hard [29, 49, 205] (according to standard complexity-theoretic assumptions). Both problems also admit an algorithm with running time “subexponential” (or “moderately exponential”) in the natural witness size. In the case of GRAPHISOMORPHISM on n-vertex graphs, the natural witness size is \( \log_2(n!) = \Theta(n \log n) \), but the best known algorithm due to Luks solves the problem in time \( 2^{O(\sqrt{n \log n})} \) [33].

In the same breath we might mention the problems GAPSVP\(\sqrt{n}\) (approximating the shortest vector in an n-dimensional lattice to factor \(\sqrt{n}\)) and UNIQUEGAMES\(\epsilon\) (the Unique Games problem proposed by Khot [136]). The former is not NP-hard subject to standard complexity-theoretic assumptions [95, 4], though we don’t know any subexponential-time algorithm. The latter has a subexponential-time algorithm [16]; whether it is NP-hard or in P (or neither) is hotly contested. The potential hardness of FACTORING and GAPSVP — even under certain average-case distributions — is well enough entrenched that many cryptographic protocols are based on it. (The same is true of random 3-XOR with noise, more on which later.) On the other hand, for GRAPHISOMORPHISM and UNIQUEGAMES\(\epsilon\)
we do not know any way of generating “hard-seeming instances”; indeed, some experts have speculated that GRAPHISOMORPHISM may be in \( \mathbb{P} \), or at least have a \( 2^{\text{polylog}(n)} \)-time algorithm.

In this chapter we investigate hardness results for the GISO problem. Since GISO may well be in \( \mathbb{P} \), let us discuss what this may mean. One direction would be to show that GISO is hard for small complexity classes. This has been pursued most successfully by Torán [217], who has shown that GISO is hard for the class DET. This is essentially the class of problems equivalent to computing the determinant; it contains \#L and is contained in TC\(^1\). It is not known whether GISO is \( \mathbb{P} \)-hard.

### 7.1.1 Our contributions

In this chapter, however, we are concerned with hardness results well above \( \mathbb{P} \). Our main contribution is that solving GISO via the Lasserre/SOS hierarchy requires \( 2^{\Omega(n)} \) time (i.e., \( \Omega(n) \) rounds/degree). This generalizes the result of Cai, Fürer, and Immerman [55] showing that the frequently effective \( o(n) \)-dimensional Weisfeiler–Lehman algorithm fails to solve GISO; it also gives even more evidence that any subexponential-time algorithm for GISO requires algebraic, non-local techniques.

Another result among our contributions is concerned with the problem of robust graph isomorphism, ROBUSTGISO. Recall from the introduction part of this that ROBUSTGISO is the following problem: given two graphs which are almost isomorphic, find an “almost-isomorphism”. ROBUSTGISO is strictly harder than GISO and the fact that it concerns “isomorphisms with noise” seems to rule out all algebraic techniques. We show that ROBUSTGISO is at least as hard as random 3-XOR with noise; hence ROBUSTGISO has no polynomial-time algorithm assuming the well-known R3XOR Hypothesis of Feige [88]. In fact, it’s possible that the R3XOR problem requires \( 2^{n^{1-o(1)}} \) time, which would mean that ROBUSTGISO is much harder than GISO itself.

### 7.1.2 SOS/Lasserre gaps

The most well-known heuristic for GRAPHISOMORPHISM (and the basis of most practical algorithms — e.g., “nauty” [171]) is the Weisfeiler–Lehman (WL) algorithm [222] and its “higher dimensional” generalizations. To describe the basic algorithm we need the notion of a colored graph. This is simply a graph, together with a function mapping the vertices to a finite set of colors; equivalently, a graph with its vertices partitioned into “color classes”. Isomorphisms involving colored graphs are always assumed to preserve colors. Let \( G \) be
a colored graph on the $n$-vertex set $V$. A color refinement step refers to the following procedure: for each $v \in V$, one determines the multiset $C_v$ of colors in the neighborhood of $v$; then one recolors each $v$ with color $C_v$. Now the basic WL algorithm, when given graphs $G$ and $H$, repeatedly applies refinement to each of them until the colorings stabilize. (Initially, the graphs are treated as having just one color class.) At the end, if $G$ and $H$ have the same number of vertices of each color the WL algorithm outputs that they are “maybe isomorphic”; otherwise, it (correctly) outputs that they are “not isomorphic”.

Note that after the initial refinement step, a graph’s vertices are colored according to their degree. Thus two $d$-regular graphs are always reported as “maybe isomorphic” by the basic WL algorithm. On the other hand, the heuristic is powerful enough to work correctly for all trees and for almost all $n$-vertex graphs in the Erdős–Rényi $G(n, 1/2)$ model [31, 32]. (We say the heuristic “works correctly” on a graph $G$ if the stabilized coloring for $G$ is distinct from the stabilized coloring of any graph not isomorphic to $G$.) To overcome WL’s failure for regular graphs, several researchers independently introduced the “$k$-dimensional generalization” WL$^k$ (see [222, 55] for discussion). Briefly, in the WL$^k$ heuristic, each $k$-tuple of vertices has a color, and color refinement involves looking at all “neighbors” of each $k$-tuple of vertices $(v_1, \ldots, v_k)$ (where the neighbors are all tuples of the form $(v_1, \ldots, v_{i-1}, u_i, v_{i+1}, \ldots, v_k)$, where $\{u_i, v_i\}$ is an edge). The WL$^k$ heuristic can be performed in time $n^{k+O(1)}$ and is thus a polynomial-time algorithm for any constant $k$.

The WL$^k$ heuristic is very powerful. For example, it is known to work correctly in polynomial time for all graphs which exclude a fixed minor [102], a class which includes all graphs of bounded treewidth or bounded genus. Spielman’s $2^{O(n^{1/3})}$-time graph isomorphism algorithm [213] for strongly regular graphs is achieved by WL$^k$ with $k = \tilde{O}(n^{1/3})$. The WL$^k$ algorithm with $k = O(\sqrt{n})$ is also a key component in the $2^{O(\sqrt{n \log n})}$-time GISO algorithm [33]. Throughout the ’80s there was some speculation that GISO might be solvable on all graphs by running the WL$^k$ algorithm with $k = O(\log n)$ or even $k = O(1)$. However this was disproved in the notable work of Cai, Fürer, and Immerman [55], which showed the existence of nonisomorphic $n$-vertex graphs $G$ and $H$ which are not distinguished by WL$^k$ unless $k = \Omega(n)$ [1].

The GRAPHISOMORPHISM problem can be thought of as kind of constraint satisfaction problem (CSP), and readers familiar with LP/SDP hierarchies for CSPs might see an analogy between $k$-dimensional WL and level-$k$ LP/SDP relaxations. A very interesting recent work of Atserias and Maneva [27] (see also [103]) shows that this is more than

---

1 Actually, $G$ and $H$ are colored graphs in [55]’s construction, with each color class having size at most 4. It is often stated that the colors can be replaced by gadgets while keeping the number of vertices $O(n)$. We do not find this to be immediately obvious. However it does follow from the asymmetry of random graphs, as we will see later in this chapter.
just an analogy — it shows that the power of WL\(^k\) is precisely sandwiched between that of the \(k\)th and \((k + 1)\)st level of the canonical Sherali–Adams LP hierarchy \([207]\). (In fact, it had long been known \([196]\) that WL\(^1\) is equivalent in power to the basic LP relaxation of GISO.) This gives a very satisfactory connection between standard techniques in optimization algorithms and the best known non-algebraic/local techniques for GISO.

This connection raises the question of whether stronger LP/SDP hierarchies might prove more powerful than WL\(^k\) in the context of GISO. The strongest such hierarchy known is the “SOS (sum-of-squares) hierarchy” due to Lasserre \([156]\) and Parrilo \([185]\). Very recent work \([34, 184, 133]\) (part of which included in this thesis) in the field of CSP approximability has shown that \(O(1)\) levels of the SOS hierarchy can succeed where \(\omega(1)\) levels of weaker SDP hierarchies fail; in particular, this holds for the hardest known instances of UNIQUEGAMES, \([34]\). This raises the question of whether there might be a subexponential-time algorithm based SOS which solves GRAPHISOMORPHISM.

We answer this question negatively. Our first main result is that a variant of the Cai–Fürer–Immerman instances also fools \(\Omega(\sqrt{n})\) levels of the SOS hierarchy. In fact, we achieve a “constant factor Lasserre gap with perfect completeness”. To explain this, recall the definition of \(\alpha\)-isomorphism from \[\text{Definition 2.1.5}\] which we restate as follows.

**Definition 7.1.1 (Re-statement of Definition 2.1.5).** Let \(G\) and \(H\) be nonempty \(n\)-vertex graphs. For \(0 \leq \beta \leq 1\), we say that a bijection \(\pi : V(G) \to V(H)\) is an \(\alpha\)-isomorphism if

\[
\frac{|\{(u, v) \in E(G) : (\pi(u), \pi(v)) \in E(H)\}|}{\max\{|E(G)|, |E(H)|\}} \geq \alpha.
\]

In this case we say that \(G\) and \(H\) are \(\alpha\)-isomorphic.

Observe that this definition is symmetric in \(G\) and \(H\). The two graphs are isomorphic if and only if they are 1-isomorphic. We will almost always consider the case that \(G\) and \(H\) have the same number of edges. We prove:

**Theorem 7.1.2.** For infinitely many \(n\), there exist pairs of \(n\)-vertex, \(O(n)\)-edge graphs \(G\) and \(H\) such that:

- \(G\) and \(H\) are not \((1 - 10^{-14})\)-isomorphic;

- any SOS refutation of the statement “\(G\) and \(H\) are isomorphic”\(^2\) requires degree \(\Omega(n)\).

\(^2\)When naturally encoded.
A word on our techniques. The essence of the Cai–Fürer–Immerman construction is to take a 3-regular expander graph and replace each vertex by a certain 10-vertex gadget (originally appearing in [123] and also sometimes called a “Fürer gadget”). This gadget is closely related to 3-variable equations modulo 2 (as observed by several authors, e.g. [217]); indeed, it may be described as the “label-extended graph” of the 3-XOR constraint. The reader may therefore recognize the [55] WL^k lower bound as stemming from the difficulty of refuting unsatisfiable, expanding 3-XOR CSP instances by “local” means. This should make our Theorem 7.1.2 look plausible in light of the Grigoriev [99] and Schoenebeck [204] SOS/Lasserre lower bounds.

Nevertheless, obtaining Theorem 7.1.2 is not automatic. For one, we still lack a complete theory of reductions within the SOS hierarchy (though see [220]). Second, the pair of graphs constructed by [55] only differ by one edge. More tricky is the issue of removing the “colors” from the [55] construction. We do not see an easy gadget-based way of doing this without sacrificing on the \( \Omega(n) \) degree. To handle this we have to: a) modify the [55] construction somewhat to make the two graphs differ by a constant fraction; b) prove that random (hyper)graphs are “robustly asymmetric” — i.e., “far from having nontrivial automorphisms”; c) use the robust asymmetry property to remove the “colors”. The result in b), described below in Section 7.1.4 qualitatively generalizes work of Erdős and Rényi [81] and may be of independent interest.

Comparison to the work by Snook et al. In an independent work, Schoenebeck, Codenotti, and Snook [211] have shown a conclusion similar to our Theorem 7.1.2. Their main result is that there are expander graphs \( G \) and \( H \) which are not isomorphic, but any SOS refutation of the statement “\( G \) and \( H \) are isomorphic” requires degree \( \Omega(n) \). As in our work, their proof combines the Cai–Fürer–Immerman construction with the Schoenebeck [204] SOS/Lasserre lower bounds.

### 7.1.3 Robust graph isomorphism

Our second main result concerns the ROBUSTGISO problem. Recall the definition of ROBUSTGISO from Definition 2.2.7 which we restate as follows.

**Definition 7.1.3** (Re-statement of Definition 2.2.7). We say an algorithm \( \mathcal{A} \) solves the ROBUSTGISO problem if there is a function \( r : [0, 1] \rightarrow [0, 1] \) satisfying \( r(\epsilon) \rightarrow 0 \) as \( \epsilon \rightarrow 0^+ \) such that whenever \( \mathcal{A} \) is given any pair of graphs which are \((1-\epsilon)\)-isomorphic, \( \mathcal{A} \) outputs a \((1-r(\epsilon))\)-isomorphism between them.

\(^3\)We could also consider the easier task of distinguishing pairs which are \((1-\epsilon)\)-isomorphic from pairs.
To understand the motivation of this definition, we might imagine an algorithm trying to recover an isomorphism between $G$ and $H$, where $H$ is formed by permuting the vertices of $G$ and then introducing a small amount of noise — say, adding and deleting an $\epsilon$ fraction of edges. Thinking of GISO as a CSP, we are concerned with finding “almost-satisfying” solutions on “almost-satisfiable” instances. For example, suppose we are given graphs $G$ and $H$ which are promised to be $(1 - \epsilon)$-isomorphic. Can we efficiently find a $(1 - 2\epsilon)$-isomorphism? A $(1 - \sqrt{\epsilon})$-isomorphism? A $(1 - \frac{1}{\log(1/\epsilon)})$-isomorphism? Therefore the notion of ROBUSTGISO naturally comes in.

**Remark 7.1.1.** In particular, $A$ in [Definition 7.1.3](#) must solve the GISO problem, because given isomorphic graphs with at most $m$ edges we can always take $\epsilon > 0$ small enough so that $r(\epsilon) < 1/m$.

The analogous problem of robust satisfaction algorithms for CSPs over constant-size domains was introduced by Zwick [228] and has proved to be very interesting. Guruswami and Zhou [113] conjectured that the CSPs which have efficient robust algorithms (subject to $P \neq NP$) are precisely those of “bounded width” — roughly speaking, those that do not encode equations over abelian groups. This conjecture was recently confirmed by Barto and Kozik [38], following partial progress in [153] [72].

The graph isomorphism seems to share some of the flavor of “unbounded width” CSPs such as 3-XOR; these CSPs have the property that special algebraic methods (namely, Gaussian elimination) are available on satisfiable instances, but these methods break down once there is a small amount of noise. Indeed, the $2^O(\sqrt{n \log n})$-time algorithm for GISO is a somewhat peculiar mix of group theory and “local methods” (namely, Weisfeiler–Lehman). Generalizing from GISO to ROBUSTGISO seems like it might rule out applicability of group-theoretic methods, thereby making the problem much harder. Our second main theorem in a sense confirms this. Roughly speaking, it shows that ROBUSTGISO is hard assuming it is hard to distinguish random 3-XOR instances from random instances with a planted solution and slight noise:

**Theorem 7.1.4.** Assume Feige’s R3XOR Hypothesis [38]. Then there is no polynomial-time algorithm for ROBUSTGISO. More precisely, there exists $\epsilon_0 > 0$, such that suppose there exists $\epsilon > 0$ and a $t(n)$-time algorithm which can distinguish $(1 - \epsilon)$-isomorphic $n$-vertex, $m$-edge graph pairs from pairs which are not even $(1 - \epsilon_0)$-isomorphic (where $m = O(n)$). Then there is a universal constant $\Delta \in \mathbb{Z}^+$ and a $t(O(n))$-time algorithm which outputs “typical” for almost all $n$-variable, $\Delta n$-constraint instances of the 3-XOR problem, yet which never outputs “typical” on instances which are $(1 - \Theta(\epsilon))$-satisfiable, which are not $(1 - r(\epsilon))$-isomorphic. In fact, our hardness result will hold even for this easier problem.
Here we refer to:

**Feige’s R3XOR Hypothesis.** For every fixed $\epsilon > 0$, $\Delta \in \mathbb{Z}^+$, there is no polynomial time algorithm which on almost all 3-XOR instances with $n$ variables and $m = \Delta n$ constraints outputs “typical”, but which never outputs “typical” on instances which an assignment satisfying at least $(1 - \epsilon)m$ constraints.

**Remark 7.1.2.** The reader may think of the output “typical” as a certification that the 3-XOR instance has no $(1 - \epsilon)$-satisfying solution. Note that with high probability the random 3-XOR instance will not even have a $0.51$-satisfying solution. Feige originally stated his hypothesis for the random 3SAT problem rather than the random 3XOR problem, but he showed the conjectures are equivalent. See also the work of Alekhnovich [5].

Feige’s R3XOR Hypothesis is a fairly well-believed conjecture. The variation in which the XOR constraints may involve any number of variables (not just 3) is called LPN (Learning Parities with Noise) and is believed to be hard even with any $m = \text{poly}(n)$ constraints. The further variation which has linear equations modulo a large prime rather than modulo 2 is called LWE (Learning With Errors) and forms the basis for a large body of cryptography. (See [200] for more on LPN and LWE.) The fastest known algorithm for solving Feige’s R3XOR problem seems to be the $2^{O(n/\log n)}$-time algorithm of Blum, Kalai, and Wasserman [44]. Thus it’s plausible that ROBUSTGISO requires $2^{n^{1-o(1)}}$ time, which would make it a much more difficulty problem than G1SO.

We close by mentioning some related literature on approximate graph isomorphism. The problem of finding a vertex permutation which maximizes the number of edge overlaps (or minimizes the number of edge/nonedge overlaps) was perhaps first discussed by Arora, Frieze, and Kaplan [17]. They gave an additive quasi-PTAS in the case of dense graphs ($m = \Omega(n^2)$) (as we previously discussed in Chapter 4). Arvind et al. [25] recently defined and studied several variants of the approximate graph isomorphism problem. Some of their results concern the case in which $G$ and $H$ have noticeably different numbers of edges and one isn’t “punished” for uncovered edges in $H$. This kind of variant is more like approximate subgraph isomorphism, and is much harder. (E.g., when $G$ is a $k$-clique and $H$ is a general graph the problem is roughly equivalent to the notorious DENSEST-kSUBGRAPH problem.) The result of theirs which is most relevant to the present work involves hardness of finding approximate isomorphisms in colored graphs. In particular, Arvind et al. prove the following:

**Theorem 7.1.5.** ([25]) There is a linear-time reduction from 2XOR (modulo 2) instances $I$ to pairs of colored graphs $G, H$ such that $G$ and $H$ are $\alpha$-isomorphic if and
only if \( I \) has a solution satisfying at least an \( \alpha \)-fraction of constraints. In particular it is NP-hard to approximate \( \alpha \)-isomorphism for colored graphs to a factor exceeding \( \frac{11}{12} \) and UNIQUEGAMES-hard to approximate it to a factor exceeding .878 (by results of [219, 116], [141, 174] respectively).

In particular, the theorem holds for colored graphs in which each color class contains at most 4 vertices. However, we do not see any way of eliminating the colors and getting the analogous inapproximability results for the usual G\( ISO \) problem without using gadgets that would destroy the constant-factor gap.

### 7.1.4 Robust asymmetry of random graphs

One of our main technical contributions is showing that random graphs are “robustly asymmetric”. In doing so, we generalize the concept of an asymmetric graph, which is a graph whose only automorphism is the trivial identity automorphism. A line of research (see, e.g., [81, 47, 172, 145]) has shown that several distributions of random graphs produce asymmetric graphs with high probability. In their well-known \( G(n, p) \) model, Erdős and Rényi [81] proved that for \( \frac{\ln n}{n} \leq p \leq 1 - \frac{\ln n}{n} \), \( G(n, p) \) is asymmetric with high probability. If we instead consider a uniformly random \( d \)-regular \( n \)-vertex graph, the sequence of works [47, 172, 145] shows that we get an asymmetric graph with high probability for any \( 3 \leq d \leq n - 4 \). In this work we will work with a third variant, the \( G_{n,m} \) model, in which a graph is chosen uniformly at random from all simple graphs with \( n \) vertices and \( m \) edges.

Given a graph \( G \) and a permutation \( \pi \) over \( V(G) \), we call \( \pi \) an \( \alpha \)-automorphism if the application of \( \pi \) on \( G \) preserves at least an \( \alpha \) fraction of the edges. A graph \( G \) is \((\beta,\gamma)\)-asymmetric if any \( \gamma \)-automorphism \( \pi \) has more than a fraction of \( (1 - \beta) \) fixed points. Intuitively, when \( \beta = 1/n, \gamma = 0 \), the property is exactly the asymmetry property; when \( \beta \) and \( \gamma \) become larger, the property requires that any permutation that is far from identity is far from an automorphism for the graph. We encourage the reader to refer to Section 7.2 for the precise definitions.

In this chapter, we show the following robust asymmetric property of \( G_{n,m} \).

**Theorem 7.1.6.** For large enough \( n \), suppose that \( m = cn \), where \( 10^4 \leq c \leq n/10^{10} \). Let \( \beta^* = \max\{e^{-c/6}, \frac{1}{n}\} \). With probability \( (1 - n^{-15}) \), for all \( \beta \) such that \( \beta^* \leq \beta \leq 1 \), \( G_{n,m} \) is \((\beta, \beta/240)\)-asymmetric.

A couple of comments are in order. First, an \( \exp(-O(c)) \) lower bound on \( \beta \) is necessary. This is because there are at least \( \lfloor \exp(-O(c)) \cdot n \rfloor \) isolated vertices in \( G_{n,m} \) with
high probability. The permutations which only permute these isolated vertices are 1-automorphisms. Therefore, with high probability, $G_{n,m}$ is not $(\exp(-\omega(c)), 0)$-asymmetric. Second, it is possible to extend our theorem to the $G(n, p)$ model by showing that there exists a constant $C > 0$, such that for $\frac{C}{n} < p < \frac{1}{n}$, with high probability, $G(n, p)$ is $(\beta, \beta/(240))$-asymmetric for all $\beta \geq \max\{\exp(-\frac{pn}{C}), \frac{1}{n}\}$. Third, when $c \geq 6 \ln n$ (or, when $p \geq \frac{C \ln n}{n}$ in the $G(n, p)$ model), we can let $\beta = \frac{1}{n}$ and obtain that $G_{n,m}$ ($G(n, p)$, respectively) is asymmetric with high probability—a result in the flavor of [81]. Finally, we do not work hard to optimize the constants in the theorem statement; we believe a more careful analysis would bring them down to more civilized numbers, but it is still interesting to explore the limits of these constants.

Now we briefly explain our proof techniques. Let us consider the case where $c$ is a big constant and $\beta = 1$, so that we only need to worry about the permutations without fixed points. We would like to show that, for every such permutation $\pi$,

$$\Pr_{G \sim G_{n,m}}[\pi \text{ is a } \frac{1}{240}\text{-automorphism for } G] \ll \frac{1}{n!},$$

and therefore we can union bound over all such possible permutations. In order to do this, from all $\binom{n}{2}$ possible edges, we construct $\Omega(n^2)$ disjoint pairs of edges $(e, e')$, which we will refer to as “bins”, such that $\pi(e) = e'$. We call a bin “half-full” if exactly one edge in the pair is selected in $G$. It is easy to see that whenever there are more than $\frac{m}{120}$ half-full bins, $\pi$ cannot be a $\frac{1}{240}$-automorphism. At this point, we would like for $\Pr_G[\#\text{half-full bins} < \frac{m}{120}] \ll \frac{1}{n!}$, and this is easy to show. Unfortunately, this method does not work when $\beta = \frac{1}{2}$. To see why, let $\pi$ be a permutation with $\frac{n}{2}$ fixed points. The probability that every edge in $G$ has fixed points of $\pi$ for its endpoints is roughly $2^{-2m} = 2^{-O(n)}$. Therefore we have $\Pr_G[\pi \text{ is an automorphism for } G] \geq 2^{-O(n)}$, and this is not enough for the application of union bound (since there are more than $(n/2)! = 2^{\Omega(n \log n)}$ such permutations). A possible fix to this problem is: we first show that with high probability $(1 - n^{-\omega(1)})$, for every $\pi$ with $\frac{n}{2}$ fixed points, there are many edges of $G$ with at least one end point not fixed by $\pi$; then, conditioned on this event, we show the probability that a fixed $\pi$ is not a $\frac{1}{480}$-automorphism is small enough for the union bound method. The actual proof is more involved, and it is also technically challenging to work with $c$ as large as $\Omega(n)$, and $\beta$ as small as $\frac{1}{n}$.

Finally, for our application to the GRAPHISOMORPHISM problem, we need to extend Theorem 7.1.6 to hypergraphs. More details on robust asymmetry of random hypergraphs can be found in Section 7.6.
7.1.5 Organization

In Section 7.2, we introduce the notations and the SOS/Lasserre hierarchy. In Section 7.3, we describe a reduction from 3-XOR to GI$\text{so}$. The completeness and soundness lemmas for reduction are proved in Section 7.4 and Section 7.5 respectively. In Section 7.6, we prove robust asymmetry property for random graphs and random hypergraphs.

Proofs of the main theorems. Theorem 7.1.2 follows from Theorem 7.2.2, Lemma 7.4.2 and Lemma 7.5.1, by choosing $c = 10^6$. Theorem 7.1.4 follows from Lemma 7.4.1 and Lemma 7.5.1, by choosing $c = \max\{10^6, \Delta\}$.

7.2 Preliminaries

We will be working with undirected graphs and hypergraphs, both of which will be denoted by $G = (V, E)$. Here, an undirected edge $e \in E$ is a set of 2 vertices $\{i, j\}$ for graphs and a set of $k$ vertices $\{i_1, i_2, \ldots, i_k\}$ for $k$-uniform hypergraphs. When $G$ is an directed graph, we use $(i, j)$ to denote a directed edge. We also use the notation $V(G)$ to denote the vertex set of $G$, and $E(G)$ to denote the edge set of $G$.

For any two undirected graphs (or hypergraphs) $G = (V(G), E(G))$ and $H = (V(H), E(H))$ with the same number of vertices, and for any bijection $\pi : V(G) \rightarrow V(H)$, let

$$GIso(G, H; \pi) = \frac{|\{e \in E(G) : \pi(e) \in E(H)\}|}{\max\{|E(G), E(H)|\}},$$

where $\pi(e)$ is the edge obtained by applying $\pi$ on each vertex incident to $e$. Let

$$GIso(G, H) = \max_{\pi : V(G) \rightarrow V(H)} Iso(G, H; \pi).$$

We say that an edge $e \in E(G)$ is satisfied by $\pi$ if $\pi(e) \in E(H)$. We call $\pi$ an $\alpha$-isomorphism for $G$ and $H$ if $GIso(G, H; \pi) \geq \alpha$, and we say $G$ and $H$ are $\alpha$-isomorphic if $GIso(G, H) \geq \alpha$.

For any permutation $\pi : V(G) \rightarrow V(G)$, let

$$AUT(G; \pi) = GIso(G, G; \pi).$$

We say that $\pi$ is an $\alpha$-automorphism for $G$ if $AUT(G; \pi) \geq \alpha$.

Given a permutation $\pi$ over the set $V$, an element $i \in V$ is a fixed point of $\pi$ if $\pi(i) = i$. 118
Definition 7.2.1. A graph (possibly hypergraph) $G$ is $(\beta, \gamma)$-asymmetric if, for any permutation $\pi$ on the vertex set of $G$ that has at most $(1 - \beta)$ fraction of the vertices as fixed points, we have $\text{AUT}(G; \pi) < 1 - \gamma$.

We extend the $G_{n,m}$ random graph model to hypergraphs as follows. Let $G^{(k)}_{n,m}$ be the uniform distribution over all $\left(\begin{array}{c} n \\ m \end{array}\right)$ simple $k$-uniform hypergraphs with $n$ vertices and $m$ edges.

A 3-XOR instance $C$ is a collection of equations $C_1, C_2, \ldots, C_m$ over the variable set $X$. Each equation $C_i$ is of the form $x_{j_1} + x_{j_2} + x_{j_3} = b$ where $x_{j_1}, x_{j_2}, x_{j_3}$ are the variables from $X$, $b \in \mathbb{Z}_2$. Given an assignment $\tau : X \to \mathbb{Z}_2$, let $\text{val}(C; \tau)$ be the fraction of equations in $C$ satisfied by $\tau$. Let $\text{val}(C) = \max_{\tau : X \to \mathbb{Z}_2} \text{val}(C; \tau)$.

7.2.1 SOS/Lasserre hierarchy

One way to formulate the SOS/Lasserre hierarchy is via the pseudo-expectation view. We briefly recall the formulation as follows. More discussion about this view can be found in [34].

We consider the feasibility of a system over $n$ variables $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ with the following constraints: $P_i(x) = 0$ for $i = 1, 2, \ldots, m_P$ and $Q_i(x) \geq 0$ for $j = 1, 2, \ldots, m_Q$, where all the $P_i, Q_i$ polynomials are of degree at most $d$. For $r \geq d$, the degree-$r$ SOS/Lasserre hierarchy finds a pseudo-expectation operator $\tilde{E}[]$ defined on the space of real polynomials of degree at most $r$ over intermediates $x_1, x_2, \ldots, x_n$ such that

- $\tilde{E}[1] = 1$;
- $\tilde{E}[\alpha p + \beta q] = \alpha \tilde{E}[p] + \beta \tilde{E}[q]$ for all real numbers $\alpha, \beta$ and all polynomials $p$ and $q$ of degree at most $r$;
- $\tilde{E}[p^2] \geq 0$ for all polynomials $p$ of degree at most $r/2$;
- $\tilde{E}[P_i \cdot q] = 0$ for all $i = 1, 2, \ldots, m_P$ and all polynomials $q$ such that $P_i \cdot q$ is of degree at most $r$;
- $\tilde{E}[Q_i \cdot p^2] \geq 0$ for all $i = 1, 2, \ldots, m_Q$ and all polynomials $p$ such that $Q_i \cdot p^2$ is of degree at most $r$.

We call any operator $\tilde{E}[]$ a normalized linear operator if it has the first two properties listed above.
**SOS/Lasserre hierarchy for 3-XOR.** Let $C$ be a 3-XOR instance on variable set $\mathcal{X}$. The degree-$r$ SOS/Lasserre hierarchy for the natural integer programming for (the satisfiability of) $C$ is to find a normalized linear pseudo-expectation operator $\tilde{E}[:]$ defined on the space of polynomials of degree at most $r$ over the indeterminates $(A[x \mapsto a])_{x \in \mathcal{X}, a \in \mathbb{Z}_2}$ associated to $C$, such that

1. $\tilde{E}[(A[x \mapsto a]^2 - A[x \mapsto a]) \cdot q] = 0$ for all $x \in \mathcal{X}$, $a \in \mathbb{Z}_2$, and polynomials $q$;
2. $\tilde{E}[(A[x \mapsto 0] + A[x \mapsto 1] - 1) \cdot q] = 0$ for all $x \in \mathcal{X}$ and polynomials $q$;
3. $\tilde{E}[(\sum_{\alpha_C \text{ satisfying } C} A[x_1 \mapsto \alpha_C(x_1)]A[x_2 \mapsto \alpha_C(x_2)]A[x_3 \mapsto \alpha_C(x_3)] - 1) \cdot q] = 0$ for each $C \in \mathcal{C}$ involving variables $x_1, x_2, x_3$ and all polynomials $q$;
4. $\tilde{E}[p^2] \geq 0$ for all polynomials $p$.

We say there is degree-$r$ SOS refutation for the satisfiability of $C$ if the pseudo-expectation operator with properties listed above does not exist.

**SOS/Lasserre hierarchy for GIso.** Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be two graphs such that $|V(G)| = |V(E)|$, $|E(G)| = |E(H)|$. The degree-$r$ SOS/Lasserre hierarchy for the natural integer programming the isomorphism problem between $G$ and $H$ is to find a normalized linear pseudo-expectation operator $\tilde{E}[:]$ on the space of real polynomials of degree at most $r$ over the indeterminates $(\Pi[u \mapsto v])_{u \in V(G), v \in V(H)}$ such that:

a. $\tilde{E}[(\Pi[u \mapsto v]^2 - \Pi[u \mapsto v]) \cdot q] = 0$ for all $u \in V(G)$, $v \in V(H)$, and polynomials $q$;

b. $\tilde{E}[(\sum_{v \in V(H)} \Pi[u \mapsto v] - 1) \cdot q] = 0$ for all $u \in V(G)$ and polynomials $q$; and similarly, $\tilde{E}[(\sum_{u \in V(G)} \Pi[u \mapsto v] - 1) \cdot q] = 0$ for all $v \in V(H)$ and polynomials $q$;

c. $\tilde{E}[(\sum_{(u, u') \in E(G)} \sum_{v, v'} \Pi[u \mapsto v] \Pi[u' \mapsto v'] - |E(G)|) \cdot p^2] \geq 0$ for all polynomials $p$;

d. $\tilde{E}[p^2] \geq 0$ for all polynomials $p$.

We say there is degree-$r$ SOS refutation for the isomorphism between $G$ and $H$ if the pseudo-expectation operator with properties listed above does not exist.

**Remark 7.2.1.** It is equivalent to replace (c) by “$\tilde{E}[(\sum_{v, v'} \Pi[u \mapsto v] \Pi[u' \mapsto v'] - |E(G)|) \cdot p^2] \geq 0$ for all $(u, u') \in E(G)$ and all polynomials $q$.”
7.2.2 Random 3-XOR

A random 3-XOR instance with \( n \) variables and \( m \) equations is sampled by choosing \( m \) unordered 3-tuples of variables from all possible \( \binom{n}{3} \) ones, and making each 3-tuple \((x_{j1}, x_{j2}, x_{j3})\) into a 3-XOR constraint \( x_{j1} + x_{j2} + x_{j3} = b \) with an independent random \( b \in \mathbb{Z}_2 \).

**Theorem 7.2.2.** [204] For every constant \( c > 1 \), there is exists \( \eta > 0 \) such that with probability \( 1 - o(1) \), the satisfiability of a random 3-XOR instance\(^4\) with \( n \) variables and \( cn \) equations cannot be refuted by degree-\((\eta n)\) SOS/Lasserre hierarchy.

7.3 Reduction from 3-XOR to GIso

We define a slight variant of the basic gadget from [55]:

**Definition 7.3.1.** Let \( C \) be a 3-XOR constraint involving variables \( x_1, x_2, x_3 \). The associated gadget graph \( G_C \) consists of: 6 “variable vertices” with names “\( x_i \mapsto a \)” for each \( i \in [3], a \in \mathbb{Z}_2 \); and, 4 “constraint vertices” with names “\( x_1 \mapsto a_1, x_2 \mapsto a_2, x_3 \mapsto a_3 \)” for each partial assignment which satisfies the constraint \( C \). Regarding edges, each pair of variable vertices \( x_i \mapsto 0, x_i \mapsto 1 \) is connected by an edge; the four constraint vertices are connected by a clique; and, each constraint vertex \( \alpha \) is connected to the three variable vertices it is consistent with.

Now we describe how an entire instance of 3-XOR is encoded by a graph:

**Definition 7.3.2.** Let \( C \) be a collection of 3-XOR constraints over variable set \( \mathcal{X} \). We define the associated graph \( G_C \) as follows: For each constraint \( C \in \mathcal{C} \), the graph \( G_C \) contains a copy of the gadget graph \( G_C \). However we identify all of the variable vertices \( x \mapsto a \) across \( x \in \mathcal{X}, a \in \mathbb{Z}_2 \) as well as the variable edges \( (x \mapsto 0, x \mapsto 1) \). The constraint vertices associated to \( C \), on the other hand, are left as-is, and will be named \( \alpha_C \). We denote the set of vertices \( \{x \mapsto 0, x \mapsto 1\} \) by \( V_x \) for every variable \( x \), denote the set of vertices corresponding to \( C \) by \( V_C \) for every variable \( C \).

**Remark 7.3.1.** If \( C \) is a 3-XOR instance with \( n \) vertices and \( m \) constraints then the graph \( G_C \) has \( N = 4m + 2n \) vertices and \( M = 18m + n \) edges.

Finally, we introduce the following notation:

\(^4\)The random 3-XOR distribution used in [204] is slightly different, but the theorem still holds for our distribution.
Notation 7.3.1. Let $C$ be a 3-XOR constraint involving variables $x_i, x_j, x_k$. We write $\overline{C}$ for its homogeneous version, $x_i + x_j + x_k = 0$. Given a collection of 3-XOR constraints $\mathcal{C}$ we write $\overline{\mathcal{C}} = \{\overline{C} : C \in \mathcal{C}\}$.

The reduction. Given a collection of 3-XOR constraints $\mathcal{C}$, the corresponding GISO instance $i (G_{\mathcal{C}}, G_{\overline{\mathcal{C}}})$.

7.4 Completeness

Lemma 7.4.1 (Completeness). If $\mathcal{C}$ is a 3-XOR instance such that $\text{val}(\mathcal{C}) \geq 1 - \epsilon$, then $\text{GIso}(G_{\mathcal{C}}, G_{\overline{\mathcal{C}}}) \geq 1 - 2\epsilon/3$.

Proof. Let $\tau$ be an assignment to the variables in $\mathcal{C}$ such that $\text{val}(\mathcal{I}; \tau) \geq 1 - \epsilon$. Now we define a bijection $\pi$ from the vertices in $G_{\mathcal{C}}$ to the ones in $G_{\overline{\mathcal{C}}}$ as follows.

For each variable vertex $x_j \mapsto b$, let $\pi(x_j \mapsto b) = x_j \mapsto b + \tau(x_j)$. For any equation vertex $\alpha_{C_i}$, if $C_i$ is not satisfied by $\tau$, map it to an arbitrary vertex in $V_{\overline{C_i}}$. If $C_i$ is satisfied by $\tau$, let us suppose that $C_i : x_{j_1} + x_{j_2} + x_{j_3} = b$, let $\alpha'$ be an assignment such that $\alpha'(x_{j_t}) = \alpha(x_{j_t}) + \tau(x_{j_t})$ for all $t \in \{1, 2, 3\}$. Observe that

$$\alpha'(x_{j_1}) + \alpha'(x_{j_2}) + \alpha'(x_{j_3}) = (\alpha(x_{j_1}) + \alpha(x_{j_2}) + \alpha(x_{j_3})) + (\tau(x_{j_1}) + \tau(x_{j_2}) + \tau(x_{j_3})) = b + b = 0.$$

Therefore $\alpha'_{\overline{G}_{\mathcal{C}}}$ is a vertex in $G_{\overline{\mathcal{C}}}$. We let $\pi$ map $\alpha_{C_i}$ to $\alpha'_{\overline{C_i}}$.

It is straightforward to check that all the edges between equation vertices and between variable vertices are satisfied. Now we consider an edge between a equation vertex and a variable vertex, namely between $\alpha_{C_i}$ and $x_j \mapsto b$ where $x_j$ is an variable in equation $C_i$ and $\alpha(x_j) = b$. We show that the edge is satisfied by $\pi$ whenever $C_i$ is satisfied by $\tau$. Let $\alpha'$ and $b'$ be such that $\pi(\alpha_{C_i}) = \alpha'_{\overline{C_i}}$, $\pi(x_j \mapsto b) = x_j \mapsto b'$. Observe that

$$\alpha'(x_j) = \alpha(x_j) + \tau(x_j) = b + \tau(x_j) = b',$$

and this implies that there is an edge between $\alpha'_{\overline{C_i}}$ and $x_j \mapsto b'$.

We see that the only edges in $G_{\mathcal{C}}$ which might not be satisfied by $\pi$ are the ones between equation vertices and variable vertices where the corresponding equation vertex is not satisfied by $\tau$. For each equation not satisfied, there are at most 12 such edges. Therefore there are at most $12\epsilon m$ edges not satisfied. We have

$$\text{GIso}(G_{\mathcal{C}}, G_{\overline{\mathcal{C}}}) \geq \text{GIso}(G_{\mathcal{C}}, G_{\overline{\mathcal{C}}}; \pi) \geq \frac{M - 12\epsilon m}{M} \geq 1 - \frac{2}{3}\epsilon. \quad \square$$

122
7.4.1 SOS completeness

Lemma 7.4.2 (SOS completeness). Let \( C \) be a 3-XOR instance on variable set \( \mathcal{X} \) and suppose that every SOS refutation of \( C \) requires degree exceeding \( r \). Then every SOS refutation of the statement “\( G_C \) and \( G_{\bar{C}} \) are isomorphic” requires degree exceeding \( r/3 \).

Proof. Since \( C \) cannot be refuted in degree \( r \), there is a pseudo-expectation operator \( \tilde{E}_C[\cdot] \) defined on the space of real polynomials of degree at most \( r \) over the indeterminates \( (A[x \mapsto a])_{x \in \mathcal{X}, a \in \mathbb{Z}_2} \) associated to \( C \). This \( \tilde{E}_C[\cdot] \) is normalized, linear, and satisfies:

\[
\begin{align*}
  i & \quad \tilde{E}_C[(A[x \mapsto a]^2 - A[x \mapsto a]) \cdot q] = 0 \text{ for all } x \in \mathcal{X}, a \in \mathbb{Z}_2, \text{ and polynomials } q; \\
  ii & \quad \tilde{E}_C[(A[x \mapsto 0] + A[x \mapsto 1] - 1) \cdot q] = 0 \text{ for all } x \in \mathcal{X} \text{ and polynomials } q; \\
  iii & \quad \tilde{E}_C[(\sum_{\alpha_C \text{ satisfying } C} A[x_1 \mapsto \alpha_C(x_1)]A[x_2 \mapsto \alpha_C(x_2)]A[x_3 \mapsto \alpha_C(x_3)] - 1) \cdot q] = 0 \text{ for each } C \in \mathcal{C} \text{ involving variables } x_1, x_2, x_3 \text{ and all polynomials } q; \\
  iv & \quad \tilde{E}_C[p^2] \geq 0 \text{ for all polynomials } p.
\end{align*}
\]

Our task is to define a normalized linear pseudo-expectation operator \( \tilde{E}_G[\cdot] \) on the space of real polynomials of degree at most \( r/3 \) over the indeterminates \( (\Pi[u \mapsto v])_{u \in V(G_C), v \in V(G_{\bar{C}})} \) such that:

\[
\begin{align*}
  \text{I} & \quad \tilde{E}_G[(\Pi[u \mapsto v]^2 - \Pi[u \mapsto v]) \cdot q] = 0 \text{ for all } u \in V(G_C), v \in V(G_{\bar{C}}), \text{ and polynomials } q; \\
  \text{II} & \quad \tilde{E}_G[(\sum_{v \in V(G_C)} \Pi[u \mapsto v] - 1) \cdot q] = 0 \text{ for all } u \in V(G_C) \text{ and polynomials } q; \text{ and similarly, } \tilde{E}_G[(\sum_{u \in V(G_C)} \Pi[u \mapsto v] - 1) \cdot q] = 0 \text{ for all } v \in V(G_C) \text{ and polynomials } q; \\
  \text{III} & \quad \tilde{E}_G[(\sum_{\{u,u'\} \in E(G_C)} \sum_{v,v' : \{v,v'\} \in E(G_{\bar{C}})} \Pi[u \mapsto v] \Pi[u' \mapsto v'] - M) \cdot p^2] \geq 0 \text{ for all polynomials } p; \\
  \text{IV} & \quad \tilde{E}_G[p^2] \geq 0 \text{ for all polynomials } p.
\end{align*}
\]

Here \( M \) denotes the number of edges in \( G_C \) (and also in \( G_{\bar{C}} \)).

The idea is to formally define each indeterminate \( \Pi[u \mapsto v] \) as a certain degree-3 multilinear polynomial of the indeterminates \( A[x \mapsto a] \). Then \( \tilde{E}_G[\cdot] \) is automatically defined in terms of \( \tilde{E}_C[\cdot] \) for all polynomials of degree at most \( r/3 \). The natural definition is as follows:
1. Let \( x \in \mathcal{X} \) and \( a \in \mathbb{Z}_2 \). We define \( \Pi[(x \mapsto a) \mapsto (x \mapsto b)] = A[x \mapsto (a - b)] \).

2. Let \( C \subseteq \mathcal{C} \), let \( \alpha_C = (x_1 \mapsto a_1, x_2 \mapsto a_2, x_3 \mapsto a_3) \) be constraint vertex in \( G_C \) corresponding to \( C \), and let \( \beta_C = (x_1 \mapsto b_1, x_2 \mapsto b_2, x_3 \mapsto b_3) \) be a constraint vertex in \( G_C \) corresponding to \( C \). We define \( \Pi[\alpha_C \mapsto \beta_C] \) to be the following degree-3 monomial:

\[
A[x_1 \mapsto (a_1 - b_1)]A[x_2 \mapsto (a_2 - b_2)]A[x_3 \mapsto (a_3 - b_3)].
\]

3. All other indeterminates \( \Pi[u \mapsto v] \) are formally defined to be 0.

It is clear that \( \tilde{E}_C[\cdot] \) is normalized and linear by the same property of \( \tilde{E}_C[\cdot] \). It remains to show that the induced pseudo-expectation operator \( \tilde{E}_C[\cdot] \) satisfies (I)-(IV) using the fact that \( \tilde{E}_C[\cdot] \) satisfies (I)-(IV). Most of these are easy; for example, the implication (IV) \( \Rightarrow \) (IV) is immediate. Almost as easy is that (I) \( \Rightarrow \) (II) and that (I), (III) \( \Rightarrow \) (II). We illustrate some of these implications, leaving the rest to the reader. For example, let’s verify (II) for indeterminates of type \( \Pi[\alpha_C \mapsto \beta_C] \). For brevity we’ll write \( \Pi[\alpha_C \mapsto \beta_C] \) as \( A_1A_2A_3 \). Now for any polynomial \( q \) over the \( \Pi \)’s,

\[
\tilde{E}_C([\Pi[\alpha_C \mapsto \beta_C]^2 - \Pi[\alpha_C \mapsto \beta_C]) \cdot q]
= \tilde{E}_C([A_1^2A_2^2A_3^2 - A_1A_2A_3] \cdot q')
= \tilde{E}_C([A_1^2 - A_1]A_2^2A_3^2 \cdot q'] + \tilde{E}_C[A_1(A_2^2 - A_2)A_3^2 \cdot q'] + \tilde{E}_C[A_1A_2(A_3^2 - A_3) \cdot q']
= 0 \quad \text{(by (I).)}
\]

And let’s verify (II) when \( u \) is a variable vertex \( x \mapsto a \):

\[
\tilde{E}_C \left[ \left( \sum_{v \in \mathcal{V}(G_C)} \Pi[(x \mapsto a) \mapsto v] - 1 \right) \cdot q \right]
= \tilde{E}_C[(A[x \mapsto a - 0] + A[x \mapsto a - 1] - 1) \cdot q']
= 0 \quad \text{(by (III))}
\]

The main effort is to establish (III). In fact we will show

\[
\tilde{E}_C \left[ \sum_{(u,v') \in E(G_C)} \Pi[u \mapsto v] \Pi[u' \mapsto v'] - 1 \right] \cdot p^2 = 0 \quad \text{(7.1)}
\]

for all edges \( \{u, u'\} \in E(G_C) \) and all \( p \), whence (III) follows by summing. We will omit the (easy) verification of this for the edges \( x \mapsto 0, x \mapsto 1 \). Instead we will first verify
that (7.1) holds for a typical clique edge associated to constraint $C$, say $(\alpha_C, \alpha'_C)$, on variables $x_1, x_2, x_3$. Only the indeterminates of corresponding constraints, say $\Pi[\alpha_C \mapsto \beta_C]$, are nonzero. Writing $A_i[\alpha - \beta] = A[x_i \mapsto \alpha_C(x_i) - \beta_C(x_i)]$ for brevity (and similarly with primes), the quantity in (7.1) is

\[
\tilde{E}_C \left[ \left( \sum_{\beta_C, \beta'_C \text{ satisfying } C} A_1[\alpha - \beta] A_2[\alpha - \beta] A_3[\alpha - \beta] A_2[\alpha' - \beta'] A_3[\alpha' - \beta'] - 1 \right) \cdot p'^2 \right]
\]

\[
= \tilde{E}_C \left[ \left( \sum_{\beta_C} A_1[\alpha - \beta] A_2[\alpha - \beta] A_3[\alpha - \beta] \right) \left( \sum_{\beta'_C} A_2[\alpha' - \beta'] A_3[\alpha' - \beta'] \right) - 1 \right) \cdot p'^2. \tag{7.2}
\]

Now for fixed $\alpha_C$, as $\beta_C$ ranges over all satisfying assignments to $C$, the assignment $\alpha_C - \beta_C$ ranges over all satisfying assignments to $C$. The analogous statement holds also for $\alpha'_C$. It’s now straightforward to see that the vanishing of (7.2) follows from (iii).

Our final task is to verify (7.1) also for edges between variable vertices and constraint vertices. Fix a typical such edge, say one connecting $x_1 \mapsto a_1$ to $\alpha_C$. (We’ll use the same notation as before for $\alpha_C$; in particular, note that we must have $\alpha_C(x_1) = a_1$.) Now in this case, the quantity in (7.1) is

\[
\tilde{E}_C \left[ \left( \sum_{b \in \mathbb{Z}_2, \beta_C \text{ satisfying } C} A[x_1 \mapsto (a_1 - b)] A_1[\alpha - \beta] A_2[\alpha - \beta] A_3[\alpha - \beta] - 1 \right) \cdot p'^2 \right]
\]

\[
= \tilde{E}_C \left[ \left( \sum_{c \in \mathbb{Z}_2} A[x_1 \mapsto c] \right) \left( \sum_{\beta_C} A_1[\alpha - \beta] A_2[\alpha - \beta] A_3[\alpha - \beta] - 1 \right) \right) \cdot p'^2. \tag{7.3}
\]

Again, the fact that (7.3) vanishes now easily follows from (ii), (iii).

\section*{7.5 Soundness}

In this section, we prove the following soundness lemma.

\begin{lemma}[Soundness] \label{lem:soundness}
Let $C = \{C_1, C_2, \ldots, C_m\}$ be a random 3-XOR instance with $n$ variables and $m = cn$ equations where $c \geq 10^{10}$. With probability $1 - o(1)$, we have

\[
\text{GIso}(G_C, G_C') < 1 - \frac{1}{95c^2}.
\]

\end{lemma}

Before proving \ref{lem:soundness} we first introduce the following definition.
Definition 7.5.2. A graph (possibly hypergraph) \( G \) is \((\epsilon, D)\)-degree bounded if the average degree of every set of \( \epsilon \) fraction of vertices is at most \( D \).

Claim 7.5.3. Suppose \( c \geq 3 \). A random 3-uniform hypergraph \( H \) drawn from \( G^{(3)}_{n,m} \), where \( m = cn \), is \((1/c, 100c)\)-degree bounded with probability \( 1 - o(1) \).

Claim 7.5.4. Given an \((\epsilon, D)\)-degree bounded graph \( G \) with \( n \) vertices, every set of \( \beta n \) vertices has at most \((\epsilon + \beta)Dn\) edges incident to them.

Lemma 7.5.1 is directed implied by the following two lemmas.

Lemma 7.5.5. Let \( H = ([n], E = \{e_i\}) \) be a 3-uniform hypergraph with \( n \) vertices and \( m = cn \) hyperedges. the constraint graph of a 3-XOR instance with \( n \) variables and \( m = cn \) equations. Suppose \( H \) is \((\epsilon, 100c)\)-degree bounded and \((\beta, \gamma)\)-asymmetric, where \( \gamma \geq 200\epsilon \). Let \( C = \{C_1, C_2, \ldots, C_m\} \) be an arbitrary 3-XOR with \( n \) variables and \( m \) constraints based on \( H \). (In other words, each hyperedge \( e_i \) of \( H \) connects the indices of the 3 variables used by \( C_i \).) If we set

\[
\delta := \delta(\epsilon, \beta, \gamma) = \min \left\{ \frac{1}{200}, \frac{\gamma}{48}, \frac{\epsilon}{95c} \right\},
\]

when \( GIso(G_C, G_{\bar{C}}) \geq 1 - \delta \), we have \( val(C) \geq .9 - 100(\epsilon + \beta) \).

Lemma 7.5.6. If \( C \) is a random 3-XOR instance with \( n \) variables and \( m \geq 10000n \) equations, then with probability \( 1 - o(1) \), we have \( val(C) < .51 \).

Proof of Lemma 7.5.1. Set \( \epsilon = \frac{1}{c}, \gamma = \frac{200}{c}, \beta = \frac{48000}{c}. \) Combining Claim 7.5.3, Lemma 7.5.6, and Theorem 7.6.9, we know that with probability \( (1 - o(1)) \), all of the following hold:

1. \( H \) is \((\epsilon, 100c)\)-degree bounded,
2. \( H \) is \((\beta, \gamma)\)-asymmetric, and
3. \( val(C) < .51 \).

Given that these hold, assume for sake of contradiction that \( GIso(G_C, G_{\bar{C}}) \geq 1 - \frac{1}{95c^2} \). Then because \( G \) satisfies Item 1 and Item 2, Lemma 7.5.5 implies that

\[
val(C) \geq .9 - 100 \left( \frac{1}{c} + \frac{48000}{c} \right) \geq .8,
\]

where the last step follows because \( c \geq 10^{10} \). However, this contradicts Item 3. Therefore, \( GIso(G_C, G_{\bar{C}}) < 1 - \frac{1}{95c^2} \) with probability \( 1 - o(1) \).
The proof of Lemma 7.5.6 is standard.

**Proof of Lemma 7.5.6** Fix an assignment to the \( n \) variables. The probability that the assignment satisfies at least \( .51m \) equations of a random 3-XOR instance is at most \( \exp(-.02^2 \cdot .5m/2) = \exp(-.0001m) \) by the Chernoff bound. Since there are only \( 2^n \) assignments, the probability that no assignment satisfies more than \( .5m \) equations is at least \( 1 - \exp(-.0001m) \cdot 2^n = 1 - o(1) \) when \( m \geq 10000 \).

The rest of the section is devoted to the proof of Lemma 7.5.5.

**Proof of Lemma 7.5.5** Let \( \pi \) be a bijection mapping the vertices in \( G_C \) to the vertices in \( G_C' \) such that \( GIso(G_C, G_C'; \pi) \geq 1 - \delta \). We first prove that for most \( i \)'s, \( \pi(V_{C_i}) = V_{C_i'} \) for some \( i' \), and for most \( j \)'s, \( \pi(V_{x_j}) = V_{x_j'} \) for some \( j' \). Formally, let \( A \) be the set of \( i \in [m] \) such that \( \pi(V_{C_i}) = V_{C_i'} \) for some \( i' \), and let \( B \) be the set of \( j \in [n] \) such that \( \pi(V_{x_j}) = V_{x_j'} \) for some \( j' \). We show that

**Claim 7.5.7.** \( |A| \geq (1 - 19\delta)m, |B| \geq (1 - 95c\delta)n. \)

Now we are able to define a permutation \( \sigma \) on the variables in \( C \) (as well as \( C' \) since the set of variables is shared). We let \( \sigma \) to be an arbitrary permutation so that for each \( j \in B \), we have \( \sigma(j) = j' \) where \( \pi(V_{x_j}) = V_{x_j'} \). Now we show that \( \sigma \) is an almost automorphism for the constraint graph (i.e. the hypergraph \( H = ([n], E) \)).

**Claim 7.5.8.** \( AUT(H; \sigma) \geq (1 - 100\epsilon - 24\delta)m. \)

By our setting of \( \delta \), we have \( 24\delta < \gamma/2 \). Since we also assume that \( 100\epsilon < \gamma/2 \), we have \( AUT(G; \sigma) \geq (1 - \gamma)m \), and therefore we know that \( \sigma \) has at least \( (1 - \beta)n \) fixed points.

Now we are ready to define an assignment \( \tau : \{x_j\} \rightarrow \mathbb{Z}_2 \) which certifies that \( \text{val}(I) \geq .9 \). For each \( j \) which is not a fixed point of \( \sigma \), define \( \tau(x_j) \) arbitrarily. For each \( j \) being a fixed point of \( \sigma \), we know that \( \pi(V_{x_j}) = V_{x_j} \). We let \( \tau(x_j) = b \) where \( \pi(x_j \mapsto 0) = x_j \mapsto b \). We conclude the proof by showing the following claim.

**Claim 7.5.9.** \( \text{val}(C; \tau) \geq .9 - 100(\epsilon + \beta). \)
7.5.1 Proof of the claims

**Proof of Claim 7.5.7** Observe that the only 4-cliques in $G_C$ are $V_{C_{i'}}$ ($i' \in [m]$). Therefore, if $\pi(V_{C_i}) \neq V_{C_{i'}}$ for every $i'$, we know that at least one of the edges in the clique $V_{C_i}$ is not satisfied. Therefore we have $m - |A| \leq \delta M$ (recall that $M = 18m + n$ is the number of edges in $G_C$), i.e. $|A| \geq m - \delta M \geq (1 - 19\delta)m$.

We now know that at least $(1 - 19\delta)m \cdot 4$ equation vertices in $G_C$ are mapped from equation vertices in $G_C$. Therefore there are at most $4m - (1 - 19\delta)m \cdot 4 = 76\delta m$ equation vertices in $G_C$ being mapped from variable vertices in $G_C$. In other words, $\pi$ maps at least $2n - 76\delta m = (2 - 76\delta)n$ variable vertices to variable vertices. Let $B'$ be the set of $j$'s such that both vertices in $V_{x_j}$ is mapped to a variable vertex. We have $B' \supseteq B$ and $|B'| \geq (2 - 76\delta)n - n = (1 - 76\delta)n$. For each $j \in B' \setminus B$, we know that the edge in $V_{x_j}$ is not satisfied. Therefore $|B' \setminus B| \leq \delta M$. Therefore, $|B| = |B'| - |B' \setminus B| \geq (1 - 76\delta)n - \delta M \geq (1 - 95\delta)n$. 

**Proof of Claim 7.5.8** Let $E'$ be the set of hyperedges in $E$ whose vertices are all in $B$. Since $G$ is $(\epsilon, 100\epsilon)$-degree bounded, and by our setting of parameters $95\epsilon < \epsilon$, by Claim 7.5.4, we know that $|E'| \geq m - 100\epsilon cn = (1 - 100\epsilon)m$. Now let us consider the hyperedges in $E'' = E' \cap A$. (Also observe that $|E''| \geq (1 - 100\epsilon - 19\delta)m$.) We claim that most of the hyperedges in $E''$ are satisfied by $\pi$. For every hyperedge $e_i = \{j_1, j_2, j_3\} \in E''$ that is not satisfied, we know that $\{\sigma(j_1), \sigma(j_2), \sigma(j_3)\} \notin T$. Since $i \in E'' \subseteq A$, let $i'$ be the equation index such that $\pi(V_{C_i}) = V_{C_{i'}}$. Since $\{\sigma(j_1), \sigma(j_2), \sigma(j_3)\} \notin E$, we have $e_{i'} \neq \{\sigma(j_1), \sigma(j_2), \sigma(j_3)\}$. Let us assume w.l.o.g. that $\sigma(j_1) \notin e_{i'}$. Then there is no edge between $V_{C_{i'}}$ and $V_{x_{\sigma(j_1)}}$ in $G_C$. Therefore the 4 edges between $V_{C_i}$ and $V_{x_{j_1}}$ in $G_C$ are not satisfied.

We have proved that whenever there is an hyperedge in $E''$ not satisfied by $\pi$, there are at least 4 edges in $G_C$ not satisfied by $\pi$. Since $\pi$ satisfies $(1 - \delta)M$ edges, there are at most $\delta M/4$ hyperedges in $E''$ not satisfied by $\pi$. Therefore, we have

\[ \text{AUT}(H; \sigma) \geq |E''| - \delta M/4 \geq (1 - 100\epsilon - 19\delta - 5\delta)m = (1 - 100\epsilon - 24\delta)m. \]

**Proof of Claim 7.5.9** Let $E'$ be the set of hyperedges in $E$ whose vertices are all fixed points of $\sigma$. Since $\sigma$ has at least $(1 - \beta)n$ fixed points, and $H$ is $(\epsilon, 100\epsilon)$-degree bounded, by Claim 7.5.4, we know that $|E'| \geq m - (\epsilon + \beta) \cdot 100cn = (1 - 100(\epsilon + \beta))m$. Now consider any $e_i \in E'$, if all the edges incident to $V_{C_i}$ are satisfied, let $E''$ contain $i$. Since there are at most $\delta M \leq 19\delta m$ edges not satisfied, we know that $|E''| \geq |E'| - 19\delta m \geq (1 - 100(\epsilon + \beta) - 19\delta)m$. We claim that for all $e_i \in E''$, the equation $C_i$ is satisfied. Therefore we have $\text{val}(\mathcal{I}; \tau) \geq 1 - 100(\epsilon + \beta) - 19\delta \geq .9 - 100(\epsilon + \beta)$, since $\delta < 1/200$. 

\[ \text{128} \]
Now we show that $C_i$ is satisfied by $\tau$ when $e_i \in E''$. Using similar argument in the proof of \textbf{Claim 7.5.8}, one can show that when $e_i = \{j_1, j_2, j_3\} \in E''$, we have $\pi(V_{C_i}) = V_{C_i}$. Also we have $\pi(V_{x_{j_t}}) = V_{x_{j_t}}$ for all $t \in \{1, 2, 3\}$, by the definition of $E''$. Let $H$ be the induced subgraph $G_{C, V_{C_i} \cup (\bigcup_{t \in \{1, 2, 3\}}V_{x_{j_t}})}$, let $J$ be the induced subgraph $G_{C, [V_{C_i} \cup (\bigcup_{t \in \{1, 2, 3\}}V_{x_{j_t}})]}$. We use the following claim to conclude the proof.

\textbf{Claim 7.5.10}. If $\pi$ \textit{(after projected on the suitable vertices)} is an isomorphism between $J$ and $J$, $C_i$ is satisfied by $\tau$.

\[
\square
\]

It remains to prove \textbf{Claim 7.5.10}. We first claim the following property about our construction $L(\cdot)$.

\textbf{Claim 7.5.11}. Let $C$ be a 3-XOR instance, let $C : x_{j_1} + x_{j_2} + x_{j_3} = b$ be an equation from $C$. For any $b_1, b_2, b_3 \in \mathbb{Z}_2$, and any vertex $\alpha_C$, the parity of the number of neighbors of $\alpha_C$ in $\{x_{j_1} \mapsto b_1, x_{j_2} \mapsto b_2, x_{j_3} \mapsto b_3\}$ is $b + b_1 + b_2 + b_3$.

Now we are ready to prove \textbf{Claim 7.5.10}.

\textbf{Proof of Claim 7.5.10} Suppose the equation $C_i$ is $x_{j_1} + x_{j_2} + x_{j_3} = b$. Consider $\alpha_{C_i} \in V_{C_i}$, let $\alpha'_{C_i} = \pi(\alpha_{C_i})$. By the construction of $G_C$ and $G_{\underline{C}}$, we know that $\alpha(x_{j_1}) + \alpha(x_{j_2}) + \alpha(x_{j_3}) = b$, and $\alpha'(x_{j_1}) + \alpha'(x_{j_2}) + \alpha'(x_{j_3}) = 0$. Now let the set $A = \{x_{j_1} \mapsto 0, x_{j_2} \mapsto 0, x_{j_3} \mapsto 0\}$. By \textbf{Claim 7.5.11}, we know that the parity of the number of neighbors of $\alpha_{C_i}$ in $A$ is $b + 0 + 0 + 0 = b$. Therefore, by isomorphism, the parity of the number of neighbors of $\alpha'_{C_i}$ in $\pi(A)$ is also $b$. On the other hand, by the definition of $\tau$, we know that $\pi(A) = \{x_{j_1} \mapsto \tau(x_{j_1}), x_{j_2} \mapsto \tau(x_{j_2}), x_{j_3} \mapsto \tau(x_{j_3})\}$. By \textbf{Claim 7.5.11} again, we know that the parity of the number of neighbors of $\alpha'_{C_i}$ in $\pi(A)$ is $0 + \tau(x_{j_1}) + \tau(x_{j_2}) + \tau(x_{j_3})$. Therefore, we have that $\tau(x_{j_1}) + \tau(x_{j_2}) + \tau(x_{j_3}) = b$, i.e. $C_i$ is satisfied by $\tau$. \[
\square
\]

### 7.6 Random graphs are robustly asymmetric

In this section we prove \textbf{Theorem 7.1.6}.

We first set up some definitions. For any graph $G = (V, E)$, let $\pi$ be a permutation over the vertices in $V$, we write $\id(\pi)$ as the number of fixed points in the permutations, that is, $\id(\pi) = |\{v \in V : \pi(v) = v\}|$. We define $\triangle(G, \pi(G)) = \{e : e \in E, \pi(e) \not\in E\} \cup \{e : e \not\in E, \pi(e) \in E\}$. Note that $\text{AUT}(G; \pi) = |E| - \frac{1}{2}|\triangle(G, \pi(G))|$. 

129
For any permutation \( \pi \) over the vertex set \( V \), we define a directed graph \( G_\pi = (\binom{V}{2}, E_\pi) \), and \((e_1, e_2) \in E_\pi \) if and only if \( e_2 = \pi(e_1) \). Since each \( e = \{u, v\} \in \binom{V}{2} \) has in-degree and out-degree exactly 1, we can divide \( G_\pi \) into disjoint unions of directed cycles. We call each directed cycle a bin, and the size of the bin is the number of elements in the cycle.

**Fact 7.6.1.** For any size-1 bins, there are only two situations:

- \( e = \{u, v\} \) where \( u \) and \( v \) are both fixed points of \( \pi \). We call these bins type-1 size-1 bins. The number of type-1 size-1 bins is at most \( \left\lceil \frac{\text{id}(\pi)}{2} \right\rceil \).

- \( e = \{u, v\} \) where \( \pi(u) = v \) and \( \pi(v) = u \). We call these classes type-2 size-1 bins. The number of type-2 size-1 bins is at most \( n - \text{id}(\pi) \) for any permutation \( \pi \).

Now let us consider \( G \sim G_{n,m} \) where \( m = cn \). Let \( Z = \binom{n}{2} \) be the number of possible edges from which we choose \( m \) edges. We first prove the following lemma.

**Lemma 7.6.2.** Let \( A \) be the event that for any permutation \( \pi \) such that \( \text{id}(\pi) = (1 - \beta)n \), the number of the edges in \( G \) that fall into the bins of size \( \geq 2 \) is at least \( \beta m/60 \) and at most \( 2\beta Z/10^9 \). Whenever \( \beta \geq \exp(-c/6) \), we have

\[
\Pr_{G \sim G_{n,m}}[A] = 1 - n^{-\omega(1)}.
\]

By a union bound over all the \( \beta : \beta \geq \beta^* \) (where there are at most \( n \) of them), we get the following corollary.

**Corollary 7.6.3.** Let \( B \) be the event that for every \( \beta \geq \beta^* \), any permutation \( \pi \) such that \( \text{id}(\pi) = (1 - \beta)n \), the number of the edges in \( G \) that fall into the bins of size \( \geq 2 \) is at least \( \beta m/60 \) at most \( 2\beta Z/10^9 \). We have

\[
\Pr_{G \sim G_{n,m}}[B] = 1 - n^{-\omega(1)}.
\]

Before we prove Lemma 7.6.2 we need the following lemma.

**Lemma 7.6.4.** Let \( G \sim G_{n,m} \). Suppose that \( \beta \geq \beta^* \). With probability \( 1 - n^{-\omega(1)} \), for any \( T \subseteq V, |T| = \beta n \), the number of edges incident to \( T \) is at least \( c\beta n/40 \).

Now we prove this lemma.

For any vertex \( v \in V \), its expected degree in \( G_{n,m} \) is \( 2c \). We would like to prove that the probability that the degree is at most \( c/10 \) is very low. Indeed, we claim a more general statement.
Claim 7.6.5. Let $W$ be a set of $w$ possible edges from $\binom{V}{2}$, where $\lfloor (n-1)/2 \rfloor \leq w \leq n$,
\[ \Pr_{G=(V,E) \sim \mathcal{G}_{n,m}} \left[ |E \cap W| \leq c/10 \right] \leq \exp(-c/2). \]

Observe that when $W$ is the set of possible edges incident to $v$, Claim 7.6.5 says that $\Pr[\deg(v) \leq c/10] \leq \exp(-c/3)$.

For each possible edge $e$, we define a random variable $X_e$ as the indicator variable for the event that $e$ is selected as an edge in $G$. Claim 7.6.5 would be a direct application of Chernoff bound if the $X_e$ variables were independent. However, the following claim states that Chernoff bound still holds since the variables are negatively associated. (Please refer to e.g. [126] for the definition of negatively associated random variables.)

Claim 7.6.6. The $Z$ variables $X_e$ are negatively associated.

Proof. Since $\{X_e\}$ follows the permutation distribution over $m$ 1’s and $(Z - m)$ 0’s, by Theorem 2.11 in [126], the claim holds. \hfill \square

Proof of Claim 7.6.5 Since $\mathbb{E}[X_e] = 2c/(n-1)$ for every $e$, we have $\mathbb{E}[|E \cap W|] = \sum_{e \in W} \mathbb{E}[X_e] = 2c|W|/(n-1) \geq c$. The claim follows by Chernoff bound for negatively associated variables. \hfill \square

Similarly we show that

Claim 7.6.7. Let $W$ be a set of possible edges, when $c \leq n/10^9$,
\[ \Pr_{G=(V,E) \sim \mathcal{G}_{n,m}} \left[ |E \cap W| \geq |W|/10^9 \right] \leq \exp(-3|W|/10^{10}). \]

Proof. Since $\mathbb{E}[X_e] = 2c/(n-1)$ for every $e$, we have $\mathbb{E}[|E \cap W|] = \sum_{e \in W} \mathbb{E}[X_e] = 2c|W|/(n-1) < 3|W|/10^{10}$. The claim follows by Chernoff bound for negatively associated variables. \hfill \square

Now we are ready to prove Lemma 7.6.4.

Proof of Lemma 7.6.4 Suppose the vertices of the random graph $G \sim \mathcal{G}_{n,m}$ are numbered from 1 to $n$. Let $X_i = \sum_{j=i+1}^{i+\lfloor (n-1)/2 \rfloor} X_e = (i,(j-1) \mod n+1)$. By Claim 7.6.5 we know that $\Pr[X_i \leq c/10] \leq \exp(-c/2)$. 

131
Since the random variables \{X_i\} are sums of disjoint sets of negatively associated random variables \(X_e\)'s, we know that the \(X_i\)'s are also negatively associated. Let \(U\) be the set of vertices \(i\) such that \(X_i \leq c/10\). We have \(\mathbb{E}[|U|] \leq n \cdot \exp(-c/2)\). Using Chernoff bound for negatively associated random variables, we have

\[
\Pr\left[|U| \geq \frac{1}{2} \beta n\right] \leq \exp\left(-\frac{1}{3} \cdot \left(\frac{\beta n}{2 \cdot \mathbb{E}[|U|]} - 1\right)^2 \mathbb{E}[|U|]\right) = \exp\left(-\frac{1}{3} \cdot \left(\frac{\beta n}{2 \cdot \mathbb{E}[|U|]} - 1\right) \left(\frac{\beta n}{2} - \mathbb{E}[|U|]\right)\right).
\]

Using \(\beta \geq \exp(-c/6)\) and \(c \geq 10\), we have \(\beta n/2 - \mathbb{E}[|U|] \geq \beta n/4\). Therefore,

\[
\Pr\left[|U| \geq \frac{1}{2} \beta n\right] \leq \exp\left(-\frac{1}{3} \cdot \left(\frac{\beta n}{2} - \exp(c/2) - 1\right) \cdot \frac{\beta n}{4}\right) \leq \exp\left(-\frac{1}{3} \cdot \frac{1}{4} \cdot \exp(c/6) \cdot \frac{\beta n}{4}\right) \leq \exp\left(-\frac{n}{48}\right) = n^{-\omega(1)},
\]

where the second and fourth inequalities are because of \(\beta \geq \exp(-c/6)\), and the third inequality is because of \(\exp(c/3)/2 - 1 \geq \exp(c/6)/4\) for \(c \geq 10\).

Therefore, with probability \(1 - n^{-\omega(1)}\), there are at most \(\beta n/2\) vertices with degree at most \(c/10\) (since \(\deg(i) \geq X_i\) for every vertex \(i\)). When this happens, for any \(T \subseteq V\), \(|T| = \beta \cdot n\), there are at least \((|T| - \beta n/2\) vertices in \(T\) with degree at least \(c/10\), the sum of degrees of vertices in \(T\) is at least \((|T| - \beta n/2 \cdot c/10 = c\beta n/20\), which means the number of edges incident to any vertex in \(T\) is at least \(c\beta n/40\).

**Proof of Lemma 7.6.2** By Lemma 7.6.4, we know that with probability \((1 - n^{-\omega(1)})\), the number of edges in \(G\) that is incident to \(T\) is at least \(c\beta n/40\), for every \(T \subseteq V\) and \(|T| = \beta n\). Therefore, for any \(\pi\) with \(\text{id}(\pi) = (1 - \beta)n\), let \(T^*\) be the non-fixed points of \(\pi\). We have \(|T^*| = \beta n\). As the number of edges in size-1 bins which are incident to \(T\) is at most \(|T^*|/2 = \beta n/2\), the number of selected edges that in bins of size \(\geq 2\) is at least \(c\beta n/40 - \beta n/2 \geq \beta m/60\), when \(c \geq 100\).

We also need to show that with probability \((1 - n^{-\omega(1)})\), the number of selected edges in bins of size \(\geq 2\) (denote this number by random variable \(X\)) is at most \(2\beta^*Z/10^9\). For every \(\pi\) such that \(\text{id}(\pi) = (1 - \beta)n\), let \(W\) be the set of possible edges whose end vertices are not both fixed point of \(\pi\). We have \(|W| = Z - (1 - \beta)^m\), therefore \(\beta Z \leq W \leq 2\beta Z\) (for large enough \(n\)) and \(X \leq |E \cap W|\). By Claim 7.6.7 we have \(\Pr[|E \cap W| \geq |W|/10^9] \leq
Lemma 7.6.8. Conditioned on event $\mathcal{B}$, for $10^4 \leq c \leq n/10^7$, $\beta_0 \geq \beta^*$, with probability $(1 - n^{-17})$, $G$ is $(\beta_0, \beta_0/240)$-asymmetric.

Proof. For any permutation $\pi$, we define a set of more fine-grained bins. We start with the bins we defined before, and split the bins of size $\geq 4$ into bins of size $2$ and at most one bin of size $3$ as follows. Suppose the original bin contains $\{e_1, e_2, \ldots, e_l\}$, where $\pi(e_i) = e_{i+1}$ and $\pi(e_l) = e_1$, $l \geq 4$, the we have new bins which contains $\{e_1, e_2\}, \ldots, \{e_{l-1}, e_l\}$ if $l$ is even, and $\{e_1, e_2\}, \ldots, \{e_{l-4}, e_{l-3}\}, \{e_{l-2}, e_{l-1}, e_1\}$ if $l$ is odd.

For each bin of size $2$ and size $3$, if all the edges are in $G$, we call it a full bin; if none of them are in $G$, we call it an empty bin; otherwise, we call it a half-full bin. Fix a permutation $\pi$, let $s_\pi$ be the number of half-full bins. For each half-full bin, it contributes at least one to $\Delta(G, \pi(G))$, therefore $s_\pi \leq |\Delta(G, \pi(G))| = 2(m - \text{AUT}(G, \pi))$. We have

$$
\Pr[G \text{ is } (\beta_0, \beta_0/240)-\text{asymmetric}|\mathcal{B}] = \Pr[\forall \pi : \text{id}(\pi) \leq (1 - \beta_0)n, \text{AUT}(G, \pi) \leq (1 - \beta_0/240)m|\mathcal{B}] \\
\geq \Pr[\forall \pi : \text{id}(\pi) \leq (1 - \beta_0)n, s_\pi \geq \beta_0m/120|\mathcal{B}].
$$

Now we turn to show that

$$
\Pr[\forall \pi : \text{id}(\pi) \leq (1 - \beta_0)n, s_\pi \geq \beta_0m/120|\mathcal{B}] \geq 1 - n^{-17}.
$$

To show this, we only have to prove for every $\beta$ where $\beta \geq \beta_0$,

$$
\Pr[\forall \pi : \text{id}(\pi) = (1 - \beta)n, s_\pi \geq \beta m/120|\mathcal{B}] \geq 1 - n^{-18}, \quad (7.4)
$$

and take a union bound over (at most $n$ possible) $\beta$’s.

Fix $\beta : \beta \geq \beta_0$ and fix a permutation $\pi$ such that $\text{id}(\pi) = (1 - \beta)n$. Let $\mathcal{C}_t$ be the event that $\mathcal{B}$ happens and there are $t$ edges in $G$ fall into the bins of size $2$ and $3$. Since $\mathcal{B}$ is a disjoint union of $\mathcal{C}_t$ for all $t : \beta m/60 \leq t \leq 2\beta Z/10^9$, we have

$$
\Pr[s_\pi \leq \beta m/120|\mathcal{B}] = \sum_{t = \beta m/60}^{m} \Pr[s_\pi \leq \beta m/120|\mathcal{C}_t] \cdot \Pr[\mathcal{C}_t|\mathcal{B}].
$$

We will prove that

$$
\Pr[s_\pi \leq \beta m/120|\mathcal{C}_t] = \left(\left(\frac{n}{\beta n}\right)(\beta n)!\right)^{-1} \cdot n^{-18}, \quad (7.5)
$$
and by taking a union bound over all \( \binom{n}{bn} \beta n! \) possible \( \pi \)'s, we prove (7.4).

Let \( \gamma = \beta/120 \). Let \( L \) be the number of possible edges in bins of size 2 and 3. We have \( L = Z - (1 - \beta)n - \frac{\beta n}{2} \geq \beta n^2/4 \) (for large enough \( n \)). Together with \( t \leq 2\beta Z/10^9 \), we have \( t \leq 4L/10^9 \leq L/10^8 \). Let \( B \) be the number of bins of size 2 and 3. Fix \( t \) such that \( \beta m/60 \leq t \leq L/10^8 \). Conditioned on \( C_t \), the \( \binom{B_t}{i} \) ways to select these \( t \) edges are uniformly distributed. Now we compute the number of ways such that there are at most \( 2\gamma m \) half-full bins. Suppose that there are \( i \) half-full bins (for \( i \leq 2\gamma m \leq t/2 \)). There are \( \binom{B_t}{i} \) ways to choose these bins. There are at most \( \binom{B_t}{(t-i)/2} \) ways to choose the full bins (since \( t/2 < L/2 \cdot 10^8 < B/2 \)). For each half-full bin, there are at most 6 ways to choose the edges in the bin. Therefore,

\[
\Pr[s_\pi = i | C_t] \leq 6^i \binom{B_t}{i} \binom{B_t}{t-i} \left( \frac{L}{t} \right)^{-1} \leq 6^i \cdot \frac{L^{t-i}}{i!} \left( \frac{t}{L} \right)^t = 6^i L^{-\frac{t-i}{i}} \frac{t^t}{i!} \left( \frac{t-i}{2} \right)!.
\]

Since \( i! \geq (i/e)^i \), we have

\[
(7.6) \leq 6^i L^{-\frac{t-i}{i}} e^{\frac{t^t}{i}} \cdot \frac{t^t}{i!} \left( \frac{t-i}{2} \right)! \leq 6^i L^{-\frac{t-i}{2}} (2e)^{\frac{t^t}{i}} \cdot \frac{t^t}{i! (t-i)^{\frac{t-i}{2}}}.
\]

Using \( i^t (t-i)^{\frac{t-i}{2}} \geq \left( \frac{t}{4} \right)^{t-i} \), we have

\[
(7.7) \leq 6^i (8e)^{\frac{t^t}{2}} \left( \frac{t}{L} \right)^{\frac{t-i}{2}} \leq (48e)^{\frac{t^t}{2}} \left( \frac{t}{L} \right)^{\frac{t}{2}} \leq \left( \frac{10^7 t}{L} \right)^{\frac{t}{4}}.
\]

Since \( \beta m/60 \leq t \leq L/10^8 \), and \( (10^7 t/L)^{t/4} \) is monotonically decreasing when \( t < L/(10^7 e) \), we have

\[
(7.8) \leq \left( \frac{10^7 \beta m}{60L} \right)^{\frac{\beta m}{240}} \leq \left( \frac{10^7 \beta cn}{60\beta n^2/4} \right)^{\frac{\beta cn}{240}} \leq \left( \frac{10^6 c}{n} \right)^{\frac{\beta cn}{240}} \leq \left( \frac{10n^4}{1} \right)^{\frac{\beta cn}{240}} \leq \left( \frac{1}{n} \right)^{30\beta n},
\]

where the second last inequality is because \( 10^4 \leq c \leq n/10^7 \) and \( (10^6 c/n)^{\beta cn/240} \) is monotonically decreasing in this range, and the last inequality is for large enough \( n \). Observing that \( \binom{n}{bn} \beta n! \leq n^{2\beta n} \) and \( \beta \geq \beta^* \geq 1/n \), we proved that

\[
\Pr[s_\pi = i | C_t] \leq \left( \binom{n}{\beta n} (\beta n)! \right)^{-1} \cdot n^{-20}.
\]

By taking a union bound over all (at most \( n^2 \) many) \( i \)'s such that \( i \leq 2\gamma m \), we prove (7.5).

Theorem 7.1.6 is proved by combining Corollary 7.6.3, Lemma 7.6.8 and taking a union bound over all possible \( \beta = \beta_0 \) (where there are at most \( n \) of them).
7.6.1 Generalization to hypergraphs

In this subsection, we generalize Theorem 7.1.6 to random \( k \)-uniform hypergraphs for any constant \( k \geq 3 \).

**Theorem 7.6.9.** For any constant \( k \), there exists constant \( \kappa_k \), such that for \( m = cn \) where \( \kappa_k \leq c \leq \binom{n}{k}/\kappa_k^3 \), \( n \) large enough, if we set \( \beta^* = \max\{\exp(-c/6), 1/n\} \), with probability \( (1 - n^{-15}) \), for all \( \beta : \beta^* \leq \beta \leq 1 \), a random graph \( H \) from the distribution \( G_{n,m}^{(k)} \) is \((\beta, \beta/240)\)-asymmetric. For \( k = 3 \), \( \kappa_3 = 10^4 \) suffices.

The proof of Theorem 7.6.9 mostly follows the lines of the proof of Theorem 7.1.6. But we need some small modifications. For simplicity, we only prove the theorem for \( k = 3 \). For higher \( k \), we encourage the readers to check by themselves.

Now we work with \( k = 3 \). For any permutation \( \pi \) over the vertex set \( V \), we define a directed graph \( G^{(3)}_{\pi} = ((V)_3, E_{\pi}) \), and \((e_1, e_2) \in E_{\pi} \) if and only if \( e_2 = \pi(e_1) \). Since each \( e \in (V)_3 \) has in-degree and out-degree exactly 1, we can divide \( G^{(3)}_{\pi} \) into disjoint unions of directed cycles. Similarly as in the ordinary graph case, we call each directed cycle a bin, and the size of the bin is the number of elements in the cycle. Let \( Z_3 = \binom{n}{3} \).

**Fact 7.6.10.** For any size-1 bins, there are only three situations:

- \( e = \{u, v, w\} \) where \( u, v \) and \( w \) are all fixed points of \( \pi \). We call these bins type-1 size-1 bins. The number of type-1 size-1 bins is at most \( \binom{\text{id}(\pi)}{3} \).

- \( e = \{u, v, w\} \) where one of them is a fixed point of \( \pi \) and the other two map to each other under \( \pi \). We call these classes type-2 size-1 bins. The number of type-2 size-1 bins is at most \( \text{id}(\pi) \cdot n^{\text{id}(\pi)} - \binom{\text{id}(\pi)}{2} = O(n^2) \) for any permutation \( \pi \).

- \( e = \{u, v, w\} \) where \( \pi(u) = v \) and \( \pi(v) = w \) and \( \pi(w) = u \). We call these classes type-3 size-1 bins. The number of type-3 size-1 bins is at most \( \frac{n^{\text{id}(\pi)}}{3} \) for any permutation \( \pi \).

The following lemma is an analogue of Lemma 7.6.2.

**Lemma 7.6.11.** For any fixed \( \beta \) where \( \beta \geq \beta^* \), let \( \mathcal{D}' \) be the event that for any permutation \( \pi \) such that \( \text{id}(\pi) = (1 - \beta)n \), the number of the hyperedges in \( H \) that fall into the bins of size \( \geq 2 \) is at least \( \beta m/60 \) and at most \( 2\beta Z_3/10^9 \). We have

\[
\Pr_{H \sim H_{n,m}^{(3)}}[\mathcal{D}'] = 1 - n^{-\omega(1)}.
\]
By a union bound over all the $\beta : \beta \geq \beta^*$ (there are at most $n$ of them), we get the following corollary.

**Corollary 7.6.12.** Let $D$ be the event that for every $\beta \geq \beta^*$, any permutation $\pi$ such that $\text{id}(\pi) = (1 - \beta)n$, the number of the hyperedges in $H$ that fall into the bins of size $\geq 2$ is at least $\beta m/60$ at at most $2\beta Z_3/10^9$. We have:

$$\Pr_{H \sim \mathcal{H}_{n,m}^{(3)}} [D] = 1 - n^{-\omega(1)}.$$ 

The proof of [Lemma 7.6.11](#) is similar to that of [Lemma 7.6.2](#), except that now we also need to take care of type-2 size-1 bins.

**Lemma 7.6.13.** With probability $1 - n^{-\omega(1)}$, for any permutation $\pi$ with $\text{id}(\pi) \leq (1 - \beta^*)n$, the number of selected hyperedges in type-2 size-1 bins is at most $c\beta n/1000$.

**Proof.** We first prove that for every $\beta \geq \beta^*$, with probability $1 - n^{-\omega(1)}$, for every permutation $\pi$ with $\text{id}(\pi) = (1 - \beta)n$, the number of selected hyperedges in type-2 size-1 bins is at most $c\beta n/1000$. By a union bound over all possible $\beta$’s (where there are at most $n$ of them), we get the desired statement.

For any fixed permutation $\pi$ with $\text{id}(\pi) = (1 - \beta)n$, the number of type-2 size-1 bins is at most $(1 - \beta)n\beta n/2 \leq \beta n^2/2$. For each possible hyperedge $e$, we define the random variable $X_e$ as the indicator variable for the event that $e$ is selected as an hyperedge in $H$. Note that $E[X_e] = cn/Z_3 \leq 7c/n^2$. Define random variable $X = \sum_{e \text{ in type-2 size-1 bin}} X_e$ as the number of selected hyperedges in type-2 size-1 bins, by linearity of expectation,

$$E[X] \leq \frac{\beta n^2}{2} \cdot \frac{7c}{n^2} < 4c\beta.$$ 

On the other hand, we can also show that all these random variables are negative associated, therefore through Chernoff bound for negative associated random variables, we have

$$\Pr_{H \sim \mathcal{G}_{n,m}^{(3)}} [X \geq \beta cn/1000] \leq \exp(-1/3 \cdot (n/250 - 1)^2 \cdot 4c \cdot \beta) \leq \exp(-c\beta n^2/10^5)$$

By a union bound over at most $n!/(\beta n)! \leq n^{2\beta n}$ such permutations, the probability that there exists $\pi$ with $\text{id}(\pi) = (1 - \beta)n$ such that the number of type-2 size-1 bins is more than $c\beta n$, is at most

$$\exp(-c\beta n^2/10^5) \cdot n^{2\beta n} = \exp(-c\beta n^2/(10^5) + 2\beta n \log n) \leq \exp(-c\beta n^2/(10^6)) \leq n^{-\omega(1)}.$$ 

\[\square\]
The following lemma is an analogue of Lemma 7.6.4 and the proof is almost identical.

**Lemma 7.6.14.** Let $H \sim G_{n,m}^{(3)}$. Suppose that $\beta \geq \beta^*$. With probability $1 - n^{-\omega(1)}$, for any $T \subseteq V$, $|T| = \beta n$, the number of hyperedges incident to $T$ is at least $c\beta n/40$.

**Proof of Lemma 7.6.14**. We only establish the lower bound $(\beta m/60)$. The proof for upper bound is almost identical to that in Lemma 7.6.2.

Let $T$ be the set of non-fixed points of $\pi$, then $|T| = \beta n$. By Lemma 7.6.14, we know that with probability $(1 - n^{-\omega(1)})$, there are at least $c\beta n/40$ hyperedges incident to $T$ – all these edges are either in bins of size $\geq 2$, or edges in size-1 bins of type-2 or type-3. By Lemma 7.6.13, we know that with probability $1 - n^{\omega(1)}$ the number of selected hyperedges that fall into type-2 size-1 bins is at most $c\beta n/1000$. Finally we recall that there are at most $n - \text{id}(\pi)/3 = \beta n/3$ type-3 size-1 bins.

Therefore, with probability $(1 - n^{-\omega(1)})$, the number of selected hyperedges that fall into bins of size $\geq 2$ is at least $c\beta n/40 - \beta n/3 - c\beta n/1000 \geq c\beta n/60$. \(\square\)

Finally we state the following analogue of Lemma 7.6.8 (whose proof is also almost identical).

**Lemma 7.6.15.** Conditioned on event $D$, for $10^4 \leq c \leq n/10^7$, $\beta \geq \beta^*$, with probability $(1 - n^{-17})$, $H \sim G_{n,m}^{(3)}$ is $(\beta, \beta/240)$-asymmetric.

The $k = 3$ case in Theorem 7.6.9 follows from Corollary 7.6.12 and Lemma 7.6.15.
Part II

A proof complexity view of the Parrilo–Lasserre hierarchy and the success of Lasserre on hard instances for weaker hierarchies
Chapter 8

Introduction and SOS preliminaries

8.1 Introduction

In a typical constraint satisfaction problem (CSP) we are given a set of variables $V$ to be assigned values from some finite domain $\Omega$ (often $\{0, 1\}$); we are also given a set of local constraints specifying how various small groups of variables should be assigned. The task is to find an assignment to the variables which minimizes the number of unsatisfied constraints. Sometimes there may also be inviolable global constraints; for example, that no domain element is assigned to too many variables. A canonical example is the $\frac{1}{3}$ vs. $\frac{2}{3}$ BALANCEDSEPARATOR problem as we defined earlier in this thesis: given is a graph $(V, E)$ with $n$ vertices which must be partitioned into two “balanced” parts, each of cardinality at least $n/3$; the goal is to minimize the number of edges crossing the cut. Throughout this chapter, we will also call this problem BALANCEDSEPARATOR.

For such problems, certifying that there is a good solution is in NP; for example, given a graph we can efficiently prove that it has a balanced cut of size at most $\alpha$ simply by exhibiting the cut. But what about the opposite problem, certifying that every balanced cut has size at least $\beta$? Since this problem is coNP-complete it is unlikely that there are efficient certifications for every instance; however there may be efficient certifications for specific instances or classes of instances. For example, if we consider a linear programming relaxation of a given BALANCEDSEPARATOR instance and then exhibit a dual solution of value $\beta$, this constitutes a proof that every balanced cut in the instance has size at least $\beta$.

The question is also interesting for problems in P, especially when the complexity of the proof system is taken into account. For example, given an unsatisfiable instance
Ax = b of the 3Lin2 CSP (meaning the equations are over \(\mathbb{F}_2\) and each involves at most 3 variables), there is always an easy-to-verify proof of unsatisfiability: a vector \(y\) such that \(y^\top A = 0\) but \(y^\top b \neq 0\). However finding such a proof requires a rather specialized algorithm, Gaussian Elimination. By contrast, unsatisfiable instances of the 2Lin2 CSP have simple proofs of unsatisfiability (an unsatisfiable “cycle” of variables) which can be found by a very generic “local consistency” algorithm. Indeed, one can view this algorithm as searching for all constant-width Resolution proofs of unsatisfiability; the same algorithm works for any “bounded-width CSP” [26].

**Positivstellensatz proofs.** In this chapter we consider a certain strong proof system for CSPs. It belongs to the well-studied class of algebraic proof systems, in which local constraints are represented by polynomial equations. To handle global constraints we also allow for polynomial inequalities; this is also natural in the context of the linear programs and semidefinite programs used by optimization algorithms. To give an example, suppose we have a BALANCEDSEPARATOR instance \((V,E)\) with \(V = [n]\). We introduce a real variable \(X_i\) for each \(i \in V\). Now to say that the optimum value of the instance is larger than \(\beta\) is precisely equivalent to saying the following system of polynomial equations and inequalities (each of degree at most 2) is infeasible:

\[
A = \{ X_i^2 = X_i \forall i \in [n] \} \cup \left\{ \sum_{i=1}^{n} X_i \geq n/3, \sum_{i=1}^{n} X_i \leq 2n/3 \right\} \cup \left\{ \sum_{(i,j) \in E} (X_i - X_j)^2 \leq \beta \right\}.
\]

Here the first set of equations enforces \(X_i \in \{0, 1\}\), encoding a cut. The second set of inequalities enforces that the cut is balanced. The final inequality states that at most \(\beta\) edges cross the cut. Now what would constitute a proof that \(A\) has no real solutions; i.e., that the BALANCEDSEPARATOR value exceeds \(\beta\)? One certificate would be a formal identity in the polynomial ring \(\mathbb{R}[X_1, \ldots, X_n]\) of the following form:

\[
-1 = \sum_{i=1}^{n} P_i(X_i^2 - X_i) + U\left(\sum_{i=1}^{n} X_i - n/3\right) + U'\left(2n/3 - \sum_{i=1}^{n} X_i\right) + V \cdot (\beta - \sum_{(i,j) \in E} (X_i - X_j)^2) + W,
\]

where \(P_1, \ldots, P_n \in \mathbb{R}[X_1, \ldots, X_n]\) and where \(U, U', V, W \in \mathbb{R}[X_1, \ldots, X_n]\) are each sums of squares (SOS), meaning of the form \(Q_1^2 + Q_2^2 + \cdots + Q_m^2\) for some \(Q_1, \ldots, Q_m \in \mathbb{R}[X_1, \ldots, X_n]\). Such an identity would indeed imply that \(A\) is infeasible, since substituting any solution of \(A\) into (8.1) would give a nonnegative right-hand side.

In fact, a certain refinement [187] of the Positivstellensatz of Krivine [151] and Stengle [215] guarantees that if \(A\) is infeasible then there is always a proof of the form (8.1). A generic “SOS proof system” based on the Positivstellensatz was introduced around 1999
by Grigoriev and Vorobjov [101]. As with most algebraic proof systems it can be difficult
to place an a priori upper bound on the degree of the polynomials needed for a proof; if we
insist on a fixed degree bound $d$ then the proof system becomes incomplete. On the other
hand this incomplete system has the advantage of being efficiently automatizable, meaning
that if a proof exists it can be found in time $\text{poly}(n^d)$. The algorithm uses semidefinite pro-
gramming and follows from the work of Shor [210], Nesterov [177], Lasserre [156, 157]
and Parrilo [185]. See Section 8.1.1 for more details.

The power of SOS. Most of the previous relevant work focused on showing SOS-degree
(equivalently, Lasserre-round) lower bounds. However, in this thesis, we bring to light the
importance of SOS degree upper bounds for the study CSP approximability. We consider
the strong integrality gap instances known for the notorious UNIQUEGAMESCSP [192,
[142, 35] and will (essentially) show that degree-8 SOS proofs can certify that the instances
have value close to 0. Thus the generic $\text{poly}(n)$-time “level-4 Lasserre SDP” algorithm
refutes their having large optimal value. This is despite the fact that the instances still
have value near 1 after $\Theta(\log \log n)^{1/4}$ rounds of the rather powerful Sherali–Adams SDP
hierarchy [192].

We will also further explore the relevance of SOS proof complexity to the algorithmic
theory of CSP approximation. Specifically, we show that the Devanur–Khot-Saket–
Vishnoi [79] instances of BALANCEDSEPARATOR can have their optimal value well-
certified by a degree-4 SOS proof. We also investigate the problem of SOS proofs for
the Khot–Vishnoi (KV) [144] instances of MAXCUT.

8.1.1 History

We review here some of what is known about SOS proofs and SDP hierarchies; for a much
more thorough discussion we recommend the monograph by Laurent [162].

Throughout this work we write $X = (X_1, \ldots, X_n)$ for a sequence of indeterminates,
with the number $n$ being clear from context. We say that the real multivariate polynomial
$u \in \mathbb{R}[X]$ is sum of squares (SOS) if $u = s_1^2 + \cdots + s_m^2$ for some $s_1, \ldots, s_m \in \mathbb{R}[X]$.
Any SOS polynomial is nonnegative on all of $\mathbb{R}^n$; however, as Hilbert [117] showed in
1888 there exist nonnegative polynomials which are not SOS. The first explicit example,
$X_1^2X_2^2(X_1^2 + X_2^2 − 3) + 1$, was given by Motzkin in the mid-'60s. Hilbert’s 17th
Problem [118] asks whether every nonnegative polynomial $q$ is the quotient of SOS polyno-
mials; this was solved affirmatively by Artin [23].

Artin’s result also follows from the Positivstellensatz, first proved (essentially) by Kriv-
The duality theory of linear programming. We state a special case appearing in [45]:

**Positivstellensatz.** Let \( A \) be a finite set of real multivariate polynomial equations and inequalities,

\[
A = \{ p_1 = 0, p_2 = 0, \ldots, p_m = 0 \} \cup \{ q_1 \geq 0, q_2 \geq 0, \ldots, q_{m'} \geq 0 \},
\]

with each \( p_i, q_j \in \mathbb{R}[X] \). Then \( A \) is infeasible if and only if there exist polynomials \( r_1, \ldots, r_m \) and SOS polynomials \( (u_J)_{J \subseteq [m']} \) in \( \mathbb{R}[X] \) such that

\[
-1 = \sum_{i=1}^m r_i p_i + \sum_{J \subseteq [m']} u_J \prod_{j \in J} q_j.
\]

One interesting further special case occurs when \( A \) contains only equations, not inequalities. In this case the Positivstellensatz says that \( p_1, \ldots, p_m \) have no common real roots if and only if the ideal they generate contains \( 1 + u \) for some SOS \( u \). This special case arises whenever one wants to show that a CSP (with no global constraints) is not perfectly satisfiable. (As noted by Shor [209], one can actually reduce to this case in general by replacing \( q \geq 0 \) with \( q - Y^2 = 0 \), where \( Y \) is a new indeterminate; indeed, by further substitutions of new indeterminates one can reduce to the case where all equations are quadratic.)

**Proof complexity.** Extending the Nullstellensatz proof system of Beame, Impagliazzo, Krajíček, Pitassi, and Pudlák [39], Grigoriev and Vorobjov [102] proposed in 1999 the natural propositional proof system based on the Positivstellensatz. The complexity measure is degree: i.e., \( \max_{i,J} \{ r_ip_i, u_J \prod_{j \in J} q_j \} \) in (8.2). This is a static proof system, meaning that one simply exhibits the refutation (8.2). Grigoriev and Vorobjov showed that refuting the single equation

\[
(1 - X_0X_1)^2 + (X_1^2 - X_2)^2 + (X_2^2 - X_3)^2 + \cdots + (X_{n-1}^2 - X_n)^2 + X_n^2 = 0
\]

requires a proof of degree at least \( 2^{n-1} \). Relying on some ideas from the work of Buss, Grigoriev, Impagliazzo, and Pitassi [54], Grigoriev showed in 1999 [97, 99] that refuting any unsatisfiable system of \( \mathbb{F}_2 \)-linear equations requires degree at least \( D/2 \), where \( D \) is

---

1Grigoriev and Vorobjov also proposed a certain dynamic version of the proof system, analogous to Polynomial Calculus [68]. Indeed, [161] had earlier proposed a dynamic proof system based on Positivstellensatz. We do not discuss dynamic proof systems further in this paper.
the least width needed to give a Resolution refutation. As a consequence he showed that degree $\Omega(n)$ is necessary to prove Tseitin tautologies on $n$-vertex regular expander graphs and to prove that the graph $K_n$ has no perfect matching when $n$ is odd. Grigoriev also subsequently [98] showed that the “$r$-Knapsack tautology” requires a proof of degree $n+1$ for any real $r \in \left( \frac{n}{2} - \frac{1}{2}, \frac{n}{2} + \frac{1}{2} \right)$; this is the infeasibility of the system

$$\{X_1^2 = X_1, \ldots, X_n^2 = X_n, X_1 + \cdots + X_n = r\},$$

for $r$ a non-integer. For more on algebraic proof complexity with inequalities, see e.g. [100].

**Optimization.** We now discuss algorithmic issues. Let $u \in \mathbb{R}[x]$ be a real $n$-variate polynomial of degree $d$. A most basic optimization problem is to determine $\inf_{x \in \mathbb{R}^n} u(x)$. Roughly speaking, this is equivalent (by binary search) to the problem of deciding whether $u(x) \geq \alpha$; further, there is no loss of generality in assuming $\alpha = 0$. Unfortunately, the problem of deciding whether $u \geq 0$ is NP-hard as soon as $d \geq 4$. In 1987, Shor [210] pioneered the idea of replacing the condition $u \geq 0$ with the stronger condition that $u$ is SOS, and noted that this can be tested in $\text{poly}(n^d)$ by solving an SDP feasibility problem. (Here we ignore the issue of precision in solving SDPs; see Section 8.2 for more details.) Shor made the connection to Hilbert’s 17th Problem but not to Positivstellensatz.

Beginning in 2000, Parrilo [185] and Lasserre [156, 157] independently published several works taking the idea further. Parrilo emphasized the viewpoint of Positivstellensatz as a refutation system for polynomial inequalities, while Lasserre focused significant attention on the dual SDP “problem of moments”. Both proposed using $\text{poly}(n^{d})$-time SDPs to search for degree-$d$ Positivstellensatz refutations, for larger and larger $d$.

Lasserre also proposed using certain variant forms of Positivstellensatz. For example, if one is optimizing a polynomial on a compact semialgebraic set $K$ then one can use SDP optimization directly (as opposed to using binary search and feasibility testing), thanks to a version of the Positivstellensatz due to Schmüdgen [203]. Furthermore, Putinar [187] showed that if $K$ is explicitly compact (“Archimedean”) — say, one of its defining inequalities is $\sum_{i=1}^{n} X_i^2 \leq B$ — then the Positivstellensatz certificates (8.2) only require $u_J$’s with $|J| \leq 1$. (Both [203, 187] contained a bug, fixed in [223].) On one hand, in practice there is rarely any harm in adding an inequality $\sum_{i=1}^{n} X_i^2 \leq B$ with large $B$; on the other hand, eliminating the $u_J$’s with $|J| > 1$ may cause the refutation degree to increase. In any case, Lasserre focused on the polynomial optimization problem

$$\inf\{p(x) \mid x \in K\}, \quad K = \{x \in \mathbb{R}^n \mid q_1(x) \geq 0, \ldots, q_m(x) \geq 0\},$$

(8.3)
and proposed a hierarchy of SDP relaxations for increasing $d$,

$$\inf \{ L(p) \mid L : \mathbb{R}[X]_d \to \mathbb{R} \text{ is a linear map}, L(1) = 1, \text{ and } L(u), L(uq_i) \geq 0 \text{ for all SOS } u \} ,$$

where $\mathbb{R}[X]_d$ denotes the ring $\mathbb{R}[X]$ restricted to polynomials of degree at most $d$. This is a relaxation because one can take $L$ to be the evaluation map $p \mapsto p(x^*)$ for any optimal solution $x^*$. We refer to (8.4) as the degree-$d$ Lasserre moment SDP; when $d$ is even it is also known as the level-$d/2$ (or sometimes $d/2 - 1$) Lasserre hierarchy SDP. The semidefinite dual of (8.4) is

$$\sup \{ \beta \mid p - \beta = u_0 + u_1 q_1 + \cdots + u_m q_m \text{ for some SOS } u_0, \ldots, u_m \text{ with } \deg(u_0), \deg(u_i q_i) \leq d \} ,$$

which we refer to as the degree-$d$ Lasserre SOS SDP. (One can also allow for polynomial equalities in the description of $K$, either by replacing them with pairs of inequalities, extending the SDP formulations as in (8.2), or by factoring out by the ideal they generate [161].)

Assuming $K$ is explicitly compact, Lasserre [157] showed that the SOS SDP’s value tends to the optimal value as the degree increases. If furthermore $K$ has a nonempty interior then there is no duality gap between (8.4) and (8.5). Generally $K$ has empty interior for discrete optimization problems (e.g., if it includes the constraints $X_i^2 = X_i$); however, the duality gap issue is algorithmically irrelevant since the Ellipsoid Algorithm can’t distinguish an empty interior from a small interior anyway. This issue is discussed briefly in Section 8.2.

Prior optimization results. We conclude by mentioning some known positive and negative results for the Lasserre moment SDP relaxation. Around 2001, Laurent [160] considered the Lasserre hierarchy for MAXCUT with negative edge weights allowed (i.e., the 2Lin2 CSP). She showed that degree-2 Lasserre optimally solves all instances whose underlying graph is a tree, and conversely that there are non-tree instances which degree-2 Lasserre does not solve optimally. She similarly characterized the underlying graphs which degree-4 Lasserre solves optimally: the $K_5$-minor-free graphs. Around 2002, Laurent [159] showed that when $n$ is odd, the degree-$(n - 1)$ moment SDP relaxation for the MAXCUT problem on $K_n$ still has value $n^2/4$ (whereas the optimum value is $n^2 - 1$); i.e., the $\left\lceil \frac{n+1}{2} \right\rceil$th level of the Lasserre hierarchy is required to obtain the optimal solution. Around 2005, Cheung [65] considered the Knapsack problem and showed that in the optimization problem

$$\inf \{ X_1 + \cdots + X_n \mid X_i^2 = X_i \forall i, \quad X_1 + \cdots + X_n \geq r \} ,$$

if $r = r(n) \in (0, 1)$ is sufficiently small then the Lasserre moment SDP does not find the optimal solution (namely, 1) until the degree is “maximal”, namely $2n + 2$. In 2008,
Schoenbeck essentially rediscovered Grigoriev’s result on $\mathbb{F}_2$-linear equations from the moment side, showing that there are $n$-variable 3Lin2 instance of value $\frac{1}{2} + o_n(1)$ for which the degree-$\Omega(n)$ Lasserre moment relaxation still has value 1. Building on this work, Tulsiani [220] showed degree-$\Omega(n)$ integrality gap instances matching the known \textsc{NP}-hardness factors for a number of CSPs. Guruswami, Sinop, and Zhou [112] showed a degree-$\Omega(n)$ integrality gap instance for the \textsc{BalancedSeparator} problem with factor $\alpha > 1$, even though this level of \textsc{NP}-hardness is not known. They also showed a degree-$\Omega(n)$ integrality gap instance for the \textsc{MaxCut} problem with factor $\frac{17}{18}$. Around 2010, Karlin, Mathieu, and Nguyen [131] showed that the degree-2\textit{t} Lasserre moment relaxation achieves approximation ratio $1 - \frac{1}{t}$ for the general Knapsack problem.

8.1.2 Our contributions and organization of this part

In this part, we study the power of the $O(1)$-degree SOS SDP hierarchy for several central combinatorial optimization problems.

\textsc{UniqueGames}. We first study the $O(1)$-degree SOS SDP hierarchy for the \textsc{UniqueGames} instances considered in the literature (i.e. in [192, 154, 142], obtained by composing the “quotient noisy cube” instance of [144] with the long-code alphabet reduction of [141]). In Chapter 9, we prove the following theorem.

**Theorem 8.1.1** (Pre-statement of Theorem 9.0.8). Let $G$ be an $n$-variable \textsc{UniqueGames} instance with label-size $q$ of the type considered in [192, 154, 142] obtained by composing the “quotient noisy cube” instance of [144] with the long-code alphabet reduction of [141] so that the best assignment to $G$’s variables satisfies at most an $\epsilon$ fraction of the constraints. When $\epsilon$ is sufficiently small and $n$ is sufficiently large, there is a degree-8 SOS refutation for the statement that the best assignment to $G$’s variables satisfy at least $1/100$ fraction of the constraints.

Thus just the level-4 Lasserre SDP hierarchy (essentially) solves the \textsc{UniqueGames} instances.

We also investigate whether the $O(1)$-degree SOS SDP hierarchy can solve known integrality gap instances of problems that are essentially \textit{harder} than \textsc{UniqueGames}. We focus on two such problems: \textsc{BalancedSeparator} and \textsc{MaxCut}.
**BalancedSeparator.** Building on work of Khot–Vishnoi [144] and Krauthgamer–Rabani [150], Devanur, Khot, Saket, and Vishnoi (DKSV) [79] gave a family of $n$-vertex BalancedSeparator instances in which the optimal balanced separator cuts an $\Omega\left(\frac{\log \log n}{\log n}\right)$ fraction of the edges, but for which the SDP with triangle inequalities has value $O\left(\frac{1}{\log n}\right)$. This is a factor-$\Theta(\log \log n)$ integrality gap. Raghavendra and Steurer [192] show that a factor-$\Omega(1)$ gap persists for these instances even for $\Omega(1)$ rounds of the “LH SDP hierarchy”. The key to analyzing the optimum value of their instances is the KKL Theorem [129] from analysis of boolean functions. In this work we give a degree-4 SOS proof of the KKL Theorem. In turn, this is used in Chapter 10 to show the following:

**Theorem 8.1.2 (Pre-statement of Theorem 10.3.3).** The degree-4 SOS relaxation for the DKSV BalancedSeparator instances has value $\Omega\left(\frac{\log \log n}{\log n}\right)$.

Thus just the level-2 Lasserre SDP hierarchy (essentially) solves the DSKV BalancedSeparator instances.

**MaxCut.** Khot and Vishnoi [144] gave integrality gap instances for the MaxCut problem, by composing their UniqueGames instances with the Khot–Kindler–Mossel–O’Donnell [141] MaxCut reduction. When this reduction is executed with parameter $\rho \in (-1, 0)$, one obtains $n$-vertex MaxCut instances with optimal value at most $\left(\arccos \rho / \pi + o_n(1)\right)$, but for which the SDP with triangle inequalities has value $\frac{1}{2} - \frac{1}{2} \rho - o_n(1)$. In particular, for $\rho = \rho_0 \approx -0.689$, this is a factor-.878 integrality gap (worst possible, by the Goemans–Williamson algorithm [94]). Khot and Saket [154] subsequently showed that this gap persists even for $\Omega(1)$ rounds of the Sherali–Adams SDP hierarchy. The key to analyzing the optimum value of the KV MaxCut instances is the Majority Is Stablest Theorem from [174]. This theorem is in turn based on an Invariance Principle for nonlinear forms of random variables, together with a Gaussian isoperimetric theorem of Borell [50]. We are able to “SOS-ize” Kindler–O’Donnell’s recent new proof of the latter [147] (it essentially only needs the triangle inequality); however we do not know how to prove the former for non-polynomial functionals. Thus we currently do not know how to give an SOS proof of the Majority Is Stablest Theorem.

We turn then to a weaker version of Majority Is Stablest known as the “$\frac{2}{3}$ Theorem”, proved in [140]. This proof relies on just the Central Limit Theorem (more precisely, the Berry–Esseen Theorem). We are able to give an SOS proof of the CLT Theorem, although not with a fixed constant degree bound. Rather, we are able to prove it up to an additive error of $\delta$ using an SOS proof of degree $\tilde{O}(1/\delta^2)$. Using this, as well as the SOS analysis of the KV UniqueGames instances, we are able to show the following in Chapter 11.
Theorem 8.1.3. There exists a universal constant $C \in \mathbb{N}^+$ such that the degree-$C$ SOS relaxation for the KV MAXCUT instances (with parameter $\rho_0 \approx -0.689$) is within a factor $.952 (> .878)$ of the optimum value. For general $\rho$, the relaxation is within a factor of $.931$ of the optimum.

A guide to the SOS proofs. Since even conceptually simple SOS proofs can sometimes look a little complicated, we give here a brief guide to our SOS proofs. Both of our results rely on the hypercontractive inequality for $\{-1, 1\}^n$ due to [48]. We give the SOS proofs for various hypercontractive inequalities in Section 8.4. For the simplest $(2, 4)$-hypercontractive inequality (Theorem 8.4.3), the only trick is that to evade the use of Cauchy–Schwarz in the standard proofs one needs to move to a “two-function” version of the inequality. We also need SOS proofs of a few other forms of the hypercontractive inequality. Though the notation is heavy, the proofs are essentially straightforward. On the other hand, we remark that we currently do not have an SOS proof of the $2 \to 2k$ version of the inequality with sharp constant for any integer $k > 2$.

For the UNIQUEGAMES instances, we will first use hypercontractivity to prove the basic “quotient noisy cube” instances by Khot and Vishnoi [144] does not have great solution. The bulk of the remaining technical work is in lifting the soundness proof of the KKMO gadget. On a high level this proof involves the following components: (1) The invariance principle of [174], saying that low influence functions cannot distinguish between the cube and the sphere; this allows us to argue that functions that perform well on the gadget must have an influential variable (an analog of the “Majority Is Stablest” theorem), and (2) the influence decoding procedure of [141] that maps these influential functions on each local gadget into a good global assignment for the original “quotient noisy cube” instance. The invariance principle poses a special challenge, since the proof of [174] uses so called “bump” functions which are not at all low-degree polynomials. We use a weaker invariance principle, only showing that the $4$ norm of a low influence function remains the same between two probability spaces that agree on the first $2$ moments. Please refer to Chapter 9 for a more detailed description.

In KKL, hypercontractivity is used to prove the “Small-Set Expansion (SSE) in the Noisy Hypercube” theorem. The usual proof of this is very short, but presents a couple of challenges for SOS proofs. One challenge is the use of Hölder’s inequality with exponents $4, \frac{4}{3}$. We are able to get around the fractional powers with a couple of tricks, one which is the following: if one needs to SOS-prove, say, $p \leq \sqrt{q}$ for some nonnegative polynomial $q$, instead prove that $p \leq \frac{\epsilon}{2} + \frac{1}{2\epsilon} \cdot q$ for all real $\epsilon > 0$. The other challenge is that the standard proof of the SSE Theorem involves division by a polynomial quantity, something we don’t see how to do with SOS proofs. Still, we manage to give a short
SOS-proof of a weaker version of the SSE Theorem which is good enough for our purposes. Finally, to obtain the BalancedSeparator result, the last step is to SOS-prove the KKL Theorem. Even the statement of the theorem involves logarithms, which does not look SOS-friendly. We get around this with a variant of the square-root trick just mentioned.

Moving to our proof of the $\frac{2}{\pi}$ Theorem, as stated, we need an SOS-proof of the Central Limit Theorem (with error bounds). Alternately phrased, we need an Invariance Theorem for linear forms of polynomials, specifically with the absolute-value functional. Although this functional is not polynomial, we can replace the required statement with something that is: namely, when $a_1, \ldots, a_n$ are indeterminates assumed to satisfy $a_1^2 + \cdots + a_n^2 = 1$, we want to upper-bound

$$E_{x \sim \{-1,1\}^n} [f(x)(a_1 x_1 + \cdots + a_n x_n)] \leq \sqrt{\frac{2}{\pi}} + e,$$

where $e$ is an error term involving $\sum_i a_i^4$, which is small when all $a_i$'s are small. Our SOS proof of this is somewhat technically difficult. To proceed, we upper-bound the absolute-value functional to within $\delta$ by a polynomial $Q$ of high degree; using real approximation theory, $\tilde{O}(1/\delta^2)$ suffices. Then we prove an Invariance Theorem for linear forms with a high-degree functional; this is feasible for linear forms (but not higher-degree ones) because of their subgaussian tails. Unlike in the usual proof of the Berry–Esseen Theorem, we need the hypercontractive inequality for high norms here.

8.2 The SOS proof system and the SDP hierarchy for optimization

In this section we give formal details of the Positivstellensatz proof system of Grigoriev–Vorobjov and the associated hierarchy of SDP algorithms due to Lasserre and Parrilo. For brevity we refer to these as “SOS proofs and hierarchies”.

**Definition 8.2.1.** Let $X = (X_1, \ldots, X_n)$ be indeterminates, let $q_1, \ldots, q_m, r_1, \ldots, r_{m'} \in \mathbb{R}[X]$, and let

$$A = \{q_1 \geq 0, \ldots, q_m \geq 0\} \cup \{r_1 = 0, \ldots, r_{m'} = 0\}.$$

Given $p \in \mathbb{R}[X]$ we say that $A$ SOS-proves $p \geq 0$ with degree $k$, written

$$A \vdash_k p \geq 0,$$
whenver
\[ \exists v_1, \ldots, v_{m'} \text{ and SOS } u_0, u_1, \ldots, u_m \text{ such that} \]
\[ p = u_0 + \sum_{i=1}^{m} u_i q_i + \sum_{j=1}^{m'} v_j r_j, \quad \text{with } \deg(u_0), \deg(u_i q_i), \deg(v_j r_j) \leq k \forall i \in [m], j \in [m']. \]

(Recall we say that \( w \in \mathbb{R}[X] \) is SOS if \( w = s_1^2 + \cdots + s_t^2 \) for some \( s_i \in \mathbb{R}[X] \).)

We say that \( A \) has a degree-\( k \) SOS refutation if
\[ A \vdash_k -1 \geq 0. \]

Finally, when \( A = \emptyset \) we will sometimes use the shorthand
\[ \vdash_k p \geq 0, \]
which simply means that \( p \) is SOS and \( \deg(p) \leq k \).

Our notation here is suggestive of a dynamic proof system, and indeed it can be helpful to think of SOS proofs this way. For example, adding deductions is not a problem:

**Fact 8.2.2.** If
\[ A \vdash_k p \geq 0, \quad A' \vdash_{k'} p' \geq 0, \]
then
\[ A \cup A' \vdash_{\max(k, k')} p + p' \geq 0. \]

However using transitivity or multiplying together two deductions leads to a worse degree bound when applied generically:

**Fact 8.2.3.** Suppose that
\[ A \vdash_k q_1' \geq 0, \ldots, q_{\ell}' \geq 0 \]
(meaning \( A \vdash_k q_i' \geq 0 \) for each \( i \in [\ell] \)). Further suppose that
\[ \{ q_1' \geq 0, \ldots, q_{\ell}' \geq 0 \} \vdash_{k'} p \geq 0. \]
Then
\[ A \vdash_{k+k'} p \geq 0. \]
Fact 8.2.4. Let \( A = \{ q_1 \geq 0, \ldots, q_m \geq 0 \} \), \( A' = \{ q'_1 \geq 0, \ldots, q'_{m'} \geq 0 \} \). If
\[
A \vdash_k p \geq 0, \quad A' \vdash_{k'} p' \geq 0,
\]
then
\[
A \cup A' \cup (A \cdot A') \vdash_{k+k'} p \cdot p' \geq 0,
\]
where \( A \cdot A' \) denotes \( \{ q_i \cdot q'_j \geq 0 : i \in [m], j \in [m'] \} \).

Notice that in the above fact we had to explicitly include product inequalities into the hypotheses. This is because in general we do not have \( \{ q \geq 0, q' \geq 0 \} \vdash qq' \geq 0 \). For example:

Proposition 8.2.5. In \( \mathbb{R}[Y, Z] \), for every \( k \in \mathbb{N} \),
\[
\{ Y \geq 0, Z \geq 0 \} \nvdash_k YZ \geq 0.
\]
Indeed, for all real \( \beta \geq 0 \),
\[
\{ Y \geq 0, Z \geq 0 \} \nvdash_k YZ \geq -\beta.
\]

Proof. Suppose to the contrary that
\[
YZ + \beta = u_1 + Y u_2 + Z u_3 \tag{8.6}
\]
for some SOS \( u_1, u_2, u_3 \in \mathbb{R}[Y, Z] \). We think of the right-hand side of (8.6) as being in \( \mathbb{R}[Z][Y] \). Let \( k_j \) be the degree of \( Y \) in \( u_j \) for \( j = 1, 2, 3 \); note that \( k_1, k_3 \) are even and \( k_2 \) is odd. Suppose first that \( \max\{ k_1, k_2, k_3 \} = k_1 \). Then we must in fact have \( k_2 = k_1 \) in order to cancel the \( Y^{k_1} \) term in the RHS of (8.6). But in fact such a cancelation is impossible because the coefficient on \( Y^{k_1} \) in \( u_1 \) will be an even-degree polynomial in \( Z \), but the coefficient on \( Y^{k_3} \) in \( u_3 \) will be an odd-degree polynomial in \( Z \). The remaining possibility is that \( k_2 > k_1, k_3 \). In this case we must have \( k_2 = 1 \), or else the degree of \( Y \) on the RHS of (8.6) will exceed 1. Thus \( u_1, u_2, u_3 \) depend only on \( Z \); but then (8.6) forces \( u_2 = Z \), contradicting the fact that \( u_2 \) is SOS. \qed

For more simple examples of the weakness of SOS proofs, see [170, Chap. 2.7]. Here is another one: we cannot directly prove \( Y^4 \geq 1 \Rightarrow Y^2 \geq 1 \).

Proposition 8.2.6. In \( \mathbb{R}[Y] \), for every \( k \in \mathbb{N} \),
\[
Y^4 \geq 1 \nvdash_k Y^2 \geq 1.
\]
Proof. Suppose to the contrary that one can write
\[ Y^2 - 1 = u + v(Y^4 - 1) \]  \hspace{1cm} (8.7)
with \( u, v \in \mathbb{R}[Y] \) being SOS. One cannot have \( v = 0 \) because \( Y^2 - 1 \) is not SOS (consider that \( 0^2 - 1 \) is negative). Therefore the highest-degree term in \( v \) is of the form \( cY^{2j} \) for some real \( c > 0 \) and some integer \( j \). This gives a term \( cY^{2j+4} \) on the right-hand side of (8.7) which must be canceled by \( u \). This is impossible if \( \deg(u) = 2j + 4 \) because the leading coefficient on \( u \) will be positive too. So \( \deg(u) > 2j + 4 \), but then its highest-degree term remains uncanceled on the right-hand side of (8.7).

On the other hand, one can easily SOS-prove \( Y^4 \leq 1 \Rightarrow Y^2 \leq 1 \); see Fact 8.3.3. Furthermore, one can \( Y^4 \geq 1 \Rightarrow Y^2 \geq 1 \) by contradiction:

**Proposition 8.2.7.** In \( \mathbb{R}[Y] \), for any \( \epsilon > 0 \) we have
\[ \{ Y^4 \geq 1, Y^2 \leq 1 - \epsilon \} \vdash d - 1 \geq 0. \]  \hspace{1cm} (8.8)

Proof. We leave the case of \( \epsilon \geq 1 \) to the reader. Otherwise, write \( c = 1 - \epsilon \in (0,1) \); then
\[ -1 = \frac{1}{1-c^2}(c + Y^2)(c - Y^2) + \frac{1}{1-c^2}(Y^4 - 1) \]
and both \( \frac{1}{1-c^2}(c + Y^2) \) and \( \frac{1}{1-c^2} \) are SOS. \( \square \)

These observations reveal that when fixing the degree of SOS proofs, the SDP simplifications explored by Lasserre (see Section 8.1.1) can be damaging: it may help to multiply together constraint inequalities, and direct optimization can be worse than binary searching for refutations. Thus we propose that for optimizations problems, one should generically use the SDP hierarchy proposed by Parrilo. I.e., for
\[ \inf \{ p(x) \mid x \in K \}, \quad K = \{ x \in \mathbb{R}^n \mid q_1(x) \geq 0, \ldots, q_m(x) \geq 0 \}, \]
one should assume that \( K \) is “explicitly compact” (say, contains the inequality \( X^2_1 + \cdots + X^2_n \leq 2^{\text{poly}(n)} \)) and then use binary search to (approximately) find the largest \( \beta \) for which
\[ \{ q_{i_1}q_{i_2} \cdots q_{i_t} \geq 0 : \deg(q_{i_1}q_{i_2} \cdots q_{i_t}) \leq d \} \cup \{ p \leq \beta \} \vdash d - 1 \geq 0. \]  \hspace{1cm} (8.8)
This can be carried out in \( \text{poly}(n^d, m) \) time using the Ellipsoid Algorithm\(^2\).

\(^2\)Determining (8.8) amounts to checking if a matrix of variables can be PSD while satisfying some equalities. One relaxes the equalities to two-sided inequalities with some small tolerance \( \delta = 2^{-\text{poly}(n)} \), allowing one to run Ellipsoid. If Ellipsoid returns a feasible solution it can be made truly PSD at the expense of adding slightly more slack in the equalities. By virtue of the compactness, this can adjusted to give a valid SOS proof of \( -1 + \delta' \geq 0 \).
8.3 A few simple SOS preliminaries

A well-known basic fact (following from the Fundamental Theorem of Algebra) is that every nonnegative univariate polynomial is SOS:

**Fact 8.3.1.** Suppose \( p \in \mathbb{R}[X_1] \) is a univariate real polynomial such that \( p(t) \geq 0 \) for all real \( t \). Then \( p \) is SOS; i.e., \( \vdash_{\text{deg}(p)} p \geq 0 \).

The following related result is credited in [162] to Fekete and Markov–Lukács, with reference also to [170]:

**Fact 8.3.2.** Suppose \( p \in \mathbb{R}[Y] \) is a univariate real polynomial of degree \( k \) such that \( p(t) \geq 0 \) for all real \( a \leq t \leq b \).

If \( k \) is odd then \( Y \geq a, b \geq Y \vdash_k p \geq 0 \).

If \( k \) is even then \( (b - Y)(Y - a) \geq 0 \vdash_k p \geq 0 \).

We now give some additional simple SOS proofs:

**Fact 8.3.3.** \( Y^2 \leq 1 \vdash_2 Y \leq 1, Y \geq -1 \).

*Proof.* The first follows from \( 1 - Y = \frac{1}{2}(1 - Y)^2 + \frac{1}{2}(1 - Y^2) \). The second follows by replacing \( Y \) by \(-Y\). \( \square \)

**Fact 8.3.4.** \( \{Y \leq 1, Y \geq -1\} \vdash_3 Y^2 \leq 1 \).

*Proof.* \( 1 - Y^2 = \frac{1}{2}(1 + Y)^2(1 - Y) + \frac{1}{2}(1 - Y)^2(1 + Y) \). \( \square \)

**Fact 8.3.5.** If \( A \vdash_k Y^2 \leq Y (k \geq 2) \), then \( A \vdash_k Y \leq 1 \).

*Proof.* Since \( 1 - Y = Y - Y^2 + (1 - Y)^2 \) and there is a degree-\( k \) SOS proof for \( Y - Y^2 \geq 0 \) (assuming \( A \)), we have a degree-\( k \) SOS proof for \( 1 - Y \geq 0 \) assuming \( A \). \( \square \)

We will need an SOS proof of the fact that \( Y, Z \in \{-1, 1\} \Rightarrow \frac{Y - Z}{2} \in \{-1, 0, 1\} \):

**Fact 8.3.6.** \( Y^2 = 1, Z^2 = 1 \vdash_3 \left(\frac{Y - Z}{2}\right) = \left(\frac{Y - Z}{2}\right)^3 \).

*Proof.* \( \left(\frac{Y - Z}{2}\right) - \left(\frac{Y - Z}{2}\right)^3 = \left(\frac{3}{8}Z - \frac{1}{8}Y\right)(Y^2 - 1) + \left(\frac{1}{8}Z - \frac{3}{8}Y\right)(Z^2 - 1) \). \( \square \)

154
Fact 8.3.7. Suppose that $A \vdash_k Y \geq -1, Y \leq 1$ and that $B \vdash_{\ell} Z \geq W, Z \geq -W$. Then $A \cup B \vdash_{k+\ell} Z \geq YW$.

Proof. $Z - YW = \frac{1}{2}(Z - W)(1 + Y) + \frac{1}{2}(Z + W)(1 - Y)$.

Fact 8.3.8. Suppose that $A \vdash_k Y' \geq Y$ and $B \vdash_{\ell} Z' \geq Z$. Further suppose $A' \vdash_{k'} Y'' \geq 0$ and $B' \vdash_{\ell'} Z'' \geq 0$. Then $A \cup B \cup A' \cup B' \vdash_{\max\{k+\epsilon, k'+\epsilon\}} Y'Z' \geq YZ$.

Proof. This follows from $Y'Z' - YZ = Y'(Z' - Z) + Z(Y' - Y)$.

We now move to Hölder-type inequalities.

Fact 8.3.9. $\vdash 2YZ \leq \frac{1}{2}Y^2 + \frac{1}{2}Z^2$.

Proof. $\frac{1}{2}Y^2 + \frac{1}{2}Z^2 - YZ = \frac{1}{2}(Y - Z)^2$.

More generally, by replacing $Y$ with $\epsilon^{1/2}Y$ and $Z$ with $\epsilon^{-1/2}Z$, we obtain:

Fact 8.3.10. $\vdash 2YZ \leq \frac{\epsilon}{2}Y^2 + \frac{1}{\epsilon}Z^2$ for any real $\epsilon > 0$.

We would also like Young’s inequality for conjugate Hölder exponents $(\frac{4}{3}, 4)$, but stating it needs a trick:

Fact 8.3.11. $\vdash_4 Y^3Z \leq \frac{3}{4}Y^4 + \frac{1}{4}Z^4$.

Proof. $\frac{3}{4}Y^4 + \frac{1}{4}Z^4 - Y^3Z = (\frac{3}{4}Y^2 + \frac{1}{2}YZ + \frac{1}{4}Z^2)(Y - Z)^2$

$= (\frac{1}{4}Y^2 + \frac{1}{4}(Y + Z)^2)(Y - Z)^2 = \frac{1}{4}Y^2(Y - Z)^2 + \frac{1}{8}(Y^2 - Z^2)^2$.

By replacing $Y$ with $\epsilon^{1/4}Y$ and $Z$ with $\epsilon^{-3/4}Z$, we obtain:

Fact 8.3.12. $\vdash_4 Y^3Z \leq \frac{3\epsilon}{4}Y^4 + \frac{1}{4\epsilon^3}Z^4$ for any real $\epsilon > 0$.

Fact 8.3.13. $\{Y^2 \leq Y\} \vdash_4 YZ \leq \frac{3\epsilon}{4}Y + \frac{1}{4\epsilon^3}Z^4$ for any real $\epsilon > 0$.

Proof. Since we have the assumption $Y^2 \leq Y$, it suffices to prove that

$\{Y^2 \leq Y\} \vdash_4 YZ \leq \frac{3\epsilon}{4}Y + \frac{\epsilon}{4}Y^2 + \frac{1}{4\epsilon^3}Z^4$.

This is true because

$\frac{3\epsilon}{4}Y + \frac{\epsilon}{4}Y^2 + \frac{1}{4\epsilon^3}Z^4 - YZ = \frac{1}{4}(\sqrt{\epsilon}Y - \frac{1}{\sqrt{\epsilon}}Z^2)^2 + \frac{1}{4\epsilon}Y^2(\frac{1}{\sqrt{\epsilon}}Z - \sqrt{\epsilon})^2 + \frac{1}{2}(Y - Y^2)(\frac{1}{\sqrt{\epsilon}}Z - \sqrt{\epsilon})^2$.
Fact 8.3.14. If $A \vdash_k Y \geq 0$ and $A \vdash_k Y \leq Z$, then $A \vdash_{2k} Y^2 \leq Z^2$.

Proof. We can deduce $A \vdash_k Z \geq 0$ and therefore $A \vdash_k Z + Y \geq 0$ using Fact 8.2.2. The result now follows from Fact 8.2.4 applied to $Z^2 - Y^2 = (Z + Y)(Z - Y)$.

Fact 8.3.15. $\vdash_2 \text{avg}_{i \in [n]}[X_i^2] \geq (\text{avg}_{i \in [n]}[X_i])^2$.

Proof. $\text{avg}_{i \in [n]}[X_i^2] - (\text{avg}_{i \in [n]}[X_i])^2 = \text{avg}_{i,j \in [n]}[\frac{1}{2}(X_i - X_j)^2]$.

8.4 SOS proofs of hypercontractivity

In the remainder of the work we will use some standard notions from analysis of Boolean functions; see, e.g., [181]. All of our main results will require SOS proofs of the well-known hypercontractivity theorems on $\{-1, 1\}^n$, first proved by Bonami [48]. To state them, recall that any function $f : \{-1, 1\}^n \rightarrow \mathbb{R}$ can be viewed as a multilinear polynomial,

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{i \in S} x_i,$$

where $\hat{f}(S) = \mathbb{E}_{x \sim \{-1, 1\}^n} [f(x) \prod_{i \in S} x_i]$. (8.9)

Then for $\rho \in \mathbb{R}$, the linear operator $T_{\rho}$ is defined by mapping the above function to

$$T_{\rho} f(x) = \sum_{S \subseteq [n]} \rho^{|S|} \hat{f}(S) \prod_{i \in S} x_i.$$

Now the $p = 2, q \geq 2$ cases of hypercontractivity can be stated as follows:

Theorem 8.4.1. Let $f : \{-1, 1\}^n \rightarrow \mathbb{R}$. Then for any real $q \geq 2$,

$$\mathbb{E}_{x \sim \{-1, 1\}^n} [|T_{\frac{1}{\sqrt{q}}} f(x)|^q] \leq \mathbb{E}_{x \sim \{-1, 1\}^n} [f(x)^2]^{q/2}. $$

Theorem 8.4.2. Let $f : \{-1, 1\}^n \rightarrow \mathbb{R}$ have degree at most $k$. Then for any real $q \geq 2$,

$$\mathbb{E}_{x \sim \{-1, 1\}^n} [|f(x)|^q] \leq (q - 1)^{(q/2)k} \cdot \mathbb{E}_{x \sim \{-1, 1\}^n} [f(x)^2]^{q/2}.$$

Note that Theorem 8.4.2 follows immediately from Theorem 8.4.1 in case $f$ is homogeneous of degree $k$. It is also known that Theorem 8.4.1 and Theorem 8.4.2 (even its homogeneous version) are “equivalent”, in the sense that one can be derived from the other using various analytic tricks.
As mentioned, we would ideally like to give SOS proofs of these theorems. In order to even state the theorems as polynomial inequalities it is required that $q$ be an even integer. For example, when $q = 4$ we may try to SOS-prove

$$\mathbb{E}_{x \sim \{-1,1\}^n} [(T_{-\sqrt{q}} \cdot f(x))^4] \leq \mathbb{E}_{x \sim \{-1,1\}^n} [f(x)^2]^2.$$

The meaning of this is that the $2^n$ Fourier coefficients of $f$ are the indeterminates; i.e., we work over the ring $\mathbb{R}[\hat{f}(\emptyset), \hat{f}(\{1\}), \ldots, \hat{f}([n])]$ and would like to show that

$$\mathbb{E}_{x \sim \{-1,1\}^n} [f(x)^2]^2 - \mathbb{E}_{x \sim \{-1,1\}^n} [(T_{-\sqrt{q}} \cdot f(x))^4]$$

is a sum of squares of polynomials over the indeterminates $\hat{f}(S)$. Sometimes we will instead use the $2^n$ indeterminates “$f(x)$” for $x \in \{-1,1\}^n$ — note that this is completely equivalent because the $f(x)$’s are homogeneous linear forms in the $\hat{f}(S)$’s and vice versa; see (8.9).

When $q$ is an even integer it is well known that [Theorem 8.4.2] has a much simpler, “almost combinatorial” proof. For example, Bonami’s original paper proved the homogeneous version of [Theorem 8.4.2] for even integer $q$ using nothing more “analytic” than absolute values and Cauchy–Schwarz. (Her proof even obtains a slightly sharper constant than $(q - 1)^{(q/2)k}.$) The inductive proof of [Theorem 8.4.2] for $q = 4$ presented in [174] is simpler still, using only Cauchy–Schwarz. It is not hard to check that these remarks also apply to [Theorem 8.4.1].

Nevertheless, it’s not completely trivial to obtain SOS proofs of [Theorem 8.4.1] and [Theorem 8.4.2] when $q$ is an even integer, simply because the Cauchy–Schwarz inequality, $\mathbb{E}[fg] \leq \sqrt{\mathbb{E}[f^2] \cdot \mathbb{E}[g^2]}$, has square-roots in it. The natural substitute is the inequality $\mathbb{E}[fg] \leq \frac{1}{2} \mathbb{E}[f^2] + \frac{1}{2} \mathbb{E}[g^2]$ (see Fact 8.3.9). However fitting this into the known proof of, say, the $q = 4$ case of [Theorem 8.4.2] seems to require an extra trick: moving to a “two-function” version of the statement.

**Theorem 8.4.3.** (SOS proof of the two-function, $q = 4$ version of [Theorem 8.4.2])

Let $n, k_1, k_2 \in \mathbb{N}$. For each $j = 1, 2$ and each $S \subseteq [n]$ of cardinality at most $k_j$, introduce an indeterminate $\hat{f}_j(S)$. For $x \in \{-1,1\}^n$, let $f_j(x)$ denote $\sum_S \hat{f}_j(S) \prod_{i \in S} x_i$. Then

$$\mathbb{E}_{x \sim \{-1,1\}^n} [f_1(x)^2 f_2(x)^2] \leq 3^{k_1+k_2} \cdot \mathbb{E}_{x \sim \{-1,1\}^n} [f_1(x)^2] \cdot \mathbb{E}_{x \sim \{-1,1\}^n} [f_2(x)^2].$$
However, we choose to omit the proof of Theorem 8.4.3 but give an SOS proof of the $q = 4$ case of Theorem 8.4.1 using the “two-function” idea, which will imply Theorem 8.4.3. We will need a more general statement which allows for some of the $\pm 1$ random variables to be replaced by Gaussians; this idea is also from [174].

**Theorem 8.4.4.** (SOS proof of the two-function, $q = 4$ version of Theorem 8.4.1)

Let $n \in \mathbb{N}$. For each $j = 1, 2$ and each $S \subseteq [n]$, introduce an indeterminate $\hat{f}_j(S)$. For each $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$, let

\[
\hat{f}_j(z) = \sum_{S \subseteq [n]} \hat{f}_j(S) \prod_{i \in S} z_i, \quad T_{\sqrt[d]{3}} f_j(z) = \sum_{S \subseteq [n]} (\frac{1}{\sqrt[d]{3}})^{|S|} \hat{f}_j(S) \prod_{i \in S} z_i;
\]

these are homogeneous linear polynomials in the indeterminates. Let $z = (z_1, \ldots, z_n)$ be a random vector in which the components $z_i$ are independent and satisfy $E[z_i] = E[z_i^2] = 0$, $E[z_i^3] = 1$, $E[z_i^4] \leq 9$. (For example, Rademachers and standard Gaussians qualify.)

Then

\[
\vdash_4 E[(T_{\sqrt[d]{3}} f_1(z))^2 \cdot (T_{\sqrt[d]{3}} f_2(z))^2] \leq E[f_1(z)^2] \cdot E[f_2(z)^2].
\]

In particular,

\[
\vdash_4 E[(T_{\sqrt[d]{3}} f_1(z))^4] \leq E[f_1(z)^2]^2.
\]

Just as Theorem 8.4.2 follows immediately from Theorem 8.4.1, Theorem 8.4.3 also follows immediately from Theorem 8.4.4.

**Proof of Theorem 8.4.4** The proof of the theorem is by induction on $n$. For $n = 0$ we need to show $\vdash_4 f_1(\theta)^2 f_2(\theta)^2 \leq \hat{f}_1(\theta)^2 \hat{f}_2(\theta)^2$, which is trivial. For general $n \geq 1$ and $(z_1, \ldots, z_n) \in \mathbb{R}^n$ we can express $f_j(z_1, \ldots, z_n) = z_n d_j(z') + e_j(z')$, where $z' \in \mathbb{R}^{n-1}$ denotes $(z_1, \ldots, z_{n-1})$.

\[
d_j(z') = \sum_{S \not\supseteq n} \hat{f}_j(S) \prod_{i \in S \setminus \{n\}} z_i,
\]

\[
e_j(z') = \sum_{S \not\supseteq n} \hat{f}_j(S) \prod_{i \in S} z_i.
\]

Now

\[
E_{x \sim \{-1, 1\}^n} [(T_{\sqrt[d]{3}} f_1(z))^2 \cdot (T_{\sqrt[d]{3}} f_2(z))^2]
\]

\[
= E_z \left[ \left( \frac{1}{\sqrt[d]{3}} z_n \cdot T_{\sqrt[d]{3}} d_1(z') + T_{\sqrt[d]{3}} e_1(z') \right)^2 \left( \frac{1}{\sqrt[d]{3}} z_n \cdot T_{\sqrt[d]{3}} d_2(z') + T_{\sqrt[d]{3}} e_2(z') \right)^2 \right]
\]

158
= \mathbb{E}_z \left[ \left( \frac{1}{3} z_n^2(Td_1)^2 + \frac{2}{\sqrt{3}} z_n(Td_1)(Te_1) + (Te_1)^2 \right) \left( \frac{1}{3} z_n^2(Td_2)^2 + \frac{2}{\sqrt{3}} z_n(Td_2)(Te_2) + (Te_2)^2 \right) \right],

where we introduced the shorthand $(Td_j)$ for $T \frac{1}{\sqrt{3}} d_j(z')$ (and similarly for $e_j$). We continue by expanding the product and using linearity of expectation, $\mathbb{E}[z_n] = \mathbb{E}[z'_n] = 0$, $\mathbb{E}[z_n^2] = 1$; thus the above equals

$$\frac{1}{9} \mathbb{E}_n^2 \mathbb{E}_z[(Td_1)^2(Td_2)^2] + \frac{1}{3} \mathbb{E}_z[(Td_1)^2(Te_2)^2] + \frac{1}{3} \mathbb{E}_z[(Td_2)^2(Te_1)^2] + \frac{4}{3} \mathbb{E}_z[(Td_1)(Te_2) \cdot (Td_2)(Te_1)].$$

Using Fact 8.3.9 we have

$$\vdash_4 \frac{1}{3} \mathbb{E}_z[(Td_1)(Te_2) \cdot (Td_2)(Te_1)] \leq \frac{2}{3} \mathbb{E}_z[(Td_1)^2(Te_2)^2] + \frac{2}{3} \mathbb{E}_z[(Td_2)^2(Te_1)^2].$$

By our assumption $\mathbb{E}[z_n^2] \leq 9$ we have $\vdash_4 \frac{1}{9} \mathbb{E}_n^2 \mathbb{E}_z[(Td_1)^2(Td_2)^2] \leq \mathbb{E}_z[(Td_1)^2(Td_2)^2]$; here we are using the fact that $\mathbb{E}_z[(Td_1)^2(Td_2)^2]$ is SOS. Thus we have shown

$$\vdash_4 \mathbb{E}_z[(T \frac{1}{\sqrt{3}} f_1(z))^2 \cdot (T \frac{1}{\sqrt{3}} f_2(z))^2] \leq \mathbb{E}_z[(Td_1)^2(Td_2)^2] + \mathbb{E}_z[(Td_1)^2(Te_2)^2] + \mathbb{E}_z[(Td_2)^2(Te_1)^2] + \mathbb{E}_z[(Te_1)^2(Td_e)^2].$$

We use induction on each of the four terms above and deduce

$$\vdash_4 \mathbb{E}_z[(T \frac{1}{\sqrt{3}} f_1(z))^2 \cdot (T \frac{1}{\sqrt{3}} f_2(z))^2] \leq \mathbb{E}_z'[d_1(z')^2] \mathbb{E}_z'[d_2(z')^2] + \mathbb{E}_z'[d_1(z')^2] \mathbb{E}_z'[e_2(z')] + \mathbb{E}_z'[d_2(z')^2] \mathbb{E}_z'[e_1(z')] + \mathbb{E}_z'[e_1(z')]^2 \mathbb{E}_z'[e_2(z')]^2 \leq \mathbb{E}_z'[d_1(z')^2 + e_1(z')^2] \mathbb{E}_z'[d_2(z')^2 + e_2(z')]^2$$

But it is easily verified that $\mathbb{E}_z[f_j(z')] = \mathbb{E}_z'[d_j(z')^2 + e_j(z')^2]$, completing the induction.

From this we can deduce Theorem 8.4.3 with the more general class of random variables.

3When $z'$ is a discrete random vector this is obvious. In the general case, note that the coefficients of the polynomial in question are finite mixed moments of $z'$. By Carathéodory’s convex hull theorem we can match any finite number of moments of $z'$ using some discrete random vector $z''$, thereby reducing SOS-verification to the discrete case. We will use this observation in the sequel without additional comment.
Corollary 8.4.5. Theorem 8.4.3 also holds with the more general type of random vector \( z \) from Theorem 8.4.4 in place of \( x \sim \{-1, 1\}^n \).

Proof. Begin by defining
\[
\tilde{g}_j(S) = \begin{cases} 
0 & \text{if } |S| > k_j, \\
\sqrt{3} |S| \hat{f}_j(S) & \text{if } |S| \leq k_j
\end{cases}
\]
for \( j = 1, 2 \), and then applying Theorem 8.4.4 to \( g_1, g_2 \). This yields
\[
\vdash_4 \mathbb{E}_z[f_1(z)^2 f_2(z)^2] \leq \mathbb{E}_z[T_{\sqrt{3}} f_1(z)^2] \cdot \mathbb{E}_z[T_{\sqrt{3}} f_2(z)^2].
\]
By a standard computation we have
\[
\mathbb{E}_z[T_{\sqrt{3}} f_j(z)^2] = \sum_{i=0}^{k_j} 3^i \cdot W_j^i, \quad \text{where} \quad W_j^i = \sum_{|S|=i} \hat{f}_j(S)^2, \quad j = 1, 2.
\]
We also have
\[
\mathbb{E}_z[f_j(z)^2] = \sum_{i=0}^{k_j} W_j^i.
\]
Thus to complete the proof it remains to show
\[
\vdash_4 3^{i+i'} \cdot W_j^{i} \cdot W_2^{i'} \leq 3^{k_1+k_2} \cdot W_1^i \cdot W_2^{i'}
\]
for each \( 0 \leq i \leq k_1, 0 \leq i' \leq k_2 \).

We would also like to have an SOS proof of Theorem 8.4.2 for even integers \( q > 4 \). We content ourselves with the following slightly weaker result, the proof of which follows easily from Corollary 8.4.5:

Theorem 8.4.6. (SOS proof of a weakened version of the two-function, even integer \( q \) case of Theorem 8.4.6)

Let \( n, r, k_1, k_2, \ldots, k_{2r} \in \mathbb{N} \). For each \( j \in [2r] \) and each \( S \subseteq [n] \) of cardinality at most \( k_j \), introduce an indeterminate \( \hat{f}_j(S) \). Let \( f_1(z), \ldots, f_{2r}(z) \) and random vector \( z \) be as in Theorem 8.4.4. Then
\[
\vdash_{2r+1} \mathbb{E}_z \left[ \prod_{j=1}^{2r} f_j(z)^2 \right] \leq 3^{r(k_1+\cdots+k_{2r})} \cdot \prod_{j=1}^{2r} \mathbb{E}_z[f_j(z)^2].
\]
Proof. The proof is by induction on \( r \). The \( r = 0 \) case is trivial. For \( r \geq 1 \), define

\[
F_1(z) = \prod_{j=1}^{2^{r-1}} f_j(z), \quad F_2(z) = \prod_{j=2^{r-1}+1}^{2^r} f_j(z).
\]

Note these are degree-\( 2^r-1 \) in the indeterminates. Further, one may express

\[
F_1(z) = \sum_{T \subseteq [n]} \hat{f}(T) \prod_{i \in T} z_i,
\]

where \( \hat{f}(T) \) denotes a degree-\( 2^r-1 \) polynomial in the indeterminates, and similarly for \( F_2 \).

Thus we may apply Corollary 8.4.5 to \( F_1 \) and \( F_2 \) and deduce

\[
\vdash_{2^{r+1}} \mathbb{E}_z \left[ \prod_{j=1}^{2^r} f_j(z)^2 \right] \leq 3^{k_1 + \cdots + k_{2r}} \cdot \mathbb{E}_z \left[ \prod_{j=1}^{2^{r-1}} f_j(z)^2 \right] \cdot \mathbb{E}_z \left[ \prod_{j=2^{r-1}+1}^{2^r} f_j(z)^2 \right]. \quad (8.10)
\]

By induction we have

\[
\vdash_{2^r} \mathbb{E}_z \left[ \prod_{j=1}^{2^{r-1}} f_j(z)^2 \right] \leq 3^{(r-1)(k_1 + \cdots + k_{2r-1})} \cdot \prod_{j=1}^{2^{r-1}} \mathbb{E}_z [f_j(z)^2],
\]

\[
\vdash_{2^r} \mathbb{E}_z \left[ \prod_{j=2^{r-1}+1}^{2^r} f_j(z)^2 \right] \leq 3^{(r-1)(k_{2r-1} + \cdots + k_{2r})} \cdot \prod_{j=2^{r-1}+1}^{2^r} \mathbb{E}_z [f_j(z)^2],
\]

and all four expressions above are SOS of degree \( 2^r \). Combining these via Fact 8.3.8 yields

\[
\vdash_{2^{r+1}} \mathbb{E}_z \left[ \prod_{j=1}^{2^{r-1}} f_j(x)^2 \right] \cdot \mathbb{E}_z \left[ \prod_{j=2^{r-1}+1}^{2^r} f_j(x)^2 \right] \leq 3^{(r-1)(k_1 + \cdots + k_{2r})} \cdot \prod_{j=1}^{2^{r-1}} \mathbb{E}_z [f_j(z)^2] \cdot \prod_{j=2^{r-1}+1}^{2^r} \mathbb{E}_z [f_j(z)^2],
\]

which taken together with \((8.10)\) completes the induction. \( \square \)

Corollary 8.4.7. (SOS proof of a weakened version of the even integer \( q \) case of Theorem 8.4.6)
Let $n, k \in \mathbb{N}$. For each $S \subseteq [n]$ of cardinality at most $k$, introduce an indeterminate $\hat{f}(S)$. Let $f(z)$ and random vector $z$ be as in Theorem 8.4.4. Then for any even integer $q \geq 2$,

$$
\mathbb{E}_{z} [f(z)^q] \leq \sqrt{3}^{q \lceil \log_2 q \rceil - q} \cdot \mathbb{E}_{z} [f_j(z)^2]^{q/2}.
$$

Proof. Take $r = \lceil \log_2 q \rceil - 1$, $f_1 = \cdots = f_{q/2} = f$, $f_{q/2+1} = \cdots = f_{2^r} = 1$ in Theorem 8.4.6. \qed
Chapter 9

Analysis of the UNIQUEGAMES instances

In this section we prove the following theorem.

**Theorem 9.0.8.** Let \( G \) be an \( n \)-variable UNIQUEGAMES instance with label-size \( q \) of the type considered in [154, 142] obtained by composing the “quotient noisy cube” instance of [144] with the long-code alphabet reduction of [141] so that the best assignment to \( G \)'s variables satisfies at most an \( \epsilon \) fraction of the constraints. When \( \epsilon \) is sufficiently small and \( n \) is sufficiently large, there is a degree-8 SOS refutation for the statement that the best assignment to \( G \)'s variables satisfy at least \( 1/100 \) fraction of the constraints.

The more formal version of *Theorem 9.0.8* is *Theorem 9.4.1* later in this section. Now we give an overview of the proof.

The proof is very technical, as it is obtained by taking the already rather technical proofs of soundness for these instances, and “lifting” each step into the SOS hierarchy, a procedure that causes additional difficulties. The high level structure of all integrality gap instances constructed in the literature was the following: start with a basic integrality gap instance of UNIQUEGAMES where the Basic SDP outputs \( 1 - o(1) \) but the true optimum is \( o(1) \), the alphabet size of \( G \) is (necessarily) \( N = \omega(1) \). Then, apply an alphabet-reduction gadget (such as the long code, or in the recent work [35] the so called “short code”) to transform \( G \) into an instance \( G \) with some constant alphabet size \( q \). The soundness proof of the gadget guarantees that the true optimum of \( G \) is small, while the analysis of previous works managed to “lift” the completeness proofs, and argue that the instance \( G \) survives a number of rounds that tends to infinity as \( \epsilon \) tends to zero, where \( (1 - \epsilon) \) is the completeness value in the gap constructions, and exact tradeoff between number of rounds and \( \epsilon \) depends on the paper and hierarchy.

The fact that the basic instance \( G \) has small integral value can be shown by appealing to
hypercontractivity of low-degree polynomials, and hence can be “lifted” to the SOS world using tools developed in Section 8.4. This part of the proof is presented in Section 9.3. The bulk of the remaining technical work is in lifting the soundness proof of the gadget. On a high level this proof involves the following components: (1) The invariance principle of [174], saying that low influence functions cannot distinguish between the cube and the sphere (related proofs presented in Section 9.1); this allows us to argue that functions that perform well on the gadget must have an influential variable (related proofs presented in Section 9.2), and (2) the influence decoding procedure of [141] that maps these influential functions on each local gadget into a good global assignment for the original instance \( G \) (related proofs presented in Section 9.4).

The invariance principle poses a special challenge, since the proof of [174] uses so-called “bump” functions which are not at all low-degree polynomials.\(^1\) We use a weaker invariance principle, only showing that the 4 norm of a low influence function remains the same between two probability spaces that agree on the first 2 moments. Unlike the usual invariance principle, we do not move between Bernoulli variables and Gaussian space, but rather between two different distributions on the discrete cube. It turns out that for the purposes of these UNIQUEGAMES integrality gaps, the above suffices. The lifted invariance principle is proven via a “hybrid” argument similar to the argument of [174], where hypercontractivity of low-degree polynomials again plays an important role.

The soundness analysis of [141] is obtained by replacing each local function with an average over its neighbors, and then choosing a random influential coordinate from the new local function as an assignment for the original UNIQUEGAMES instance. We follow the same approach. It turns out that by making appropriate modification to the analysis, it can be lifted to complete the proof of the theorem.

### 9.1 An invariance principle for the fourth moment

In this section, we will be interested in \( f(x) \) that is a multilinear polynomial over the doubly-indexed set of indeterminates \( \{x_{i,j}\}_{i \in [N], j \in \mathbb{N}} \) with degree at most \( \ell \), i.e., \( f(x) \) can be written in the form of

\[
    f(x) = \sum_{\alpha \in \mathbb{N}^N, |\alpha| \leq \ell} \hat{f}(\alpha)x_\alpha,
\]

where \( x_\alpha = \prod_{i=1}^{N} x_{i,\alpha_i} \), and \( |\alpha| \), the degree of \( \alpha \), is defined to be \( |\{i \in [N] : \alpha_i > 0\}| \).

\(^1\)A similar, though not identical, challenge arises in [35] where they need to extend the invariance principle to the “short code” setting. However, their solution does not seem to apply in our case, and we use a different approach.
We will use the following definition introduced in [174].

**Definition 9.1.1** (Definition 3.1 from [174]). We call a collection of finitely many orthonormal random variables, one of which is the constant 1, an orthonormal ensemble. We will write a typical sequence of $n$ orthonormal ensembles as $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$, where $\mathbf{X}_i = \{X_{i,0} = 1, X_{i,1}, \ldots, X_{i,m_i}\}$. We call a sequence of orthonormal ensembles $\mathbf{X}$ independent if the ensembles are independent families of random variables.

In this section, we are only interested in independent sequences of orthonormal ensembles, and we will call these sequences of ensembles for brevity.

Let $q = 2^t$ for some $t \in \mathbb{N}$. Let $z_i$ be a random variable uniformly distributed over $\{\pm 1\}^t$, and let $z = (z_1, z_2, \ldots, z_N)$ be the random variable uniformly distributed over $\{\pm 1\}^N$.

Let $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$ be a sequence of ensembles such that $\mathbf{X}_i = \{X_{i,0}, X_{i,1}, \ldots, X_{i,q-1}\}$ and $X_{i,j} = \chi_j(z_i)$, where $\{\chi_0 \equiv 1, \chi_1, \ldots, \chi_{q-1}\}$ is the set of characters for $\{\pm 1\}^t$.

Let $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_N)$ be a sequence of ensembles such that $\mathbf{Y}_i = \{Y_{i,0}, Y_{i,1}, \ldots, Y_{i,q-1}\}$ where $Y_{i,0} \equiv 1$ and $Y_{i,j}$ is an independent uniform sample from $\{\pm 1\}$ for $j > 0$.

Now we are ready to state our invariance principle for the fourth moment of $f$.

**Theorem 9.1.2.** Let $\hat{f}(\alpha)$ be the indeterminates. Define $\text{Inf}_i(f) = \sum_{\alpha : \alpha_i > 0} \hat{f}(\alpha)^2$. For every constant $\beta > 0$, we have

$$\left\{ \sum_{\alpha} \hat{f}(\alpha)^2 \leq 1 \right\} \implies \mathbb{E} f(\mathbf{X})^4 \leq \mathbb{E} f(\mathbf{Y})^4 + \beta \cdot q^{5t} + \left(1 + \frac{1}{\beta}\right) \sum_{i \in [N]} \text{Inf}_i(f)^2.$$

**Proof.** Let $\mathbf{Z}^{(i)} = (\mathbf{X}_1, \ldots, \mathbf{X}_i, \mathbf{Y}_{i+1}, \ldots, \mathbf{Y}_N)$ for $i = 0, 1, 2, \ldots, N$. Also for each $i = 1, 2, \ldots, N$, let

$$E_i f(x) = \sum_{\alpha : \alpha_i = 0} \hat{f}(\alpha)x_\alpha,$$

and

$$D_i f(x) = \sum_{\alpha : \alpha_i > 0} \hat{f}(\alpha)x_\alpha.$$

For all $i = 1, 2, \ldots, N$, we have the following polynomial identities,

$$\mathbb{E} f(\mathbf{Z}^{(i)})^4 - \mathbb{E} f(\mathbf{Z}^{(i-1)})^4 = \mathbb{E} \left[ E_i f(\mathbf{Z}^{(i)}) + D_i f(\mathbf{Z}^{(i)}) \right]^4 - \mathbb{E} \left[ E_i f(\mathbf{Z}^{(i-1)}) + D_i f(\mathbf{Z}^{(i-1)}) \right]^4.$$
\[ E \left[ E_i f(Z^{(i)}) D_i f(Z^{(i)})^3 + D_i f(Z^{(i)})^4 \right] - E \left[ E_i f(Z^{(i-1)}) D_i f(Z^{(i-1)})^3 + D_i f(Z^{(i-1)})^4 \right], \]

where in the second identity we use the fact that the first two moments of \( X_i \) and \( Y_i \) match.

Now, for every constant \( \beta > 0 \), we use Fact 8.3.10 and the simple fact that \( \beta \cdot \frac{1}{2} t^2 \leq 0 \), and get

\[ \vdash_4 E f(Z^{(i)})^4 - E f(Z^{(i-1)})^4 \]

\[ \leq E \left[ \frac{\beta}{2} E_i f(Z^{(i)})^2 D_i f(Z^{(i)})^2 + \left( 1 + \frac{1}{2\beta} \right) D_i f(Z^{(i)})^4 \right] 
+ E \left[ \frac{\beta}{2} E_i f(Z^{(i-1)})^2 D_i f(Z^{(i-1)})^2 + \frac{1}{2\beta} D_i f(Z^{(i-1)})^4 \right]. \tag{9.1} \]

We view \( E_i f(Z^{(i)}) \) as a multilinear polynomial over \( \{\pm 1\} \) inputs with degree at most \( t\ell \), and view \( D_i f(Z^{(i)}) \) as a multilinear polynomial over the same set of inputs with degree at most \( t\ell \). Therefore, by hypercontractivity inequality (Theorem 8.4.3), we have

\[ \vdash_4 E E_i f(Z^{(i)}) D_i f(Z^{(i)})^2 \leq \frac{9}{t\ell} \left( E E_i f(Z^{(i)})^2 \right) \left( E D_i f(Z^{(i)})^2 \right) 
= \frac{9}{t\ell} \left( E E_i f(X)^2 \right) \left( E D_i f(X)^2 \right) = \frac{9}{t\ell} \left( E E_i f(X)^2 \right) \text{Inf}_i(f), \tag{9.2} \]

where the first equality is because the first two moments of \( Z^{(i)} \) match those of \( X \), and the second one is by the definition of \( \text{Inf}_i(f) \). Similarly, we have

\[ \vdash_4 E E_i f(Z^{(i-1)})^2 D_i f(Z^{(i-1)})^2 \leq \frac{9}{t\ell} \left( E E_i f(Z^{(i-1)})^2 \right) \left( E D_i f(Z^{(i-1)})^2 \right) 
= \frac{9}{t\ell} \left( E E_i f(X)^2 \right) \left( E D_i f(X)^2 \right) = \frac{9}{t\ell} \left( E E_i f(X)^2 \right) \text{Inf}_i(f). \tag{9.3} \]

We also view \( D_i f(Z^{(i)})^4 \) as a multilinear polynomial over \( \{\pm 1\} \) inputs with degree at most \( t\ell \). Using Theorem 8.4.3, we get

\[ \vdash_4 E D_i f(Z^{(i)})^4 \leq \frac{9}{t\ell} \left( E D_i f(Z^{(i)})^2 \right)^2 = \left( E D_i f(X)^2 \right)^2 = \left( \text{Inf}_i(f) \right)^2. \tag{9.4} \]

Again, the first equality is because the first two moments of \( Z^{(i)} \) match those of \( X \), while the second equality is by the definition of \( \text{Inf}_i(f) \). Similarly, we have

\[ \vdash_4 E D_i f(Z^{(i-1)})^4 \leq \frac{9}{t\ell} \left( E D_i f(Z^{(i-1)})^2 \right)^2 = \left( E D_i f(X)^2 \right)^2 = \left( \text{Inf}_i(f) \right)^2. \tag{9.5} \]
Now we incorporate (9.2), (9.3), (9.4), and (9.5) into (9.1), and get
\[ \vdash_4 \mathbb{E} f(Z^{(i)})^4 - \mathbb{E} f(Z^{(i-1)})^4 \leq \beta \cdot 9^\ell \left( \mathbb{E} E_i f(X)^2 \right) \text{Inf}_i(f) + \left( 1 + \frac{1}{\beta} \right) (\text{Inf}_i(f))^2. \]

Therefore, for every constant $\beta > 0$,
\[
\left\{ \sum_{\alpha} \hat{f}(\alpha)^2 \leq 1 \right\} \vdash_4 \mathbb{E} f(Z^{(i)})^4 - \mathbb{E} f(Z^{(i-1)})^4 \leq \beta \cdot 9^\ell \cdot \text{Inf}_i(f) + \left( 1 + \frac{1}{\beta} \right) (\text{Inf}_i(f))^2. \]

Now we sum up (9.6) over all $i = 1, 2, \ldots, N$, and get
\[
\left\{ \sum_{\alpha} \hat{f}(\alpha)^2 \leq 1 \right\} \vdash_4 \mathbb{E} f(X)^4 - \mathbb{E} f(Y)^4 \leq \beta \cdot 9^\ell \sum_{i \in [N]} \text{Inf}_i(f) + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} (\text{Inf}_i(f))^2
\[
\leq \beta \ell \cdot 9^\ell + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} (\text{Inf}_i(f))^2 \leq \beta \cdot q^5 + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} (\text{Inf}_i(f))^2. \]

\[ \square \]

The following corollary will be useful.

**Corollary 9.1.3.** Let $\hat{f}(\alpha)$ be the indeterminates. Define $\text{Inf}_i(f) = \sum_{\alpha: \alpha_i > 0} \hat{f}(\alpha)^2$. For every constant $\beta > 0$, we have
\[
\left\{ \sum_{\alpha} \hat{f}(\alpha)^2 \leq 1 \right\} \vdash_4 \mathbb{E} f(X)^4 \leq 9^\ell \left( \mathbb{E} f(X)^2 \right)^2 + \beta \cdot q^5 + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}_i(f)^2.
\]

**Proof.** Since $f(Y)$ is a multilinear polynomial over $\{\pm 1\}$ inputs with degree at most $\ell$, by hypercontractivity inequality (Theorem 8.4.3), we have
\[ \vdash_4 \mathbb{E} f(Y)^4 \leq 9^\ell \left( \mathbb{E} f(Y)^2 \right)^2. \]

We apply this together with the polynomial identity $\left( \mathbb{E} f(Y)^2 \right)^2 = \left( \mathbb{E} f(X)^2 \right)^2$ (since the first two moments of $X$ and $Y$ match) to Theorem 9.1.2, and get the desired conclusion. \[ \square \]
9.2 Analysis of the dictatorship test gadget for UNIQUEGAMES in the SOS proof system

In this section, we are going to prove the following theorem (Theorem 9.2.1) in the SOS proof system. The theorem can be viewed as an analog of the “Majority Is Stablest” theorem in [174], and serves the same purpose – to show that any function with no influential coordinates succeeds with very small probability in the dictatorship test for UNIQUEGAMES.

Let \( q = t^t \) for some \( t \in \mathbb{N} \) throughout this section.

**Theorem 9.2.1.** Let the entries in \( f(x) \) where \( x \in \mathbb{Z}_q^N \) be indeterminates. Let \( T_\rho \) be the operator on \( f : \mathbb{Z}_q^N \to \mathbb{R} \) such that \( T_\rho f(x) = \mathbb{E}_{y \sim \rho x} f(y) \), where \( y \sim \rho x \) means each \( y_i \) independently takes the value \( x_i \) with probability \( \rho \), and takes a uniform random value in \( \mathbb{Z}_q \) with probability \( (1 - \rho) \). Define \( \langle f, T_\rho f \rangle = \mathbb{E}_{x \in \mathbb{F}_q^N} f(x) T_\rho f(x) \).

For all constants \( \delta, \gamma \in [0, 1] \) and \( \beta > 0 \), we have

\[
\{ f(x)^2 \leq f(x) : \forall x, \mathbb{E} f \leq \delta \} \vdash_4 \langle f, T_{1-\gamma} f \rangle \leq \langle f, P_{>\lambda} f \rangle + \lambda \mathbb{E} f^2.
\]

**Proof.** For every constant \( \lambda : 0 < \lambda < 1 \), let \( P_{>\lambda} \) be the operator so that \( P_{>\lambda} f \) is the projection of \( f \) on to the eigenspace of \( T_{1-\gamma} \) with eigenvalue greater than \( \lambda \). The following SOS statement is easy to deduce.

\[
\vdash_2 \langle f, T_{1-\gamma} f \rangle \leq \langle f, P_{>\lambda} f \rangle + \lambda \mathbb{E} f^2.
\]

On the other hand, by Fact 8.3.13, we know that for every constant \( \epsilon > 0 \), we have

\[
\{ f(x)^2 \leq f(x) : \forall x \} \vdash_4 \langle f, P_{>\lambda} f \rangle \leq \frac{3\epsilon}{4} \mathbb{E} f + \frac{1}{4\epsilon^2} \mathbb{E} (P_{>\lambda} f)^4.
\]

Observe that both \( f \) and \( P_{>\lambda} f \) can be written in the multilinear forms of

\[
f(x) = \sum_{\alpha \in \mathbb{N}^N} \hat{f}(\alpha) \prod_{i=1}^n \chi_{\alpha_i}(x_i)
\]

\[
P_{>\lambda} f(x) = \sum_{\alpha \in \mathbb{N}^N, |\alpha| \leq \ln(\lambda 1/\gamma)} \hat{f}(\alpha) \prod_{i=1}^n \chi_{\alpha_i}(x_i),
\]

168
where \( \{ \chi_0 \equiv 1, \chi_1, \ldots, \chi_{q-1} \} \) is the set of characters for \( \{ \pm 1 \} \). Also, \( \hat{f}(\alpha) \) can be written as linear combinations of the indeterminates \( f(x) \); and \( \{ f(x)^2 \leq f(x) : \forall x, E f \leq \delta \} \vdash_2 E f^2 \leq 1 \) (when \( \delta \leq 1 \)), and \( \{ E f^2 \leq 1 \} \vdash_2 \sum_\alpha P_{> \lambda} f(\alpha)^2 \leq 1 \). Therefore, let

\[
\ell \overset{\text{def}}{=} \frac{2}{\gamma} \log \frac{1}{\lambda}
\]

which upper bounds the degree of \( P_{> \lambda} f \), by Corollary 9.1.3 for every constant \( \beta > 0 \),

\[
\{ f(x)^2 \leq f(x) : \forall x, E f \leq \delta \} \vdash_4 9^\ell \left( E (P_{> \lambda} f)^2 \right)^2 + \beta \cdot q^{5\ell} + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}^\leq_{\ell}(f)^2,
\]

(9.9)

where \( \text{Inf}^\leq_{\ell}(f) = \sum_{|\alpha| \leq \ell} \hat{f}(\alpha)^2 \) is the low-degree influence of \( f \). Since

\[
\vdash_4 \left( E (P_{> \lambda} f)^2 \right)^2 \leq \left( E f^2 \right)^2,
\]

and

\[
\{ f(x)^2 \leq f(x) : \forall x \} \vdash_2 E f^2 \leq E f,
\]

(9.10)

together with (9.8) and (9.9), we get for all constants \( \epsilon, \beta > 0 \),

\[
\{ f(x)^2 \leq f(x) : \forall x, E f \leq \delta \} \vdash_4 \langle f, P_{> \lambda} f \rangle \leq \frac{3\epsilon}{4} E f + \frac{9^\ell}{4 \epsilon^3} (E f)^2 + \frac{\beta \cdot q^{5\ell}}{4 \epsilon^3} + \frac{1}{4 \epsilon^3} \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}^\leq_{\ell}(f)^2
\]

\[
\leq \frac{3\epsilon \delta}{4} + \frac{9^\ell \delta^2}{4 \epsilon^3} + \frac{\beta \cdot q^{5\ell}}{4 \epsilon^3} + \frac{1}{4 \epsilon^3} \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}^\leq_{\ell}(f)^2.
\]

(9.11)

We combine (9.11), (9.10), and (9.7), and get

\[
\{ f(x)^2 \leq f(x) : \forall x, E f \leq \delta \} \vdash_4 \langle f, T_{1-\gamma} f \rangle \leq \lambda \delta + \frac{3\epsilon \delta}{4} + \frac{9^\ell \delta^2}{4 \epsilon^3} + \frac{\beta \cdot q^{5\ell}}{4 \epsilon^3} + \frac{1}{4 \epsilon^3} \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}^\leq_{\ell}(f)^2.
\]

(9.12)

Now we set \( \lambda = \delta^{.01\gamma} \) (and therefore \( \ell = .02 \log \frac{1}{\delta} \)), and \( \epsilon = \delta^{-1} \). We have \( \frac{3\epsilon \delta}{4} + \frac{9^\ell \delta^2}{4 \epsilon^3} \leq \delta^{1.1} \). Therefore, (9.12) implies
\{ f(x)^2 \leq f(x) : \forall x, \mathbb{E} f \leq \delta \} \vdash 4

\langle f, T_{1-\gamma} f \rangle \leq \delta^{1+0.1\gamma} + \delta^{1.1} + \frac{1}{4\delta^{0.3}} \left( \beta \cdot q^{1.1 \log \frac{1}{\delta}} + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}_i^{\leq 0.2 \log \frac{1}{\delta}} (f)^2 \right)

for every constant \( \beta > 0. \)

9.3 The KV UNIQUEGAMES instance and its SOS analysis

Let us recall the UNIQUEGAMES problem with label-size \( N \in \mathbb{N}^+ \). Given is a regular weighted graph \( G = (\mathcal{V}, \mathcal{E}) \) (self-loops allowed) with weights summing to 1. Also, given for each edge \((u, v)\) is a permutation \( \pi_{uv} : [N] \to [N] \). We write \((u, v, \pi) \sim \mathcal{E}\) to denote that edge \((u, v)\) with permutation \( \pi = \pi_{uv} \) is chosen with probability equal to its edge weight. The goal is to give a labeling \( F : \mathcal{V} \to [N] \) so as to maximize

\[
\Pr((u, v, \pi) \sim \mathcal{E} | F(u) = F(v)).
\]

The natural polynomial optimization formulation has an indeterminate \( X_{u,i} \) for each \( u \in \mathcal{V}, i \in [k] \):

\[
\max_{(u, v, \pi) \sim \mathcal{E}} \mathbb{E} \left[ \sum_{i=1}^{N} X_{u,i} X_{v,\pi(i)} \right] = \mathbb{E} \left[ \sum_{u \in \mathcal{V}} X_{u,i} \cdot \mathbb{E}_{\pi \sim u} [X_{v,\pi(i)}] \right]
\]

s.t.

\[
X_{u,i} = X_{u,i}, \quad \forall u \in \mathcal{V}, i \in [N],
\]

\[
\sum_{i=1}^{N} X_{u,i} = 1, \quad \forall u \in \mathcal{V},
\]

where we write \((v, \pi) \sim u\) in place of \((u, v, \pi) \sim \mathcal{E}|_{u=u}\) for brevity. Thus the degree-d SOS SDP hierarchy will use binary search to compute the smallest \( \beta \) for which

\[
\{ X_{u,i} = X_{u,i} : \forall u \in \mathcal{V}, i \in [N] \} \cup \{ \sum_{i=1}^{N} X_{u,i} = 1 : \forall u \in \mathcal{V} \}
\]

\[
\cup \left\{ \mathbb{E}_{u \in \mathcal{V}} \left[ \sum_{i=1}^{N} X_{u,i} \cdot \mathbb{E}_{(v, \pi) \sim u} [X_{v,\pi(i)}] \right] \geq \beta \right\} \vdash_{d} -1 \geq 0. \tag{9.13}
\]

The Khot–Vishnoi UNIQUEGAMES instance is defined as follows. Fix \( k \in \mathbb{N} \). Let \( \mathcal{F} = \{ \{\pm 1\}^k \to \{\pm 1\} \} \) be the family of Boolean functions on \( \{\pm 1\}^k \). Consider the following equivalence relation \( \equiv \) on \( \mathcal{F} \). For any two Boolean functions \( f, g \in \mathcal{F} \), we say \( f \equiv g \) if and only if there is an \( S \subseteq [k] \) such that \( f = g x_S \) where \( x_S(i) = \prod_{i \in S} x_i \). Now this relation partitions \( \mathcal{F} \) into equivalence classes \( \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m \). We denote by [\( \left\lfloor \mathcal{P}_i \right\rfloor \) an
arbitrary representative in $\mathcal{P}_i$. For each $f \in \mathcal{F}$, let $\mathcal{P}(f)$ be the $\mathcal{P}_i$ which contains $f$. Now we are ready to describe the UNIQUEGAMES instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with parameter $\eta > 0$. The vertex set $\mathcal{V}$ is simply the set of equivalence classes. The label set is $[N]$. For each vertex $\mathcal{P}_i$, we decide an arbitrary one-to-one correspondence between the elements in $\mathcal{P}_i$ and $[N]$ so that the permutations $\pi$ can be described as mappings between the elements in $\mathcal{P}_i$ and $\mathcal{P}_j$. We define the edge set together with the weight distribution $(\mathcal{P}_i, \mathcal{P}_j, \pi) \sim \mathcal{E}$ as follows,

- choose $f$ as a uniform random function in $\mathcal{F}$, and let $\mathcal{P}_i = \mathcal{P}(f)$;
- choose $g \sim 1 - \eta f$, i.e. we obtain $g$ by flipping each entry of $f$ independently with probability $2\eta$. Let $\mathcal{P}_j = \mathcal{P}(g)$;
- finally, since $\mathcal{P}_i = \{f \chi_S : \forall S \subseteq [k]\}$ and $\mathcal{P}_j = \{g \chi_S : \forall S \subseteq [k]\}$, we define $\pi: \mathcal{P}_i \rightarrow \mathcal{P}_j$ as $\pi(f \chi_S) = g \chi_S$ for every $S \subseteq [k]$.

We are going to show degree-4 SOS proofs to the statement that the UNIQUEGAMES instance $\mathcal{G}$ defined as above does not have a great solution. In fact we are going to show something stronger than (9.13), in the sense that we only assume $X^2_{u,i} \leq X_{u,i}$ and $E_{u \in \mathcal{V}} E_{i \in [N]} X_{u,i} \leq 1/N$, and we will give an SOS proof instead of an SOS refutation. We prove

**Theorem 9.3.1.** Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as the instance described above, for every constant $\delta \in (0, 1)$, we have

$$\{X^2_{u,i} \leq X_{u,i} : \forall u \in \mathcal{V}, i \in [N]\} \cup \{ E_{u \in \mathcal{V}} E_{i \in [N]} X_{u,i} \leq \delta \} \vdash_4$$

$$E_{u \in \mathcal{V}} \left[ \sum_{i=1}^{N} X_{u,i} \cdot \sum_{(v, \pi) \sim u} \left[ X_{v,\pi(i)} \right] \right] \leq N \cdot \delta^{1+\Omega(\eta)}.$$

Therefore, when $\delta = 1/N$, we have

$$\{X^2_{u,i} \leq X_{u,i} : \forall u \in \mathcal{V}, i \in [N]\} \cup \{ E_{u \in \mathcal{V}} E_{i \in [N]} X_{u,i} \leq \frac{1}{N} \} \vdash_4$$

$$E_{u \in \mathcal{V}} \left[ \sum_{i=1}^{N} X_{u,i} \cdot \sum_{(v, \pi) \sim u} \left[ X_{v,\pi(i)} \right] \right] \leq N^{-\Omega(\eta)}.$$

**Proof.** We first perform some notational manipulations in order to make the representation easier. Observe that each pair $(u, i) \in \mathcal{V} \times [N]$ is uniquely mapped to a function $f \in \mathcal{F}$. Therefore, we will talk about $X(f) : f \in \mathcal{F}$ instead of $X_{u,i} : (u, i) \in \mathcal{V} \times [N]$. Given
$u \in V$ and $i \in [N]$ (therefore $f \in F$), let $g \in F$ be the function corresponding to $(v, \pi(i))$ where $(v, \pi) \sim u$. Observe that $g$ in fact follows the distribution $g \sim 1 - \eta f$, and therefore

$$
\mathbb{E}_{u \in V} \left[ \sum_{i=1}^{N} X_{u,i} \cdot \mathbb{E}_{(v,\pi) \sim u} [X_{v,\pi(i)}] \right] = N \mathbb{E}_{f \in F, g \sim 1 - \eta f} X(f)X(g)
$$

(9.14)

if we assign $X(f)$ according to $X_{u,i}$.

Therefore, to prove the theorem, it suffices to show that

$$
\{ X(f)^2 \leq X(f) : \forall f \in F \} \cup \{ \mathbb{E}_{f \in F} X(f) \leq \delta \} \vdash 4 \langle X, T_{1-\eta}X \rangle \leq \delta^{1+\Omega(n)}.
$$

(9.15)

where $T_{1-\eta}$ is the operator so that $T_{1-\eta}X(f) = \mathbb{E}_{g \sim 1 - \eta f} X(g)$ and we define $\langle X, T_{1-\eta}X \rangle = \mathbb{E}_{f \in F} X(f)T_{1-\eta}X(f)$.

The rest of the proof is to prove (9.15). For every $d \in \mathbb{N}$, it is easy to show that

$$
\vdash_2 \langle X, T_{1-\eta}X \rangle \leq \langle X, P_{\geq (1-\eta)^d}X \rangle + (1-\eta)^d \mathbb{E}_{f \in F} X(f)^2,
$$

(9.16)

where $P_{\geq (1-\eta)^d}$ is the operator so that $P_{\geq (1-\eta)^d}X$ is the projection of $X$ on to the span of the characters $\chi_S$ where $|S| \leq d$. Now we apply Fact 8.3.13 to (9.16) and have that for every constant $\epsilon > 0$, we have

$$
\vdash_4 \langle X, T_{1-\eta}X \rangle \leq 3\epsilon 4 \mathbb{E}_{f \in F} X(f) + \frac{1}{4\epsilon^3} \mathbb{E}_{f \in F} (P_{\geq (1-\eta)^d}X(f))^4 + (1-\eta)^d \mathbb{E}_{f \in F} X(f)^2.
$$

(9.17)

Observe that $P_{\geq (1-\eta)^d}X(f)$ is a multilinear polynomial of $f(x)$ with degree at most $d$. Therefore, by hypercontractivity inequality (Theorem 8.4.3) we have

$$
\vdash_4 \langle X, T_{1-\eta}X \rangle \leq 3\epsilon 4 \mathbb{E}_{f \in F} X(f) + \frac{9d}{4\epsilon^3} \left( \mathbb{E}_{f \in F} (P_{\geq (1-\eta)^d}X(f))^2 \right)^2 + (1-\eta)^d \mathbb{E}_{f \in F} X(f)^2.
$$

(9.18)

With some simple SOS facts, (9.18) implies

$$
\{ X(f)^2 \leq X(f) : \forall f \in F \} \cup \{ \mathbb{E}_{f \in F} X(f) \leq \delta \} \vdash_4 \langle X, T_{1-\eta}X \rangle \leq \delta \left( \frac{3\epsilon}{4} + (1-\eta)^d \right) + \frac{9d\delta^2}{4\epsilon^3}
$$

for every $\epsilon > 0$ and $d \in \mathbb{N}$.

We choose $d = \left\lceil \frac{\ln(1/\delta)}{\ln 81} \right\rceil$ and $\epsilon = N^{-1}$ to establish (9.15). 

\[ 172 \]
Corollary 9.3.2. Given $G = (V, E)$ as the instance in Theorem 9.3.1, for every constant $\delta \in (0, 1)$, we have

$$\{X_{u,i}^2 \leq X_{u,i} : \forall u \in V, i \in [N] \} \cup \{ \mathbb{E}_{u \in V} \mathbb{E}_{i \in [N]} X_{u,i} \leq \delta \} \vdash 4$$

$$\mathbb{E}_{u \in V} \left[ \sum_{i=1}^{N} \left( \mathbb{E}_{(v, \pi) \sim u} [X_{v, \pi(i)}] \right)^2 \right] \leq N \cdot \delta^{1+\Omega(\eta)}.$$

Proof. The proof of this corollary is almost the same as that of Theorem 9.3.1. The only difference is that in (9.14), instead we observe that

$$\mathbb{E}_{u \in V} \left[ \sum_{i=1}^{N} \left( \mathbb{E}_{(v, \pi) \sim u} [X_{v, \pi(i)}] \right)^2 \right] = \mathbb{E}_{f \in F} \mathbb{E}_{g \sim 1-\eta} X(g^2),$$

and in (9.15), we turn to prove

$$\{X(f)^2 \leq X(f) : \forall f \in F \} \cup \{ \mathbb{E}_{f \in F} X(f) \leq \frac{1}{N} \} \vdash 4$$

$$\langle T_{1-\eta}X, T_{1-\eta}X \rangle = \langle X, T_{(1-\eta)^2}X \rangle \leq \delta^{1+\Omega(\eta)}.$$

instead.

In the rest of the proof, we use $(1 - \eta)^2$ whenever $(1 - \eta)$ occurs, and reach the same conclusion.

9.4 Influence decoding and putting everything together

Now let us recall that the UniqueGames instances we are interested in are obtained by composing the KKMO “noise stability” reduction from [141] with the KV integrality gap instances analyzed in the previous section. Let us fix an UniqueGames instance $G = (V, E)$ with label-size $N$ (in this section, we are interested in the KV integrality gap instance presented in the previous section with parameter $\eta$). The KKMO reduction, parameterized by $\gamma \in [0, 1]$, creates a new UniqueGames instance $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with label-size $q$ in the vertex set $V$, there is a vertex $w_{u,x}$ for each $u \in V$ and each $x \in \mathbb{Z}_q^N$. For each $c \in \mathbb{Z}_q$, let $\sigma_c(x) = x + c$. The probability distribution $E$ on edges and permutations is given as follows.
• draw \( u \in \mathcal{V} \) uniformly;
• independently draw \((u, v_1, \pi_1)\) and \((u, v_2, \pi_2)\) from the marginal of \( E \) which is first vertex \( u \);
• draw \( x \in \mathbb{Z}_q^N \) uniformly;
• draw \( y \sim_{1-\gamma} x \), i.e. each \( y_i \) independently becomes \( x_i \) with probability \( 1 - \gamma \) and a uniform random element from \( \mathbb{Z}_q \) with probability \( \gamma \);
• pick a random element \( c \in \mathbb{Z}_q \);
• output the edge \((w_{u_1.(x \circ \pi_1)}, w_{u_2.(y-c) \circ \pi_2})\) together with \( \sigma_c \) as the permutation.

Here \( x \circ \pi \) is defined to be the string in \( \mathbb{Z}_q^N \) such that \((x \circ \pi)_i = x_{\pi(i)}\) for all \( i \in [N] \); and \( y - c \) is the string in \( \mathbb{Z}_q^N \) such that \((y - c)_i = y_i - c\) for all \( i \in [N] \).

In this section we will prove the following theorem using the tools developed in the previous sections. The theorem directly implies Theorem 9.0.8 (when \( q \) is sufficiently large and \( \log N \gg (\log q)^2/\eta \)).

**Theorem 9.4.1.** For each \( u \in \mathcal{V}, x \in \mathbb{Z}_q^N, a \in \mathbb{Z}_q \), let \( f_u^{(a)}(x) \) be the indeterminate which is intended to be the indicate variable for the event that \( w_{u,x} \) takes the label \( a \). Let

\[
A = \left\{ f_u^{(a)}(x)^2 = f_u^{(a)}(x) : \forall u \in \mathcal{V}, x \in \mathbb{Z}_q^N, a \in \mathbb{Z}_q \right\} \cup \left\{ \sum_{a \in \mathbb{Z}_q} f_u^{(a)}(x) = 1 : \forall u \in \mathcal{V}, x \in \mathbb{Z}_q^N \right\}.
\]

We have

\[
A \vdash_8 \text{val}(f) \overset{\text{def}}{=} \mathbb{E}_{u \in \mathcal{V}} \mathbb{E}_{(v_1, \pi_1) \sim u} \mathbb{E}_{x \in \mathbb{Z}_q^N} \mathbb{E}_{c \in \mathbb{Z}_q} \sum_{a \in \mathbb{Z}_q} f_{v_1}^{(a)}(x \circ \pi_1) f_{v_2}^{(a+c)}((y - c) \circ \pi_2)
\]

\[
\leq q^{-\Omega(\gamma)} + q^{O(\log q)} N^{-\Omega(\eta)}.
\]

**Proof.** For each \( u \in \mathcal{V}, x \in \mathbb{Z}_q^N, a \in \mathbb{Z}_q \), let \( \tilde{f}_u^{(a)}(x) = \mathbb{E}_{c \in \mathbb{Z}_q} f_u^{(a+c)}(x - c) \); let \( h_u^{(a)}(x) = \mathbb{E}_{(v, \pi) \sim u} \tilde{f}_v^{(a)}(x \circ \pi^{-1}) \). Let \( T_{1-\gamma} \) be the operator such that \( T_{1-\gamma} h(x) = \mathbb{E}_{y \sim_{1-\gamma} x} h(y) \) when \( h \) is a function defined over \( \mathbb{Z}_q^N \). The following polynomial identity is easy to verify.

\[
\text{val}(f) = \sum_{a \in \mathbb{Z}_q} \mathbb{E}_{u \in \mathcal{V}} \langle h_u^{(a)}, T_{1-\gamma} h_u^{(a)} \rangle.
\]
Therefore, to prove the theorem, we only need to show that for every \( a \in \mathbb{Z}_q \),
\[
A \vdash_4 \mathbb{E}_{u \in V} \langle h_u^{(a)}, T_{1-\gamma} h_u^{(a)} \rangle \leq q^{-1-\Omega(\gamma)} + q^{O(\log q) N^{-\Omega(\eta)}}. \tag{9.19}
\]

Now fix an arbitrary \( a \in \mathbb{Z}_q \), by the definition of \( h \) (and using Fact 8.3.15) we have
\[
\forall u \in V, x \in \mathbb{Z}_q^n, A \vdash 2 h_u^{(a)}(x)^2 \leq h_u^{(a)}(x), \tag{9.20}
\]
and
\[
\forall u \in V, A \vdash \mathbb{E}_{x \in \mathbb{Z}_q^n} h_u^{(a)}(x) = \frac{1}{q}. \tag{9.21}
\]

Therefore, by Theorem 9.2.1, for every \( u \in V \) and every constant \( \beta > 0 \), we have²
\[
A \vdash_4 \langle h_u^{(a)}, T_{1-\gamma} h_u^{(a)} \rangle \leq \frac{1}{q^{1+\Omega(\gamma)}} + \frac{1}{4q^3} \left( \beta \cdot q^{1+\log q} + \left( 1 + \frac{1}{\beta} \right) \sum_{i \in [N]} \text{Inf}_{i}^{\leq 0.02 \log q} (h_u^{(a)})^2 \right). 
\tag{9.22}
\]

Therefore, for every constant \( \beta > 0 \),
\[
A \vdash_4 \mathbb{E}_{u \in V} \langle h_u^{(a)}, T_{1-\gamma} h_u^{(a)} \rangle \\
\leq \frac{1}{q^{1+\Omega(\gamma)}} + \frac{1}{4q^3} \left( \beta \cdot q^{1+\log q} + \left( 1 + \frac{1}{\beta} \right) \mathbb{E}_{u \in V} \sum_{i \in [N]} \text{Inf}_{i}^{\leq 0.02 \log q} (h_u^{(a)})^2 \right). \tag{9.23}
\]

Since (for every \( \ell \in \mathbb{N} \))
\[
\text{Inf}^{\leq \ell}_{i} (h_u^{(a)}) = \sum_{\alpha:|\alpha| \leq \ell, \alpha_i > 0} \widehat{h}_u^{(a)}(\alpha)^2 = \sum_{\alpha:|\alpha| \leq \ell, \alpha_i > 0} \left( \mathbb{E}_{(v, \pi) \sim u} \widehat{f}_{v}^{(a)}(\alpha \circ \pi) \right)^2,
\]
by Fact 8.3.15 we have
\[
\vdash_2 \text{Inf}^{\leq \ell}_{i} (h_u^{(a)}) \leq \sum_{\alpha:|\alpha| \leq \ell, \alpha_i > 0} \mathbb{E}_{(v, \pi) \sim u} \widehat{f}_{v}^{(a)}(\alpha \circ \pi)^2 = \mathbb{E}_{(v, \pi) \sim u} \text{Inf}^{\leq \ell}_{\pi(i)} (\tilde{f}_v^{(a)}). \tag{9.24}
\]

²This would require degree 6 if we apply Fact 8.2.3 directly. However, if carefully examining the details, one can see that degree-4 SOS proof suffices.
Therefore,

\[ \vdash_4 E_{u \in V} \sum_{i \in [N]} \inf_{E \sim u}^{\leq 0.02 \log q} (h_u^{(a)})^2 \leq \sum_{i \in [N]} E_{v, \pi} \left( E_{u \sim u} \inf_{E \sim u}^{\leq 0.02 \log q} (f_v^{(a)}) \right)^2 \]  \hspace{1cm} (9.25)

On the other hand, for every \( i \in [N], v \in \mathcal{V} \), we have the following proof (whose proof will be deferred to the end of this section).

**Claim 9.4.2.** \( A \vdash_{4} \inf_{i}^{\leq 0.02 \log q} (f_v^{(a)}) \leq \inf_{i}^{\leq 0.02 \log q} (f_v^{(a)}) \).

Since

\[ E_{v \in V} E_{i \in [N]} \inf_{i}^{\leq 0.02 \log q} f_v^{(a)} = \frac{1}{N} E_{v \in V} \sum_{\alpha:|\alpha| \leq 0.02 \log q} \alpha f_v^{(a)}(\alpha)^2, \]

We also have

\[ \vdash_2 E_{v \in V} E_{i \in [N]} \inf_{i}^{\leq 0.02 \log q} f_v^{(a)} \leq \frac{0.02 \log q}{N} E_{v \in V} E_{x} f_v^{(a)}(x)^2. \]  \hspace{1cm} (9.26)

It’s easy to deduce that \( \vdash_2 f_v^{(a)}(x)^2 \leq f_v^{(a)}(x) \) (for all \( v \in \mathcal{V}, x \in \mathbb{Z}_q^N \)). Therefore, (9.26) implies

\[ \vdash_2 E_{v \in V} E_{i \in [N]} \inf_{i}^{\leq 0.02 \log q} f_v^{(a)} \leq \frac{0.02 \log q}{N} E_{v \in V} E_{x} f_v^{(a)}(x). \]  \hspace{1cm} (9.27)

Therefore,

\[ A \vdash_2 E_{v \in V} E_{i \in [N]} \inf_{i}^{\leq 0.02 \log q} f_v^{(a)} \leq \frac{0.02 \log q}{N q}. \]  \hspace{1cm} (9.28)

**Claim 9.4.2** (9.28), Corollary 9.3.2, and Fact 8.2.3 imply

\[ A \vdash_8 \sum_{i \in [N]} E_{u \in V} \left( E_{v, \pi \sim u} \inf_{E \sim u}^{\leq 0.02 \log q} (f_v^{(a)}) \right)^2 \leq N \left( \frac{0.02 \log q}{N q} \right)^{1+\Omega(\eta)} \leq \log q \frac{1}{N^{\Omega(\eta)} q}. \]  \hspace{1cm} (9.29)

Now, we put (9.23), (9.25), and (9.29) together, and get (for every \( \beta > 0 \))

\[ A \vdash_8 \frac{1}{q^{1+\Omega(\gamma)}} + \frac{1}{4q^3} \left( \beta \cdot q^{-1 \log q} + \left( 1 + \frac{1}{\beta} \right) \frac{\log q}{N^{\Omega(\eta)} q} \right). \]  \hspace{1cm} (9.30)

We establish (9.19) by choosing \( \beta = N^{c \eta} \) where \( c > 0 \) is a constant depending on the coefficient hidden in the \( \Omega(\eta) \) notation in (9.30). \( \square \)
9.4.1 Proof of Claim 9.4.2

It remains to prove Claim 9.4.2.

Proof of Claim 9.4.2. By fact Fact 8.3.5, we know that

\[ A \vdash f_u^c(x) \leq 1, \]

and hence

\[ A \vdash f_u^c(x)^2 \leq 1, \]

for all \( u \in V, x \in \mathbb{Z}_q^N, c \in \mathbb{Z}_q \). Therefore, by Fact 8.3.15, we have

\[ A \vdash \tilde{f}_v^a(x)^2 \leq 1 \] (9.31)

for all \( x \in \mathbb{Z}_q^N \).

Since \( \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)} = \sum_{\alpha : \alpha_i > 0, |\alpha| \leq 0.02 \log q} \tilde{f}_v^a(\alpha)^2 \), we have

\[ A \vdash \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)} \leq \sum_{\alpha} \tilde{f}_v^a(\alpha)^2 = \mathbb{E}_{x} \tilde{f}_v^a(x)^2. \]

Together with (9.31), we have

\[ A \vdash \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)} \leq 1. \] (9.32)

On the other hand, since \( \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)} \) is a sum of squared linear forms, we can multiply \( \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)} \) on the both sides of (9.32), and get

\[ A \vdash \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)}^2 \leq \text{Inf}_i^{\leq 0.02 \log q (\tilde{f}_v^a)}, \]

which is the desired statement. \( \square \)

9.5 Refuting Instances based on Short Code

In this section, we consider the UNIQUEGAMES instances obtained by composing the KV UNIQUEGAMES integrality gap instances with the “short code” gadget reduction constructed in [35].

The following analog of Theorem 9.0.8 holds.
Theorem 9.5.1. Let $G$ be an $n$-variable Unique Games instance with label-size $q$ obtained by composing the KV “quotient noisy cube” Unique Games integrality gap instances with the “short code” gadget reduction constructed in [35] so that the best assignment to $G$’s variables satisfies at most an $\epsilon$ fraction of the constraints. When $\epsilon$ is sufficiently small and $n$ is sufficiently large, there is a degree-8 SOS refutation for the statement that the best assignment to $G$’s variables satisfy at least $1/100$ fraction of the constraints.

The proof of Theorem 9.5.1 is almost literally the same as the proof of Theorem 9.0.8. In the following, we sketch the main arguments why the proof doesn’t have to change. First, several of the results of the previous sections apply to general graphs and instances of Unique Games. In particular, the proofs in Section 9.3 does not need to change. The proofs in 9.2 will still go through assuming the invariance principle result (in Section 9.1) for the type of graphs we are interested in; and the proofs in Section 9.4 apply to general gadget-composed instances of unique games assuming a “Majority Is Stablest” result for the gadget. In fact, the only parts that require further justification are the invariance principle (the proofs in Section 9.1) and hypercontractivity bound (Theorem 8.4.3). Both the invariance principle and the hypercontractivity bound are about the fourth moment of a low-degree Fourier polynomial (whose coefficients are fictitious random variables). For the construction of [35], we need to argue about the fourth moment with respect to a different distribution over inputs. (Instead of the uniform distribution, [35] considers a distribution over inputs related to the Reed–Muller code.) However, this distribution happens to be $k$-wise independent for $k/4$ larger than the degree of our Fourier polynomial. Hence, as a degree-4 polynomial in Fourier coefficients, the fourth moment with respect to the [35]-input distribution is the same as with respect to the uniform distribution, which considered here.
Chapter 10

SOS proofs of SSE in the Noisy Hypercube, KKL, and the analysis of the DKSV \textsc{BalancedSeparator} instances

In this chapter, we given an SOS proof for the KKL theorem, and use this to show that degree-4 SOS proofs certify the lower bound for the optimal value of the DKSV \textsc{BalancedSeparator} instances up to only a constant factor.

10.1 An SOS proof of small-set expansion in the noisy hypercube

The following well-known theorem concerning small-set expansion (SSE) in the hypercube is due to Kahn, Kalai, and Linial \cite{129}:

Noisy Hypercube SSE Theorem. Let $f : \{-1, 1\}^n \rightarrow \{-1, 0, 1\}$. Then for any $0 \leq \rho \leq 1$,

$$\text{Stab}_\rho[f] \leq E[|f|^2]^{2/(1+\rho)},$$

where $\text{Stab}_\rho[f]$ denotes $\langle f, T_\rho f \rangle = \|T_\sqrt{\rho} f\|^2_2$.

Proof:

$$\text{Stab}_\rho[f] = \|T_\sqrt{\rho} f\|^2 \leq \|f\|^2_{1+\rho} = E[|f|^{1+\rho}]^{2/(1+\rho)} = E[|f|^2]^{2/(1+\rho)},$$

179
where the inequality is hypercontractivity (the Hölder dual of Theorem 8.4.1).

We remark on two special cases:

$$\text{Stab}_{\frac{1}{3}}[f] \leq E[f^2]^{3/2}, \quad \text{Stab}_{\frac{1}{\sqrt{3}}}[f] \leq E[f^2]^{3-\sqrt{3}} \leq E[f^2]^{1.2679}.$$ 

We do not know how to obtain a low-degree SOS proof of either inequality. Nevertheless, we come close in the following theorem. We remark that its proof is very similar to the one in Section 9.3.

**Theorem 10.1.1.** (SOS proof of a weakened special case of the Noisy Hypercube SSE Theorem.)

Let $n \in \mathbb{N}$, and for each $x \in \{-1, 1\}^n$ let $f(x)$ be an indeterminate. Then for any real $\epsilon > 0$,

$$\{ f(x) = f(x)^3 : \forall x \} \vdash \text{Stab}_{\frac{1}{\sqrt{3}}}[f] \leq E_x[f(x)^2] \left( \frac{3\epsilon}{4} + \frac{1}{4\epsilon^3} E_x[f(x)^2] \right).$$

**Remark 10.1.1.** From this we can deduce that if $f : \{-1, 1\}^n \to \{-1, 0, 1\}$ is an ordinary function then $\text{Stab}_{\frac{1}{\sqrt{3}}}[f] \leq E[f(x)^2]^{5/4}$, by taking $\epsilon = E[f(x)^2]^{1/4}$.

**Proof.** From Fact 8.3.12 (and the trivial fact $Y = Y^3 \vdash Y p^2 = Y^4$) we may easily deduce

$$Y = Y^3 \vdash \ Y Z \leq \frac{3\epsilon}{4} Y^2 + \frac{1}{4\epsilon^3} Z^4.$$ 

Since $\text{Stab}_{\frac{1}{\sqrt{3}}}[f] = E_x[f(x) T_{\frac{1}{\sqrt{3}}} f(x)]$ we may therefore obtain

$$\{ f(x) = f(x)^3 : \forall x \} \vdash \text{Stab}_{\frac{1}{\sqrt{3}}}[f] \leq \frac{3\epsilon}{4} E_x[f(x)^2] + \frac{1}{4\epsilon^3} E_x[T_{\frac{1}{\sqrt{3}}} f(x)^4].$$

The result now follows from Theorem 8.4.4.

10.2 The KKL Theorem

With the Noisy Hypercube SSE Theorem in hand, we can now give an SOS proof of the famed KKL Theorem [129], the key ingredient in the analysis of the DKSV BALANCED-SEPARATOR instances.
**Theorem 10.2.1.** (SOS proof of the KKL Theorem.)

Let \( n \in \mathbb{N} \), and for each \( x \in \{-1, 1\}^n \) let \( f(x) \) be an indeterminate. Let \( \tau \) be an indeterminate. Then for any reals \( \epsilon > 0 \), \( K \geq 2 \),

\[
\{ f(x)^2 = 1 : \forall x \} \cup \{ \text{Inf}_i[f] \leq \tau : \forall i \in [n] \} \vdash_{4} \text{Var}[f] \leq \left( \frac{\sqrt{\tau}K-1}{K} \left( \frac{3\epsilon}{4} + \frac{\tau}{4\epsilon} \right) + \frac{1}{K} \right) I[f].
\]

**Remark 10.2.1.** From this we can deduce that if \( f : \{-1, 1\}^n \rightarrow \{-1, 1\} \) is an ordinary function and \( \tau \leq \frac{1}{9} \) is a positive real such that \( \text{Inf}_i[f] \leq \tau \) for all \( i \), then \( I[f] \geq \frac{1}{2} \log_9(\frac{9}{\tau}) \cdot \text{Var}[f] \). This follows by taking \( \epsilon = \tau^{1/4} \) and \( K = \log_9(\frac{9}{\tau}) \).

**Proof.** We may apply Theorem 10.1.1 to each of the derivative “functions”

\[
D_i f(x) = \frac{f(x^{(i\rightarrow 1)}) - f(x^{(i\rightarrow -1)})}{2}.
\]

(These are actually sets of indeterminates, each of which is a homogeneous linear form in the indeterminates \( f(x) \).) We can obtain the hypothesis \( D_i f(x) = D_i f(x)^3 \) from the hypotheses \( f(x)^2 = 1 \) via Fact 8.3.6. We deduce

\[
\{ f(x)^2 = 1 : \forall x \} \vdash_{4} \text{Stab}_{\frac{1}{\sqrt{3}}} [D_i f] \leq E[D_i f(x)] (\frac{3\epsilon}{4} + \frac{1}{4\epsilon} E[D_i f(x)^2]) \]

\[
\Leftrightarrow \sum_{S \ni i} (\frac{1}{\sqrt{3}})^{|S|-1} \tau(S)^2 \leq \text{Inf}_i[f] (\frac{3\epsilon}{4} + \frac{1}{4\epsilon} \text{Inf}_i[f])
\]

for each \( i \in [n] \). Further, since \( \text{Inf}_i[f] \) is SOS and of degree 2 we have

\[
\text{Inf}_i[f] \leq \tau \vdash_{4} \text{Inf}_i[f] \cdot (\frac{\tau}{4\epsilon^3} - \frac{1}{4\epsilon} \text{Inf}_i[f]) \geq 0.
\]

Adding the previous two deductions yields

\[
\{ f(x)^2 = 1 : \forall x \} \cup \{ \text{Inf}_i[f] \leq \tau : \forall i \in [n] \} \vdash_{4} \sum_{S \ni i} (\frac{1}{\sqrt{3}})^{|S|-1} \tau(S)^2 \leq \text{Inf}_i[f] (\frac{3\epsilon}{4} + \frac{\tau}{4\epsilon})
\]

for each \( i \). Now adding over all \( i \in [n] \) gives

\[
\{ f(x)^2 = 1 : \forall x \} \cup \{ \text{Inf}_i[f] \leq \tau : \forall i \in [n] \} \vdash_{4} \sum_{S \subseteq [n]} |S| (\frac{1}{\sqrt{3}})^{|S|-1} \tau(S)^2 \leq (\frac{3\epsilon}{4} + \frac{\tau}{4\epsilon}) I[f].
\]

Moreover, since \( s(\frac{1}{\sqrt{3}})^{s-1} \geq K(\frac{1}{\sqrt{3}})^{K-1} - s(\frac{1}{\sqrt{3}})^{K-1} \) holds for all \( s \in [n] \) (consider \( s \leq K \) and \( s \geq K \)), it follows that

\[
\vdash_{2} \sum_{S \subseteq [n]} |S| (\frac{1}{\sqrt{3}})^{|S|-1} \tau(S)^2 \geq \frac{K}{\sqrt{3}^{K-1}} \text{Var}[f] - \frac{1}{\sqrt{3}^{K-1}} I[f].
\]
By combining the previous two deductions and doing some rearranging, we obtain
\[ \{ f(x)^2 = 1 : \forall x \} \cup \{ \text{Inf}_i[f] \leq \tau : \forall i \in [n] \} \vdash_4 \text{Var}[f] \leq \left( \frac{\sqrt{2^{K-1}}}{K} \left( \frac{3\epsilon}{4} + \frac{\tau}{4\epsilon} \right) + \frac{1}{K} \right) I[f] , \]
as claimed. \hfill \Box

We can now easily deduce (an SOS proof of) the fact that if \( f : \{-1, 1\}^n \to \{-1, 1\} \) has constant variance and all its influences equal then its total influence is \( \Omega(\log n) \). For the application to \textsc{BalancedSeparator}, we will in fact need a slightly more technical statement:

**Corollary 10.2.2.** \textit{(SOS proof of KKL for equal-influence functions.)}

Let \( n \geq 81 \) be an integer and for each \( x \in \{-1, 1\}^n \) let \( f(x) \) be an indeterminate. Define
\[
A = \{ f(x)^2 = 1 : \forall x \} \cup \{ \text{Inf}_i[f] \leq \tau : \forall i \in [n] \}
\cup \{ \text{Inf}_i[f] = \text{Inf}_j[f] : \forall i, j \in [n] \} \cup \{ \text{Var}[f] \geq \frac{3}{4} \} \cup \{ I[f] \leq \frac{1}{20} \ln n \}.
\]
Then \( A \vdash_4 -1 \geq 0 \).

In fact, the result holds even if we change the equal-influences assumption \( \{ \text{Inf}_i[f] = \text{Inf}_j[f] : \forall i, j \in [n] \} \) to the weaker pair of assumptions \( \{ \text{Inf}_i[f] = \text{Inf}_j[f] : \forall i, j \leq n/2 \} \) and \( \{ \text{Inf}_i[f] = \text{Inf}_j[f] : \forall i, j > n/2 \} \) (assume \( n \) even).

**Proof.** We will prove the “in fact” statement, assuming \( n \) is even. (The reader will see why the original statement is also true when \( n \) is odd.) Define \( I^{(1)}[f] = \sum_{i \leq n/2} \text{Inf}_i[f] \) and \( I^{(2)}[f] = \sum_{i > n/2} \text{Inf}_i[f] \), so \( I[f] = I^{(1)}[f] + I^{(2)}[f] \). Note that
\[
\{ \text{Inf}_i[f] = \text{Inf}_j[f] : \forall i, j \leq n/2 \} \vdash_2 \text{Inf}_i[f] = \frac{2}{n} I^{(1)}[f]
\]
for each \( i \leq n/2 \), and similarly for \( i > n/2 \). Since \( I^{(1)}[f], I^{(2)}[f] \) are themselves SOS and of degree 2, we get
\[
\{ \text{Inf}_i[f] = \text{Inf}_j[f] : \forall i, j \leq n/2 \& \forall i, j > n/2 \} \vdash_2 \text{Inf}_i[f] \leq \frac{2}{n} I^{(1)}[f] + \frac{2}{n} I^{(2)}[f] = \frac{2}{n} I[f]
\]
for each \( i \in [n] \). (Note that with the basic equal-influences assumption we can obtain the even stronger conclusion \( \text{Inf}_i[f] = \frac{1}{n} I[f] \) for each \( i \in [n] \).) We can now employ [Theorem 10.2.1](#) replacing \( \tau \) by \( \frac{2}{n} I[f] \). Using also \( \text{Var}[f] \geq \frac{3}{4} \), we obtain that for any reals \( \epsilon > 0, K \geq 2 \),
\[
A \setminus \{ I[f] \leq \frac{1}{20} \ln n \} \vdash_4 \frac{3}{4} \leq \left( \frac{\sqrt{2^{K-1}}}{K} \left( \frac{3\epsilon}{4} + \frac{1}{2^4 n} I[f] \right) + \frac{1}{K} \right) I[f].
\]

182
Select \( K = \log_9(9n^{1/2}) \) and \( \epsilon = n^{-1/8} \) to obtain

\[
A \setminus \{ I[f] \leq \frac{1}{20} \ln n \} \vdash 4 \quad \frac{3}{4} \leq \left( \frac{n^{1/8}}{\log_9(9n^{1/2})} \left( \frac{3}{4} n^{-1/8} + \frac{1}{2} n^{-5/8} I[f] \right) + \frac{1}{\log_9(9n^{1/2})} \right) I[f]
\]

\[
= \frac{7}{2 \log_9(81n)} I[f] + \frac{1}{n^{1/2} \log_9(81n)} I[f]^2. \tag{10.1}
\]

We now employ \( I[f] \leq \frac{1}{20} \ln n \). Since \( I[f] \) is SOS and of degree 2 we also have

\[
I[f] \leq \frac{1}{20} \ln n \quad \vdash 4 \quad I[f]^2 \leq \frac{1}{400} \ln^2 n.
\]

Substituting this into (10.1) yields

\[
A \vdash 4 \quad \frac{3}{4} \leq \frac{7}{2 \log_9(81n)} \cdot \frac{1}{20} \ln n + \frac{1}{n^{1/2} \log_9(81n)} \cdot \frac{1}{400} \ln^2 n \leq \frac{7}{20} \ln(3) \leq 0.4,
\]

whence \( A \vdash 4 \geq 0. \)

\[\square\]

### 10.3 Analysis of the DKS\(B\)ALANCEDSEPARATOR instances

We recall the BALANCEDSEPARATOR problem: Given is an undirected multigraph \( G = (V, E) \). It is required to find a cut \( S \subseteq V \) with \( \frac{1}{3} \leq \frac{|S|}{|V|} \leq \frac{2}{3} \) so as to minimize \( \frac{E(S, \overline{S})}{|E|} \). The natural polynomial optimization formulation has an indeterminate \( f(x) \) for each vertex \( x \in V \):

\[
\min \quad \frac{1}{|E|} \sum_{(x,y) \in E} \left( \frac{f(x) - f(y)}{2} \right)^2
\]

s.t. \( f(x)^2 = 1 \quad \forall x \in V \),

\[
\left( \frac{1}{|V|} \sum_{x \in V} f(x) \right)^2 \leq \frac{1}{9}.
\]

Thus as discussed in Section 8.2, the degree-4 SOS SDP hierarchy will use binary search to compute the largest \( \beta \) for which

\[
\{ f(x)^2 = 1 : \forall x \in V \} \cup \left\{ \left( \frac{1}{|V|} \sum_{x \in V} f(x) \right)^2 \leq \frac{1}{9} \right\} \cup \left\{ \frac{1}{|E|} \sum_{(x,y) \in E} \left( \frac{f(x) - f(y)}{2} \right)^2 \leq \beta \right\}
\]

\( \vdash 4 \quad -1 \geq 0. \)

183
The DKSV instances. We now recall the DKSV BalancedSeparator instances [79]. The instances $G = (V, E) = (V_N, E_N)$ are parameterized by primes $N$. Let $\mathcal{F} = \{-1, 1\}^N \times \{-1, 1\}^N$, thought of as the $2N$-dimensional hypercube graph. Let $\sigma$ act on elements $(x, y) \in \mathcal{F}$ by cyclic rotation of both halves:

$$\sigma(x, y) = (x_N, x_1, \ldots, x_{N-1}, y_N, y_1, \ldots, y_{N-1}).$$

The elements $\sigma, \sigma^2, \ldots, \sigma^N = id$ form a group acting on $\mathcal{F}$, partitioning it into orbits $O_1, \ldots, O_m$; $4$ of these orbits have cardinality $1$ and the remaining $(2^{2N} - 4)/N$ have cardinality $N$. A cardinality-$N$ orbit $O$ is called “nearly orthogonal” if for all distinct $(x, y), (x', y') \in O$ it holds that $|\langle (x, y), (x', y') \rangle| \leq 8\sqrt{N \log N}$. Presuming that $N$ is sufficiently large, [79] shows that the number $n$ of nearly orthogonal orbits satisfies

$$(1 - 4/N^2)m \leq n \leq m.$$ 

(This implies $N = \Theta(\log n)$.) For typographic simplicity the nearly orthogonal orbits are assumed to be $\{O_1, \ldots, O_n\}$, and this set is taken to be the vertex set $V$. We write $L \subseteq \mathcal{F}$ for the “leftover” elements contained in orbits $O_{n+1}, \ldots, O_m$; writing $\epsilon = |L|/2^n$, we have $\epsilon = O(1/N^2)$. The edges $E$ in $G$ are given by the usual hypercube edges in $\mathcal{F}$. More precisely, any pair $O, O' \in V$ have either $N$ or $0$ edges between them, according to whether or not there exist $(x, y) \in O, (x', y') \in O'$ at Hamming distance $1$ in $\mathcal{F}$. There are no self-loops in $G$ because of the near orthogonality property. The set of edges $E$ is in one-to-one correspondence with a subset of (almost all the) hypercube edges in $\mathcal{F}$; specifically, all those not incident on $L$. The authors of [79] use the KKL Theorem to prove:

**Theorem 10.3.1.** The DKSV BalancedSeparator instances have optimum value $\Omega(\log \log n/\log n)$.

(Although we haven’t formally verified it, it’s very likely that the optimum value of these instances is also $O(\log \log n/\log n)$, at least for infinitely many $N$. The reason is that there is a $\sigma$-invariant function $f : \mathcal{F} \to \{-1, 1\}$ of constant variance and total influence $\Omega(\log N)$; namely, $f(x, y) = 1$ if $x \in \{-1, 1\}^N$ contains a “run” (with wraparound) of length $\lfloor \log_2 N - \log_2 \log \log N \rfloor$.)

On the other hand, the main result of [79] is the following:

**Theorem 10.3.2.** The standard SDP relaxation with triangle inequalities for the DKSV BalancedSeparator instances has value $O(1/\log n)$.

We show here that this factor $\Theta(\log \log n)$ gap is eliminated when the degree-4 SOS relaxation is used.

**Theorem 10.3.3.** The degree-4 SOS relaxation for the DKSV BalancedSeparator instances has value $\Omega(\log \log n/\log n)$.

**Proof.** We need to show
\[ \{ f(O)^2 = 1 : \forall O \in V \} \cup \left\{ \left( \frac{1}{n} \sum_{O \in V} f(O) \right)^2 \leq \frac{1}{9}, \frac{1}{|E|} \sum_{(O,O') \in E} \left( \frac{f(O) - f(O')}{2} \right)^2 \leq c \frac{\log \log n}{\log n} \right\} \]

for some constant \( c > 0 \) (and \( N \) sufficiently large).

Introduce indeterminates \( g(x) \) for all \( x \in \mathcal{F} = \{-1, 1\}^N \times \{-1, 1\}^N \). By Corollary 10.2.2 it is possible to write

\[ -1 = u_0 + \sum_{x \in \mathcal{F}} v_x(g(x)^2 - 1) + \sum_{1 \leq i < j \leq N} w_{ij}(\text{Inf}_i[g] - \text{Inf}_j[g]) + u_1(\text{Var}[g] - \frac{3}{4}) + u_2\left( \frac{1}{20} \ln(2N) - 1[g] \right), \quad (10.3) \]

where \( u_0, u_1, u_2 \) are SOS (in the variables \( g(x) \)) and all summands have degree at most 4. Now substitute into this identity \( g(x) = f(O) \) for each \( x \in O \in V \), and also substitute \( g(x) = 1 \) for each \( x \in \mathcal{F} \) which is not contained in any \( O \in V \). We now consider what happens to each term in (10.3).

First, we notice that the degree of each term cannot increase. The polynomial \( u_0 \) (now over indeterminates \( f(O) \)) remains SOS. The next term, \( \sum_{x \in \mathcal{F}} v_x(g(x)^2 - 1) \), becomes of the form \( \sum_{O \in V} v'_O(f(O)^2 - 1) \) for some polynomials \( v'_O \). We claim that each summand \( w_{ij}(\text{Inf}_i[g] - \text{Inf}_j[g]) \) in the next term drops out entirely. This is because when \( g \) is viewed as mapping from \( \mathcal{F} \) to the set of homogeneous degree-1 polynomials in the \( f(O) \)'s, it is invariant under the action of \( \sigma \), by construction. From this it follows that \( \text{Inf}_i[g] = \text{Inf}_j[g] \) formally as polynomials for all \( 1 \leq i < j \leq N \) and \( N + 1 \leq i < j \leq 2N \).

Next we come to the term \( u_1(\text{Var}[g] - \frac{3}{4}) \). We have

\[ \text{Var}[g] - \frac{3}{4} = \mathbb{E}_{x \in \mathcal{F}}[g(x)^2] - \frac{3}{4} - \mathbb{E}_{x \in \mathcal{F}}[g(x)]^2. \]

Even after our substitution, \( \mathbb{E}_{x \in \mathcal{F}}[g(x)^2] - \frac{3}{4} \) will provably equal \( \frac{1}{4} \) under the assumption \( \{ f(O)^2 = 1 : \forall O \in V \} \), so it remains to focus on

\[ \mathbb{E}_{x \in \mathcal{F}}[g(x)]^2 = \left( \epsilon + (1 - \epsilon) \frac{1}{n} \sum_{O \in V} f(O) \right)^2. \]

Recalling that \( \vdash_2 (Y + Z)^2 \leq 2Y^2 + 2Z^2 \), we deduce

\[ \left( \frac{1}{n} \sum_{O \in V} f(O) \right)^2 \leq \frac{1}{9}, \quad \vdash_2 \left( \epsilon + (1 - \epsilon) \frac{1}{n} \sum_{O \in V} f(O) \right)^2 \leq 2\epsilon^2 + 2(1 - \epsilon)^2 \cdot \frac{1}{9} \leq \frac{1}{4} \]
(for $N$ sufficiently large, since $\epsilon = O(1/N^2)$), as needed.

Finally we come to the term $u_2(\frac{1}{20} \ln(2N) - I[g])$. Let $\epsilon'$ denote the fraction of hypercube edges in $\mathcal{F}$ which are incident on $L$; note that $\epsilon' \leq 2\epsilon = O(1/N^2)$. After our substitution, we have

$$\frac{1}{20} \ln(2N) - I[g] = \frac{1}{20} \ln(2N) - (2N) \sum_{\text{edge } (x,y) \in \mathcal{F}} \left( \frac{g(x) - g(y)}{2} \right)^2$$

$$= \frac{1}{20} \ln(2N) - (2N)(1 - \epsilon') \cdot \frac{1}{|E|} \sum_{(O,O') \in E} \left( \frac{f(O) - f(O')}{2} \right)^2 - \epsilon' \cdot (\ast),$$

where $(\ast)$ is the average of a number of terms, some of which are $(\frac{1-1}{2})^2 = 0$ and some of which are of the form

$$\left( \frac{f(O_i) - 1}{2} \right)^2 = 1 + \frac{1}{2}(f(O_i)^2 - 1) - \frac{1}{4}(f(O_i) + 1)^2.$$

The above shows that $\{ f(O)^2 = 1 : \forall O \in V \} \vdash_2 \left( \frac{f(O_i) - 1}{2} \right)^2 \leq 1$. Hence

$$\{ f(O)^2 = 1 : \forall O \in V \} \cup \left\{ \frac{1}{|E|} \sum_{(O,O') \in E} \left( \frac{f(O) - f(O')}{2} \right)^2 \leq c \frac{\log \log n}{\log n} \right\}$$

$$\vdash_2 \left(10.4\right) \geq \frac{1}{20} \ln(2N) - (2N)(1 - \epsilon') \cdot \frac{c \log \log n}{\log n} - \epsilon',$$

which is nonnegative for $c$ sufficiently small, since $N = \Theta(\log n)$ and $\epsilon' = O(1/N^2)$. Thus we have verified $[10.2]$.
Chapter 11

SOS proofs of the CLT, the \( \frac{2}{\pi} \) Theorem, and the analysis of the KV \textsc{MaxCut} instances

In this chapter, we give an SOS proof for the \( \frac{2}{\pi} \) theorem, and use this to show that constant-degree SOS proofs certify a better upper bound on the optimal value of the KV \textsc{MaxCut} instances (than the upper bound given by the Goemans-Williamson algorithm).

## 11.1 An invariance theorem for polynomials of linear forms

**Theorem 11.1.1.** (\textit{SOS proof of an Invariance Theorem for polynomials of linear forms.})

Let \( a_1, \ldots, a_n \) be indeterminates. For any real vector \( z = (z_1, \ldots, z_n) \), let \( \ell(z) \) denote the homogeneous linear polynomial \( \ell(z) = a_1 z_1 + \cdots + a_n z_n \). Then for any even integer \( k \geq 4 \) we have

\[
a_1^2 + \cdots + a_n^2 \leq 1 \quad \vdash_{2k} E_G[\ell(G)^k] - k^{O(k)} \sum_{i=1}^{n} a_i^4 \leq E_x[\ell(x)^k] \leq E_G[\ell(G)^k],
\]

where \( G = (G_1, \ldots, G_n) \sim N(0,1)^n \) and \( x \sim \{-1,1\}^n \) is uniform.

**Remark 11.1.1.** It is easy to see that \( E_x[\ell(x)^k] = E_G[\ell(G)^k] \) formally as polynomials for \( k = 0, 1, 2, 3 \), and any odd integer \( k > 3 \).
Proof. For each integer $0 \leq i \leq n$, define the polynomial

$$P_i = E[\ell(G_1, \cdots, G_i, x_{i+1}, \cdots, x_n)^k].$$

We will show for each $1 \leq i \leq n$ that

$$a_1^2 + \cdots + a_n^2 \leq 1 \quad \vdash_{2k} \quad P_i - k^{O(k)} a_i^4 \leq P_{i-1} \leq P_i. \quad (11.1)$$

The desired result then follows by summing over $i$. So fix $1 \leq i \leq n$ and write $\ell(z) = \ell'(z') + a_i z_i$, where $z' = (z_1, \ldots, z_{i-1}, z_{i+1}, z_n)$ and

$$\ell'(z') = a_1 z_1 + \cdots + a_{i-1} z_{i-1} + a_{i+1} z_{i+1} + \cdots + a_n z_n$$

does not depend on the indeterminate $a_i$. Denoting $Z' = (G_1, \ldots, G_{i-1}, x_{i+1}, \ldots, x_n)$ we have

$$P_i - P_{i-1} = E_{Z'} \left[ E_{G_i, x_i} \left[ (\ell'(Z') + a_i G_i)^k - (\ell'(Z') + a_i x_i)^k \right] \right]$$

$$= \sum_{j=2}^{k/2} \binom{k}{2j} ((2j-1)!! - 1) a_i^{2j} E_{Z'}[\ell'(Z')^{k-2j}] \quad (11.2)$$

where we used $E[G_i^r] = E[x_i^r] = 0$ for $r$ odd and $E[G_i^r] = (r-1)!!$, $E[x_i^r] = 1$, for $r$ even. The above polynomial is evidently SOS, justifying the second inequality in (11.1). As for the first inequality in (11.1), we have

$$a_1^2 + \cdots + a_n^2 \leq 1 \quad \vdash_{2j} \quad a_i^{2j} \leq a_i^4 \quad (11.3)$$

for each $i \in [n]$ and $2 \leq j \leq k/2$ because

$$a_i^4 - a_i^{2j} = (1 - a_i^2) (a_i^4 + a_i^6 + a_i^8 + \cdots + a_i^{2j-2})$$

$$= \left(1 - \sum_{i' \neq i} a_{i'}^2 + \sum_{i' \neq i} a_{i'}^2 \right) (a_i^4 + a_i^6 + a_i^8 + \cdots + a_i^{2j-2});$$

and, we have

$$a_1^2 + \cdots + a_n^2 \leq 1 \quad \vdash_{2(k-2j)} \quad E_{Z'}[\ell'(Z')^{k-2j}] \leq k^{O(k)} E_{Z'}[\ell'(Z')^2]^{k/2-j}$$

$$= k^{O(k)} (a_1^2 + \cdots + a_n^2)^{k/2-j} \leq k^{O(k)} \quad (11.4)$$

188
by [Corollary 8.4.7], the second inequality’s SOS proof being
\[ 1 - (\sum a_i^2)^{k/2-j} = 1 + (\sum a_i^2) + (\sum a_i^2)^2 + (\sum a_i^2)^3 + \cdots + (\sum a_i^2)^{k/2-j-1}. \]
Combining (11.3) and (11.4) via [Fact 8.2.4]
\[ a_1^2 + \cdots + a_n^2 \leq 1 \quad \implies \quad a_i^{2j} E[\ell'(Z')^{k-2j}] \leq k^{O(k)} a_i^4. \]
Using this in (11.2), along with \((\binom{k}{2j})((2j-1)!! - 1) \leq k^{O(k)}\) for each \(j\), yields the first inequality in (11.1), completing the proof. \(\square\)

### 11.2 An SOS Proof of the \(\frac{2}{\pi}\) Theorem

We require the below technical lemma giving a polynomial approximator to the absolute-value function. The proof uses some standard methods in approximation theory and is deferred to [Section 11.4]

**Lemma 11.2.1.** For any sufficiently small parameter \(\delta > 0\), there exists a univariate, real, even polynomial \(P(t) = Q(t^2)\) of degree at most \(\tilde{O}(1/\delta^2)\) such that:

1. \(P(t) \geq |t|\) for all \(t \in \mathbb{R}^2\);
2. \(E[|P(\sigma g)|] \leq \sqrt{\frac{2}{\pi}} \cdot |\sigma + \delta| \leq \left(\frac{1}{2} \sigma^2 + \frac{1}{\pi}\right) + \delta\) for all \(0 \leq \sigma \leq 1\), where \(g \sim \mathcal{N}(0, 1)\);
3. Each coefficient of \(P\) is at most \(2^{O(d)}\) in absolute value.

It is not hard to show that among degree-2 polynomials \(P(t)\) with \(P(t) \geq |t|\), the lowest possible value of \(E[P(g)]\) is 1, achieved by \(P(t) = \frac{1}{2} + \frac{1}{2}t^2\). Interestingly, this is also the lowest possible value even when degree-4 is allowed:

**Theorem 11.2.2.** Suppose \(P(t)\) is a univariate real polynomial of degree at most 4 satisfying \(P(t) \geq |t|\) for all \(t\). Then \(E_{g \sim \mathcal{N}(0, 1)}[P(g)] \geq 1\).

**Proof.** Replacing \(P(t)\) by \(\frac{1}{2}(P(t) + P(-t))\) if necessary, we may assume \(P(t)\) is even; i.e., \(P(t) = a + bt^2 + ct^4\) for some real \(a, b, c\). For any \(M > 0\) we have
\[
\frac{1}{M^2} \times (P(0) \geq 0) \quad + \quad \frac{M^2-3}{M^2-1} \times (P(1) \geq 1) \quad + \quad \frac{2}{(M^2-1)M^2} \times (P(M) \geq M) \\
\implies \frac{1}{M^2} \times (a \geq 0) \quad + \quad \frac{M^2-3}{M^2-1} \times (a + b + c \geq 1) \quad + \quad \frac{2}{(M^2-1)M^2} \times (a + M^2b + M^4c \geq M) \\
\implies a + b + 3c \geq 1 - \frac{2}{M(M+1)}.
\]
This completes the proof because \(E[P(g)] = a + b + 3c\) and \(M\) may be arbitrarily large. \(\square\)
Remark 11.2.1. Once we allow degree 6 it is possible to obtain a bound strictly smaller than 1. For example, $P(t) = .333 + .815 t^2 - .136 t^4 + .01 t^6 \geq |t|$ pointwise, and $\mathbb{E}[P(g)] = .89$.

The following “$\frac{2}{\pi}$ Theorem”, due to [141], is essentially the special case of the Majority Is Stablest Theorem in which $\rho \to 0^+$. We reproduce the proof.

**Theorem 11.2.3.** Let $f : \{-1,1\}^n \to [-1,1]$ and assume $|\hat{f}(i)| \leq \epsilon$ for all $i \in [n]$. Then
\[
\sum_{i=1}^n \hat{f}(i)^2 \leq \frac{2}{\pi} + O(\epsilon).
\]

**Proof.** Let $\ell : \{-1,1\}^n \to \mathbb{R}$ be $\ell(x) = \sum_{i=1}^n \hat{f}(i)x_i$ and let $\sigma = \sqrt{\sum_{i=1}^n \hat{f}(i)^2}$. Then
\[
\sigma^2 = \mathbb{E}_{x \sim \{-1,1\}^n} [f(x)\ell(x)] \leq \mathbb{E}_{x} [\ell(x)] \leq \sigma \mathbb{E}_{g \sim N(0,1)} [|g|] + O(\sigma \epsilon) = \sigma \left( \sqrt{\frac{2}{\pi}} + O(\epsilon) \right),
\]
the inequality being Berry–Esseen. The result follows after dividing by $\sigma$ and squaring. \hfill \Box

**Theorem 11.2.4.** (SOS proof of the Berry–Esseen Theorem with $\ell_1$ functional.)

Let $a_1, \ldots, a_n$ be indeterminates, and for each $x \in \{-1,1\}^n$, let $f(x)$ be an indeterminate. Let
\[
A = \{f(x) \geq -1, f(x) \leq 1 : \forall x \} \cup \{a_1^2 + \cdots + a_n^2 \leq 1\}.
\]
Then for any small real $\delta > 0$,
\[
A \vdash_{\tilde{O}(1/\delta^2)} \mathbb{E}_{x \sim \{-1,1\}^n} [f(x)(a_1 x_1 + \cdots + a_n x_n)] \leq b + \delta + 2\tilde{O}(1/\delta^2) \sum_{i=1}^n a_i^4,
\]
where we may choose either
\[
b = \sqrt{\frac{2}{\pi}} \quad \text{or} \quad b = \frac{1}{2}(a_1^2 + \cdots + a_n^2) + \frac{1}{\pi}.
\]

**Proof.** For each $x \in \{-1,1\}^n$, let $\ell(x)$ denote $a_1 x_1 + \cdots + a_n x_n$, a homogeneous linear polynomial in the indeterminates $a_i$. Let $P(t) = Q(t^2) = \sum_{k=0,2,4,\ldots,d} c_k t^k$ be the univariate real polynomial in Lemma 11.2.1 where $d = \deg(P) \leq \tilde{O}(1/\delta^2)$. Since $P(t) \geq \pm t$ for all real $t$, Fact 8.3.1 tells us that $\vdash_{d} P(t) \geq \pm t$ in $\mathbb{R}[t]$. Using Fact 8.3.7 and substituting $t = \ell(x)$ we deduce
\[
\{f(x) \geq -1, f(x) \leq 1\} \vdash_{d+1} f(x)\ell(x) \leq P(\ell(x)).
\]
Averaging over $x$ yields

$$\{f(x) \geq -1, f(x) \leq 1 : \forall x\} \vdash_{d+1} E_x[f(x)(a_1x_1 + \cdots + a_nx_n)] \leq E_x[P(\ell(x))] = \sum_{k=0,2,4,\ldots,d} E[c_k\ell(x)^k]. \quad (11.5)$$

For each even $0 \leq k \leq d$, regardless of the sign of $c_k$, Theorem 11.1 implies that

$$a_1^2 + \cdots + a_n^2 \leq 1 \vdash_{2d} E_x[c_k\ell(x)^k] \leq E_G[c_k\ell(G)^k] + |c_k|k^{O(k)} \sum_{i=1}^n a_i^4.$$  

Summing this over $k$, using $\sum_k |c_k|k^{O(k)} \leq d^{O(d)}$, and combining with (11.5) yields

$$A \vdash_{2d} E_x[f(x)(a_1x_1 + \cdots + a_nx_n)] \leq E_G[P(\ell(G))] + d^{O(d)} \sum_{i=1}^n a_i^4. \quad (11.6)$$

Let $\sigma^2$ be shorthand for $\sum_{i=1}^n a_i^2$. Note that if we treat $a_1, \ldots, a_n$ as arbitrary real numbers, we have

$$E_G[P(\ell(G))] = \mathbb{E}_{g \sim \mathcal{N}(0,1)} [P(\sigma g)] = \mathbb{E}_{g \sim \mathcal{N}(0,1)} [Q(\sigma^2 g^2)], \quad (11.7)$$

by the rotational symmetry of multivariate Gaussians. Since the left and right sides are polynomials in $a_1, \ldots, a_n$, it follows that (11.7) also holds as a formal polynomial identity over the indeterminates $a_1, \ldots, a_n$. Now temporarily view $\sigma^2$ as an indeterminate. From Lemma 11.2.1 we have that $E_g \sim \mathcal{N}(0,1)[Q(\sigma^2 g^2)]$ is upper-bounded by both $\sqrt{\frac{2}{\pi} + \delta}$ and $\frac{1}{2}\sigma^2 + \frac{1}{\pi} + \delta$ for all real numbers $0 \leq \sigma^2 \leq 1$. Thus from Fact 8.3.2 we have the following univariate SOS proof(s):

$$(1 - \sigma^2)\sigma^2 \geq 0 \vdash_{d/2} \mathbb{E}_{g \sim \mathcal{N}(0,1)} [Q(\sigma^2 g^2)] \leq \sqrt{\frac{2}{\pi} + \delta}, \frac{1}{2}\sigma^2 + \frac{1}{\pi} + \delta$$

(note that $Q$ has even degree). Letting $\sigma^2 = \sum_{i=1}^n a_i^2$ again, we deduce that for either choice of $b$,

$$\left(1 - \sum_{i=1}^n a_i^2\right) \left(\sum_{i=1}^n a_i^2\right) \geq 0 \vdash_d \mathbb{E}_{g \sim \mathcal{N}(0,1)} [Q(\sigma^2 g^2)] \leq b + \delta$$

$$\Leftrightarrow \quad a_1^2 + \cdots + a_n^2 \leq 1 \vdash_d E_G[P(\ell(G))] \leq b + \delta.$$
using (11.7) and the fact that $\sum_{i=1}^n a_i^2$ is already SOS. Combining this with (11.6) yields

$$A \vdash_{2d} \sum_{i=1}^n f(x_i) (a_1 x_1 + \cdots + a_n x_n) \leq b + \delta + d^{O(d)} \sum_{i=1}^n a_i^4,$$

as needed.

**Corollary 11.2.5. (SOS proof of the $2/\pi$ Theorem.)**

For each $x \in \{-1, 1\}^n$, let $f(x)$ be an indeterminate. Define $\hat{f}(S)$ as usual and write $\hat{f}(i) = \hat{f}(\{i\})$ for short. Let

$$A = \{f(x) \geq -1, f(x) \leq 1 : \forall x\}.$$

Then for each small real $\delta > 0$,

$$A \vdash_{O(1/\delta^2)} \sum_{i=1}^n \hat{f}(i)^2 \leq \frac{2}{\pi} + \delta + 2 \tilde{O}(1/\delta^2) \sum_{i=1}^n \hat{f}(i)^4,$$

$$A \cup \{\hat{f}(i)^2 \leq \tau : \forall i \in [n]\} \vdash_{O(1/\delta^2)} \sum_{i=1}^n \hat{f}(i)^2 \leq \frac{2}{\pi} + \delta + 2 \tilde{O}(1/\delta^2) \cdot \tau.$$

**Proof.** We wish to apply Theorem 11.2.4 with $a_i = \hat{f}(i)$ for each $i \in [n]$. A standard proof shows that

$$\sum_{S \subseteq [n]} \hat{f}(S)^2 = \mathbb{E}_{x \sim \{-1, 1\}^n} [f(x)^2]$$

and hence, using Fact 8.3.4

$$\vdash 3 \sum_{i=1}^n \hat{f}(i)^2 \leq 1. \quad (11.8)$$

We may therefore employ Theorem 11.2.4 (with $\delta/2$ instead of $\delta$) to obtain

$$A \vdash_{\tilde{O}(1/\delta^2)}$$

$$\mathbb{E}_{x \sim \{-1, 1\}^n} [f(x) (\hat{f}(1) x_1 + \cdots + \hat{f}(n) x_n)] \leq \frac{1}{2} \sum_{i=1}^n \hat{f}(i)^2 + \frac{\delta}{2} + 2 \tilde{O}(1/\delta^2) \sum_{i=1}^n \hat{f}(i)^4.$$

But

$$\mathbb{E}_{x \sim \{-1, 1\}^n} [f(x) (\hat{f}(1) x_1 + \cdots + \hat{f}(n) x_n)] = \sum_{i=1}^n \hat{f}(i)^2$$

192
is a polynomial identity so we deduce

\[ A \vdash \tilde{O}(1/\delta^2) \sum_{i=1}^n \tilde{f}(i)^2 \leq \frac{1}{2} \sum_{i=1}^n \tilde{f}(i)^2 + \frac{1}{n} + \frac{\delta}{2} + 2\tilde{O}(1/\delta^2) \sum_{i=1}^n \tilde{f}(i)^4 \]

\[ \iff \sum_{i=1}^n \tilde{f}(i)^2 \leq \frac{2}{n} + \delta + 2\tilde{O}(1/\delta^2) \sum_{i=1}^n \tilde{f}(i)^4, \]

completing the first part of the proof. Now adding the assumptions \( \tilde{f}(i)^2 \leq \tau \) easily yields

\[ A \cup \{ \tilde{f}(i)^2 \leq \tau : \forall i \in [n] \} \vdash \sum_{i=1}^n \tilde{f}(i)^4 \leq \tau \sum_{i=1}^n \tilde{f}(i)^2 \leq \tau \]

using (11.8) again. The proof is complete.

\[ \square \]

11.3 Analysis of the KV MAXCUT instances

We recall the MAXCUT problem: Given is an undirected weighted graph \( G \) on vertex set \( V \) in which the nonnegative edge weights sum to 1. We write \((x, y) \sim E\) to denote that \((x, y)\) is a random edge chosen with probability equal to the edge weight. It is required to find a cut \( S \subseteq V \) so as to maximize \( \Pr_{(x, y) \sim E}[x \in S, y \notin S \text{ or vice versa}] \). The natural polynomial optimization formulation has an indeterminate \( f(x) \) for each vertex \( x \in V \):

\[
\max \mathbb{E}_{(x, y) \sim E} \left[ \frac{1}{2} - \frac{1}{2} f(x) f(y) \right] \\
\text{s.t.} \quad f(x)^2 = 1 \quad \forall x \in V.
\]

Thus as discussed in Section 8.2, the degree-\( d \) SOS SDP hierarchy will use binary search to compute the smallest \( \beta \) for which

\[
\{ f(x)^2 = 1 : \forall x \in V \} \cup \left\{ \mathbb{E}_{(x, y) \sim E} \left[ \frac{1}{2} - \frac{1}{2} f(x) f(y) \right] \geq \beta \right\} \vdash_{d} -1 \geq 0.
\]

**UNIQUEGAMES.** The Khot–Vishoi (KV) instances of MAXCUT [144] are given by composing the KKMO “noise stability” reduction from [141] with the KV integrality gap instances for UNIQUEGAMES (UG). Our SOS proof of the \( 2/\delta \) Theorem gives us a “blackbox” analysis of the KKMO reduction which can essentially be “plugged in” to a sufficiently strong SOS analysis of UG instances. Let us now recall the **UNIQUEGAMES**
problem with label-size \( k \in \mathbb{N}^+ \). Given is a regular weighted graph \( G = (\mathcal{V}, \mathcal{E}) \) (self-loops allowed) with weights summing to 1. Also, given for each edge \((u, v)\) a permutation \( \pi_{uv} : [k] \rightarrow [k] \). We write \((u, v, \pi) \sim \mathcal{E}\) to denote that edge \((u, v)\) with permutation \( \pi = \pi_{uv}\) is chosen with probability equal to its edge weight. The goal is to give a labeling \( F : \mathcal{V} \rightarrow [k] \) so as to maximize \( \Pr((u, v, \pi) \sim \mathcal{E} | \pi(F(u)) = F(v)) \). The natural polynomial optimization formulation has an indeterminate \( X_{u,i} \) for each \( u \in \mathcal{V}, i \in [k] \):

\[
\max_{(u, v, \pi) \sim \mathcal{E}} \mathbb{E} \left[ \sum_{i=1}^{k} X_{u,i} X_{v,\pi(i)} \right] = \mathbb{E} \left[ \sum_{u \in \mathcal{V}} \sum_{i=1}^{k} X_{u,i} \cdot \mathbb{E}_{(v, \pi) \sim \mathcal{E}} [X_{v,\pi(i)}] \right]
\]

s.t.

\[
X_{u,i} = X_{u,i} \quad \forall u \in \mathcal{V}, i \in [k]
\]

\[
\sum_{i=1}^{k} X_{u,i} = 1 \quad \forall u \in \mathcal{V},
\]

where we write \((v, \pi) \sim u\) in place of \((u, v, \pi) \sim \mathcal{E} \mid u = u\) for brevity. Thus the degree-\( d \) SOS SDP hierarchy will use binary search to compute the smallest \( \beta \) for which

\[
\{ X_{u,i} \geq 0 : \forall u \in \mathcal{V}, i \in [k] \} \cup \{ \sum_{i=1}^{k} X_{u,i} = 1 : \forall u \in \mathcal{V} \}
\]

\[
\cup \left\{ \mathbb{E}_{u \in \mathcal{V}} \left[ \sum_{i=1}^{k} X_{u,i} \cdot \mathbb{E}_{(v, \pi) \sim \mathcal{E}} [X_{v,\pi(i)}] \right] \geq \beta \right\} \vdash d - 1 \geq 0.
\]

In \[\text{Section 9.3}\], we have shown that the degree-4 moment SDP proves that the KV family of UG instances has a very low optimum value. In fact we have shown something stronger; one only needs the hypotheses \( X_{u,i}^2 \leq X_{u,i} \) and \( (\text{avg}_{u,i} X_{u,i})^2 \leq 1/k^2 \). Let us make a somewhat more general definition which applies to SOS-refutations of any UG instances:

\textbf{Definition 11.3.1.} \textit{Given a UG instance }\mathcal{G} = (\mathcal{V}, \mathcal{E})\textit{ with label-size }k\textit{, we say there is a degree-}\( d \textit{ SOS refutation that the fractional assignment optimum is at least } \beta \textit{ if}

\[
\{ X_{u,i} \geq 0 : \forall u \in \mathcal{V}, i \in [k] \} \cup \{ \sum_{i=1}^{k} X_{u,i} \leq 1 : \forall u \in \mathcal{V} \}
\]

\[
\cup \left\{ \mathbb{E}_{u \in \mathcal{V}} \left[ \sum_{i=1}^{k} X_{u,i} \cdot \mathbb{E}_{(v, \pi) \sim \mathcal{E}} [X_{v,\pi(i)}] \right] \geq \beta \right\} \vdash d - 1 \geq 0.
\]

The above definition is slightly more demanding than the most natural one, in which the hypotheses \( X_{u,i}^2 = X_{u,i} \) are granted. As mentioned, in \[\text{Section 9.3}\] we have established something noticeably stronger anyway: the following theorem is a restatement of \textbf{Theorem 9.3.1}.
Theorem 11.3.2. Let $G = G(N, \eta) = (V, E)$ be the Khot–Vishnoi instance of UniqueGames parameterized by $N$ (a power of 2) and $\eta \in (0, 1)$, which has $2^N / N$ vertices, label-size $N$, and optimum value at most $N^{-\eta}$. Then there is a degree-$4$ SOS refutation that its fractional assignment optimum is at least $N - \Omega(\eta)$.

We now recall the KKMO [141] reduction from UG to MaxCut, which is parameterized by $\rho \in (-1, 0)$. Given a UG instance $G$ with label-size $N$, the reduction creates a vertex set $V$ with a vertex $w_{u,x}$ for each $u \in V$ and each $x \in \{-1, 1\}^N$. The probability distribution $E$ on edges for the MaxCut instance is given as follows:

- draw $u \sim V$;
- independently draw $(u, v_1, \pi_1)$ and $(u, v_2, \pi_2)$ from the marginal of $E$ which has first vertex $u$;
- draw “$\rho$-correlated strings” $(x, y)$ from $\{-1, 1\}^N$;
- output the edge $(w_{u_1, xo\pi_1}, w_{u_2, yo\pi_2})$.

KKMO make the following easy observation:

Proposition 11.3.3. Consider any cut $V \rightarrow \{-1, 1\}$ in the above-described MaxCut instance $(V, E)$; specifically, let us write it as a collection of functions $f_v : \{-1, 1\}^N \rightarrow \{-1, 1\}$, one for each $v \in V$. Then the value of this cut is

$$\frac{1}{2} - \frac{1}{2} \mathbb{E}_{u \sim V} \left[ \text{Stab}_\rho[g_u] \right],$$

where $g_u : \{-1, 1\}^N \rightarrow [-1, 1]$ is defined by $g_u(x) = \mathbb{E}_{(u, v, \pi) \sim E_{u = u}} [f_v(x \circ \pi)]$.

As mentioned, the KV MaxCut instances are formed by composing the KKMO reduction with the KV UG instances. Khot and Vishnoi show that for any fixed $\eta \in (0, 1)$, the optimum value of the resulting MaxCut instance is at most $(\arccos \rho) / \pi + o_N(1)$. Further, using “Majority cuts” it’s easy to show (using, e.g. [179, Theorem 3.4.2]) that the optimum values is at least $(\arccos \rho) / \pi - o_N(1)$.

The main result of this section is the following:

Theorem 11.3.4. Fix any small $\epsilon, \delta > 0$. Let $G = (V, E)$ be a UG instance with label-size $N$ for which there is a degree-$d$ SOS proof ($d \geq 2$) that its fractional assignment optimum is at most $\epsilon$. Let $G = (V, E)$ be the MaxCut instance resulting from applying the KKMO reduction with parameter $\rho \in (-1, 0)$ to $G$. Then there is a degree $d + \tilde{O}(1/\delta^2)$ SOS refutation of the claim that the optimum value of $G$ is at least $\frac{1}{2} - \frac{1}{\pi} \rho - (\frac{1}{2} - \frac{1}{\pi}) \rho^3 + \delta + \epsilon \cdot 2\tilde{O}(1/\delta^2)$.  

195
Together with Theorem 11.3.2 this implies:

**Corollary 11.3.5.** Fix any small \( \delta > 0 \). Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) be a Khot-Vishnoi UG instance with label-size \( N \) and noise parameter \( \eta \). Let \( G = (V, E) \) be the MAXCUT instance resulting from applying the KKMO reduction with parameter \( \rho \in (-1, 0) \) to \( \mathcal{G} \). Then there is a degree \( \tilde{O}(1/\delta^2) \) SOS refutation of the claim that the optimum value of \( G \) is at least

\[
\frac{1}{2} - \frac{1}{\pi} \rho - \left( \frac{1}{2} - \frac{1}{\pi} \right) \rho^3 + \delta + 2\tilde{O}(1/\delta^2) \cdot N^{-\Omega(1)}.
\]

**Corollary 11.3.6.** Consider the KV MAXCUT instances with parameter \( \rho_0 \approx -0.689 \). The degree-\( O(1) \) SOS SDP certifies they have value at most \( 0.779 \), which is within a factor \( 0.952 \) of the optimum. For general \( \rho \), the degree-\( O(1) \) SOS SDP certifies a value for the KV MAXCUT instances which is within a factor \( 0.931 \) of the optimum, where

\[
0.931 \approx \min_{\rho \in (-1, 0)} \frac{(\arccos \rho)/\pi}{\frac{1}{2} - \frac{1}{\pi} \rho - \left( \frac{1}{2} - \frac{1}{\pi} \right) \rho^3}.
\]

Before proving Theorem 11.3.4 we prove a lemma which gives an alternative way to refute a UG instance having a good solution: roughly, for most vertices \( v \in \mathcal{V} \), its neighbors cannot agree well on what \( v \)'s label should be.

**Lemma 11.3.7.** Let \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) be a UG instance with label-size \( N \) and suppose there is a degree-\( d \) SOS refutation that its fractional assignment optimum is at least \( \epsilon \). Then also

\[
A \cup \left\{ \mathbb{E}_{u \sim \mathcal{V}} \left[ \sum_{i=1}^{N} \left( \mathbb{E}_{(v,\pi) \sim u} [X_{v,\pi(i)}] \right)^2 \right] \geq 4\epsilon \right\} \vdash_d -1 \geq 0,
\]

where

\[
A = \{ X_{u,i} \geq 0 : \forall x \in \mathcal{V}, i \in [N] \} \cup \{ \sum_{i=1}^{N} X_{u,i} \leq 1 : \forall x \in \mathcal{V} \}.
\]

**Proof.** Given the indeterminates \( X_{u,i} \), define for each \( u \in \mathcal{V} \) and \( i \in [N] \) the homogeneous linear forms

\[
Y_{u,i} = \frac{1}{2} X_{u,i} + \frac{1}{2} \mathbb{E}_{(v,\pi) \sim u} [X_{v,\pi(i)}].
\]

We will apply the assumption regarding the degree-\( d \) SOS refutation for \( \mathcal{G} \) to the \( Y_{u,i} \)'s. Certainly we have

\[
A \vdash_1 Y_{u,i} \geq 0, \sum_{j=1}^{N} Y_{u,j} \leq 1
\]
for every \( u \in V, i \in [N] \). Indeed, it’s not hard to check that to complete the proof we need only verify

\[
A \vdash _2 \mathbb{E}_{u \sim V} \left[ \sum_{i=1}^{N} Y_{u,i} \cdot E_{\pi} [Y_{v,\pi(i)}] \right] \geq \frac{1}{4} \mathbb{E}_{u \sim V} \left[ \sum_{i=1}^{N} \left( E_{\pi} [X_{v,\pi(i)}] \right)^2 \right].
\]

But this follows from

\[
\mathbb{E}_u \left[ \sum_{i} Y_{u,i} \cdot E_{(v,\pi) \sim u} [Y_{v,\pi(i)}] \right] = \frac{1}{2} \mathbb{E}_u \left[ \sum_{i} X_{u,i} \cdot E_{(v,\pi) \sim u} [Y_{v,\pi(i)}] \right] + \frac{1}{2} \mathbb{E}_u \left[ \sum_{i} E_{(v,\pi) \sim u} [X_{v,\pi(i)}] \cdot E_{(v,\pi) \sim u} [Y_{v,\pi(i)}] \right]
\]

\[
= \frac{1}{2} \mathbb{E}_u \left[ \sum_{i} X_{u,i} \cdot E_{(v,\pi) \sim u} [Y_{v,\pi(i)}] \right] + \frac{1}{4} \mathbb{E}_u \left[ \sum_{i} E_{(v,\pi) \sim u} [X_{v,\pi(i)}] \cdot E_{(v',\pi') \sim u} [X_{v',\pi'(i)}] \right] + \frac{1}{4} \mathbb{E}_u \left[ \sum_{i} E_{(v,\pi) \sim u} [X_{v,\pi(i)}^2] \right]
\]

(where we do not even need the assumptions \( \sum_{i=1}^{N} X_{u,i} \leq 1 \)).

We now give the proof of Theorem 11.3.4

**Proof.** It is not hard to deduce the following result from Corollary 11.2.5:

**Corollary 11.3.8.** In the setting of Corollary 11.2.5, for any \( \rho \in (-1, 0) \) we have

\[
A \vdash _\Theta(1/\delta^2) \ Stab_{\rho}[f] \geq \frac{2}{\pi} \cdot \rho + (1 - \frac{2}{\pi}) \cdot \rho^3 - \delta - 2^\Theta(1/\delta^2) \cdot \sum_{i=1}^{N} \hat{f}(i)^4.
\]

It is also easy to check using Fact 8.3.3 that

\[
\{ f_v(x)^2 = 1 : \forall v \in V, x \in \{-1, 1\}^N \} \vdash _4 g_v(x) \geq -1, g_v(x) \leq 1
\]

for all \( v \in V, x \in \{-1, 1\}^N \). Thus using the above corollary and Proposition 11.3.3 we obtain

\[
\{ f_v(x)^2 = 1 : \forall v \in V, x \in \{-1, 1\}^N \} \vdash _\Theta(1/\delta^2) \ \frac{1}{2} - \frac{1}{2} \mathbb{E}_{u \sim V} [Stab_{\rho}[g_u]] \leq \frac{1}{2} - \frac{1}{2} \rho - (\frac{1}{2} - \frac{1}{\pi}) \rho^3 + \delta + 2^\Theta(1/\delta^2) \cdot \mathbb{E}_{u \sim V} \left[ \sum_{i=1}^{N} \hat{g}_u(i)^4 \right].
\]

(11.9)
Now we bound the error term $2^{\tilde{O}(1/\delta^2)} \cdot E_{u \sim V}[\sum_{i=1}^{N} \hat{g}_u(i)^4]$ as follows. Using the polynomial identity $\hat{g}_u(i) = E_{(v, \pi) \sim u}[\hat{f}_v(\pi(i))]$ together with Fact 8.3.14 and Fact 8.3.15, we have

$$\vdash_{d+4} E_{u \sim V} \left[ \sum_{i=1}^{N} \hat{g}_u(i)^4 \right] \leq E_{u \sim V} \left[ \sum_{i=1}^{N} \left( E_{(v, \pi) \sim u} [\hat{f}_v(\pi(i))] \right)^2 \right].$$  \hspace{1cm} (11.10)

On the other hand, it is easy to check that for all $i \in [N]$ and $v \in V$, we have

$$\{ f_v(x)^2 = 1 : \forall x \in \{-1, 1\}^N \} \vdash_2 \hat{f}_v(i)^2 \geq 0, \sum_{i=1}^{N} \hat{f}_v(i)^2 \leq 1. \hspace{1cm} (11.11)$$

Since there is a degree-$d$ refutation for $G$ having a fractional assignment of value at least $\epsilon$, implementing Lemma 11.3.7 with $X_{v,i} = \hat{f}_v(i)^2$, we have

$$\{ f_v(x)^2 = 1 : \forall v \in V, x \in \{-1, 1\}^N \} \cup \{ E_{u \sim V} \left[ \sum_{i=1}^{N} \left( E_{(v, \pi) \sim u} [\hat{f}_v(\pi(i))] \right)^2 \right] \geq 4\epsilon \} \vdash_{d+2} -1 \geq 0. \hspace{1cm} (11.12)$$

By Fact 8.2.3 (11.11) and (11.12) give

$$\{ f_v(x)^2 = 1 : \forall v \in V, x \in \{-1, 1\}^N \} \cup \left\{ E_{u \sim V} \left[ \sum_{i=1}^{N} \left( E_{(v, \pi) \sim u} [\hat{f}_v(\pi(i))] \right)^2 \right] \geq 4\epsilon \right\} \vdash_{d+4} -1 \geq 0. \hspace{1cm} (11.13)$$

Combining (11.13) and (11.10), we get

$$\{ f_v(x)^2 = 1 : \forall v \in V, x \in \{-1, 1\}^N \} \cup \{ E_{u \sim V} \left[ \sum_{i=1}^{N} \hat{g}_v(i)^4 \right] \geq 4\epsilon \} \vdash_{d+2} -1 \geq 0. \hspace{1cm} (11.14)$$

Finally, combining (11.14) and (11.9), we get

$$\{ f_v(x)^2 = 1 : \forall v \in V, x \in \{-1, 1\}^N \} \cup \left\{ \frac{1}{2} - \frac{1}{2} E_{u \sim V} [\text{Stab}_\rho[g_u]] \geq \frac{1}{2} - \frac{1}{\pi} \rho - \left( \frac{1}{2} - \frac{1}{\pi} \right) \rho^3 + \delta + 2^{\tilde{O}(1/\delta^2)} \right\} \vdash_{d+\tilde{O}(1/\delta^2)} -1 \geq 0. \hspace{1cm} \square$$
11.4 An approximator for the absolute-value function

Here we restate and prove Lemma 11.2.1. A key tool will be the polynomial approximator to the \( \text{sgn} \) function constructed in [80].

**Lemma 11.2.1.** For any sufficiently small parameter \( \delta > 0 \), there exists a univariate, real, even polynomial \( P(t) = Q(t^2) \) of degree at most \( \tilde{O}(1/\delta^2) \) such that:

1. \( P(t) \geq |t| \) for all \( t \in \mathbb{R} \);
2. \( E[P(\sigma g)] \leq \sqrt{2\pi} \cdot \sigma + \delta \) for all \( 0 \leq \sigma \leq 1 \), where \( g \sim N(0,1) \);
3. Each coefficient of \( P \) is at most \( 2^{O(\deg(P))} \) in absolute value.

**Proof.** We will use the following result from [80, Theorem 3.10]:

**Theorem 11.4.1.** For every \( 0 < \epsilon < 1 \) there is an odd integer \( d = d(\epsilon) = \Theta(\log^2(1/\epsilon)/\epsilon) \) and a univariate polynomial \( p(t) \) of degree \( d \) satisfying:

- \( p(t) \in [\text{sgn}(t) - \epsilon, \text{sgn}(t) + \epsilon] \) for all \( |t| \in [\epsilon, 1] \);
- \( p(t) \in [-1 - \epsilon, 1 + \epsilon] \) for all \( |t| \leq \epsilon \);
- \( p(t) \) is monotonically increasing on the intervals \((-\infty, -1]\) and \([1, +\infty)\).

We can assume without loss of generality that \( p(t) \) is odd since the odd part of \( p(t) \) (i.e. \( (p(t) - p(-t))/2 \)) also satisfies the properties in Theorem 11.4.1.

Given \( p(t) \) as in Theorem 11.4.1, define

\[
p_0(t) = (1 + 2\epsilon)p(t/M), \quad \text{where } M = \frac{c \log^2(1/\epsilon)}{\sqrt{\epsilon}}
\]

and \( c > 1 \) is a universal constant to be chosen later. The polynomial \( p_0(t) \) has the following properties:

- \( p_0(t) \in [1, 1+4\epsilon] \) when \( t \in [M, M] \), \( p_0(t) \in [-1+4\epsilon, -1] \) when \( t \in [-M, -M] \);
- \( p_0(t) \in [-1+4\epsilon, 1+4\epsilon] \) for all \( |t| \leq M \);
- \( p_0(t) \geq 1 \) when \( t \geq M \), \( p_0(t) \leq 1 \) when \( t \leq -M \).
Finally, define

\[ P(t) = \int_0^t p_0(x)dx + 2M\epsilon. \]

an even polynomial of degree \(d + 1\). We will show that the following hold assuming \(c\) is taken sufficiently large and then \(\epsilon\) is sufficiently small:

(a) \(P(t) \geq |t|\) for all \(t \in \mathbb{R}\);

(b) \(\mathbb{E}[P(\sigma g)] \leq \sqrt{\frac{2}{\pi}} \cdot \sigma + O(M\epsilon)\) for all \(0 \leq \sigma \leq 1\);

(c) Each coefficient of \(P\) is at most \(2^{O(d)}\) in absolute value.

The proof is then completed by taking \(\epsilon = \delta^2 / \text{polylog}(1/\delta)\).

Properties (\((a)\)) follows easily from the definition of \(P(t)\). It also follows easily from the definition that \(|P(t)| \leq 1 + O(M\epsilon) \leq 2\) for all \(|t| \leq 1\). It is a standard fact in approximation theory (see, e.g., [208, 176]) that if \(P\) is a degree \(d+1\) polynomial satisfying \(|P(t)| \leq b\) for all \(|t| \leq 1\) then each coefficient of \(P(t)\) is at most, say, \(b(4\epsilon)^{d+1} = 2^{O(d)}\) in magnitude. This verifies (\((c)\)). It remains to establish property (\((b)\)). For this we have

\[
\mathbb{E}[P(\sigma g)] = \mathbb{E}[P(\sigma g) \cdot 1\{\sigma g \leq M\}] + \mathbb{E}[P(\sigma g) \cdot 1\{\sigma g > M\}] \tag{11.15}
\]

Regarding the first term in (11.15) we use that for \(|t| \leq M\) we have \(|p_0(t)| \leq 1 + 4\epsilon\) and hence

\[
P(t) \leq (1 + 4\epsilon)|t| + 2M\epsilon = |t| + O(M\epsilon) \quad \forall |t| \leq M. \tag{11.16}
\]

Thus

\[
\mathbb{E}[P(\sigma g) \cdot 1\{\sigma g \leq M\}] \leq \mathbb{E}[|\sigma g| \cdot 1\{|\sigma g| \leq M\}] + O(M\epsilon)
\leq \mathbb{E}[|\sigma g|] + O(M\epsilon) = \sqrt{\frac{2}{\pi}} \cdot \sigma + O(M\epsilon).
\]

To complete the verification of (\((b)\)) it therefore suffices to bound the second term in (11.15) by \(O(M\epsilon)\). In fact we will show

\[
\mathbb{E}[P(\sigma g) \cdot 1\{\sigma g > M\}] \ll M\epsilon. \tag{11.17}
\]

Using evenness of \(P\) and the fact that it is evidently increasing on \([M, \infty)\) we have

\[
\mathbb{E}[P(\sigma g) \cdot 1\{\sigma g > M\}] = 2 \mathbb{E}[P(\sigma g) \cdot 1\{\sigma g > M\}] \leq 2 \mathbb{E}[P(g) \cdot 1\{g > M\}]. \tag{11.18}
\]

We upper-bound \(P\)'s value on large inputs using a well-known fact from approximation theory (and a corollary of the theorem in \(\S\)33 of [169]):
**Fact 11.4.2.** Let \( q(t) \) be a polynomial of degree at most \( k \) satisfying \( |q(t)| \leq b \) for all \( |t| \leq 1 \). Then \( |q(t)| \leq b \cdot |3t|^k \) for all \( |t| \geq 1 \).

Applying this fact to \( p(t) \) we obtain \( p(t) \leq (1 + \epsilon)(3t)^d \) for all \( t \geq 1 \), whence \( p_0(t) \leq 2(3t/M)^d \) for all \( t \geq M \), whence

\[
P(t) = \int_0^t p_0(x)dx + 2M\epsilon \leq O(M) + t \cdot 2(3t/M)^d \leq O(1) \cdot (3t/M)^{d+1}
\]

for all \( t \geq M \) (we also used \( M = o(2^d) \)). Thus

\[
(11.18) \leq O\left(\frac{3}{M}\right)^{d+1} \cdot \mathbb{E}[g^{d+1} \cdot 1\{g > M\}] \leq O\left(\frac{3}{M}\right)^{d+1} \cdot O(dM)^{d+1} \exp(-M^2/2) \\
\leq 2^{\text{polylog}(1/\epsilon)/\epsilon} \exp(-M^2/2) = 2^{\text{polylog}(1/\epsilon)/\epsilon} \exp(-c^2 \log^2(1/\epsilon)/2\epsilon) \ll M\epsilon
\]

if we choose \( c \) to be a large enough universal constant. This completes the justification of (11.17) and the overall proof.
Part III

Robust algorithms
Chapter 12

Robust satisfiability algorithms for width 1 CSPs

12.1 Introduction

Constraint satisfaction problems (CSPs) constitute a broad and important subclass of algorithmic tasks. One approach to studying the complexity of CSPs centers around the Feder–Vardi Dichotomy Conjecture \cite{86} and the use of algebra \cite{125} to classify all CSP decision problems. Another approach to the study of CSPs involves quantifying the extent to which natural CSPs can be approximately solved \cite{127}; this approach has been characterized by more “analytic” methods. Recently there has been interest in melding the two approaches (see, e.g., \cite{155, 128, 113}); the present work takes another step in this direction.

Almost-satisfiable instances. The algebraic approach to CSPs is mainly concerned with what we’ll call the decision problem for CSPs: given an instance, is it completely satisfiable? The Dichotomy Conjecture states that for every CSP this task is either in \( P \) or is \( \text{NP}-\text{hard} \); the Algebraic Dichotomy Conjecture of Bulatov, Jeavons, and Krokhin \cite{53} refines this by conjecturing a precise algebraic characterization of the tractable CSP decision problems. However when it comes to approximability, not all tractable CSPs are “equally tractable”. E.g., for Max-Cut, not only can one efficiently find a completely satisfying assignment when one exists, the Goemans–Williamson algorithm \cite{94} efficiently finds an \textit{almost}-satisfying assignment whenever an \textit{almost}-satisfying assignment exists. (Specifically, it finds a \((1 - \tilde{O}(\sqrt{\epsilon}))\)-satisfying assignment whenever a \((1 - \epsilon)\)-satisfying assign-
ment exists.) Contrast this with the $k\text{Lin}(\text{mod } 2)$ problem, $k \geq 3$: again, one can efficiently find a completely satisfying assignment whenever one exists; however Håstad [116] has shown that finding even a somewhat-satisfying assignment whenever an almost-satisfying assignment exists is NP-hard. (Specifically, $\forall \epsilon > 0$ it is hard to find a $(1/2 + \epsilon)$-satisfying assignment when a $(1 - \epsilon)$-satisfying assignment exists.)

Prior work on robust decidability. In 1997, Zwick [228] initiated the study of the following very natural problem: which CSPs are efficiently robustly decidable? By this we mean that the algorithm should find $(1 - o(1))$-satisfying assignments whenever $(1 - \epsilon)$-satisfying assignments exist (formal definitions are given in Section 12.2). Zwick gave a linear programming (LP)-based algorithm for finding $(1 - O(1/\log(1/\epsilon)))$-satisfying assignments for Horn-$k$Sat (for any fixed $k$); he also gave a semidefinite programming (SDP)-based algorithm for finding $(1 - O(\epsilon^{1/3}))$-satisfying assignments for 2Sat (since improved to $1 - O(\epsilon^{1/2})$ [60]). Later, Khot [136] gave an SDP-based algorithm for finding $(1 - \tilde{O}(\epsilon^{1/5}))$-satisfying assignments for the notorious Unique-Games problem over domains $D$ with $|D| = O(1)$ (since improved to $1 - O(\epsilon^{1/2})$ [59]). On the other hand, the only tractable CSPs for which the robust decision problem seems to be NP-hard are the ones that can encode linear equations over groups.

Bounded width. If we wish to classify the CSPs which are efficiently robustly decidable, we seek a property that is shared by Horn-$k$Sat, 2Sat, and Unique-Games but not by $3\text{Lin}(\text{mod } p)$. From the algebraic viewpoint on CSPs there is a very obvious candidate: the former CSPs have bounded width while the latter does not. Briefly, a CSP is said to have bounded width if unsatisfiable instances can always be “refuted” in a proof system that only allows for constant-sized partial assignments to be kept “in memory” (again, more formal definitions are in Section 12.2). Recent independent works of Barto–Kozik [37] and Bulatov [52] have connected this notion to algebra by showing that bounded-width CSPs coincide with those which cannot encode linear equations over groups. Thus by Håstad’s work we know that any CSP which is efficiently robustly decidable must have bounded width (assuming $P \neq NP$). As mentioned at the beginning of this thesis, the Guruswami–Zhou Conjecture [113] states that the converse also holds: every bounded-width CSP has an efficient robust decision algorithm.

1We emphasize that in this chapter, we always treat the domain size $|D|$ as a fixed constant, with $\epsilon \to 0$ independently.
**Linear and semidefinite programming.** Essentially the only known way to produce CSP approximation algorithms is through the use of LPs and SDPs. Indeed, recent work of Raghavendra [189] shows that if one believes Khot’s Unique Games Conjecture [136], then a CSP $\Pi$ is efficiently robustly decidable if and only if the basic SDP relaxation robustly decides it. However understanding and solving SDPs can be difficult, and as Zwick’s Horn-$k$Sat algorithm illustrates, sometimes only the power of linear programming is needed for robust decision algorithms.

### 12.1.1 Our contributions

As a step towards the Guruswami–Zhou Conjecture, we show that a special case of the bounded width CSPs are robustly decidable by the basic linear programming relaxation. Somewhat informally stated, our main theorem is the following:

**Theorem 12.1.1** (Informal version of Theorem 12.3.1). Let $\Pi$ be any (finitely presented) CSP. Then the basic LP relaxation robustly decides $\Pi$ if $\Pi$ has width 1.

(Formal definitions of the terms in this theorem are given in Section 12.2.)

In slightly more details, our proof of Theorem 12.1.1 gives an efficient deterministic “LP-rounding” algorithm for actually finding the required almost-satisfying assignments. Quantitatively, it finds $(1 - O(1/\log(1/\epsilon)))$-satisfying assignments for $(1 - \epsilon)$-satisfiable instances, matching the performance of Zwick’s Horn-$k$Sat algorithm. As we describe below, this is best possible. Our rounding algorithm is also simpler than Zwick’s.

Independently and concurrently, Dalmau and Krokhin have also shown that width 1 CSPs have efficient robust decision algorithms. Their proof is different from ours; it is by a black-box reduction to Zwick’s Horn-Sat algorithm.

**The quantitative dependence on $\epsilon$.** As mentioned, our LP-based algorithm for width-1 CSPs finds $(1 - O(1/\log(1/\epsilon)))$-satisfying assignments to $(1 - \epsilon)$-satisfiable instances. One might hope for a better (say, polynomial) dependence on $\epsilon$ here. Unfortunately, this is not possible. Zwick [228] already showed that for Horn-3Sat there are “gap instances” where the basic LP has value $1-\epsilon$ but the optimum value is only $1-\Omega(1/\log(1/\epsilon))$. Indeed Guruswami and Zhou [113] extended this by showing there are equally bad gap instances for the basic SDP relaxation of Horn-3Sat. Assuming the Unique Games Conjecture, Raghavendra’s work [189] in turn implies that no polynomial-time algorithm can find $(1 - o(1/\log(1/\epsilon)))$-satisfying assignments to $(1 - \epsilon)$-satisfiable instances. On a positive
note, in Section [12.3.1] we show that for the special case of width-1 CSPs called “lattice CSPs”, the basic LP relaxation can be used to find \((1 - O(\epsilon))\)-satisfying assignments to \((1 - \epsilon)\)-satisfiable instances.

12.2 Preliminaries

12.2.1 CSP preliminaries

Definitions. Let \(D\) be a nonempty finite domain of values, and let \(\Gamma\) be a nonempty finite set of relations over \(D\), each of positive finite arity. We write such a \(k\)-ary relation as \(R : D^k \rightarrow \{0, 1\}\). An instance \(\mathcal{I}\) of the constraint satisfaction problem \(CSP(\Gamma)\) consists of a set \(V\) of \(n\) variables, along with a list of \(m\) constraints. Each constraint \(C\) is a pair \((S, R)\), where \(S\) is a tuple of some \(k\) variables (the scope of the constraint), and \(R\) is a \(k\)-ary relation in the set \(\Gamma\). We say that \(\mathcal{I}'\) is a sub-instance of \(\mathcal{I}\) if contains just a subset of the variables and constraints in \(\mathcal{I}\); it is induced by the variable set \(V' \subseteq V\) if it includes all constraints in \(\mathcal{I}\) involving just the variables in \(V'\). An assignment \(\alpha\) for an instance of \(CSP(\Gamma)\) is any mapping \(\alpha : V \rightarrow D\). The assignment satisfies a constraint \(C = (S, R)\) if \(R(\alpha(S)) = 1\) (where \(\alpha\) operates on \(S_t\) component-wise). The value of the assignment, \(\text{val}_{\mathcal{I}}(\alpha) \in [0, 1]\), is the fraction of constraints it satisfies. We define the optimum value of the instance \(\mathcal{I}\) to be \(\text{opt}(\mathcal{I}) = \max_{\alpha} \{\text{val}_{\mathcal{I}}(\alpha)\}\). We say the instance is satisfiable if \(\text{opt}(\mathcal{I}) = 1\).

CSP width. An important parameter of a \(CSP(\Gamma)\) problem is its width. This notion, dating back to Feder and Vardi [86], can be given many equivalent definitions (in terms of, e.g., pebble games, Datalog, logic, tree-width, proof complexity...). Roughly speaking, \(CSP(\Gamma)\) has width \(k\) if unsatisfiable instances of \(CSP(\Gamma)\) can always be refuted while only keeping \(k\) partial assignments “in memory”. More formally, given an instance \(\mathcal{I}\) of \(CSP(\Gamma)\), consider the following \((k, \ell)\) pebble game with \(1 \leq k < \ell\) integers: Alice begins by placing each of \(\ell\) pebbles on variables in \(V\). Bob must respond with a partial assignment to the pebbled variables which satisfies all constraints in which they participate. On each subsequent turn, Alice may move \(\ell - k\) of the pebbles to different vertices. Bob must respond with a partial assignment to the newly pebbled variables which again satisfies all constraints in which the pebbled variables participate, and which is consistent with the assignment to the \(k\) unmoved pebbles from the previous turn. If ever Bob cannot respond, Alice wins the game; if Bob can always play forever, he wins the game. If \(\mathcal{I}\) is a satisfiable instance then Bob can always win regardless of \(k\) and \(\ell\); on the other hand, if \(\mathcal{I}\)
is unsatisfiable, then Alice may or may not be able to win. We say that $CSP(\Gamma)$ has width $(k, \ell)$ if Alice can win the $(k, \ell)$ pebble game on all unsatisfiable instances; and, we say that $CSP(\Gamma)$ has width $k$ if it has width $(k, \ell)$ for some finite $\ell$. In particular, we say that $CSP(\Gamma)$ has bounded width if it has width $k$ for some finite $k$. Bounded width CSPs can be solved in polynomial time using a simple enumeration over Bob’s possible strategies. As examples, Horn-$k$Sat has width $1$, 2-Colorability has width $2$ (but not width $1$), and 3Lin(mod 2) does not have bounded width.

Tree duality and width 1. It is well known [86] that the CSPs of width 1 can be precisely characterized as those which have tree duality. We say that $CSP(\Gamma)$ has tree duality if for every unsatisfiable instance $I$ there is a unsatisfiable “tree” instance $T$ which “witnesses” this. By “witness” we mean that there is a homomorphism from $T$ to $I$; i.e., a map from $T$’s variables into $I$’s variables which preserves all relations. The definition of a “tree” instance is the natural one in case all relations in $\Gamma$ have arity 2; in general, we must make more careful definitions. We define a walk in instance $I$ of $CSP(\Gamma)$ to be a sequence $x_1, C_1 = (S_1, R_1), t_1, u_1, x_2, C_2 = (S_2, R_2), t_2, u_2, \ldots, x_{\ell+1}$ where each $x_i$ is a variable in $I$, each $C_i$ is a constraint in $I$, the indices $t_i$ and $u_i$ are distinct, and $(S_i)_{t_i} = x_i$, $(S_i)_{u_i} = x_{i+1}$ for all $i \in [\ell]$. We say the walk proceeds from starting point $x_1$ to endpoint $x_\ell$. We say the walk is non-backtracking if for every $i \in [\ell]$ either $C_i$ differs from $C_{i+1}$ or $u_i \neq t_{i+1}$. We say that $I$ is connected if there is a walk from $x$ to $y$ for all pairs of distinct variables $x$ and $y$ in $I$. Finally, we say that $I$ is a tree if it is connected and it does not contain any non-backtracking walk with the same starting point and endpoint.

### 12.2.2 Algorithmic preliminaries

**Approximation algorithms.** For real numbers $0 \leq s \leq c \leq 1$, we say an algorithm $(c, s)$-approximates $CSP(\Gamma)$ if it outputs an assignment with value at least $s$ on any input instance with value at least $c$. For $c = s = 1$ we simply say that the algorithm decides $CSP(\Gamma)$; this means the algorithm always finds a satisfying assignment given a satisfiable instance. We say that an algorithm robustly decides $CSP(\Gamma)$ if there is an error function $r : [0, 1] \to [0, 1]$ with $r(\epsilon) \to 0$ as $\epsilon \to 0$ such that the algorithm $(1 - \epsilon, 1 - r(\epsilon))$-approximates $CSP(\Gamma)$ for all $\epsilon \in [0, 1]$. In particular, the algorithm must decide $CSP(\Gamma)$.

**The basic integer program.** For any instance $\mathcal{I}$ of $CSP(\Gamma)$ there is an equivalent canonical 0-1 integer program we denote by $IP(\mathcal{I})$. It has variables $p_v(j)$ for each $v \in V$, $j \in D$, as well as variables $q_{C_i}(J)$ for each arity-$k$ constraint $C_i = (S_i, R_i)$ and tuple.
\( J \in D^{k_i} \). The interpretation of \( p_v(j) = 1 \) is that variable \( v \) is assigned value \( j \); the interpretation of \( q_{C_i}(J) = 1 \) is that the \( k_i \)-tuple of variables \( S_i \) is assigned the \( k_i \)-tuple of values \( J \). More formally, IP(\( \mathcal{I} \)) is the following:

\[
\text{maximize } \frac{1}{m} \sum_{i=1}^{m} \sum_{J: R_i(J) = 1} q_{C_i}(J)
\]

subject to:

\[
\sum_{j \in D} p_v(j) = 1 \quad \text{for all } v \in V, \tag{12.1}
\]

\[
\sum_{J \in D^{k_i}: J_t = j} q_{C_i}(J) = p_v(j) \quad \text{for all } C_i \text{ and } t \text{ such that } (S_i)_t = v. \tag{12.2}
\]

The optimum value of IP(\( \mathcal{I} \)) is precisely opt(\( \mathcal{I} \)). Note that the size of this integer programming formulation is \( \text{poly}(n, m) \) (as we are assuming that \( D \) and \( \Gamma \) are of constant size).

**The basic linear program.** If we relax IP(\( \mathcal{I} \)) by having the variables take values in the range \([0, 1]\) rather than \(\{0, 1\}\), we obtain the basic linear programming relaxation which we denote by LP(\( \mathcal{I} \)). An optimal solution of LP(\( \mathcal{I} \)) can be computed in \( \text{poly}(n, m) \) time; the optimal value, which we denote by \( \text{opt}_{\text{LP}}(\mathcal{I}) \), always satisfies \( \text{opt}(\mathcal{I}) \leq \text{opt}_{\text{LP}}(\mathcal{I}) \leq 1 \).

We interpret any feasible solution to LP(\( \mathcal{I} \)) as follows: For each \( v \in V \), the quantities \( p_v(j) \) form a discrete probability distribution on \( D \) (because of (12.1)), denoted \( p_v \). For each \( k_i \)-ary constraint \( C_i = (S_i, R_i) \), the quantities \( q_{C_i}(J) \) form a probability distribution on \( D^{k_i} \), denoted \( q_{C_i} \). Furthermore (because of (12.2)), the marginals of the \( q_{C_i} \) distributions are “consistent” with the \( p_v \) distributions, in the sense that whenever \( (S_i)_t = v \) it holds that \( \Pr_{J \sim q_{C_i}}[J_t = j] = p_v(j) \) for all \( j \in D \). Finally, the objective value to be optimized in LP(\( \mathcal{I} \)) is

\[
\text{val}_{\text{LP}}(\{p_v\}, \{q_{C_i}\}) = \frac{1}{m} \sum_{i=1}^{m} \Pr_{J \sim q_{C_i}}[R_i(J) = 1];
\]

the optimum value of this over all feasible solutions is \( \text{opt}_{\text{LP}}(\mathcal{I}) \).

### 12.2.3 Algebraic preliminaries

**Polymorphisms.** The Dichotomy Conjecture of Feder and Vardi [86] asserts that for each \( \Gamma \), the problem of deciding CSP(\( \Gamma \)) is either in P or is NP-complete. The most successful approach towards this conjecture has been the algebraic one initiated by Jeavons...
and coauthors [125] in which the problem is studied through the polymorphisms of $\Gamma$. We say $f : D^\ell \to D$ is an $\ell$-ary polymorphism for the $k$-ary relation $R$ if $R(f(x^1), \ldots, f(x^k)) = 1$ whenever $R(x^1, \ldots, x^j) = 1$ for all $i \in [\ell]$ (here each $x^j$ is a tuple in $D^\ell$). We say that $f$ is a polymorphism for $\Gamma$ if it is a polymorphism for each relation in $\Gamma$. We say that $\Gamma$ is a core, if all of its 1-ary polymorphisms are bijections (at a high level, this means that there are no superfluous values in $D$ for $CSP(\Gamma)$). Finally, we call a polymorphism $f$ idempotent, if $f(j, \ldots, j) = j$ for all $j \in D$.

**Polymorphisms and width.** Recently, independent works of Barto–Kozik [37] and Bulatov [52] managed to characterize bounded-width CSPs in terms of their polymorphisms. Specifically, they showed that $CSP(\Gamma)$ has bounded width (for $\Gamma$ a core) if and only if $\Gamma$ has an $\ell$-ary weak near-unanimity (WNU) polymorphism for all $\ell \geq 3$. Here a polymorphism $f$ is said to be WNU if it is idempotent and has the following symmetry: $f(x, x, \ldots, x, y) = f(x, \ldots, x, y, x) = \cdots = f(x, y, x, \ldots, x) = f(y, x, x, \ldots, x)$.

Much earlier, Dalmau and Pearson [73] gave a straightforward characterization the class of width-1 CSPs in terms of their polymorphisms. Specifically, they showed that $CSP(\Gamma)$ has width 1 if and only if $\Gamma$ is preserved by a set operation $g : \mathcal{P}(D) \to D$. This means that $f : D^\ell \to D$ defined by $f(x_1, \ldots, x_\ell) = g(\{x_1, \ldots, x_\ell\})$ is a polymorphism for all $\ell \geq 1$. Note that all these polymorphisms are symmetric, meaning invariant under all permutations of the inputs. We will use a simple lemma about width-1 CSPs which first requires a definition.

**Definition 12.2.1.** Let $J$ be a subset of a cartesian product $B_1 \times \cdots \times B_k$ of nonempty sets. We say $J$ is subdirect, written $J \subseteq S B_1 \times \cdots \times B_k$, if for each $i \in [k]$ the projection of $J$ to the $i$’th coordinate is all of $B_i$.

**Lemma 12.2.2.** Say $g$ is a set operation for $CSP(\Gamma)$, $R$ is an arity-$k$ relation in $\Gamma$, and $B_1, \ldots, B_k \subseteq D$. Assume there is a $J \subseteq S B_1 \times \cdots \times B_k$ all of whose members satisfy $R$. Then $R(\{g(B_1), \ldots, g(B_k)\}) = 1$.

**Proof.** For each $t \in [k]$ and $j \in B_t$, select some $J^{t,j} \in J$ whose $t$’th coordinate is $j$. Think of the $\ell = \sum |B_i| \geq 1$ tuples $J^{t,j}$ as column vectors, and adjoin them in some order to form a $k \times \ell$ matrix $X$. Let $x^t$ be the $t$th row of $X$. It is clear that the set of values appearing in $x^t$ is precisely $B_t$. Thus if $f$ is the $\ell$-ary polymorphism defined by $g$, we have $f(x^t) = g(B_t)$. But since $f$ is a polymorphism and each $J^{t,j}$ satisfies $R$, it follows that $R(\{g(B_1), \ldots, g(B_k)\}) = 1$. \qed
12.3 Width 1 implies robust decidability by LP

The following theorem shows that a simple rounding algorithm for the basic linear program robustly decides any width-1 CSP.

**Theorem 12.3.1.** Let $\Gamma$ be a finite set of relations over the finite domain $D$, each relation having arity at most $K$. Assume that $\text{CSP}(\Gamma)$ has width 1. Then there is a \text{poly}(n,m)-time algorithm for $\text{CSP}(\Gamma)$ which when given an input $\mathcal{I}$ with $\text{val}^{\text{LP}}(\mathcal{I}) = 1 - \epsilon$ outputs an assignment $\alpha : V \rightarrow D$ with $\text{val}_I(\alpha) \geq 1 - O(K^2|D| \log(2|D|))/\log(1/\epsilon)$. (In particular, $\text{val}_I(\alpha) = 1$ if $\text{val}^{\text{LP}}(\mathcal{I}) = 1$.)

**Proof.** The first step of the algorithm is to solve the LP relaxation of the instance, determining an optimal solution $\{p_v : v \in V\}$, $\{q_{C_i} : i \in [m]\}$ which obtains $\text{val}^{\text{LP}}(\mathcal{I}) = 1 - \epsilon$.

For technical reasons we will now assume without loss of generality that $K \geq 2$ and that $2^{-\text{poly}(n,m)} \leq \epsilon \leq 1/4(2^{|D|})^{2(K-1)/\log(1/\epsilon)}$. (12.3)

The assumption (12.3) is also without loss of generality. We may assume the upper bound by adjusting the constant in the $O(\cdot)$ of our theorem. As for the lower bound, since linear programming is in polynomial time, $\epsilon$ will be either 0 or at least $2^{-\text{poly}(n,m)}$. In the former case, we replace $\epsilon$ with a sufficiently small $2^{-O(m)}$ so that the theorem’s claimed lower bound on $\text{val}_I(\alpha)$ exceeds $1 - 1/m$; then $\text{val}_I(\alpha) > 1 - 1/m$ implies $\text{val}_I(\alpha) = 1$ as required when $\text{val}^{\text{LP}}(\mathcal{I}) = 1$.

For a particular constraint $C_i$, let $\epsilon_i = \sum_{J:R_i(J) = 0} q_{C_i}(J)$. Since $\text{val}^{\text{LP}}(\mathcal{I}) \geq 1 - \epsilon$ we have $\text{avg} \{\epsilon_i\} \leq \epsilon$. The next step is to “give up” on any constraint having $\epsilon_i > \sqrt{\epsilon}$. By Markov’s inequality the fraction of such constraints is at most $\sqrt{\epsilon}$, which is negligible compared to the $O(1/\log(1/\epsilon))$ error guarantee of our algorithm. For notational simplicity, we now assume that $\epsilon_i \leq \sqrt{\epsilon}$ for all $i \in [m]$.

We now come to the main part of the algorithm. Since $\text{CSP}(\Gamma)$ has width 1, it has a set operation $g : \mathcal{P}(D) \rightarrow D$. We first describe a simple randomized “LP-rounding” algorithm based on $g$:

1. Let $r = (2|D|)^{K-1}$ and let $b = |\log_r(1/2\sqrt{\epsilon})|$. We have $b \geq 1$ by (12.3).

2. Choose $\theta \in \{r^{-1}, r^{-2}, \ldots, r^{-b}\}$ uniformly at random. Note that $r^{-b} \geq 2\sqrt{\epsilon}$.

3. Output the assignment $\alpha : V \rightarrow D$ defined by $\alpha(v) = g(\text{supp}_\theta(p_v))$, where $\text{supp}_\theta(p_v)$ denotes $\{j \in D : p_v(j) \geq \theta\}$.
We will show for each constraint \( C_i = (S_i, R_i) \) that
\[
\Pr[R_i(\alpha(S_i)) = 0] \leq K|D|/b. \tag{12.4}
\]

It follows from linearity of expectation that the expected fraction of constraints not satisfied by \( \alpha \) is at most \( K|D|/b = O(K^2|D| \log(2|D|)) / \log(1/\epsilon) \). This would complete the proof, except for the fact that we have given a randomized algorithm. However we can easily make the algorithm deterministic and efficient by trying all choices for \( \theta \) (of which there are at most \( b \leq \text{poly}(n, m) \) by (12.3)) and selecting the best resulting assignment.

We now give the analysis justifying (12.4) for each fixed constraint \( C_i = (S_i, R_i) \). For simplicity we henceforth write \( C = C_i, S = S_i, R = R_i \) and suppose that \( R \) has arity \( k \leq K \). Let us say that a choice of \( \theta \) is bad if it falls into the interval \((p_{S_t}(j)/r, p_{S_t}(j))\) for some \( t \in [k] \) and \( j \in D \). For each choice of \( t \) and \( j \) there is at most one bad choice of \( \theta \) for the associated interval; hence the overall probability \( \theta \) is bad is at most \( K|D|/b \). Thus it suffices to show that whenever \( \theta \) is not bad, \( C \) is satisfied by \( \alpha \).

For each \( t \in [k] \) let \( B_t = \text{supp}_\theta(p_{S_t}) \); these sets are nonempty because \( \theta \leq r^{-1} \leq |D|^{-1} \). Also, let \( \mathcal{J} = \{ J \in B_1 \times \cdots \times B_k : R(j) = 1 \} \). By Lemma 12.2.2 to show that \( C \) is satisfied by \( \alpha \), we only need to show that \( J \subseteq S \times B_1 \times \cdots \times B_k \), i.e., that for all \( t \in [k] \) and all \( j \in B_t \) there exists a tuple \( J \in \mathcal{J} \) such that \( J_t = j \). We show this is true for \( t = k \) and the statement for other values of \( t \) follows in the same way. For any \( j \in B_k \), we have \( \theta \leq p_{S_k}(j) \) by the definition of \( B_t \). Since \( \theta \) is not bad, we know that \( \theta \not\in (p_{S_t}(j)/r, p_{S_t}(j)) \). Therefore we have \( \theta \leq p_{S_t}(j)/r \). Now since all but at most \( \epsilon \leq \sqrt{\epsilon} \) of the probability mass in \( q_C \) is on assignments satisfying \( R \), we conclude
\[
\sum_{J' \in D^{k-1}, R(J', j) = 1} q_C(J', j) \geq p_{S_t}(j) - \sqrt{\epsilon} \geq p_{S_t}(j)/2.
\]

Here we used \( 2\sqrt{\epsilon} \leq r^{-b} \leq \theta \leq p_{S_t}(j) \). Now the pigeonhole principle implies there exists some \( J' \in D^{k-1} \) with \( R(J', j) = 1 \) and \( q_C(J', j) \geq p_{S_t}(j)/(2|D|^{k-1}) \geq p_{S_t}(j)/r \). By consistency of marginals this certainly implies \( p_{S_t}(J'/r) \geq p_{S_t}(j)/r \geq \theta \) for all \( t' \in [k-1] \). Now for all \( t' \in [k-1] \) we know that \( J_{S_{t'}} \in B_{t'} \). Therefore, if we let \( J = (J', j) \) we have that \( J \in \mathcal{J} \) and \( J_k = j \).

### 12.3.1 Lattice CSPs: better quantitative dependence on \( \epsilon \)

As discussed at the end of Section 12.1.1, one cannot hope to improve the approximation guarantee of \( 1 - O(1/\log(1/\epsilon)) \) given by our LP-rounding algorithm, even in the case of Horn-3Sat. On the other hand, for Horn-2Sat it is known [135] that on \((1 - \epsilon)\)-satisfiable
instances one can efficiently find \((1 - O(\epsilon))\)-satisfying assignments (indeed, \((1 - 2\epsilon)\)-satisfying \([113]\)). One might ask what the algebraic difference is between Horn-2Sat and Horn-3Sat. A notable difference is that the former is a \textit{lattice} CSP.

**Subclasses of width-1: lattice and semilattice CSPs.** A broad natural subclass of the width-1 CSPs is the class of \textit{semilattice} CSPs. These are CSPs which have a \textit{semilattice polymorphism}, meaning a binary polymorphism \(\wedge\) which is associative, commutative, and idempotent. Horn-Sat CSPs are not just width-1 but are in fact semilattice; thus we cannot hope for improved dependence on \(\epsilon\) even for semilattice CSPs.\(^2\)

An even further subclass is that of \textit{lattice} CSPs. These are CSPs whose relations are preserved by \textit{two} semilattice operations \(\wedge\) and \(\vee\) which additionally satisfy the “absorption” identity: \(\vee(x, \wedge(x, y)) = \wedge(x, \vee(x, y)) = x\). Note that \(\vee\) and \(\wedge\) extend naturally to polymorphisms of every arity. Good examples of lattice CSPs are “lattice retraction problems”. Here there is a fixed lattice poset \(L\); the CSP’s domain is \(L\) and its constraints are the poset constraint “\(\leq\)” along with all unary constraints “\(=a\)” for \(a \in L\).

**Robust decidability for lattice CSPs.** In this subsection we prove a variant of our Theorem \([12.3.1]\) which shows an efficient LP-based algorithm for finding \((1 - O(\epsilon))\)-satisfying assignments to \((1 - \epsilon)\)-satisfiable lattice CSP instances.

We first describe the characterization of lattice CSPs we need. Carvalho, Dalmau, and Krokhin \([58]\) have observed that if \(CSP(\Gamma)\) has lattice polymorphisms then it is preserved by what they call an \textit{absorptive block-symmetric operation}. This is an operation \(f\) which takes as input tuples (of any positive length) of nonempty subsets of \(D\), outputs an element of \(D\), and has the following properties:

- (Block-symmetry.) \(f(B_1, \ldots, B_\ell)\) only depends on \(\{B_1, \ldots, B_\ell\}\).
- (Absorption.) If \(B \supseteq B_1\) then \(f(B, B_1, \ldots, B_\ell) = f(B_1, \ldots, B_\ell)\).
- (Preservation.) Let \(R\) be an arity-\(k\) relation \(\Gamma\) and let \((B_i^j)_{i=1}^\ell\) be nonempty subsets of \(D\). Assume that for each \(i \in [\ell]\) there is a \(J_i \subseteq_R B_1^i \times \cdots \times B_k^i\) all of whose members satisfy \(R\). Then \(R(f(B_1^1, \ldots, B_1^\ell), \ldots, f(B_1^k, \ldots, B_\ell^k)) = 1\).

\(^2\)There are CSPs which are width-1 but not semilattice; e.g., the CSP over domain \(\{a, b, c, d\}\) with all unary relations and also the binary relations \((a, b), (b, a), (c, a), (c, b), (c, d), (d, c),\) and \((d, d)\).
Indeed, the operation $f$ is simply $f(B_1, \ldots, B_\ell) = \bigvee \{ \land B_i : i \in [\ell] \}$.

We now show:

**Theorem 12.3.2.** Let $\Gamma$ be a finite set of relations over the finite domain $D$, each relation having arity at most $K$. Assume that CSP($\Gamma$) has lattice polymorphisms. Then there is a poly($n, m$)-time algorithm for CSP($\Gamma$) which when given an input $I$ with val$^{LP}(I) = 1 - \epsilon$ outputs an assignment $\alpha : V \rightarrow D$ with val$_I(\alpha) \geq 1 - O(K2^{D})\epsilon$.

**Proof.** As in Theorem 12.3.1, the first step of the algorithm is to solve the LP relaxation of the instance, determining an optimal solution $\{ p_v : v \in V \}$, $\{ q_c : i \in [m] \}$ which obtains val$^{LP}(I) = 1 - \epsilon$. Since CSP($\Gamma$) has lattice polymorphisms, it has some absorptive block-symmetric operation $f$. We next describe a randomized LP-rounding algorithm:

1. Set $r = (2K2^{[D]}m)^{-1}$ and choose
   \[ \theta \in \{1/r, 2/r, 3/r, \ldots, 1\} \] uniformly at random.

2. For each $v \in V$, define $B_v = \{ B \subseteq D : p_v(B) \geq \theta \}$, a nonempty family of nonempty sets. (Here we introduce the notation $p_v(B) = \sum_{b \in B} p_v(b)$.)

3. Output the assignment $\alpha : V \rightarrow D$ defined by $\alpha(v) = f(B_v)$.

We will show for each constraint $C_i = (S_i, R_i)$ that

\[ \Pr[R_i(\alpha(S_i)) = 0] \leq K2^{[D]}(\epsilon_i + 1/r) = K2^{[D]}\epsilon_i + 1/2m, \]  \hspace{1cm} (12.5)

where $\epsilon_i = \sum_{J : R_i(J) = 0} q_{c_i}(J)$ as in the previous proof. It then follows from linearity of expectation that the expected fraction of constraints not satisfied by $\alpha$ is at most $K2^{[D]} \text{avg}\{ \epsilon_i \} + 1/2m = K2^{[D]}\epsilon + 1/2m$. We can therefore efficiently deterministically find an $\alpha$ with value at least $1 - K2^{[D]}\epsilon - 1/2m$ by trying all $O(m)$ possible values for $\theta$. This is sufficient to prove the theorem: if $\epsilon < (2K2^{[D]}m)^{-1}$ then $\alpha$’s value exceeds $1 - 1/m$ and hence is in fact 1; if $\epsilon \geq (2K2^{[D]}m)^{-1}$ then the $O(\cdot)$ in the theorem statement covers the loss of $1/2m$.

We now give the analysis justifying (12.5) for each fixed constraint $C_i = (S_i, R_i)$. For simplicity we henceforth write $C = C_i$, $S = S_i$, $R = R_i$ and suppose that $R$ has arity $k \leq K$. It suffices to show that $R(\alpha(S)) = 1$ holds assuming

\[ \theta \not\in (p_{S_i}(B), p_{S_i}(B) + \epsilon_i) \quad \forall t \in [k], \forall B \subseteq D. \]  \hspace{1cm} (12.6)

The reason is that the probability of (12.6) not holding is at most $K2^{[D]}(\epsilon_i + 1/r)$. Note that with assumption (12.6), whenever we have $p_{S_i}(B) \geq \theta - \epsilon_i$ it follows that in fact $p_{S_i}(B) \geq \theta$ and thus $B \in B_{S_i}$.
Claim 12.3.3. For all $t \in [k]$ and $B \in \mathcal{B}_{S_t}$, there exist $B_1, \ldots, B_k$ with $B_u \in \mathcal{B}_{S_u}$ such that: a) $B_t \subseteq B$; b) there exists $J \subseteq S B_1 \times \cdots \times B_k$ with $R(J) = 1$ for all $J \in J$.

Proof. Suppose $B \in \mathcal{B}_{S_t}$, so $p_{S_t}(B) \geq \theta$. Letting $J' = \{ J \in D^k : J_t \subseteq B \}$, it follows from consistency of marginals that $q_{C}(J') \geq \theta$. Thus if $J$ is the subset of $J'$ for which $R$ holds, it follows that $q_{C}(J) \geq \theta - \epsilon_i$. For $u \in [k]$, we define $B_u = \{ J_u : J \in J \}$. Certainly $B_t \subseteq B$, and by consistency of marginals we obtain from $q_{C}(J) \geq \theta - \epsilon_i$ that $p_{S_u}(B_u) \geq \theta - \epsilon_i$ for each $u \in [k]$. Thus it follows from assumption (12.6) that $B_u \in \mathcal{B}_{S_u}$ for each $u$, completing the proof of the claim.

For each choice of $t \in [k]$ and $B \in \mathcal{B}_{S_t}$, take the (names of the) sets $B_1, \ldots, B_k$ given by the above claim and arrange them in a height-$k$ column. Adjoin all of these columns to form a $k \times \ell$ matrix $M$, where $\ell = \sum_{t=1}^{k} |\mathcal{B}_{S_t}|$. The matrix $M$ has the following properties: (i) each entry in row $u$ is a set in $\mathcal{B}_{S_u}$; (ii) for each set $B \in \mathcal{B}_{S_u}$, some subset of it appears in the $u^{th}$ row of $M$; (iii) for each column $(B_1, \ldots, B_k)$ of $M$ there is a $J \subseteq S B_1 \times \cdots \times B_k$ all of whose members satisfy $R$.

Suppose we now apply the absorptive block-symmetric operation $f$ to the rows of $M$, with the $u^{th}$ row producing $j_u \in D$. By (iii), $R(j_1, \ldots, j_k) = 1$. Thus the justification of (12.5) is complete if we can show $j_u = f(B_{S_u}) = \alpha(S_u)$. But this follows from (i), (ii), and the absorptive property of $f$. \qed
Chapter 13

An algorithm for

ROBUSTMAXBISECTION

13.1 Introduction

In the MAXCUT problem, we are given a graph and the goal is to partition the vertices into two parts so that a maximum number of edges cross the cut. As one of the most basic problems in the class of constraint satisfaction problems, the study of MAXCUT has been highly influential in advancing the subject of approximability of optimization problems, from both the algorithmic and hardness sides. The celebrated Goemans-Williamson (GW) algorithm for MAXCUT [24] was the starting point for the immense and highly successful body of work on semidefinite programming (SDP) based approximation algorithms. This algorithm guarantees finding a cut whose value (i.e., fraction of edges crossing it) is at least 0.878 times the value of the maximum cut. On graphs that are “almost-bipartite” and admit a partition such that most edges cross the cut, the algorithm performs much better — in particular, if there is a cut such that a fraction \((1 - \epsilon)\) of edges cross the cut, then the algorithm finds a partition cutting at least a fraction \(1 - O(\sqrt{\epsilon})\) of edges. (All through the discussion in the paper, think of \(\epsilon\) as a very small positive constant.)

On the hardness side, the best known NP-hardness result shows hardness of approximating MAXCUT within a factor greater than 16/17 [116, 219]. Much stronger and in fact tight inapproximability results are now known conditioned on the Unique Games Conjecture of Khot [136]. In fact, one of the original motivations and applications for the formulation of the UGC in [136] was to show that finding a cut of value larger than \(1 - o(\sqrt{\epsilon})\) in a graph with Max-Cut value \((1 - \epsilon)\) (i.e., improving upon the above-mentioned perfor-
mance guarantee of the GW algorithm substantially) is likely to be hard. This result was strengthened in [141] to the optimal $1 - O(\sqrt{\epsilon})$ bound, and this paper also showed that the 0.878 approximation factor is the best possible under the UGC. (These results relied, in addition to the UGC, on the Majority is Stablest conjecture, which was proved shortly afterwards in [174].)

There are many other works on MAXCUT, including algorithms that improve on the GW algorithm for certain ranges of the optimum value and integrality gap constructions showing limitations of the SDP based approach. This long line of work on MAXCUT culminated in the paper [183] which obtained the precise integrality gap and approximation threshold curve as a function of the optimum cut value.

**Maximum Bisection.** Let us consider a closely related problem called MAXBISECTION, which is MAXCUT with a global “balanced cut” condition. In the MAXBISECTION problem, given as input a graph with an even number of vertices, the goal is to partition the vertices into two equal parts while maximizing the fraction of cut edges. Despite the close relation to MAXCUT, the global constraint in MAXBISECTION changes its character substantially, and the known algorithms for approximating MAXBISECTION have weaker guarantees. While MAXCUT has a factor 0.878 approximation algorithm [94], the best known approximation factor for MAXBISECTION equals 0.7027 [91] improving on previous bounds of 0.6514 [93], 0.699 [226], and 0.7016 [115].

In terms of inapproximability results, it is known that MAXBISECTION cannot be approximated to a factor larger than $15/16$ unless $\text{NP} \subseteq \bigcap_{\gamma > 0} \text{TIME}(2^{\gamma n})$ [120]. Note that this hardness factor is slightly better than the inapproximability factor of $16/17$ known for MAXCUT [116, 219]. A simple approximation preserving reduction from MAXCUT shows that MAXBISECTION is no easier to approximate than MAXCUT (the reduction is simply to take two disjoint copies of the MAXCUT instance). Therefore, the factor 0.878 Unique-Games hardness for MAXCUT [141] also applies for MAXBISECTION. Further, given a graph that has a bisection cutting $1 - \epsilon$ of the edges, it is Unique-Games hard to find a bisection (or even any partition in fact) cutting $1 - O(\sqrt{\epsilon})$ of the edges.

An intriguing question is whether MAXBISECTION is in fact harder to approximate than MAXCUT (so the global condition really changes the complexity of the problem), or whether there are algorithms for MAXBISECTION that match (or at least approach) what

---

1 At the time when this chapter was published. This ratio was later improved by [195] and subsequently [28], as we will mention soon.
is known for \textsc{MaxCut}\textsuperscript{2} None of the previously known algorithms\textsuperscript{3} for \textsc{MaxBisection} \cite{93,226,115,91} are guaranteed to find a bisection cutting most of the edges even when the graph has a near-perfect bisection cutting $(1 - \epsilon)$ of the edges (in particular, they may not even cut 75\% of the edges). These algorithms are based on rounding a vector solution of a semidefinite programming relaxation into a cut (for example by a random hyperplane cut) and then balancing the cut by moving low-degree vertices from the larger side to the smaller side. After the first step, most of the edges are cut, but the latter rebalancing step results in a significant loss in the number of edges cut. In fact, as we will illustrate with a simple example in \textbf{Section 13.2}, the standard SDP for \textsc{MaxBisection} has a large integrality gap: the SDP optimum could be 1 whereas every bisection might only cut less than 0.95 fraction of the edges.

Thus an interesting “qualitative” question is one can one efficiently find an \textit{almost-complete} bisection when promised that one exists. Formally, we ask the following question.

\textbf{Question 13.1.1.} \textit{Is there a polynomial time algorithm that given a graph $G = (V,E)$ with a \textsc{MaxBisection} solution of value $(1 - \epsilon)$, finds a bisection of value $(1 - g(\epsilon))$, where $g(\epsilon) \to 0$ as $\epsilon \to 0$?}

Indeed, this question motivates the problem of \textsc{RobustMaxBisection} defined previously in \textbf{Definition 2.2.6}. We restate the definition here for reader’s convenience.

\begin{definition}[\textsc{RobustMaxBisection}, re-statement of \textbf{Definition 2.2.6}] We say that an algorithm $A$ solves the \textsc{RobustMaxBisection} problem if there exists a function $r : [0,1] \to [0,1]$ satisfying $r(\epsilon) \to 0$ as $\epsilon \to 0^+$ such that whenever $A$ is given an undirected graph $G$ with \textsc{MaxBisection} optimum at least $1 - \epsilon$, $A$ outputs a bisection with $(1 - r(\epsilon))$ of the edges across the bisection.
\end{definition}

Note that without the bisection constraint, we can achieve $g(\epsilon) = O(\sqrt{\epsilon})$, and when $\epsilon = 0$, we can find a bisection cutting all the edges (see the first paragraph of \textbf{Section 13.2} for details). Thus this question highlights the role of both the global constraint and the “noise” (i.e., $\epsilon$ fraction of edges need to be removed to make the input graph bipartite) on the complexity of the problem.

\textsuperscript{2}Note that for the problem of minimizing the number of edges cut, the global condition does make a big difference: \textsc{MinCut} is polynomial-time solvable whereas \textsc{MinBisection} is NP-hard.

\textsuperscript{3}Also by the time when this chapter was published.
13.1.1 Our contributions

In this chapter, we answer the above question in the affirmative, by proving the following theorem.

**Theorem 13.1.3.** There is a randomized polynomial-time algorithm such that for every $\epsilon > 0$, given an edge-weighted graph $G$ with a MAXBISECTION solution of value $\epsilon(1 - \epsilon)$, finds a MAXBISECTION of value $(1 - O(3\sqrt{\epsilon \log(1/\epsilon)))$.

We remark that for regular graphs any cut with most of the edges crossing it must be near-balanced, and hence we can solve MAXBISECTION by simply reducing to MAXCUT. Thus the interesting instances for our algorithm are non-regular graphs.

Our algorithms are not restricted to finding exact bisections. If the graph has a $\beta$-balanced cut (which means the two sides have a fraction $\beta$ and $1 - \beta$ of the vertices) of value $(1 - \epsilon)$, then the algorithm can find a $\beta$-balanced cut of value $1 - O(\epsilon^{1/3} \log(1/\epsilon))$.

Our results are not aimed at improving the general approximation ratio for MAXBISECTION which remains at $\approx 0.7027$ [91]. More generally, our work highlights the challenge of understanding the complexity of solving constraint satisfaction problems with global constraints. Algorithmically, the challenge is to ensure that the global constraint is met without hurting the number of satisfied constraints. From the hardness side, the Unique Games based reductions which have led to a complete understanding of the approximation threshold of CSPs [189] are unable to exploit the global constraint to yield stronger hardness results.

13.1.2 Later development

Soon after our work was published, Raghavendra and Tan [195] also studied the ROBUSTMAXBISECTION problem, improving our algorithm and giving an algorithm that finds a MAXBISECTION of value $(1 - O(\sqrt{\epsilon}))$ when there is a MAXBISECTION of value $(1 - \epsilon)$. As mentioned earlier, as a function of $\epsilon$, this approximation is best possible assuming the Unique Games Conjecture. However, one thing not very satisfactory about the Raghavendra–Tan algorithm is that their algorithm runs in time $n^{1/\epsilon^{\Theta(1)}}$, which is not completely polynomial in $n$ when $\epsilon$ is subconstant (say, $\frac{1}{\text{polylog}(n)}$). It is an interesting open question whether the same approximation guarantee can be obtained in $\text{poly}(n)$ time for every $\epsilon$.

$^4$The value of a cut in an edge-weighted graph is defined as the weight of the edges crossing the cut divided by the total weight of all edges.
Using the same technique, Raghavendra and Tan [195] also gave an algorithm substantially improving the approximation ratio of \textsc{MaxBisection} to $\approx 0.852$. This approximation ratio was later improved by [28] to $\approx 0.8776$. It remains an interesting question how much this can be improved and whether one approach (or even match) the 0.878 factor possible for \textsc{MaxCut}.

13.2 Method overview

13.2.1 Integrality gap

We begin by describing why the standard SDP for \textsc{MaxBisection} has a large gap. Given a graph $G = (V, E)$, this SDP, which is the basis of all previous algorithms for \textsc{MaxBisection} starting with that of Frieze and Jerrum [93], solves for unit vectors $v_i$ for each vertex $i \in V$ subject to $\sum_i v_i = 0$, while maximizing the objective function $E_{e=(i,j) \in E} \frac{1}{2} \|v_i - v_j\|^2$.

This SDP could have a value of 1 and yet the graph may not have any bisection of value more than 0.95 (in particular the optimum is bounded away from 1), as the following example shows. Take $G$ to be the union of three disjoint copies of $K_{2m,m}$ (the complete $2m \times m$ bipartite graph) for some even $m$. It can be seen that every bisection fails to cut at least $m^2/2$ edges, and thus has value at most $11/12$. On the other hand, the SDP has a solution of value 1. Let $\omega = e^{2\pi i/3}$ be the primitive cube root of unity. In the two-dimensional complex plane, we assign the vector/complex number $\omega^{i-1}$ (resp. $-\omega^{i-1}$) to all vertices in the larger part (resp. smaller part) of the $i^{th}$ copy of $K_{2m,m}$ for $i = 1, 2, 3$. These vectors sum up to 0 and for each edge, the vectors associated with its endpoints are antipodal.

For all CSPs, a tight connection between integrality gaps (for a certain “canonical” SDP) and inapproximability results is now established [189]. The above gap instance suggests that the picture is more subtle for CSPs with global constraints — in this work we give an algorithm that does much better than the integrality gap for the “basic” SDP. Could a stronger SDP relaxation capture the complexity of approximating CSPs with global constraints such as \textsc{MaxBisection}? It is worth remarking that we do not know whether an integrality gap instance of the above form (i.e., $1 - \epsilon$ SDP optimum vs. say 0.9 \textsc{MaxBisection} value) exists even for the basic SDP augmented with triangle inequalities.
13.2.2 Notations

Suppose we are given a graph \( G = (V, E) \). We use the following notation: \( E(U) = \{(u, v) \in E : u, v \in U\} \) denotes the set of edges within a set of vertices \( U \), \( \text{edges}(U_1, U_2) = \{(u, v) \in E : u \in U_1, v \in U_2\} \) denotes the set of edges between two sets of vertices \( U_1 \) and \( U_2 \), and \( G[U] \) denotes the subgraph of \( G \) induced by the set \( U \).

**Definition 13.2.1 (Value and bias of cuts).** For a cut \( (S, V \setminus S) \) of a graph \( G = (V, E) \), we define its value to be \( \frac{\text{edges}(S, V \setminus S)}{|E|} \) (i.e., the fraction of edges which cross the cut) if \( G \) is unweighted, and \( \frac{w(\text{edges}(S, V \setminus S))}{w(E)} \) if \( G \) is edge-weighted with weight function \( w : E \to \mathbb{R} \geq 0 \) (where for \( F \subseteq E \), \( w(F) = \sum_{e \in F} w(e) \)).

We define the bias \( \beta \in [0, 1] \) of a cut \( (S, V \setminus S) \) to be \( \beta = \frac{1}{|V|} \cdot \left| |S| - |V \setminus S| \right| \), and we say that the cut \( (S, V \setminus S) \) is \( \beta \)-biased. (Note that a 0-biased cut is a bisection.)

Recall that the normalized Laplacian of \( G \) is a matrix \( L_G \) whose rows and columns correspond to vertices of \( G \) that is defined as follows

\[
L_G(u, v) = \begin{cases} 
1, & \text{if } u = v \text{ and } d_u \neq 0, \\
-1/\sqrt{d_u d_v}, & \text{if } (u, v) \in E, \\
0, & \text{otherwise},
\end{cases}
\]

where \( d_u \) is the degree of the vertex \( u \). Let \( \lambda_2(L_G) \) be the second smallest eigenvalue of \( L_G \). We abuse the notation by letting \( \lambda_2(G) = \lambda_2(L_G) \). We define the volume of a set \( U \subseteq V \) as \( \text{vol}(U) = \text{vol}_G(U) = \sum_{u \in U} d_u \).

We will use the following version of Cheeger’s inequality.

**Theorem 13.2.2 (Cheeger’s inequality for non-regular graphs [67]).** For every graph \( G = (V, E) \),

\[
\lambda_2(G)/2 \leq \phi(G) \leq \sqrt{2\lambda_2(G)},
\]

where \( \phi(G) \) is the expansion of \( G \),

\[
\phi(G) \equiv \min_{S \subseteq V} \frac{\text{edges}(S, V \setminus S)}{\min(\text{vol}(S), \text{vol}(V \setminus S))}.
\]

Moreover, we can efficiently find a set \( A \subseteq V \) such that \( \text{vol}(A) \leq \text{vol}(V)/2 \) and \( |\text{edges}(A, V \setminus A)|/\text{vol}(A) \leq \sqrt{2\lambda_2(G)} \).

For any two disjoint sets \( X, Y \subseteq V \), let \( \text{uncut}(X, Y) = |E(X) + E(Y)|/|E(X \cup Y)| \) be the fraction of edges of \( G[X \cup Y] \) that do not cross the cut \( (X, Y) \). We say that a cut \( (X, Y) \) of \( V \) is perfect if \( \text{uncut}(X, Y) = 0 \).
13.2.3 Our approach

In this section, we give a brief overview of our algorithm. It is instructive to consider first the case when \( G \) has a perfect bisection cut. In this case, \( G \) is a bipartite graph. If \( G \) has only one connected component, each part of this component has the same number of vertices, so this is the desired bisection. Now assume that \( G \) has several connected components. Then each connected component \( C \) of \( G \) is a bipartite graph with two parts \( X_C \) and \( Y_C \). Since all edges are cut in the optimal solution, \( X_C \) must lie on one side of the optimal cut and \( Y_C \) on the other. So in order to find a perfect bisection \((X,Y)\), for every connected component \( C \) we need to either (i) add \( X_C \) to \( X \) and \( Y_C \) to \( Y \) or (ii) add \( X_C \) to \( Y \) and \( Y_C \) to \( X \) so that \(|X| = |Y| = |V|/2\). We can do that using dynamic programming.

Our algorithm for almost satisfiable instances proceeds in a similar way. Assume that the optimal bisection cuts a \((1 - \epsilon)\) fraction of edges.

1. In a preprocessing step, we use the algorithm of Goemans and Williamson \([94]\) to find an approximate maximum cut in \( G \). A fraction \( 1 - O(\sqrt{\epsilon}) \) of edges cross this cut. We remove all uncut edges and obtain a bipartite graph. We denote the parts of this graph by \( A \) and \( B \). (Of course, in general \(|A| \neq |B|\).

2. Then we recursively partition \( G \) into pieces \( W_1, \ldots, W_s \) using Cheeger’s Inequality (see Lemma 13.3.1). Every piece is either a sufficiently small subgraph, which contains at most an \( \epsilon \) fraction of all vertices, or is a spectral expander, with \( \lambda_2 \geq \epsilon^{2/3}. \) There are very few edges between different pieces, so we can ignore them later. In this step, we obtain a collection of induced subgraphs \( G[W_1], \ldots, G[W_s] \) with very few edges going between different subgraphs.

3. Now our goal is to find an “almost perfect” cut in every \( G[W_i] \), then combine these cuts and get a bisection of \( G \). Note that every \( G[W_i] \) is bipartite and therefore has a perfect cut (since \( G \) is bipartite after the preprocessing step). However, we cannot restrict our attention only to this perfect cut since the optimal solution \((S, T)\) can cut \( G[W_i] \) in another proportion. Instead, we prepare a list \( \mathcal{W}_i \) of “candidate cuts” for each \( G[W_i] \) that cut \( W_i \) in different proportions. One of them is close to the cut \((W_i \cap S, W_i \cap T)\) (the restriction of the optimal cut to \( W_i \)).

4. If \( G[W_i] \) is an expander, we find a candidate cut that cuts \( G[W_i] \) in a given proportion by moving vertices from one side of the perfect cut \((W_i \cap A, W_i \cap B)\) to the other, greedily (see Lemma 13.4.1 and Lemma 13.4.2).

5. If \( G[W_i] \) is small, we find a candidate cut that cuts \( G[W_i] \) in a given proportion using semi-definite programming (see Lemma 13.4.3 and Corollary 13.4.4). We solve an
SDP relaxation similar to the Goemans–Williamson relaxation [94] with “$\ell_2^2$-triangle inequalities”, and then find a cut by using hyperplane or threshold rounding.

In fact, the cut that we find can be more unbalanced than $(W_i \cap S, W_i \cap T)$ but this is not a problem since the set $W_i$ is small. Note however that if a cut of another piece $W_j$ is very unbalanced than we might need to find a cut of $W_i$ that is unbalanced in the other direction. So it is important that the candidate cut of $W_i$ is at least as unbalanced as $(W_i \cap S, W_i \cap T)$.

6. Finally, we combine candidate cuts of subgraphs $G[W_i]$ into one balanced cut of the graph $G$, in the optimal way, using dynamic programming (see Lemma 13.5.1).

### 13.2.4 Organization

The rest of the chapter is devoted to the full description and proof of the algorithm. In Section 13.3, we partition the graph into expanders and small pieces, after proper preprocessing. In Section 13.4, we produce a list of candidate cuts for each expander and small piece, by different methods. In Section 13.5, we show how to choose one candidate cut for each part. In Section 13.6, we put everything together to finish the proof of Theorem 13.1.3.

### 13.3 Preprocessing and partitioning graph $G$

In this section, we present the preprocessing and partitioning steps of our algorithms. We will assume that we know the value of the optimal solution $\text{opt} = 1 - \epsilon_{\text{opt}}$ (with a high precision). If we do not, we can run the algorithm for many different values of $\epsilon$ and output the best of the bisection cuts we find.

#### 13.3.1 Preprocessing: Making $G$ bipartite and unweighted

In this section, we show that we can assume that the graph $G$ is bipartite, with parts $A$ and $B$, unweighted, and that $|E| \leq O(|V|/\epsilon_{\text{opt}}^2)$.

First, we “sparsify” the edge-weighted graph $G = (V, E)$, and make the graph unweighted: we sample $O(\epsilon_{\text{opt}}^{-2}|V|)$ edges (according to the weight distribution) with replacement from $G$, then with high probability, every cut has the same cost in the original graph.
as in the new graph, up to an additive error $\epsilon_{\text{opt}}$ (by Chernoff’s bound). So we assume that $|E| \leq O(\epsilon_{\text{opt}}^2|V|)$.

We apply the algorithm of Goemans and Williamson to $G$ and find a partitioning of $G$ into two pieces $A$ and $B$ so that only an $O(\sqrt{\epsilon_{\text{opt}}})$ fraction of edges lies within $A$ or within $B$.

### 13.3.2 Partitioning

In this section, we describe how we partition $G$ into pieces.

**Lemma 13.3.1.** Given a graph $G = (V,E)$, and parameters $\delta \in (0,1)$ and $\lambda \in (0,1)$ such that $|E| = O(|V|/\delta^2)$, we can find a partitioning of $V$ into disjoint sets $U_1, \ldots, U_p$ ("small sets"), and $V_1, \ldots, V_q$ ("expander graphs"):

$$V = \bigcup_i U_i \cup \bigcup_j V_j,$$

in polynomial time, so that

1. $|U_i| \leq \delta|V|$ for each $1 \leq i \leq p$;
2. $\lambda_2(G[V_i]) \geq \lambda$ for each $1 \leq i \leq q$;
3. $\sum_i |E(U_i)| + \sum_j |E(V_j)| \geq (1 - O(\sqrt{\lambda}\log(1/\delta)))|E|.$

**Proof.** We start with a trivial partitioning $\{V\}$ of $V$ and then iteratively refine it. Initially, all sets in the partitioning are “active”; once a set satisfies conditions 1 or 2 of the lemma, we permanently mark it as “passive” and stop subdividing it. We proceed until all sets are passive. Specifically, we mark a set $S$ as passive in two cases. First, if $|S| \leq \delta|V|$ then we add $S$ to the family of sets $U_i$. Second, if $\lambda_2(G[S]) \geq \lambda$ then we add $S$ to the family of sets $V_i$.

We subdivide every active $S$ into smaller pieces by applying the following easy corollary of Cheeger’s inequality (Theorem 13.2.2) to $H = G[S]$.

**Corollary 13.3.2.** Given a graph $H = (S,E(H))$ and a threshold $\lambda > 0$, we can find, in polynomial time, a partition $S_1, S_2, \ldots, S_t$ of $S$ such that

1. $|E(S_i)| \leq |E(S)|/2$ or $\lambda_2(H[S_i]) \geq \lambda$, for each $1 \leq i \leq t$.
2. $\sum_{i<j} |\text{edges}(S_i, S_j)| \leq \sqrt{8\lambda}|E(H)|.$
3. each graph \( H[S_i] \) is connected.

Proof. If \( \lambda_2(H) \geq \lambda \) then we just output a trivial partition \( \{S\} \). Otherwise, we apply Theorem 13.2.2 to \( H_1 = H \), find a set \( S_1 \) s.t. \( \text{vol}_{H_1}(S_1) \leq \text{vol}_{H_1}(S) / 2 \) and \( |\text{edges}(S_1, S \setminus S_1)| / \text{vol}_{H_1}(S_1) \leq \sqrt{2} \lambda_2(H_1) \leq \sqrt{2} \lambda \). Then we remove \( S_1 \) from \( H_1 \), obtain a graph \( H_2 \) and iteratively apply this procedure to \( H_2 \). We stop when either \( \lambda_2(H_i) \geq \lambda \) or \( |E(H_i)| \leq |E(S)| / 2 \).

We verify that the obtained partitioning \( S_1, \ldots, S_t \) of \( S \) satisfies the first condition. For each \( i \in \{1, \ldots, t - 1\} \), we have \( |E(S_i)| \leq \text{vol}_{H_i}(S_i) / 2 \leq \text{vol}_{H_i}(V(H_i)) / 4 = E(H_i) / 2 \leq |E(H)| / 2 \). Our stopping criterion guarantees that \( S_t \) satisfies the first condition. We verify the second condition.

\[
\sum_{i<j} |\text{edges}(S_i, S_j)| = \sum_{i=1}^{t-1} |\text{edges}(S_i, V(H_i) \setminus S_i)| \\
\leq \sum_{i=1}^{t-1} \sqrt{2} \lambda \text{vol}_{H_i}(S_i) \leq \sqrt{2} \lambda \text{vol}_H(S) = 2 \sqrt{2} \lambda |E(H)|.
\]

Finally, if for some \( i \), \( H[S_i] \) is not connected, we replace \( S_i \) in the partitioning with the connected components of \( H[S_i] \).

By the definition, sets \( U_i \) and \( V_j \) satisfy properties 1 and 2. It remains to verify that

\[
\sum_{i=1}^{p} |E(U_i)| + \sum_{j=1}^{q} |E(V_j)| \geq (1 - O(\sqrt{\lambda} \log(1/\delta)))|E|.
\]

We first prove that the number of iterations is \( O(\log(1/\delta)) \). Note that if \( S \) is an active set and \( T \) is its parent then \( |E(S)| \leq |E(T)| / 2 \). Set \( V \) contains \( O(|V|/\delta^2) \) edges. Every active set \( S \) contains at least \( \delta |V| / 2 \) edges, since \( |E(S)| \geq |S| - 1 \geq \delta |V| / 2 \) (we use that \( G[S] \) is connected). Therefore, the number of iterations is \( O(\log_2((|V|/\delta^2) / (\delta |V| / 2))) = O(\log 1/\delta) \).

We finally observe that when we subdivide a set \( S \), we cut \( O(\sqrt{\lambda} |E(S)|) \) edges. At each iteration, since all active sets are disjoint, we cut at most \( O(\sqrt{\lambda} |E|) \) edges. Therefore, the total number of edges cut in all iterations is \( O(\sqrt{\lambda} \log(1/\delta))|E| \).
13.4 Finding cuts in sets $U_i$ and $V_i$

In the previous section, we showed how to partition the graph $G$ into the union of “small graphs” $G[U_i]$ and expander graphs $G[V_i]$. We now show how to find good “candidate cuts” in each of these graphs.

13.4.1 Candidate cuts in $V_i$

In this section, first we prove that there is essentially only one almost perfect maximum cut in an expander graph (Lemma 13.4.1). That implies that every almost perfect cut in the graph $G[V_i]$ should be close to the perfect cut $(V_i \cap A, V_i \cap B)$. Using that we construct a list of good candidate cuts (Lemma 13.4.2). One of these cuts is close to the restriction of the optimal cut to subgraph $G[V_i]$.

**Lemma 13.4.1.** Suppose we are given a graph $H = (V, E)$ and two cuts $(S_1, T_1)$ and $(S_2, T_2)$ of $G$, each of value at least $(1 - \delta)$. Then

$$\min\{\vol_H(S_1 \triangle S_2), \vol_H(S_1 \triangle T_2)\} \leq 4\delta|E|/\lambda_2(H).$$

**Proof.** Let

$$X = S_1 \triangle S_2 = (S_1 \cap T_2) \cup (S_2 \cap T_1);$$

$$Y = S_1 \triangle T_2 = (S_1 \cap T_1) \cup (S_2 \cap T_2).$$

Note that $V = X \cup Y$. There are at most $2\delta|E|$ edges between $X$ and $Y$, since

$$\text{edges}(X, Y) \subset E(S_1) \cup E(S_2) \cup E(T_1) \cup (T_2),$$

$$|E(S_1) \cup E(T_1)| \leq \delta|E| \text{ and } |E(S_2) \cup E(T_2)| \leq \delta|E|.$$

On the other hand, by Cheeger’s inequality (Theorem 13.2.2), we have

$$\frac{|\text{edges}(X, Y)|}{\min(\vol_H(X), \vol_H(Y))} \geq \lambda_2(H)/2.$$

Therefore,

$$\min(\vol_H(X), \vol_H(Y)) \leq 2|\text{edges}(X, Y)|/\lambda_2(H) \leq \frac{4\delta|E|}{\lambda_2(H)}.$$

\qed
Consider one of the sets $V_i$. Let $H = G[V_i]$. Denote $A_i = V_i \cap A$ and $B_i = V_i \cap B$. We sort all vertices in $A_i$ and $B_i$ w.r.t. their degrees in $H$. Now we are ready to define the family of candidates cuts $(X_0, Y_0), \ldots, (X_{|V_i|}, Y_{|V_i|})$ for $G[V_i]$. For each $j$, we define $(X_j, Y_j)$ as follows.

- If $j \leq |A_i|$ then $X_j$ consists of $j$ vertices of $A_i$ with highest degrees, and $Y_j$ consists of the remaining vertices of $H$ (i.e. $Y_j$ contains all vertices of $B_i$ as well as $|A_i| - j$ lowest degree vertices of $A_i$).
- If $j \geq |A_i|$ then $Y_j$ consists of $|V_i| - j$ vertices of $B_i$ with highest degrees, and $X_j$ consists of the remaining vertices of $H$.

Clearly, $|X_j| = j$ and $|Y_j| = |V_i| - j$. Let $(S, T)$ be the restriction of the optimal bisection of $G$ to $H$. We will show that one of the cuts $(X_j, Y_j)$ is not much worse than $(S, T)$. By Lemma 13.4.1 applied to cuts $(A_i, B_i)$ and $(S, T)$ (note that $\text{uncut}(A_i, B_i) = 0$),

$$\min\{\text{vol}_H(A_i \Delta S), \text{vol}_H(A_i \Delta T)\} \leq \frac{4 \cdot \text{uncut}(S, T)|E(H)|}{\lambda_2(H)}.$$ 

Assume without loss of generality that $\text{vol}_H(A_i \Delta S) \leq 4E(H)/\lambda_2(H)$ (otherwise, rename sets $X$ and $Y$). We show that $\text{vol}_H(A_i \Delta X_{|S|}) \leq \text{vol}_H(A_i \Delta S)$. Consider the case $|A_i| \geq |S|$. Note that by the definition of $X_{|S|}$, the set $X_{|S|}$ has the largest volume among all subsets of $A_i$ of size at most $|S|$. Correspondingly, $A_i \setminus X_{|S|}$ has the smallest volume among all subsets of $A_i$ of size at least $|A_i| - |S|$. Finally, note that $|A_i \setminus S| \geq |A_i| - |S|$. Therefore,

$$\text{vol}_H(A_i \Delta X_{|S|}) = \text{vol}_H(A_i \setminus X_{|S|}) \leq \text{vol}_H(A_i \setminus S) \leq \text{vol}_H(A_i \Delta S).$$

The case when $|A_i| \leq |S|$ is similar. We conclude that

$$\text{vol}_H(A_i \Delta X_{|S|}) \leq 4 \cdot \text{uncut}(S, T)|E_H|/\lambda_2(H).$$

Therefore, the size of the cut $(X_{|S|}, Y_{|S|})$ is at least

$$|E(H)| - \text{vol}_H(A_i \Delta X_{|S|}) \geq \left(1 - \frac{4 \cdot \text{uncut}(S, T)}{\lambda_2(H)}\right)|E(H)|.$$ 

We have thus proved the following lemma.

**Lemma 13.4.2.** There is a polynomial time algorithm that given a graph $H = G([V_i])$ finds a family of cuts $\mathcal{V}_i = \{(X_1, Y_1), \ldots, (X_{|V_i|}, Y_{|V_i|})\}$ such that for every cut $(S, T)$ of $H$ there exists a cut $(X, Y) \in \mathcal{V}_i$ with $|X| = \min(|S|, |T|)$ and

$$\text{uncut}(X, Y) \leq \frac{4 \cdot \text{uncut}(S, T)}{\lambda_2(H)}.$$ 

228
13.4.2 Candidate cuts in $U_i$

In this section, we show how to find candidate cuts for the small parts, i.e., the induced subgraphs $G[U_i]$.

**Lemma 13.4.3.** Suppose we are given a graph $H = (U, E)$ and two parameters $0 \leq \theta \leq 1/2$ and $0 < \Delta < 1$. Then in polynomial time we can find a cut $(X, Y)$ such that for every cut $(S, T)$ in $H$, with $|S| \leq \theta |U|$, we have

1. $\text{uncut}(X, Y) \leq O(\sqrt{\text{uncut}(S, T)} + \text{uncut}(S, T)/\Delta)$.
2. $|X| \leq (\theta + \Delta)|U|$.

**Proof.** Let $(S, T)$ be the maximum cut among all cuts with $|S| \leq t|U|$ (of course, our algorithm does not know $(S, T)$). Let $\epsilon_H = \text{uncut}(S, T)$. We may assume that our algorithm knows the value of $\epsilon_H$ (with high precision) — as otherwise, we can run our algorithm on different values of $\epsilon$ and output the best of the cuts the algorithm finds.

We write the following SDP program. For every vertex $i \in U$, we introduce a unit vector $v_i$. Additionally, we introduce a special unit vector $v_0$.

Maximize $\frac{1}{|U|} \sum_{i \in U} \langle v_0, v_i \rangle$

Subject to $\frac{1}{4|E|} \sum_{(i,j) \in E} \|v_i + v_j\|^2 \leq \epsilon_H$

$\|v_i\|^2 = 1 \quad \forall i \in V \cup \{0\}$

$|\langle v_i + v_j, v_0 \rangle| \leq \frac{\|v_i + v_j\|^2}{2} \quad \forall i, j \in V.$

The “intended solution” to this SDP is $v_i = v_0$ if $i \in T$ and $v_i = -v_0$ if $i \in S$ (vector $v_0$ is an arbitrary unit vector). Clearly, this solution satisfies all SDP constraints. In particular, it satisfies the last constraint (“an $\ell_2^2$-triangle inequality”) since the left hand side is positive only when $v_i = v_j$, then $|\langle v_i + v_j, v_0 \rangle| = \frac{\|v_i + v_j\|^2}{2} = 2$. The value of this solution is $(|T| - |S|)/|U| \geq 1 - 2\theta$.

We solve the SDP and find the optimal SDP solution $\{v_i\}$. Note that $\sum_{i \in U} \langle v_0, v_i \rangle \geq (1 - 2\theta)|U|$.

Let $\Delta' = 2\Delta/3$. Choose $r \in [\Delta', 2\Delta']$ uniformly at random. Define a partition of $U$ into sets $Z_k$, $0 \leq k < 1/\Delta'$, as follows: let $Z_k = \{i : k\Delta' + r < |\langle v_0, v_i \rangle| \leq (k + 1)\Delta' + r\}$.
for \( k \geq 1 \) and \( Z_0 = \{ i : -\Delta' - r \leq \langle v_0, v_i \rangle \leq \Delta' + r \} \). We bound the probability that the endpoints of an edge \((i, j)\) belong to different sets \( Z_k \). Note that if no point from the set \( \{ \pm (k \Delta' + r) : k \geq 1 \} \) lies between \( |\langle v_i, v_0 \rangle| \) and \( |\langle v_j, v_0 \rangle| \) then \( i \) and \( j \) belong to the same set \( Z_k \). The distance between \( |\langle v_i, v_0 \rangle| \) and \( |\langle v_j, v_0 \rangle| \) is at most \( |\langle v_i + v_j, v_0 \rangle| \). Therefore, the probability (over \( r \)) that \( i \) and \( j \) belong to different sets \( Z_k \) is at most \( |\langle v_i + v_j, v_0 \rangle| / \Delta' \).

So the expected number of cut edges is at most

\[
\frac{1}{\Delta'} \sum_{(i,j) \in E} |\langle v_i + v_j, v_0 \rangle| \leq \frac{1}{2\Delta'} \sum_{(i,j) \in E} \|v_i + v_j\|^2 \leq \frac{2|E|\epsilon_H}{\Delta'}. \tag{13.1}
\]

For each \( k \geq 1 \), let \( Z_k^+ = \{ i \in Z_k : \langle v_i, v_0 \rangle > 0 \} \) and \( Z_k^- = \{ i \in Z_k : \langle v_i, v_0 \rangle < 0 \} \). We use hyperplane rounding of Goemans and Williamson \cite{gwo95} to divide \( Z_0 \) into two sets \( Z_0^+ \) and \( Z_0^- \). We are ready to define sets \( X \) and \( Y \). For each \( k \), we add vertices from the smaller of the two sets \( Z_k^+ \) and \( Z_k^- \) to \( X \), and vertices from the larger of them to \( Y \).

Now we bound \( \text{uncut}(X, Y) \). Note that

\[
|\text{uncut}(X, Y)| \leq \sum_{k \leq l} |\text{edges}(Z_k, Z_l)| + \sum_{k \geq 0} (|E(Z_k^+)\rangle + |E(Z_k^-)\rangle).
\]

We have already shown that \( \sum_{k \leq l} |\text{edges}(Z_k, Z_l)| \) is less than \( 2\epsilon_H|E| / \Delta' \) in expectation. If \((i, j) \in E(Z_k^+)\) or \((i, j) \in E(Z_k^-)\) for \( k \geq 1 \) then \( |\langle v_i + v_j, v_0 \rangle| \geq \Delta' \). Therefore,

\[
\sum_{k \geq 1} (|E(Z_k^+)\rangle + |E(Z_k^-)\rangle) \leq \frac{2\epsilon_H|E|}{\Delta'} = \frac{3\epsilon_H|E|}{\Delta}.
\]

Finally, note that when we divide \( Z_0 \), the fraction of edges of \( E(Z_0) \) that do not cross the random hyperplane is \( O(\sqrt{\epsilon_0}) \) (in expectation) where

\[
\epsilon_0 = \frac{1}{4|E(Z_0)|} \sum_{(i,j) \in E(Z_0)} \|v_i + v_j\|^2 \leq \frac{1}{4|E(Z_0)|} \sum_{(i,j) \in E} \|v_i + v_j\|^2 \leq \frac{\epsilon_H \cdot |E|}{|E(Z_0)|}.
\]

Thus,

\[
E[|E(Z_0^+)\rangle + |E(Z_0^-)\rangle| r] \leq O \left( \sqrt{\epsilon_H|E| / |E(Z_0)|} \right) |E(Z_0)| \leq O(\sqrt{\epsilon_H}|E|).
\]

Combining the above upper bounds, we conclude that

\[
E[\text{uncut}(X, Y)] \leq O \left( \frac{\epsilon_H}{\Delta} + \sqrt{\epsilon_H} \right).
\]

230
Finally, we estimate the size of the set $X$. Note that if $v_i \in \mathbb{Z}_k^+$ then $\langle v_i, v_0 \rangle - k\Delta' \leq 3\Delta'$, if $v_i \in \mathbb{Z}_k^-$ then $\langle v_i, v_0 \rangle + k\Delta' \leq 3\Delta'$. Therefore, $\sum_{i \in \mathbb{Z}_k} \langle v_i, v_0 \rangle \leq k(|\mathbb{Z}_k^+| - |\mathbb{Z}_k^-|)\Delta' + 3\Delta'|\mathbb{Z}_k|$, which implies

$$\sum_k k(|\mathbb{Z}_k^+| - |\mathbb{Z}_k^-|)\Delta' \geq \sum_{i \in U} \langle v_i, v_0 \rangle - 3\Delta'|U| \geq (1 - 2\theta - 3\Delta')|U|.$$ 

Therefore,

$$|Y| - |X| = \sum_k |Z_k^+| - |Z_k^-| \geq \sum_{k:|Z_k^+| - |Z_k^-| > 0} (|Z_k^+| - |Z_k^-|)$$

$$\geq \sum_k k(|Z_k^+| - |Z_k^-|)\Delta' \geq (1 - 2\theta - 3\Delta')|U|,$$

implying $|X| \leq (\theta + 3\Delta'/2)|U| = (\theta + \Delta)|U|$. \hfill \qed

We apply this algorithm to every graph $G[U_i]$ and every $\theta = k/|U_i|$, $0 < k \leq |U_i|/2$, and obtain a list of candidate cuts. We get the following corollary.

**Corollary 13.4.4.** There is a polynomial time algorithm that given a graph $H = G([U_i])$ and a parameter $\Delta \in (0, 1)$ finds a family of cuts $\mathcal{U}_i$ such that for every cut $(S, T)$ of $H$ there exists a cut $(X, Y) \in \mathcal{U}_i$ with $|X| \leq \min(|S|, |T|) + \Delta|U_i|$ and

$$\text{uncut}(X, Y) \leq O\left(\sqrt{\text{uncut}(S, T)} + \frac{\text{uncut}(S, T)}{\Delta}\right).$$

### 13.5 Combining candidate cuts

In this section, we show how to choose one candidate cut for each set $U_i$ and $V_j$.

For brevity, we denote $W_i = U_i$ for $i \in \{1, \ldots, p\}$ and $W_{p+j} = V_j$ for $j \in \{1, \ldots, q\}$. Similarly, $W_i = U_i$ for $i \in \{1, \ldots, p\}$ and $W_{p+j} = V_j$ for $j \in \{1, \ldots, q\}$ Then $W_1, \ldots, W_{p+q}$ is a partitioning of $V$, and $\mathcal{W}_i$ is a family of cuts of $G[W_i]$.

We say that a cut $(X, Y)$ of $G$ is a combination of candidate cuts from $\mathcal{W}_i$ if the restriction of $(X, Y)$ to each $W_i$ belongs to $\mathcal{W}_i$ (we identify cuts $(S, T)$ and $(T, S)$).
Lemma 13.5.1. There exists a polynomial time algorithm that given a graph $G = (V, E)$ and a threshold $\zeta \in [0, 1/2]$, sets $W_i$ and families of cuts $\mathcal{W}_i$, finds the maximum cut among all combination cuts $(X, Y)$ with $|X|, |Y| \in [(1/2 - \zeta)|V|, (1/2 + \zeta)|V|]$.

Proof. We solve the problem by dynamic programming. Denote $H_k = G[\bigcup_{i=1}^k W_i]$. For every $a \in \{1, \ldots, p+q\}$ and $b \in \{1, \ldots, |G[H_a]|\}$, let $Q[a, b]$ be the size of the maximum cut among all combination cuts $(X, Y)$ on $H_a$ with $|X| = b$ ($Q[a, b]$ equals $-\infty$ if there are no such cuts). We loop over all value of $a$ from 1 to $p+q$ and fill out the table $Q$ using the following formula

$$Q[a, b] = \max_{(X,Y) \in W_a \text{ or } (Y,X) \in W_a} (Q[a-1, b-|X|] + |\text{edges}(X,Y)|),$$

where we assume that $Q[0, 0] = 0$, and $Q[a, b] = -\infty$ if $a \leq 0$ and $b \leq 0$ and $(a, b) \neq (0, 0)$.

Finally the algorithm outputs maximum among $T[p + q, \lceil (1/2 - \zeta)|V| \rceil], \ldots, T[p + q, \lfloor (1/2 + \zeta)|V| \rfloor]$, and the corresponding combination cut.

Finally, we prove that there exists a good almost balanced combination cut.

Lemma 13.5.2. Let $G = (V, E)$ be a graph. Let $V = \bigcup_i U_i \cup \bigcup_j V_j$ be a partitioning of $V$ that satisfies conditions of Lemma 13.3.1, and $U_i$ and $V_j$ be families of candidate cuts that satisfy conditions of Corollary 13.4.4 and Lemma 13.4.2, respectively. Then there exists a composition cut $(X, Y)$ such that

$$\left| \frac{|X|}{|V|} - \frac{1}{2} \right| \leq \max(\Delta, \delta)$$

and

$$\text{uncut}(X, Y) \leq O\left( \sqrt{\lambda \log(1/\delta)} + \sqrt{\text{uncut}(S_{\text{opt}}, T_{\text{opt}})} + \text{uncut}(S_{\text{opt}}, T_{\text{opt}}) \left( \frac{1}{\lambda} + \frac{1}{\Delta} \right) \right),$$

where $(S_{\text{opt}}, T_{\text{opt}})$ is the optimal bisection of $G$.

Proof. Consider the optimal bisection cut $(S_{\text{opt}}, T_{\text{opt}})$. We choose a candidate cut for every set $V_i$. By Lemma 13.4.2 for every $V_i$ there exists a cut $(X_i, Y_i) \in \mathcal{V}_i$ such that

$$\text{uncut}(X_i, Y_i) \leq 4\text{uncut}(S_{\text{opt}} \cap V_i, T_{\text{opt}} \cap V_i) / \lambda_2(G[V_i]) \leq 4\text{uncut}(S_{\text{opt}} \cap V_i, T_{\text{opt}} \cap V_i) / \lambda,$$

(13.2)
and \(|X_i| = \min(|S_{opt} \cap V_i|, |T_{opt} \cap V_i|)|. We define sets \(X^V\) and \(Y^V\) as follows. For each \(i\), we add \(X_i\) to \(X^V\) if \(|X_i| = |S_{opt} \cap V_i|\), and we add \(Y_i\) to \(Y^V\), otherwise (i.e. if \(|Y_i| = |S_{opt} \cap V_i|\)). Similarly, we add \(Y_i\) to \(Y^V\) if \(|Y_i| = |T_{opt} \cap V_i|\), and we add \(X_i\) to \(Y^V\), otherwise. Clearly, \((X^V, Y^V)\) is a candidate cut of \(\bigcup_i V_i\) and \(|X^V| = |S_{opt} \cap \bigcup_i V_i|\).

Assume without loss of generality that \(|X^V| \geq |Y^V|\).

Now we choose a candidate cut for every set \(U_i\). By Corollary 13.4.4 for every \(U_i\) there exists a cut \((X_i', Y_i') \in U_i\) such that

\[
\text{uncut}(X_i', Y_i') \leq O\left(\sqrt{\text{uncut}(S_{opt} \cap U_i, T_{opt} \cap U_i)} + \frac{\text{uncut}(S_{opt} \cap U_i, T_{opt} \cap U_i)}{\Delta}\right),
\]

(13.3)

and \(|X_i'| \leq \min(|S_{opt} \cap U_i|, |T \cap U_i|) + \Delta|U_i|\). We assume that \(X_i'\) is the smaller of the two sets \(X_i'\) and \(Y_i'\).

We want to add one of the sets \(X_i'\) and \(Y_i'\) to \(X^V\), and the other set to \(Y^V\) so that the resulting cut \((X, Y)\) is almost balanced. We set \(X = X^V\) and \(Y = Y^V\). Then consequently for every \(i\) from 1 to \(p\), we add \(X_i'\) to the larger of the sets \(X\) and \(Y\), and add \(Y_i'\) to the smaller of the two sets (recall that \(X_i'\) is smaller than \(Y_i'\)). We obtain a candidate cut \((X, Y)\) of \(G\).

We show that \(||X|/|V| - 1/2| \leq \max(\Delta, \delta)\). Initially, \(|X| = |X^V| \geq |Y| = |Y^V|\).

If at some point \(X\) becomes smaller than \(Y\) then after that \(||X| - |Y|| \leq \delta|V|\) since at every step \(||X| - |Y||\) does not change by more than \(|U_i| \leq \delta|V|\). So in this case \(||X|/|V| - 1/2| \leq \delta\). So let us assume that the set \(X\) always remains larger than \(Y\). Then we always add \(X_i'\) to \(X\) and \(Y_i'\) to \(Y\). We have

\[
|X| = |X^V \cup \bigcup_i X_i'|
\]

\[
\leq \sum_{j=1}^q |S_{opt} \cap V_j| + \sum_{i=1}^p \left(\min(|S_{opt} \cap U_i|, |T_{opt} \cap U_i|) + \Delta|U_i|\right)
\]

\[
\leq \sum_{j=1}^q |S_{opt} \cap V_j| + \sum_{i=1}^p |S_{opt} \cap U_i| + \Delta|V|
\]

\[
= |S_{opt}| + \Delta|V| = (1/2 + \Delta)|V|.
\]

It remains to bound \(\text{uncut}(X, Y)\). We have,

\[
\text{uncut}(X, Y)|E| \leq \sum_{1 \leq i < j \leq p} |\text{edges}(U_i, U_j)| + \sum_{1 \leq i < j \leq q} |\text{edges}(V_i, V_j)| + \sum_{1 \leq i \leq p} |\text{edges}(U_i, V_j)|
\]

233
\[ + \sum_{1 \leq i \leq p} \text{uncut}(X'_i, Y'_i)|E(U_i)| + \sum_{1 \leq j \leq q} \text{uncut}(X_j, Y_j)|E(V_j)|. \]

By Lemma 13.3.1, the sum of the first three terms is at most \(O(\sqrt{\lambda} \log(1/\delta))|E|\). From (13.3), we get
\[ \sum_{1 \leq i \leq p} \text{uncut}(X'_i, Y'_i)|E(U_i)| \leq O(1) \sum_{i=1}^{p} \left( \sqrt{\text{uncut}(S_{\text{opt}} \cap U_i, T_{\text{opt}} \cap U_i)} + \frac{\text{uncut}(S_{\text{opt}} \cap U_i, T_{\text{opt}} \cap U_i)}{\Delta} \right)|E(U_i)| \]
\[ = O(1) \sum_{i=1}^{p} \sqrt{|E(S_{\text{opt}} \cap U_i)| + |E(T_{\text{opt}} \cap U_i)|} \cdot \sqrt{|E(U_i)|} \]
\[ + O(1) \sum_{i=1}^{p} \frac{|E(S_{\text{opt}} \cap U_i)| + |E(T_{\text{opt}} \cap U_i)|}{\Delta} \]
\[ \leq O(1) \sqrt{\sum_{i=1}^{p} (|E(S_{\text{opt}} \cap U_i)| + |E(T_{\text{opt}} \cap U_i)|) \cdot \sqrt{\sum_{i=1}^{p} |E(U_i)|} + O \left( \frac{\text{uncut}(S_{\text{opt}}, T_{\text{opt}}) \cdot |E|}{\Delta} \right)} \]
\[ \leq O\left( \sqrt{\text{uncut}(S_{\text{opt}}, T_{\text{opt}}) + \frac{\text{uncut}(S_{\text{opt}}, T_{\text{opt}})}{\Delta}} \cdot |E|. \right) \]

From (13.2), we get
\[ \sum_{j} \text{uncut}(X_j, Y_j)|E(V_j)| \]
\[ \leq \sum_{j} \frac{4 \cdot \text{uncut}(S_{\text{opt}} \cap V_j, T_{\text{opt}} \cap V_j)|E(V_j)|}{\lambda} \leq \frac{4 \cdot \text{uncut}(S_{\text{opt}}, T_{\text{opt}})}{\lambda} \cdot |E|. \]

**13.6 The bisection algorithm – proof of Theorem 13.1.3**

First, we run the preprocessing step described in Section 13.3.1. Then we use the algorithm from Lemma 13.3.1 with \(\lambda = \epsilon_{\text{opt}}^{2/3}\) and \(\delta = \epsilon_{\text{opt}}\) to find a partition of \(V\) into sets \(U_1, \ldots, U_p, V_1, \ldots, V_q\). We apply Corollary 13.4.4 with \(\Delta = \sqrt{\epsilon_{\text{opt}}}\) to all sets \(U_i\), and obtain a list \(\mathcal{U}_i\) of candidate cuts for each set \(U_i\). Then we apply Lemma 13.4.2 and obtain
a list \( V_j \) of candidate cuts for each set \( V_j \). Finally, we find the optimal combination of candidate cuts using Lemma 13.5.1. Denote it by \((X, Y)\). By Lemma 13.5.2 we get that uncut \((X, Y)\) is at most

\[
O\left(\sqrt{\lambda} \log(1/\delta) + \sqrt{\epsilon_{\text{opt}}} + \frac{\epsilon_{\text{opt}}}{\lambda} + \frac{\epsilon_{\text{opt}}}{\Delta}\right) \leq O\left(\sqrt{\epsilon_{\text{opt}}} \log(1/\epsilon_{\text{opt}})\right),
\]

and

\[
\left|\frac{|X|}{|V|} - \frac{1}{2}\right| \leq \max(\Delta, \delta) = O(\sqrt{\epsilon_{\text{opt}}}).
\]

By moving at most \( O(\sqrt{\epsilon_{\text{opt}}}|V|) \) vertices of the smallest degree from the larger size of the cut to smaller part of the cut, we obtain a balanced cut. By doing so, we increase the number of uncut edges by at most \( O(\sqrt{\epsilon_{\text{opt}}}|E|) \). The obtained bisection cuts a \( 1 - O(\sqrt{\epsilon_{\text{opt}} \log(1/\epsilon_{\text{opt}}))} \) fraction of all edges.

It is easy to see that a slight modification of the algorithm leads to the following extension of Theorem 13.1.3.

**Theorem 13.6.1.** There is a randomized polynomial time algorithm that given an edge-weighted graph \( G \) with a \( \beta \)-biased cut of value \( (1 - \epsilon) \) finds a \( \beta \)-biased cut of value \( (1 - O(\sqrt{\epsilon} \log(1/\epsilon)) + \sqrt{\epsilon}/(1 - \beta)) \).

**Proof.** We use the algorithm above, while changing DP algorithm used by Lemma 13.5.1 to find the best combination with bias \( \beta \pm t \) (where \( t = O(\sqrt{\epsilon}) \)). We modify the proof of Lemma 13.5.2 to show that there exists a \( \beta \pm t \) cut of value \( 1 - O(\sqrt{\epsilon}) \). As previously, we first find sets \( X = X^V \) and \( Y = Y^V \) with \( |X^V| \geq |Y^V| \). Now, however, if \( |X| - |Y| > \beta|V| \) then we add \( X_i^t \) to \( X \) and \( Y_i^t \) to \( Y \); otherwise, we add \( X_i^t \) to \( X \) and \( Y_i^t \) to \( Y \). We argue again that if at some point the difference \( |X| - |Y| \) becomes less than \( O(\sqrt{\epsilon}|V|) \), then after that \( |X| - |Y| = O(\sqrt{\epsilon}|V|) \), and therefore, we find a cut with bias \( \beta + O(\sqrt{\epsilon}) \). Otherwise, there are two possible cases: either we always have \( |X| - |Y| > \beta|V| \), and then we always add \( X_i^t \) to \( X \) and \( Y_i^t \) to \( Y \), or we always have \( |X| - |Y| > \beta|V| \), and then we always add \( X_i^t \) to \( Y \) and \( Y_i^t \) to \( X \). Note, however, that in both cases \( |X| - |Y| - \beta|V| \) decreases by \( |Y_i^t| - |X_i^t| \geq |S_{\text{opt}} \cap U_i| - |T_{\text{opt}} \cap U_i| - 2\Delta|U_i| \) after each iteration. Thus after all iterations, the value of \( |X| - |Y| - \beta|V| \) decreases by at least

\[
\sum_{i=1}^{p} |S_{\text{opt}} \cap U_i| - |T_{\text{opt}} \cap U_i| - 2\Delta|U_i| \geq |S_{\text{opt}} \cap \bigcup_{i} U_i| - |T_{\text{opt}} \cap \bigcup_{i} U_i| - 2\Delta|\bigcup_{i} U_i|.
\]

Taking into the account that \( |X^V| - |Y^V| = |S_{\text{opt}} \cap \bigcup_{i} V_i| - |T_{\text{opt}} \cap \bigcup_{i} V_i| \), we get the following bound for the bias of the final combination cut \((X, Y)\),
\[ |X| - |Y| - \beta|V| \leq |S_{\text{opt}} \cap \bigcup_i V_i| - |T_{\text{opt}} \cap \bigcup_i V_i| - |S_{\text{opt}} \cap \bigcup_i U_i| + |T_{\text{opt}} \cap \bigcup_i U_i| + 2\Delta|\bigcup_i U_i| \leq |S_{\text{opt}}| - |T_{\text{opt}}| - \beta|V| + 2\Delta|V| = 2\Delta|V|.\]

We get the exact \(\beta\)-biased cut by moving at most \(O(\sqrt{\epsilon})|V|\) vertices of the smallest degree from the larger size of the cut to smaller part of the cut. By doing so, we lose at most \(O(\sqrt{\epsilon})|E|/(1 - \beta)\) cut edges. Therefore the theorem follows. \(\blacksquare\)
Chapter 14

A robust isomorphism algorithm for trees

14.1 Introduction

The graph isomorphism problem (GI) is arguably one of the most famous computational problems on graphs: given two graphs $G$ and $H$, we have to decide whether they are isomorphic, i.e. there exists a bijection $\pi : V(G) \rightarrow V(H)$ such that there is an edge $(u, v) \in E(G)$ if and only if there is an edge $(\pi(u), \pi(v)) \in E(H)$. Here we use $V(G)$ to denote the set of vertices in $G$, and use $E(G)$ to denote the set of edges in $G$. GI is one of the rare, intriguing problems in NP that is neither known to be polynomial-time tractable nor NP-Complete. Resolving the complexity of GI is one of major problems in graph theory and it is still open despite many decades of effort.

Graph isomorphism algorithms are also very useful in practice to test for isomorphism between any structures that can be encoded as graphs. A few examples of its applications include: image analysis [82], isomorphisms of molecule (for chemistry) [84, 85], and data mining [197]. In many settings, even when the two graphs are not completely isomorphic, we are still interested in measuring how similar the two graphs are. The following optimization problem which captures the similarity between the two given graphs was studied in literature (e.g. [17, 25]).

**Definition 14.1.1 (MAXGISO, rephrase of Definition 2.1.6).** For two graphs $G$ and $H$ with the same number of vertices and a bijection $\pi : V(G) \rightarrow V(H)$, let $\text{val}(G, H, \pi) = |\{(u, v) \in E(G) \mid (\pi(u), \pi(v)) \in E(H)\}|$. Let $\text{opt}(G, H) = \max_{\pi} \text{val}(G, H, \pi)$. In the maximum graph isomorphism problem (MAXGISO), the objective is to compute $\text{opt}(G, H)$
for given two graphs $G$ and $H$.

Observe that GI$SO$ is a special case of the above optimization variant since $\text{opt}(G, H) = \max\{|E(G)|, |E(H)|\}$ if and only if $G$ and $H$ are isomorphic, and any $\pi$ such that $\text{val}(G, H, \pi) = \max\{|E(G)|, |E(H)|\}$ is an isomorphism.

In most applications of GI$SO$, we are interested in graphs which are almost isomorphic to each other. We say that $G$ and $H$ are $(1 - \epsilon)$-isomorphic when $\text{val}(G, H, \pi) = (1 - \epsilon) \max\{|E(G)|, |E(H)|\}$. In fact, even graphs which are isomorphic to each other may get perturbed slightly in practice, due to some small noise. This motivates us to introduce the following definition of robust graph isomorphism algorithm, which certifies when two graphs are $(1 - \epsilon)$-isomorphic when $\epsilon$ is very small.

**Definition 14.1.2 (RobustGI$SO$, rephrase of Definition 2.2.7).** Given two graphs $G$ and $H$ on $n$ vertices, we say $A$ is a robust graph isomorphism algorithm for $G$ and $H$ if there exists a function $f : [0, 1] \rightarrow [0, 1]$ satisfying $\lim_{\epsilon \to 0^+} f(\epsilon) = 0$, such that $A$ outputs a bijection $\pi : V(G) \rightarrow V(H)$ with $\text{val}(G, H, \pi) \geq (1 - f(\epsilon)) \max\{|E(G)|, |E(H)|\}$ whenever $\text{opt}(G, H) = (1 - \epsilon) \max\{|E(G)|, |E(H)|\}$, for any $\epsilon \geq 0$.

A robust graph isomorphism algorithm is a graph isomorphism algorithm (when $\epsilon$ is so small that $f(\epsilon) < \frac{1}{\max\{|E(G)|, |E(H)|\}}$). Therefore, we currently do not expect an efficient robust graph isomorphism algorithm given that a polynomial-time algorithm for GI$SO$ is not yet known. However, GI$SO$ is known to be polynomial-time tractable for many special cases such as trees [134], planar graphs [122], graphs of bounded-degree [167], and graphs of bounded tree-width [46]. This leads to the following natural question:

**Question 14.1.3.** Can polynomial-time algorithms for graph isomorphism (on restricted families of instances) be made robust?

### 14.1.1 Our contributionss and overview of the proofs

In this chapter, we present a robust isomorphism algorithm for trees. The well known canonicalization approach for trees seems quite sensitive to the $\epsilon$-fraction of “noisy edges”. Our algorithm is inspired by a property testing algorithm by Newman and Sohler [178] which implies a PTAS for MAXGI$SO$ on bounded-degree trees. In Section 14.2, we first show the following much weaker statement compared to Newman and Sohler’s theorem, but with a simpler proof.1

1The simple proof can be modified to give robust isomorphism algorithm for bounded-degree planar graphs and graphs with bounded-width as well.
Theorem 14.1.4. Given two $n$-vertex $B$-degree bounded forests $G$ and $H$, suppose that $\text{opt}(G, H) \geq (1 - \epsilon)n$. There is a polynomial-time algorithm to find a bijection $\sigma : V(G) \rightarrow V(H)$ such that $\text{val}(G, H, \sigma) \geq (1 - 20\sqrt{B}\epsilon)n$.

Then we reduce the general trees to bounded-degree trees, and prove that

Theorem 14.1.5. Given two $n$-vertex trees $G$ and $H$ such that $\text{opt}(G, H) \geq (1 - \epsilon)n$, there is a polynomial-time algorithm to find a bijection $\sigma : V(G) \rightarrow V(H)$ where $\text{val}(G, H, \sigma) \geq (1 - 200\epsilon^{1/4})n$.

To prove Theorem 14.1.5 by removing a tiny fraction of edges, we decompose the two input trees into two collections of trees where each tree has at most one high-degree vertex. We match the high-degree vertices using the maximum weight bipartite graph matching algorithm with carefully designed weights. Then we use the algorithm in Theorem 14.1.4 to match up the low-degree vertices.

14.2 The algorithm

In this section, we prove Theorem 14.1.5 which is restated as follows.

Theorem 14.1.5 (restated). Given two $n$-vertex trees $G$ and $H$ such that $\text{opt}(G, H) \geq (1 - \epsilon)n$, there is a polynomial-time algorithm to find a bijection $\sigma : V(G) \rightarrow V(H)$ where $\text{val}(G, H, \sigma) \geq (1 - 200\epsilon^{1/4})n$.

We prove Theorem 14.1.5 by first proving Theorem 14.1.4 which says that when $G$ and $H$ are bounded-degree forests, there is a robust isomorphism algorithm for $G$ and $H$. Then, we reduce the general trees to bounded-degree trees.

Theorem 14.1.4 (restated). Given two $n$-vertex $B$-degree bounded forests $G$ and $H$, suppose that $\text{opt}(G, H) \geq (1 - \epsilon)n$. There is a polynomial-time algorithm to find a bijection $\sigma : V(G) \rightarrow V(H)$ such that $\text{val}(G, H, \sigma) \geq (1 - 20\sqrt{B}\epsilon)n$.

Proof sketch. Let $k = \lceil 1/\sqrt{B}\epsilon \rceil$. There is a simple way to remove at most $n/k$ edges from $G$ to obtain a forest $G'$, such that each tree in $G'$ has at most $kB$ vertices. We do the same decomposition for $H$ to obtain a forest $H'$. Since we removed at most $n/k$ edges from both $G$ and $H$, we have

$$\text{opt}(G', H') \geq \text{opt}(G, H) - 2n/k \geq (1 - \epsilon - 2/k)n.$$
Now the algorithm lets \( G'' = G', H'' = H' \). The algorithm chooses a tree \( T_G \) from \( G'' \) and a tree \( T_H \) from \( H'' \) so that \( T_G \) is isomorphic to \( T_H \), lets \( \sigma \) map the vertices in \( T_G \) to the vertices in \( T_H \) according to the isomorphism, and removes \( T_G \) from \( G'' \), \( T_H \) from \( H'' \). The algorithm iterates this process until no such pair of trees can be found in \( G'' \) and \( H'' \). Finally the algorithm extends \( \sigma \) to a bijection between vertex sets of \( G \) and \( H \).

It is easy to see that, when the algorithm terminates, the total number of trees in \( G'' \) is at most \( 4(|E(G')| + |E(H')| - 2 \operatorname{opt}(G', H')) \leq 8en \) (because \(|E(G')| \leq (1 - 2/k)n\) and \(|E(H')| \leq (1 - 2/k)n\)). Since each tree has at most \( kB \) edges, \( \sigma \) loses at most \( kB \cdot 8en \) edges. Therefore,

\[
\operatorname{val}(G, H, \sigma) \geq \operatorname{val}(G', H', \sigma) \geq \operatorname{opt}(G', H') - 8kBn \geq (1 - 2/k - 9kB\epsilon)n \geq (1 - 20\sqrt{B\epsilon})n.
\]

Now we introduce the following definition.

**Definition 14.2.1.** We call a tree \( T \) a \( B \)-tree if there is one vertex with degree at least \( B \), while other vertices have degree less than \( B \). The vertex with highest degree is referred to as the center of \( T \). Let \( \operatorname{cdeg}(T) \) be the degree of the center.

We will use Theorem 14.1.4 to prove the following Lemma 14.2.2, which says that there is a robust isomorphism algorithm for forests of \( B \)-trees and degree bounded trees.

**Lemma 14.2.2.** Let \( G \) and \( H \) be two \( n \)-vertex forests of \( B \)-trees and \( (B - 1) \)-degree bounded trees. Given that \( \operatorname{opt}(G, H) \geq |E(G)| - \epsilon n \), there is a polynomial-time algorithm to find a bijection \( \sigma : V(G) \to V(H) \) such that \( \operatorname{val}(G, H, \sigma) \geq \operatorname{opt}(G, H) - 100B\sqrt{\epsilon}n \).

We defer the proof of Lemma 14.2.2 to the next subsection. Now we prove Theorem 14.1.5 using Lemma 14.2.2.

**Proof of Theorem 14.1.5 from Lemma 14.2.2.** For any integer parameter \( B \geq 2 \), one can remove at most \( 2n/B \) edges from \( G \) to get a forest of \( B \)-trees and \( (B - 1) \)-degree bounded trees, namely \( G' \). (To see this, simply root \( G \) using an arbitrary vertex, and for each vertex whose degree is no less than \( B \), remove the edge to its parent.) We do the similar decomposition for \( H \) to get \( H' \). Since we removed at most \( n/B \) edges from both \( G \) and \( H \), we have \( \operatorname{opt}(G', H') \geq \operatorname{opt}(G, H) - 4n/B \geq (1 - \epsilon - 4/B)n \). Also observe that \(|E(G')| - \operatorname{opt}(G', H') \leq |E(G)| - \operatorname{opt}(G, H) \leq \epsilon n \).

Now we apply the algorithm in Lemma 14.2.2 to \( G' \) and \( H' \) to get a bijection \( \sigma \). We have

\[
\operatorname{val}(G, H, \sigma) \geq \operatorname{val}(G', H', \sigma) \geq \operatorname{opt}(G', H') - 100B\sqrt{\epsilon}n \geq (1 - \epsilon - 4/B - 100B\sqrt{\epsilon})n.
\]
If we take $B = \lceil \epsilon^{-1/4} \rceil$, the algorithm finds a bijection $\sigma$ such that $\text{val}(G, H, \sigma) \geq (1 - 200\epsilon^{1/4})n$. 

### 14.2.1 Robust isomorphism algorithm for $B$-trees

In this subsection, we prove **Lemma 14.2.2**. Let $G$ consist of $G_1, G_2, \ldots, G_p$ and $\tilde{G}$ where $G_1, G_2, \ldots, G_p$ are $B$-trees and $\tilde{G}$ is a forest of $(B - 1)$-degree bounded trees; let $H$ consist of $H_1, H_2, \ldots, H_q$ and $\tilde{H}$ where $H_1, H_2, \ldots, H_q$ are $B$-trees and $\tilde{H}$ is a forest of $(B - 1)$-degree bounded trees.

For any two $B$-trees $G_i$ and $H_j$, let $E_c(G_i)$ and $E_c(H_j)$ be the set of edges incident to the centers of $G_i$ and $H_j$. Now let $C(G_i, H_j)$ be the minimum number of edges in $E_c(G_i) \cup E_c(H_j)$ one has to remove from $G_i$ and $H_j$, so that the connected components where the two centers are in are isomorphic and the isomorphism maps the center of $G_i$ to the center of $H_j$. Also, let $C_e(G_i, H_j)$ be the set of edges removed from $E_c(G_i) \cup E_c(H_j)$.

It is easy to see that both $C(\cdot, \cdot)$ and $C_e(\cdot, \cdot)$ can be computed in polynomial-time.

**The algorithm.** The first step of the algorithm finds a (partial) matching between $\{G_1, G_2, \ldots, G_p\}$ and $\{H_1, H_2, \ldots, H_q\}$. We use two mappings

$$
\tau : \{G_1, G_2, \ldots, G_p\} \to \{H_1, H_2, \ldots, H_q, \bot\}, \text{ and }
\tau' : \{H_1, H_2, \ldots, H_q\} \to \{G_1, G_2, \ldots, G_p, \bot\}
$$

to denote the (partial) matching. It is satisfied that $\tau(G_i) = H_j$ iff $\tau'(H_j) = G_i$. The algorithm uses the (polynomial-time) maximum weight bipartite graph matching algorithm to find $\tau$ and $\tau'$ so that the following cost is minimized.

$$
\text{cost}(\tau, \tau') = \sum_{G_i: \tau(G_i) = \bot} (\text{cdeg}(G_i) - B + 1) + \sum_{H_j: \tau'(H_j) = \bot} (\text{cdeg}(H_j) - B + 1) + \sum_{G_i: \tau(G_i) \neq \bot} C(G_i, \tau(G_i)).
$$

For each $G_i$, let $E'_c(G_i)$ be an arbitrary subset of cardinality $(\text{cdeg}(G_i) - B + 1)$ of $E_c(G_i)$. Define $E'_c(H_j)$ similarly for each $H_j$. Let

$$
E_0 = \bigcup_{G_i: \tau(G_i) = \bot} E'_c(G_i) \bigcup_{H_j: \tau'(H_j) = \bot} E'_c(H_j) \bigcup_{G_i: \tau(G_i) \neq \bot} C_e(G_i, \tau(G_i)),
$$

241
we have \(|E_0| = \text{cost}(\tau, \tau')\).

In the second step of the algorithm, we remove the edges in \(E_0\) from \(G\) and \(H\), call the remaining graphs \(G'\) and \(H'\) respectively. Observe that \(\text{opt}(G', H') \geq \text{opt}(G, H) - |E_0| = \text{opt}(G, H) - \text{cost}(\tau, \tau')\). Both \(G'\) and \(H'\) can be divided into two parts: \(G^{(a)}, H^{(a)}\) and \(G^{(b)}, H^{(b)}\). \(G^{(a)}\) consists of the connected components where the centers of \(G_i\)'s are in, where \(\tau(G_i) \neq \perp\). Similarly, \(H^{(a)}\) consists of the connected components where the centers of \(H_j\)'s are in, where \(\tau'(H_j) \neq \perp\). \(G^{(b)}\) and \(H^{(b)}\) are the remaining parts of \(G'\) and \(H'\). Observe that \(G^{(a)}\) is isomorphic to \(H^{(a)}\), while \(G^{(b)}\) and \(H^{(b)}\) are \((B - 1)\)-degree bounded.

Finally, we use the algorithm in Theorem 14.1.4 to find an approximation to \(\text{opt}(G^{(b)}, H^{(b)})\). Suppose that each of \(G^{(b)}, H^{(b)}\) has \(n^{(b)}\) vertices. The algorithm finds a bijection \(\sigma^{(b)}\) such that

\[
\text{val}(G^{(b)}, H^{(b)}, \sigma^{(b)}) \geq \left(1 - 20 \sqrt{\frac{B(n^{(b)} - \text{opt}(G^{(b)}, H^{(b)))}{n^{(b)}}} \right) n^{(b)} \\
\geq \text{opt}(G^{(b)}, H^{(b)}) - 20 \sqrt{B(n^{(b)} - \text{opt}(G^{(b)}, H^{(b))))n^{(b)}}. \tag{14.1}
\]

Observe that since \(\text{opt}(G^{(b)}, H^{(b)}) + |E_0| + |E(G^{(a)})| \geq \text{opt}(G, H) \geq (1 - \epsilon)n\), we have

\[
n^{(b)} - \text{opt}(G^{(b)}, H^{(b)}) \leq n^{(b)} - n + |E(G^{(a)})| + |E_0| + \epsilon n \leq \epsilon n + |E_0| \\
= \epsilon n + \text{cost}(\tau, \tau'), \tag{14.2}
\]

where the last inequality is because \(|E(G^{(a)})|\) is less than the number of vertices in \(G^{(a)}\), which is \(n - n^{(b)}\). Now we combine (14.1) and (14.2), getting

\[
\text{val}(G^{(b)}, H^{(b)}, \sigma^{(b)}) \geq \text{opt}(G^{(b)}, H^{(b)}) - 20 \sqrt{B(\epsilon n + \text{cost}(\tau, \tau'))n^{(b)}} \\
\geq \text{opt}(G^{(b)}, H^{(b)}) - 20 \sqrt{B(\epsilon n + \text{cost}(\tau, \tau'))n}. \tag{14.3}
\]

Let \(\sigma^{(a)}\) be the isomorphism between \(G^{(a)}\) and \(H^{(a)}\). The algorithm lets \(\sigma = \sigma^{(a)} \cup \sigma^{(b)}\) be the final solution.

**Analysis.** The analysis of the algorithm uses the following lemmas.

**Lemma 14.2.3.** \(\text{cost}(\tau, \tau') \leq 2(B + 1)\epsilon n \leq 4B\epsilon n\).

**Lemma 14.2.4.** Let \(G = G_0 \cup G_1\) be a union of two graphs with disjoint vertex sets. Similarly, let \(H = H_0 \cup H_1\). Suppose that \(G_0\) is isomorphic to \(H_0\) and \(|V(G)| = |V(H)|\), we have \(\text{opt}(G, H) = \text{opt}(G_1, H_1) + |E(G_0)|\).
We prove both Lemma 14.2.3 and Lemma 14.2.4 in the next subsection. Lemma 14.2.4 is very intuitive: if $G$ has a connected component $G_0$ which is isomorphic to a connected component $H_0$ in $H$, there must be an optimal solution for $G$ and $H$ which maps $G_0$ to $H_0$.

Now we proceed to analyze $\text{val}(G, H, \sigma)$ using these lemmas.

\[
\text{val}(G, H, \sigma) \geq \text{val}(G^{(a)}, H^{(a)}, \sigma^{(a)}) + \text{val}(G^{(b)}, H^{(b)}, \sigma^{(b)}) \\
\geq |E(G^{(a)})| + \text{opt}(G^{(b)}, H^{(b)}) \\
- 20\sqrt{B(\epsilon n + \text{cost}(\tau, \tau'))n} \quad \text{(by (14.3))} \\
\geq |E(G^{(a)})| + \text{opt}(G^{(b)}, H^{(b)}) - 20B\sqrt{5\epsilon n} \quad \text{(by Lemma 14.2.3)} \\
= \text{opt}(G', H') - 20B\sqrt{5\epsilon n} \quad \text{(Lemma 14.2.4)} \\
\geq \text{opt}(G, H) - 4B\epsilon n - 20B\sqrt{5\epsilon n} \quad \text{(by Lemma 14.2.3 again)} \\
\geq \text{opt}(G, H) - 100B\sqrt{\epsilon n}.
\]

### 14.2.2 Proofs of Lemma 14.2.3 and Lemma 14.2.4

**Proof of Lemma 14.2.3** Let $\sigma^* : V(G) \to V(H)$ be a bijection such that $\text{val}(G, H, \sigma^*) = (1-\epsilon)n$. We define $\tau^* : \{G_1, G_2, \ldots, G_p\} \to \{H_1, H_2, \ldots, H_q, \perp\}$ and $\tau'^* : \{H_1, H_2, \ldots, H_q\} \to \{G_1, G_2, \ldots, G_p, \perp\}$ as follows.

- $\tau^*(G_i) = H_j$ when $\sigma^*$ maps the center of $G_i$ to the center of $H_j$; $\tau^*(G_i) = \perp$ when the center of $G_i$ is not mapped to the center of any $H_j$.
- $\tau'^*(H_j) = G_i$ when $\sigma^*$ maps the center of $G_i$ to the center of $H_j$; $\tau'^*(H_j) = \perp$ when none of the centers of $G_i$ is not mapped to the center of $H_j$.

Now we are going to upper bound $\text{cost}(\tau^*, \tau'^*)$ and therefore prove the lemma.

For each $G_i$ where $1 \leq i \leq p$, if the center of $G_i$ is not mapped by $\sigma^*$ to the center of any $H_j$, it must be mapped to a vertex whose degree is less than $B$. This means that at least $(\text{cdeg}(G_i) - B + 1)$ edges incident to the center of $G_i$ are not mapped to $H$. Therefore, we have

\[
\sum_{G_i: \tau^*(G_i) = \perp} (\text{cdeg}(G_i) - B + 1) \leq \epsilon n.
\]
Since \( \text{cdeg}(G_i)/(\text{cdeg}(G_i) - B + 1) \leq B \), we have

\[
\sum_{G_i: \tau^*(G_i) = \perp} \text{cdeg}(G_i) \leq \epsilon B n \tag{14.4}
\]

Similarly, we have

\[
\sum_{H_j: \tau^*(H_j) = \perp} \text{cdeg}(H_j) \leq \epsilon B n. \tag{14.5}
\]

For each \( 1 \leq i \leq p, 1 \leq j \leq q \), let \( E_{G_i} \) be the set of the edges in \( G_i \) that are not mapped to \( H \); let \( E_{H_j} \) be the set of the edges in \( H_j \) that are not mapped to \( G \). Now consider any pair of \( G_i \) and \( H_j \) such that \( \sigma^* \) maps the center of \( G_i \) to the center of \( H_j \). Recall that \( C(G_i, H_j) \) is the minimum number of edges in \( E_c(G_i) \cup E_c(H_j) \) one has to remove from \( G_i \) and \( H_j \), so that the connected components corresponding to the two centers are isomorphic. Also recall that \( E_c(G_i) \) and \( E_c(H_j) \) are the sets of edges incident to the centers of \( G_i \) and \( H_j \) respectively.

For each edge \( e \in E_{G_i} \), we remove the corresponding edge in \( E_c(G_i) \), i.e. the unique edge on the unique path connecting \( e \) to the center of \( G_i \). Similarly, for each edge \( e \in E_{H_j} \), we remove the corresponding edge in \( E_c(H_j) \), i.e. the unique edge on the unique path connecting \( e \) to the center of \( H_j \). In total, at most \(|E_{G_i}| + |E_{H_j}|\) edges are removed.

We observe that, after removing the edges, the connected components corresponding to the centers of \( G_i \) and \( H_j \) are isomorphic and the isomorphism maps the center of \( G_i \) to the center of \( H_j : \sigma^* \) defines such an isomorphism.

To summarize, we have proved that \( C(G_i, H_j) \leq |E_{G_i}| + |E_{H_j}| \). Therefore,

\[
\sum_{G_i: \tau^*(G_i) \neq \perp} C(G_i, \tau(G_i)) \leq \sum_{G_i: \tau^*(G_i) \neq \perp} (|E_{G_i}| + |E_{\tau^*(G_i)}|) \leq \sum_{i=1}^p |E_{G_i}| + \sum_{j=1}^q |E_{H_j}| \leq 2\epsilon n. \tag{14.6}
\]

Now, summing up (14.4), (14.5), and (14.6), we get

\[
\text{cost}(\tau, \tau') \leq \text{cost}(\tau^*, \tau'^*) = \sum_{G_i: \tau^*(G_i) = \perp} \text{cdeg}(G_i) + \sum_{H_j: \tau^*(H_j) = \perp} \text{cdeg}(H_j) + \sum_{G_i: \tau^*(G_i) \neq \perp} C(G_i, \tau(G_i)) \leq 2(B+1)\epsilon n.
\]  

\[\square\]

244
Proof of Lemma 14.2.4. It is straightforward to see that \( \text{opt}(G,H) \geq \text{opt}(G_1,H_1) + |E(G_0)| \). Now we prove that \( \text{opt}(G,H) \leq \text{opt}(G_1,H_1) + |E(G_0)| \).

Given \( \sigma : V(G) \to V(H) \) such that \( \text{val}(G,H,\sigma) = \text{opt}(G,H) \), we define the bijection \( \sigma' : V(G_1) \to V(H_1) \) as follows and prove that \( \text{opt}(G,H) \leq \text{val}(G_1,H_1,\sigma') + |E(G_0)| \).

Let the bijection \( \tau : V(G_0) \to V(H_0) \) be the isomorphism between \( G_0 \) and \( H_0 \). For each \( v \in V(G_1) \), use the following procedure to decide \( \sigma'(v) \) : if \( \sigma(v) \in V(H_1) \), then return \( \sigma(v) \); if \( \sigma(v) \in V(H_0) \), repeat this procedure with \( v \leftarrow \tau^{-1}(\sigma(v)) \).

One can verify that the procedure above always terminates for all \( v \in V(G_1) \), and that \( \sigma' \) is indeed a bijection. It remains to show that \( \text{val}(G_1,H_1,\sigma') \geq \text{opt}(G,H) - |E(G_0)| \).

For each edge \((u,v) \in E(G_1)\) such that \( (\sigma(u),\sigma(v)) \in E(H) \), let us write down \( u_1 = \sigma(u), u_2, u_3, \ldots, u_p = \sigma'(u) \) and \( v_1 = \sigma(v), v_2, v_3, \ldots, v_q = \sigma'(v) \) to be the sequence of vertices in \( V(H) \) visited by the procedure above when the input is \( u \) and \( v \) respectively.

We can assume w.l.o.g. that \( p \leq q \).

Suppose that \( (\sigma'(u),\sigma'(v)) \notin E(H_1) \), we know that \( u_p = \sigma'(u) \in V(H_1) \) and \( v_p \in V(H_0) \), and therefore \( (u_p,v_p) \notin E(H) \). Let \( i \) be the smallest index such that \( (u_i,v_i) \notin E(H) \). We have that \( i \geq 2 \). Therefore \( (u_{i-1},v_{i-1}) \in E(H_0) \). Since \( \tau \) is the isomorphism between \( G_0 \) and \( H_0 \), we have that \( (\tau^{-1}(u_{i-1}),\tau^{-1}(v_{i-1})) \in E(G_0) \). Also note that \( (\sigma(\tau^{-1}(u_{i-1})),\sigma(\tau^{-1}(v_{i-1}))) = (u_i,v_i) \notin E(H) \).

To summarize, for each \((u,v) \in E(G_1)\) such that \( (\sigma(u),\sigma(v)) \in E(H) \) and \( (\sigma'(u),\sigma'(v)) \notin E(H_1) \), we have set up a mapping \( f((u,v)) = (u_f,v_f) = (\tau^{-1}(u_{i-1}),\tau^{-1}(v_{i-1})) \) with the property that \( (u_f,v_f) \in E(G_0) \) and \( (\sigma(u_f),\sigma(v_f)) \notin E(H) \). By the fact that the procedure to define \( \sigma'(v) \) is reversible at any point, one can verify that \( f \) is injective. Therefore,

\[
|\{(u,v) \in E(G_1) : (\sigma(u),\sigma(v)) \in E(H), (\sigma'(u),\sigma'(v)) \notin E(H_1)\}| \leq |\{(u,v) \in E(G_0) : (\sigma(u),\sigma(v)) \notin E(H)\}|.
\]

Therefore,

\[
\text{val}(G_1,H_1,\sigma') = |E(G_1)| - |\{(u,v) \in E(G_1) : (\sigma'(u),\sigma'(v)) \notin E(H_1)\}| \geq |E(G_1)| - |\{(u,v) \in E(G_1) : (\sigma'(u),\sigma'(v)) \notin E(H_1), (\sigma(u),\sigma(v)) \in E(H)\}| - |\{(u,v) \in E(G_1) : (\sigma(u),\sigma(v)) \notin E(H)\}| \geq |E(G_1)| - |\{(u,v) \in E(G_0) : (\sigma(u),\sigma(v)) \notin E(H)\}| = |E(G_1)| - |E(G_0)| + |\{(u,v) \in E(G_0) : (\sigma(u),\sigma(v)) \in E(H)\}| - |\{(u,v) \in E(G_1) : (\sigma(u),\sigma(v)) \notin E(H)\}|.
\]

Therefore,

\[
|\{(u,v) \in E(G_1) : (\sigma(u),\sigma(v)) \in E(H), (\sigma'(u),\sigma'(v)) \notin E(H_1)\}| \leq |\{(u,v) \in E(G_0) : (\sigma(u),\sigma(v)) \notin E(H)\}|.
\]
\[ \text{val}(G, H, \sigma) - |E(G_0)| = \text{opt}(G, H) - |E(G_0)|. \]
Part IV

Other approximation and hardness of approximation results
Chapter 15

Certifying the $2 \rightarrow 4$ norm of random linear operators

15.1 Introduction

For a function $f : \Omega \rightarrow \mathbb{R}$ on a (finite) probability space $\Omega$, the $p$-norm is defined as $\|f\|_p = (\mathbb{E}_\Omega f^p)^{1/p}$. The $p \rightarrow q$ norm $\|A\|_{p \rightarrow q}$ of a linear operator $A$ between vector spaces of such functions is the smallest number $c \geq 0$ such that $\|Af\|_q \leq c \|f\|_p$ for all functions $f$ in the domain of $A$. We also define the $p \rightarrow q$ norm of a subspace $V$ to be the maximum of $\|f\|_q / \|f\|_p$ for $f \in V$; note that for $p = 2$ this is the same as the norm of the projector operator into $V$.

In this chapter, we are interested in the case $p < q$ and we will call such $p \rightarrow q$ norms hypercontractive. Roughly speaking, for $p < q$, a function $f$ with large $\|f\|_q$ compared to $\|f\|_p$ can be thought of as “spiky” or somewhat sparse (i.e., much of the mass concentrated in small portion of the entries). Hence finding a function $f$ in a linear subspace $V$ maximizing $\|f\|_q / \|f\|_2$ for some $q > 2$ can be thought of as a geometric analogue of the problem finding the shortest word in a linear code. This problem is equivalent to computing the $2 \rightarrow q$ norm of the projector $P$ into $V$ (since $\|Pf\|_2 \leq \|f\|_2$). Also when $A$ is a normalized adjacency matrix of a graph (or more generally a Markov operator), upper

---

1We follow the convention to use expectation norms for functions (on probability spaces) and counting norms, denoted as $\|v\|_p = (\sum_{i=1}^n |v_i|^p)^{1/p}$, for vectors $v \in \mathbb{R}^m$. All normed spaces here will be finite dimensional. We distinguish between expectation and counting norms to avoid recurrent normalization factors.

2We use this name because a bound of the form $\|A\|_{p \rightarrow q} \leq 1$ for $p < q$ is often called a hypercontractive inequality.
bounds on the $p \to q$ norm are known as mixed-norm, Nash or hypercontractive inequalities and can be used to show rapid mixing of the corresponding random walk (e.g., see the surveys [104, 201]). Such bounds also have many applications to theoretical computer science, which are described in the survey [43].

However, very little is known about the complexity of computing these norms. This is in contrast to the case of $p \to q$ norms for $p \geq q$, where much more is known both in terms of algorithms and lower bounds, see [214, 146, 42].

We study a natural semidefinite programming (SDP) relaxation for computing the $2 \to 4$ norm of a given linear operator which we call TensorSDP. While TensorSDP is very unlikely to provide a poly-time constant-factor approximation for the $2 \to 4$ norm in general (as shown in [34]), we do show that it provides such approximation on random linear operators, as we describe below.

We show that TensorSDP certifies a constant upper bound on the ratio $\frac{\|A\|_{2 \to 4}}{\|A\|_{2 \to 2}}$ where $A : \mathbb{R}^n \to \mathbb{R}^m$ is a random linear operator (e.g., obtained by a matrix with entries chosen as i.i.d Bernoulli variables) and $m \geq \Omega(n^2)$. In contrast, if $m = o(n^2)$ then this ratio is $\omega(1)$, and hence this result is almost tight in the sense of obtaining “good approximation” in the sense mentioned above. We find this interesting, since random matrices seem like natural instances; indeed for superficially similar problems such shortest codeword, shortest lattice vector (or even the $1 \to 2$ norm), it seems hard to efficiently certify bounds on random operators.

15.2 The TensorSDP algorithm

Observe that when the linear operator $A$ is given in the form of $A = \sum_{j=1}^{m} e_j a_j^T$, for every $x$ such that $\|x\|_2 = 1$, we have

$$\|Ax\|_4^4 = \frac{1}{m} \sum_{j=1}^{m} (a_j^T x)^4 = \frac{1}{m} \sum_{j=1}^{m} \sum_{i_1, i_2, i_3, i_4 \in [n]} \text{tr}(a_i a_i^T \otimes a_i a_i^T)(xx^T \otimes xx^T).$$

Instead of maximizing the above quantity over all $xx^T \otimes xx^T$, we maximize over all the $4$-th moment matrices. This gives the following natural semidefinite program for estimating the $2 \to 4$ norm of $A$.

3We use the name TensorSDP for this program since it will be a canonical relaxation of the polynomial program $\max_{\|x\|_2 = 1} (T, x \otimes 4)$ where $T$ is the 4-tensor such that $(T, x \otimes 4) = \|Ax\|_4^4$. Indeed, it is identical to the degree-2 Lasserre–Parrilo SDP relaxation.
Algorithm 2 The TensorSDP algorithm for $2 \to 4$ norms.

**Input:** A linear operator $A = \sum_{j=1}^{m} e_j a_j^T$.

**Output:** An estimated value for $\|A\|_{2\to4}^4$.

Find $X = (X(i_1,i_2),(i_3,i_4))_{i_1,i_2,i_3,i_4 \in [n]}$ over set of $n^2 \times n^2$ real matrices to maximize

$$\text{tr} \left( \frac{1}{m} \sum_{j=1}^{m} \sum_{i_1,i_2,i_3,i_4 \in [n]} \text{tr}(a_i a_i^T \otimes a_i a_i^T) \right) X$$

such that

- $X \succeq 0,$
- $E_{i,j \in [n]} X(i,j),(i,j) = 1,$
- $X(i_1,i_2),(i_3,i_4) = X(\pi(i_1),\pi(i_2),(\pi(i_3),\pi(i_4))$ for all permutations $\pi$ over $[4].$

Output the above quantity with the maximizer $X.$

### 15.3 Certifying the hypercontractivity of random operators

In this section we show that TensorSDP provides non-trivial approximation guarantees on the $2 \to 4$ norms of random linear operators.

Let $A = \sum_{i=1}^{m} e_i a_i^T / \sqrt{n}$, where $e_i$ is the vector with a 1 in the $i^{th}$ position, and each $a_i$ is chosen i.i.d. from a distribution $\mathcal{D}$ on $\mathbb{R}^n$. Three natural possibilities are

1. $\mathcal{D}_{\text{sign}}$: the uniform distribution over $\{-1, 1\}^n$
2. $\mathcal{D}_{\text{Gaussian}}$: a vector of $n$ independent Gaussians with mean zero and variance 1
3. $\mathcal{D}_{\text{unit}}$: a uniformly random (expectation-norm) unit vector on $\mathbb{R}^n$.

Our arguments will apply to any of these cases, or even to more general nearly-unit vectors with bounded sub-Gaussian moment (details below).

Before discussing the performance of TensorSDP, we will discuss how the $2 \to 4$-norm of $A$ behaves as a function of $n$ and $m$. We can gain intuition by considering two limits in the case of $\mathcal{D}_{\text{Gaussian}}$. If $n = 1$, then $\|A\|_{2\to4} = \|a\|_4$, for a random Gaussian vector.
a. For large $m$, $\|a\|_4$ is likely to be close to $3^{1/4}$, which is the fourth moment of a mean-zero unit-variance Gaussian. By Dvoretzky’s theorem [186], this behavior can be shown to extend to higher values of $n$. Indeed, there is a universal $c > 0$ such that if $n \leq c\sqrt{me^2}$, then w.h.p. $\|A\|_{2 \rightarrow 4} \leq 3^{1/4} + \epsilon$. In this case, the maximum value of $\|Ax\|_4$ looks roughly the same as the average or the minimum value, and we also have $\|Ax\|_4 \geq (3^{1/4} - \epsilon) \|x\|_2$ for all $x \in \mathbb{R}^n$. In the cases of $D_{\text{sign}}$ and $D_{\text{unit}}$, the situation is somewhat more complicated, but for large $n$, their behavior becomes similar to the Gaussian case.

On the other hand a simple argument (see e.g. [34]) shows that $\|A\|_{2 \rightarrow 4} \geq n^{1/2}/m^{1/4}$ for any (not only random) $m \times n$ matrix with all $\pm 1/\sqrt{n}$ entries. A nearly identical bound applies for the case when the $a_i$ are arbitrary unit or near-unit vectors. Thus, in the regime where $n \geq \omega(\sqrt{m})$, we always have $\|A\|_{2 \rightarrow 4} \geq \omega(1)$.

The following theorem shows that TensorSDP achieves approximately the correct answer in both regimes.

**Theorem 15.3.1.** Let $a_1, \ldots, a_m$ be drawn i.i.d. from a distribution $D$ on $\mathbb{R}^n$ with $D \in \{D_{\text{Gaussian}}, D_{\text{sign}}, D_{\text{unit}}\}$, and let $A = \sum_{i=1}^m e_i a_i^T / \sqrt{n}$. Then w.h.p. TensorSDP$(A) \leq 3 + c \max(\frac{n \sqrt{m}}{m}, \frac{n^2}{m^{1/4}})$ for some constant $c > 0$.

From Theorem 15.3.1 and the fact that $\|A\|_{2 \rightarrow 4} \leq \text{TensorSDP}(A)$, we obtain:

**Corollary 15.3.2.** Let $A$ be as in Theorem 15.3.1. Then $\exists c > 0$ such that w.h.p.

$$\|A\|_{2 \rightarrow 4} \leq \begin{cases} 3^{1/4} + c \frac{n \sqrt{m}}{m} & \text{if } n \leq \sqrt{m} \\ c \frac{n^{1/2}}{m^{1/4}} & \text{if } n > \sqrt{m} \end{cases}$$ \hspace{1cm} (15.1)

Before proving Theorem 15.3.1 we introduce some more notation. This will in fact imply that Theorem 15.3.1 applies to a broader class of distributions. For a distribution $D$ on $\mathbb{R}^N$, define the $\psi_p$ norm $\|D\|_{\psi_p}$ to be the smallest $C > 0$ such that

$$\max_{v \in S(\mathbb{R}^N)} \mathbb{E}_{a \sim D} e^{-\frac{|\langle v, a \rangle|^p N^{p/2}}{C p}} \leq 2,$$ \hspace{1cm} (15.2)

or $\infty$ if no finite such $C$ exists. We depart from the normal convention by including a factor of $N^{p/2}$ in the definition, to match the scale of [2]. The $\psi_2$ norm (technically a seminorm) is also called the sub-Gaussian norm of the distribution. One can verify that for each of the above examples (sign, unit and Gaussian vectors), $\psi_2(D) \leq O(1)$.

We also require that $D$ satisfies a boundedness condition with constant $K \geq 1$, defined as

$$\Pr \left[ \max_{i \in [m]} \|a_i\|_2 > K \max(1, (m/N)^{1/4}) \right] \leq e^{-\sqrt{N}}.$$ \hspace{1cm} (15.3)
Similarly, $K$ can be taken to be $O(1)$ in each case that we consider.

We will require a following result of [1, 2] about the convergence of sums of i.i.d rank-one matrices.

**Lemma 15.3.3** (2). Let $\mathcal{D}'$ be a distribution on $\mathbb{R}^N$ such that $E_{v \sim \mathcal{D}'} vv^T = I$, $\|\mathcal{D}'\|_{\psi_1} \leq \psi$ and (15.3) holds for $\mathcal{D}'$ with constant $K$. Let $v_1, \ldots, v_m$ be drawn i.i.d. from $\mathcal{D}'$. Then with probability $\geq 1 - 2 \exp(-c\sqrt{N})$, we have

\[
(1 - \epsilon)I \leq \frac{1}{m} \sum_{i=1}^{m} v_i v_i^T \leq (1 + \epsilon)I,
\]

where $\epsilon = C(\psi + K)^2 \max(N/m, \sqrt{N/m})$ with $c, C > 0$ universal constants.

The $N \leq m$ case (when the $\sqrt{N/m}$ term is applicable) was proven in Theorem 1 of [2], and the $N > m$ case (i.e. when the max is achieved by $N/m$) was proven in Theorem 2 of [2] (see also Theorem 3.13 of [1]).

**Proof of Theorem 15.3.1** Define $A_{2,2} = \frac{1}{m} \sum_{i=1}^{m} a_i a_i^T \otimes a_i a_i^T$. For $n^2 \times n^2$ real matrices $X, Y$, define $\langle X, Y \rangle := \text{tr} X^T Y / n^2 = E_{i,j \in [n]} X_{i,j} Y_{i,j}$. Additionally define the convex set $\mathcal{X}$ to be the set of $n^2 \times n^2$ real matrices $X = (X(i_1,i_2),(i_3,i_4))_{i_1,i_2,i_3,i_4 \in [n]}$ with $X \succeq 0$, $E_{i,j \in [n]} X(i,j),(i,j) = 1$ and $X(i_1,i_2),(i_3,i_4) = X(i_\pi(1),i_\pi(2)),(i_\pi(3),i_\pi(4))$ for any permutation $\pi \in S_4$. Finally, let $h_{\mathcal{X}}(Y) := \max_{X \in \mathcal{X}} \langle X, Y \rangle$. It is straightforward to show that

\[
\text{TensorSDP}(A) = h_{\mathcal{X}}(A_{2,2}) = \max_{X \in \mathcal{X}} \langle X, A_{2,2} \rangle.
\]

We note that if $\mathcal{X}$ were defined without the symmetry constraint, it would simply be the convex hull of $xx^T$ for unit vectors $x \in \mathbb{R}^n$ and $\text{TensorSDP}(A)$ would simply be the largest eigenvalue of $A_{2,2}$. However, we will later see that the symmetry constraint is crucial to $\text{TensorSDP}(A)$ being $O(1)$.

Our strategy will be to analyze $A_{2,2}$ by applying Lemma 15.3.3 to the vectors $v_i := \Sigma^{-1/2}(a_i \otimes a_i)$, where $\Sigma = E a_i a_i^T \otimes a_i a_i^T$, and $^{-1/2}$ denotes the pseudo-inverse. First, observe that, just as the $\psi_2$ norm of the distribution over $a_i$ is constant, a similar calculation can verify that the $\psi_1$ norm of the distribution over $a_i \otimes a_i$ is also constant. Next, we have to argue that $\Sigma^{-1/2}$ does not increase the norm by too much.

To do so, we compute $\Sigma$ for each distribution over $a_i$ that we have considered. Let $F$ be the operator satisfying $F(x \otimes y) = y \otimes x$ for any $x, y \in \mathbb{R}^n$; explicitly $F = \sum_{i,j=1}^{n} e_i e_j^T \otimes e_j e_i^T$. 253
Define
\[ \Phi := \sum_{i=1}^{n} e_i \otimes e_i \]  
\[ \Delta := \sum_{i=1}^{n} e_i e_i^T \otimes e_i e_i^T \]  
(15.6) (15.7)

Direct calculations (omitted) can verify that the cases of random Gaussian vectors, random unit vectors and random $\pm 1$ vectors yield respectively
\[ \Sigma_{\text{Gaussian}} = I + F + \Phi \Phi^T \]  
(15.8a)
\[ \Sigma_{\text{unit}} = \frac{n}{n+1} \Sigma_{\text{Gaussian}} \]  
(15.8b)
\[ \Sigma_{\text{sign}} = \Sigma_{\text{Gaussian}} - 2\Delta \]  
(15.8c)

In each case, the smallest nonzero eigenvalue of $\Sigma$ is $\Omega(1)$, so $v_i = \Sigma^{-1/2}(a_i \otimes a_i)$ has $\psi_1 \leq O(1)$ and satisfies the boundedness condition (15.3) with $K \leq O(1)$.

Thus, we can apply Lemma 15.3.3 (with $N = \text{rank } \Sigma \leq n^2$ and $\epsilon := c \max(n/\sqrt{m}, n^2/m)$) and find that in each case w.h.p.
\[ A_{2,2} = \frac{1}{m} \sum_{i=1}^{m} a_i a_i^T \otimes a_i a_i^T \preceq (1+\epsilon) \Sigma \preceq (1+\epsilon) (I + F + \Phi \Phi^T) \]  
(15.9)

Since $h_X(Y) \geq 0$ whenever $Y \succeq 0$, we have $h_X(A_{2,2}) \leq (1+\epsilon) h_X(\Sigma)$. Additionally, $h_X(I + F + \Phi \Phi^T) \leq h_X(I) + h_X(F) + h_X(\Phi \Phi^T)$, so we can bound each of three terms separately. Observe that $I$ and $F$ each have largest eigenvalue equal to 1, and so $h_X(I) \leq 1$ and $h_X(F) \leq 1$. (In fact, these are both equalities.)

However, the single nonzero eigenvalue of $\Phi \Phi^T$ is equal to $n$. Here we will need to use the symmetry constraint on $X$. Let $X^\Gamma$ be the matrix with entries $X^\Gamma_{(i_1,i_2),(i_3,i_4)} := X_{(i_1,i_4),(i_3,i_2)}$. If $X \in \mathcal{X}$ then $X = X^\Gamma$. Additionally, $\langle X, Y \rangle = \langle X^\Gamma, Y^\Gamma \rangle$. Thus
\[ h_X(\Phi \Phi^T) = h_X((\Phi \Phi^T)^\Gamma) \leq \| (\Phi \Phi^T)^\Gamma \|_{2 \to 2} = 1. \]
This last equality follows from the fact that $(\Phi \Phi^T)^\Gamma = F$.

Putting together these ingredients, we obtain the proof of the theorem. \[ \square \]

It may seem surprising that the factor of $3^{1/4}$ emerges even for matrices with, say, $\pm 1$ entries. An intuitive justification for this is that even if the columns of $A$ are not Gaussian vectors, most linear combinations of them resemble Gaussians. The following Lemma shows that this behavior begins as soon as $n$ is $\omega(1)$.
Lemma 15.3.4. Let $A = \sum_{i=1}^{m} e_i a_i^T / \sqrt{n}$ with $E_i \|a_i\|_2^4 \geq 1$. Then $\|A\|_{2 \to 4} \geq (3/(1 + 2/n))^{1/4}$.

To see that the denominator cannot be improved in general, observe that when $n = 1$ a random sign matrix will have $2 \to 4$ norm equal to 1.

Proof. Choose $x \in \mathbb{R}^n$ to be a random Gaussian vector such that $E_x \|x\|_2^2 = 1$. Then

$$E_x \|Ax\|_4^4 = E_x E_i n^{-2} (a_i^T x)^4 = n^2 E_x E_i \langle a_i, x \rangle^4 = 3 E_i \|a_i\|_2^4 \geq 3.$$  \hspace{1cm} (15.10)

The last equality comes from the fact that $\langle a_i, x \rangle$ is a Gaussian random variable with mean zero and variance $\|a_i\|_2^2 / n$. On the other hand, $E_x \|x\|_2^4 = 1 + 2/n$. Thus, there must exist an $x$ for which $\|Ax\|_4^4 / \|x\|_2^4 \geq 3/(1 + 2/n)$. $\square$
Chapter 16

Hardness of $\text{MAX}\Gamma-2\text{-LIN}$ and $\text{MAX}\Gamma-3\text{-LIN}$ over integers

16.1 Introduction

In this chapter we consider one of the most fundamental algorithmic tasks: solving systems of linear equations. Given a ring $R$, the $\text{MAX}k\text{-LIN}(R)$ problem is defined as follows: An input instance is a list of linear equations of the form $a_1x_{i_1} + \cdots + a_kx_{i_k} = b$, where $a_1, \ldots, a_k, b \in R$ are constants and $x_{i_1}, \ldots, x_{i_k}$ are variables from the set $\{x_1, \ldots, x_n\}$. Each equation also comes with a nonnegative rational weight; it is assumed the weights sum up to 1. The algorithmic task is to assign values from $R$ to the variables so as to maximize the total weight of satisfied equations. We say that an assignment is $\gamma$-good if the equations it satisfies have total weight at least $\gamma$. We say that an algorithm achieves $(c, s)$-approximation if, whenever the instance has a $c$-good solution, the algorithm is guaranteed to find an $s$-good solution.

16.1.1 Prior work on $\text{MAX}3\text{-LIN}(\mathbb{Z})$

It is an old result of Arora–Babai–Stern–Sweedyk [13] that for all $0 < \delta < 1$ there exists $\epsilon > 0$ and $k \in \mathbb{Z}^+$ such that it is NP-hard to $(\epsilon, \delta\epsilon)$-approximate $\text{MAX}k\text{-LIN}(\mathbb{Q})$. Håstad’s seminal work from 1997 [116] showed hardness even for very sparse, near-satisfiable instances: specifically, he showed that for all constant $\epsilon, \delta > 0$ and $q \in \mathbb{N}$, it is NP-hard to $(1 - \epsilon, 1/q + \delta)$-approximate $\text{MAX}3\text{-LIN}(\mathbb{Z}_q)$. This is optimal in the sense that it is algorithmically easy to $(1, 1)$-approximate or $(c, 1/q)$-approximate $\text{MAX}3\text{-LIN}(\mathbb{Z}_q)$. Håstad’s
hardness result even holds for the special case of \text{MAX}-3-LIN(\mathbb{Z}_q), meaning that all equations are of the form \(x_{i_1} - x_{i_2} + x_{i_3} = b\).

Håstad’s proof does not strictly generalize the ABSS [13] result on \text{MAX}-k-LIN(\mathbb{Q}) because there is no obvious reduction from hardness over \(\mathbb{Z}_q\) to hardness over \(\mathbb{Q}\). Indeed, it was not until much later, 2006, that NP-hardness of \((1 - \epsilon, \delta)\)-approximating \text{MAX}-k-LIN(\mathbb{Q}) was shown [92, 108]. Finally, in 2007 Guruswami and Raghavendra [109] generalized all of [13, 92, 108] by showing NP-hardness of \((1 - \epsilon, \delta)\)-approximating \text{MAX}^{\Gamma}-3-LIN(\mathbb{Z}). As we will see shortly, this easily implies the same hardness for \text{MAX}^{\Gamma}-3-LIN(\mathbb{Q}) and \text{MAX}^{\Gamma}-3-LIN(\mathbb{R}). Indeed, it shows a kind of “bicriteria” hardness: given a \text{MAX}^{\Gamma}-3-LIN(\mathbb{Z}) instance with a \((1 - \epsilon)\)-good solution over \(\mathbb{Z}\), it is NP-hard to find a \(\delta\)-good solution even over \(\mathbb{R}\). Guruswami and Raghavendra’s proof followed that of Håstad’s to some extent, but involved somewhat technically intricate derandomized Long Code testing, using Fourier analysis with respect to a certain exponential distribution on \(\mathbb{Z}^+\).

We would also like to mention the very recent work of Khot and Moshkovitz [139]. Motivated by proving the Unique Games Conjecture, they showed a strong NP-hardness result for a homogeneous variant of \text{MAX}3-LIN(\mathbb{R}). Specifically, they considered the case where all equations are of the form \(a_1 x_{i_1} + a_2 x_{i_2} + a_3 x_{i_3} = 0\) with \(a_1, a_2, a_3 \in [\frac{1}{2}, 2]\). Very roughly speaking, they showed there is a universal \(\delta > 0\) such that for all \(\epsilon > 0\) the following problem is NP-hard: given an instance where there is a “Gaussian-distributed” real assignment which is \((1 - \epsilon)\)-good, find a Gaussian-distributed assignment in which the weight of equations satisfied to within margin \(\delta \sqrt{\epsilon}\) is at least \(1 - \delta\). This result is incomparable to the one in [109].

16.1.2 Prior work on \text{MAX}2-LIN

Following Håstad’s work there was five years of no progress on \text{MAX}2-LIN(\mathbb{R}) for any ring \(\mathbb{R}\). To circumvent this, in 2002 Khot [136] introduced the Unique Games (UG) Conjecture, which would prove to be very influential (and notorious!). Khot showed a strong “UG-hardness” result for \text{MAX}2-LIN(\mathbb{Z}_2) (crediting the result essentially to Håstad), namely that for all \(t > 1/2\) and sufficiently small \(\epsilon > 0\) it is UG-hard to \((1 - \epsilon, 1 - \epsilon^t)\)-approximate \text{MAX}2-LIN(\mathbb{Z}_2). This result is essentially optimal due to the Goemans–Williamson algorithm [94].

In 2004, Khot–Kindler–Mossel–O’Donnell [141] (using [175]) extended this work by showing that for all \(\epsilon, \delta > 0\), there exists \(q \in \mathbb{N}\) such that \((1 - \epsilon, \delta)\)-approximating \text{MAX}^{\Gamma}-2-LIN(\mathbb{Z}_q) is UG-hard, and hence in fact UG-complete. Here \(\Gamma\)-2-LIN means that all equations are of the form \(x_{i_1} - x_{i_2} = b\). KKMO gave a quantitative dependence as
well: given $\epsilon$ and $q$ one can choose any $\delta > q\Lambda_{1-\epsilon}(1/q) \approx (1/q)^{\epsilon/(2-\epsilon)}$, where $\Lambda_{1-\epsilon}(1/q)$ is a certain correlated Gaussian quadrant probability.

The following natural question was left open by KKMO [141]:

**Question 16.1.1.** Is it true that for all $\epsilon, \delta > 0$ it is UG-hard to $(1 - \epsilon, \delta)$-approximate $\text{MAX}_2\text{-LIN}(\mathbb{Z})$?

The key technical tool used in the KKMO hardness result for $\text{MAX}_2\text{-LIN}(\mathbb{Z}_q)$, namely the Majority Is Stablest Theorem [175], has a bad dependence on the parameter $q$. Thus pushing $q$ to be “superconstantly” large seemed to pose a fundamental problem. The question above is one of the open problems posed at the end of Raghavendra’s monumental thesis [190].

### 16.1.3 Our contributions

In this chapter we show that it is relatively easy to modify the proofs of the hardness results known for $\text{MAX}_2\text{-LIN}(\mathbb{Z}_q)$ and $\text{MAX}_3\text{-LIN}(\mathbb{Z}_q)$ to obtain $(1 - \epsilon, \delta)$-approximation hardness results for $\text{MAX}_2\text{-LIN}(\mathbb{Z})$ and $\text{MAX}_3\text{-LIN}(\mathbb{Z})$. (Here $\text{MAX}_3\text{-LIN}$ means that all equations are of the form $x_{i_1} + x_{i_2} - x_{i_3} = b$.) Thus we resolve the open question about $\text{MAX}_2\text{-LIN}(\mathbb{Z})$ and give a simpler proof of the Guruswami–Raghavendra [109] result. Our results also hold over $\mathbb{R}$ and over “superconstantly large” cyclic groups $\mathbb{Z}_q$ (we are not aware of previously known hardness results over $\mathbb{Z}_q$ when $q$ is superconstant and prime). The results also have an essentially optimal quantitative tradeoff between $\epsilon, \delta$, and the magnitudes of the “right-hand side constants” $b$.

To state our two theorems, let us define $B\text{-BOUNDED-MAX}_2\text{-LIN}$ and $B\text{-BOUNDED-MAX}_3\text{-LIN}$ to be the special cases of $\text{MAX}_2\text{-LIN}$ and $\text{MAX}_3\text{-LIN}$ in which all right-hand side constants $b$ are integers satisfying $|b| \leq B$. Given an instance $\mathcal{I}$ of $\text{MAX}_k\text{-LIN}$ with integer constants $b$, we use the notation $\text{opt}_R(\mathcal{I})$ to denote the maximum weight of equations that can be satisfied when the equations are evaluated over $R$.

**Theorem 16.1.2.** For all constant $\epsilon, \gamma, \kappa > 0$ and constant $q \in \mathbb{N}$, given a $q\text{-BOUNDED-MAX}_2\text{-LIN}$ instance $\mathcal{I}$ it is UG-hard to distinguish the following two cases:

- (Completeness.) There is a $(1 - \epsilon - 3\gamma)$-good assignment over $\mathbb{Z}$; i.e., $\text{opt}_\mathbb{Z}(\mathcal{I}) \geq 1 - \epsilon - 3\gamma$. 

259
• (Soundness.) There is no \((q \Lambda_1 - (1/q) + \kappa)\)-good assignment over \(\mathbb{Z}_q\); i.e., 
\[ \text{opt}_{\mathbb{Z}_q}(I) \leq q \Lambda_1 - (1/q) + \kappa. \]

Note that \(q \Lambda_1 - (1/q) \approx \frac{1}{q} \epsilon / (2 - \epsilon)\) is the same soundness proved by KKMO [141] for 
\(\text{MAX}-2\)-LIN(\(\mathbb{Z}_q\)).

**Theorem 16.1.3.** For all constant \(\epsilon, \kappa > 0\) and \(q \in \mathbb{N}\), given a \(q\)-BOUNDED-MAX3-LIN
instance \(I\) it is NP-hard to distinguish the following two cases:

• (Completeness.) There is a \((1 - O(\epsilon))\)-good assignment over \(\mathbb{Z}\), i.e., 
\[ \text{opt}_x(I) \geq 1 - O(\epsilon). \]

• (Soundness.) There is no \((1/q + \kappa)\)-good assignment over \(\mathbb{Z}_q\); i.e., 
\[ \text{opt}_{\mathbb{Z}_q}(I) \leq 1/q + \kappa. \]

Note that \(\text{opt}_x(I) \leq \text{opt}_{\mathbb{Z}_q}(I)\) since we can convert a \(\delta\)-good assignment over \(\mathbb{Z}\) to 
a \(\delta\)-good assignment over \(\mathbb{Z}_q\) by reducing the integer solution modulo \(q\). Therefore our 
hardness results are of the strongest “bicriteria” type: even when promised that there is 
near-perfect solution over \(\mathbb{Z}\), it is hard for an algorithm to find a slightly good solution over 
\(\mathbb{Z}_q\). Indeed, by virtue of Lemma 16.6.1 in Section 16.6, by losing just a constant factor 
in the soundness, we can show that it is also hard for an algorithm to find a slightly good 
solution over any ring \(\{\mathbb{R}, \mathbb{Z}, \mathbb{Z}_{q+1}, \mathbb{Z}_{q+2}, \ldots\}\) of the algorithm’s choosing. 
Our results subsume and unify all aforementioned results on \(\text{MAX3-LIN}(\mathbb{Z}_q)\), \(\text{MAX3-LIN}(\mathbb{Z})\), and 
\(\text{MAX2-LIN}(\mathbb{Z}_q)\), and also provide an optimal UG-hardness result for \(\text{MAX}-2\)-LIN(\(\mathbb{Z}\)).

### 16.2 Preliminaries

#### 16.2.1 Notations and Definitions

We write \(\mathbb{Z}_q\) for the integers modulo \(q\), and we identify the elements with \(\{0, 1, \ldots, q - 1\} \in \mathbb{Z}\). We sometimes write \(\oplus_q\) for addition of integers modulo \(q\) and + for addition over 
the integers. For two vectors \(x, y \in \mathbb{Z}_n\), both \(x \oplus_q y\) and \(x + y\) are coordinate-wise add.
We will also write \(\Delta_q\) for the set of probability distributions over \(\mathbb{Z}_q\). We can identify \(\Delta_q\) 
with the standard \((q - 1)\)-dimensional simplex in \(\mathbb{R}^q\). We also identify an element \(a \in \mathbb{Z}_q\) 
with a distribution in \(\Delta_q\), namely, the distribution that puts all of its probability mass on \(a\).

Fix \(x \in \mathbb{Z}_n\), a random variable \(y\) is \((1 - \epsilon)\)-correlated to \(x\), i.e. 
\(y \sim_{1-\epsilon} x\), if \(y\) can be get by rerandomizing each coordinate of \(x\) independently with probability \(\epsilon\).
We recall some standard definitions from the harmonic analysis of boolean functions (see, e.g., [190]). We will be considering functions of the form \( f : \mathbb{Z}_q^n \rightarrow \mathbb{R} \). The set of all functions \( f : \mathbb{Z}_q^n \rightarrow \mathbb{R} \) forms an inner product space with inner product
\[
\langle f, g \rangle = \mathbb{E}_x \left[ f(x) \cdot g(x) \right],
\]
where \( x \sim \mathbb{Z}_q^n \) means that \( x \) is uniform randomly chosen from \( \mathbb{Z}_q^n \). We also write \( \| f \|_2 = \sqrt{\langle f, f \rangle} \) as usual.

The following Efron–Stein decomposition theorem is well-known; see [141].

**Theorem 16.2.1.** Any \( f : \mathbb{Z}_q^n \rightarrow \mathbb{R} \) can be uniquely decomposed as a sum of functions
\[
f(x) = \sum_{S \subseteq [n]} f^S(x),
\]
where

- \( f^S(x) \) depends only on \( x_S = (x_i, i \in S) \),
- for every \( S \subseteq [n] \), for every \( S' \) such that \( S \setminus S' \neq \emptyset \), and for every \( y \in \mathbb{Z}_q^n \), it holds that
  \[
  \mathbb{E}_{x \sim \mathbb{Z}_q^n} \left[ f^S(x) | x_{S'} = y_{S'} \right] = 0.
  \]

**Definition 16.2.2 (Influences).** For functions \( f : \mathbb{Z}_q^n \rightarrow \mathbb{R} \), define the influence of the \( i \)-th coordinate on \( f \) to be
\[
\text{Inf}_i(f) = \sum_{S \ni i} \| f^S \|_2^2,
\]
where \( \| f^S \|_2^2 = \mathbb{E}_x [f^S(x)^2] \). For functions \( f : \mathbb{Z}_q^n \rightarrow \Delta_m \), let
\[
\text{Inf}_i(f) = \sum_{a \in \mathbb{Z}_m} \text{Inf}_i(f_a),
\]
where \( f_a(x) = f(x)_a, \forall x \in \mathbb{Z}_q^n \).

**Definition 16.2.3 (Noise operators).** For functions \( f : \mathbb{Z}_q^n \rightarrow \mathbb{R} \), define the noise operator \( T_{1-\eta} \) to be
\[
T_{1-\eta}f(x) = \mathbb{E}_{y \sim 1-\eta^x} [f(y)].
\]
For functions \( f : \mathbb{Z}_q^n \rightarrow \Delta_m \), let \( T_{1-\eta} \) be the noise operator so that \((T_{1-\eta}f)_a = T_{1-\eta}(f_a), \forall a \in \mathbb{Z}_q^n \).
Definition 16.2.4 (Noisy-influences). For functions $f: \mathbb{Z}_q^n \to \mathbb{R}$ and functions $f: \mathbb{Z}_q^n \to \Delta_m$, define the $(1 - \eta)$-noisy-influence of the $i$-th coordinate of $f$ to be

$$\text{Inf}_i^{(1-\eta)}(f) = \text{Inf}_i(T_{1-\eta}f).$$

Fact 16.2.5. For functions $f: \mathbb{Z}_q^n \to \Delta_m$, we have

$$\sum_{i \in [n]} \text{Inf}_i^{(1-\eta)}(f) \leq 1/\eta.$$

Proposition 16.2.6. Let $f^{(1)}, \ldots, f^{(t)}$ be a collection of functions $\mathbb{Z}_q^n \to \mathbb{R}^m$. Then

$$\text{Inf}_i^{(1-\eta)} \left[ \text{avg}_{k \in [t]} \{ f^{(k)} \} \right] \leq \text{avg}_{k \in [t]} \left\{ \text{Inf}_i^{(1-\eta)}[f^{(k)}] \right\}.$$

Here for any $c_1, c_2, \ldots c_t \in \mathbb{R}$ (or $\mathbb{R}^m$), we use the notation $\text{avg}(c_1, \ldots, c_t)$ to denote their average:

$$\sum_{i=1}^t c_i / t.$$

Both of Fact 16.2.5 and Proposition 16.2.6 are easy to verify by the definition of Noisy-influences. The proofs of the facts can be found in, e.g., [180] and [181].

Definition 16.2.7 (Noise stability). For functions $f: \mathbb{Z}_q^n \to \mathbb{R}$, define its stability against $\epsilon$ noise to be

$$\text{Stab}_{1-\epsilon}[f] = \mathbb{E}_{x \sim \mathbb{Z}_q^n, y \sim 1-\epsilon x} [f(x)f(y)].$$

One tool we need is the Majority Is Stablest Theorem from [175]. (We state here a version using a small noisy-influences assumption rather than a small “low-degree influences” assumption; see, e.g., Theorem 3.2 in [190] for a sketch of the small modification to [175] needed.)

Theorem 16.2.8. For every function $f: \mathbb{Z}_q^n \to [0, 1]$ such that $\text{Inf}_i^{(1-\eta)}[f] \leq \tau$ for all $i \in [n]$, Let $\mu = \mathbb{E}[f]$. Then for any $0 < \epsilon < 1$,

$$\text{Stab}_{1-\epsilon}[f] \leq \Lambda_{1-\epsilon}(\mu) + e(\tau, q, \eta).$$

Here if we fix $\eta, q$, $e(\tau, \eta, q)$ goes to 0 when $\tau$ goes to 0.

In the above theorem, the quantity $\Lambda_{1-\epsilon}(\mu)$ is defined to be $\text{Pr}[x, y \leq t]$ when $(x, y)$ are joint standard Gaussians with covariance $1 - \epsilon$ and $t$ is defined by $\text{Pr}[x \leq t] = \mu$. 262
16.3 Review of proofs of $\text{Max}\Gamma^{-2}\text{-Lin}(\mathbb{Z}_q)$ and $\text{Max}\Gamma^{-3}\text{-Lin}(\mathbb{Z}_q)$ hardness

As mentioned, we prove \textbf{Theorem 16.1.2} and \textbf{Theorem 16.1.3} by fairly easy modifications of the known hardness results for $\text{Max}\Gamma^{-2}\text{-Lin}(\mathbb{Z}_q)$ and $\text{Max}\Gamma^{-3}\text{-Lin}(\mathbb{Z}_q)$, due respectively to Khot–Kindler–Mossel–O’Donnell \[141\] and Håstad \[116\]. In this section, we review several places in the two proofs that are related to our modifications. We also assume the reader’s familiarity with these works.

16.3.1 $\text{Max}\Gamma^{-2}\text{-Lin}$

Let us begin with $\text{Max}\Gamma^{-2}\text{-Lin}$. As shown in \[141\], to prove UG-hardness of $(1 - \epsilon, \delta)$-approximating $\text{Max}\Gamma^{-2}\text{-Lin}(\mathbb{Z}_q)$ for constant $\kappa$ and $q$, where $\delta = q\Lambda_{1-\epsilon}(1/q) + \kappa$, it suffices to construct a “Dictator vs. Small Low-Degree-Influences Test” (or, Dictator Test for short) for functions $f : \mathbb{Z}_q^L \rightarrow \Delta_q$ which uses $\Gamma$-2-Lin constraints and has completeness $1 - \epsilon$, soundness $\delta$. We recall the definition of Dictator Test as follows.

Generally speaking, a $1 - \epsilon$ vs. $\delta$ Dictator Test for functions $f : \mathbb{Z}_q^L \rightarrow \Delta_q$ is defined by a distribution over $\Gamma$-2-Lin constraints (over the entries of $f$). We say $f$ passes the test when a random constraint (from the distribution) is satisfied by $f$. At the completeness side, all the $L$ dictators (i.e., $f(x_i) = x_i$ for some $i \in L$) pass the test with probability at least $1 - \epsilon$. At the soundness side, all functions with small noisy-influences (on all coordinates) pass the test with probability at most $\delta$. KKMO indeed needs to construct a Dictator Test for functions of distributions, i.e., for $f : \mathbb{Z}_q^L \rightarrow \Delta_q$, where whenever the test refers an entry $f(x)$ for an element in $\mathbb{Z}_q$, it randomly samples an element from the distribution $f(x)$.

The Dictator Test used by KKMO is indeed a noise-stability test. Intuitively, dictator functions have high noise stability, while functions far from dictators have low noise stability. Note that this intuition is true only for balanced functions, as constant functions are far from dictators but very noise stable. Therefore, KKMO used the “folding” trick (which was introduced in \[116\]) to ensure that $f$ outputs $1, 2, \ldots, q$ with the same probability.

16.3.2 $\text{Max}\Gamma^{-3}\text{-Lin}$

Let us move on to $\text{Max}\Gamma^{-3}\text{-Lin}$ and our proof of \textbf{Theorem 16.1.3} Håstad essentially showed that to prove NP-hardness of $(1 - \epsilon, 1/q + \kappa)$-approximating $\text{Max}\Gamma^{-3}\text{-Lin}(\mathbb{Z}_q)$
for constant $q$, it suffices to construct a “Matching-Dictator Test” on two functions for $f : \mathbb{Z}_q^K \to \mathbb{Z}_q$, $g : \mathbb{Z}_q^L \to \mathbb{Z}_q$ and $\pi : L \to K$. The test is defined by a distribution over $x \in \mathbb{Z}_q^K$, $y \in \mathbb{Z}_q^L$, $z \in \mathbb{Z}_q^L$ with the check $f(x) + g(y) - g(z) = c \mod q$. Håstad’s Test has the following completeness and soundness promises:

- If $f(x) = x_i$ and $g(y) = y_j$ such that $\pi(i) = j$, then $f$ and $g$ passes with probability $1 - \epsilon$.

- If $f$ and $g$ passes the test with probability $1/q + \kappa$, then there is a randomized procedure that “decodes” $f$ into a coordinate $i \in L$ and $g$ into a coordinate $j \in K$ such that $\pi(i) = j$ with constant probability depending only on $q, \epsilon, \kappa$ and independent of $L, K, \pi$. Also note that the decoding processes for $f$ and $g$ should be independent from each other.

Håstad constructed the following test: choose $x \in \mathbb{Z}_q^K$ and $y \in \mathbb{Z}_q^L$ uniformly and independently, define $z \in \mathbb{Z}_q^L$ to be $z = y \oplus_q (x \circ \pi)$, where $(x \circ \pi)_i := x_{\pi(i)}$, let $z'$ be $(1 - \epsilon)$-correlated to $z$, and test $f(x) \oplus_q g(y) = g(z')$. Such a test does not work when $f \equiv 0$; thus Håstad introduced and used his method of folding (which was also used [141]) to ensure that $f$ outputs $1, 2, \ldots, q$ with equal probability.

### 16.4 Overview of our proofs

As mentioned, we obtain [Theorem 16.1.2](#) and [Theorem 16.1.3](#) by modifying the KKMO [141] and Håstad [116] proofs. In this section we describe the idea of the modifications.

#### 16.4.1 Active folding

The usual folding trick [116] enforces that $f$ is balanced by replacing references to $f(x_1, \ldots, x_L)$ with references to $f(x_1 \oplus_q x_{j^*}, x_2 \oplus_q x_{j^*}, \ldots, x_L \oplus_q x_{j^*}) \oplus_q (\neg x_{j^*})$ for some arbitrary $j^* \in [L]$. (I.e., the reduction only uses $q^{L-1}$ variables to represent $f$ as opposed to $q^L$. Note that this makes the test’s constraints of the form $f(x) \oplus_q b = f(x') \oplus_q b'$, but this is still of $\Gamma$-2-LIN type. We call this trick static folding.

Let us explain the alternative to “static folding” which we call active folding. Active folding is nothing more than building the folding directly into the test. We feel that this is slightly more natural than static folding, and as we will see it proves to be more flexible.
In the KKMO context of \( \text{MAX-2-LIN}(\mathbb{Z}_q) \), active folding means that the test additionally chooses \( c, c' \sim \mathbb{Z}_q \) uniformly and independently, and then it checks the \( \Gamma-2-\text{LIN} \) constraint

\[
f(x \oplus_q (c, \ldots, c)) \oplus_q (-c) = f(x' \oplus_q (c', \ldots, c')) \oplus_q (-c')
\]

rather than \( f(x) = f(x') \). To analyze the KKMO test with active folding, first note that completeness does not change. As for the soundness analysis, given a function \( f : \mathbb{Z}_q^L \rightarrow \Delta_q \) we introduce \( \tilde{f} : \mathbb{Z}_q^L \rightarrow \Delta_q \) defined by

\[
\tilde{f}(x)_a = \mathbb{E}_{c \sim \mathbb{Z}_q} [f(x \oplus_q (c, \ldots, c))_a \oplus_q c]. \tag{16.1}
\]

Then the probability \( f \) satisfies the test with active folding is precisely the probability that \( \tilde{f} \) satisfies the \( \tilde{f}(x) = \tilde{f}(x') \) test (in the sense of randomized functions), namely \( \text{Stab}_{1-\epsilon}[\tilde{f}] \). We can now proceed with the KKMO analysis; the key is that we still have \( \mathbb{E}[\tilde{f}_a] = 1/q \) for all \( a \in \mathbb{Z}_q \). To see this, take \( Q = q \) in the following lemma:

**Lemma 16.4.1.** Let \( f : \mathbb{Z}_q^L \rightarrow \Delta_q \) and suppose \( \tilde{f} : \mathbb{Z}_q^L \rightarrow \Delta_q \) is defined as in (16.1). Then \( \mathbb{E}[\tilde{f}_a] = 1/q \) for all \( a \in \mathbb{Z}_q \).

**Proof.** We have

\[
\mathbb{E}[\tilde{f}_a] = \mathbb{E}_{x \sim \mathbb{Z}_q^L, c \sim \mathbb{Z}_q} [f(x \oplus_q (c, \ldots, c))_a \oplus_q c].
\]

Write \( \tilde{x} = x \oplus_q (c, \ldots, c) \in \mathbb{Z}_q^K \). The distribution of \( \tilde{x} | (c = c) \) is uniform on \( \mathbb{Z}_q^K \) for every \( c \). In other words, \( \tilde{x} \) and \( c \) are independent. Thus

\[
\mathbb{E}[\tilde{f}_a] = \mathbb{E}_{\tilde{x}} \left[ \mathbb{E}_c [f(\tilde{x})_a \oplus_q c] \right] = \mathbb{E}_{\tilde{x}} \left[ (1/q) \sum_{b \in \mathbb{Z}_q} [f(\tilde{x})_b] \right] = \mathbb{E}_{\tilde{x}} [1/q] = 1/q. \tag{16.2}
\]

### 16.4.2 Modifying the KKMO proof

We now describe how to obtain Theorem 16.1.2. Let us first ask: Why does the KKMO reduction (with active folding) not prove Theorem 16.1.2 already? The soundness statement of Theorem 16.1.2 would hold since it is over \( \mathbb{Z}_q \). The problem is in the completeness statement: a dictator \( f : \mathbb{Z}_q^L \rightarrow \mathbb{Z}_q, f(x) = x_i \) does not satisfy the the KKMO test with probability close to 1. The reason is that folding may introduce wrap-around in \( \mathbb{Z}_q \). More specifically (and ignoring the \( \epsilon \) noise), the KKMO test with active folding will check

\[
(x_i + c \mod q) - c = (x_i + c' \mod q) - c'
\]

over the integers, and this is only satisfied if both \( x_i + c \) and \( x_i + c' \) wrap around, or neither does: probability 1/2. (The situation with static folding is similar.)
Sketch of a first fix. There is a simple way to somewhat fix the completeness: choose $c$ and $c'$ from a range smaller than $\{0, 1, \ldots, q-1\}$. E.g., if we choose $c$ and $c'$ independently and uniformly in $\{0, 1, \ldots, \lfloor q/t \rfloor \}$, then we get wrap-around in $x_i + c$ with probability at most $1/t$. Hence the dictator $f(x) = x_i$ will satisfy the test (16.2) over $\mathbb{Z}$ with probability at least $1 - 2/t$, which we can make close to 1 by taking $t$ large. Now how does this restricted folding affect the soundness analysis? If we redefine the folded function $\tilde{f}$ appropriately, it is not hard to show that we will have $\mathbf{E}[\tilde{f}_a] \leq (t/q)$ for all $a$. We could then proceed with the KKMO analysis applied to $\tilde{f}$ and obtain soundness $q\Lambda_{1-\epsilon}(t/q)$. Choosing, say, $t = \log q$ would achieve a good completeness versus soundness tradeoff; roughly $1 - \epsilon$ versus $O(1/q)^{\epsilon/(2-\epsilon)}$.

A better fix. A slight twist on this idea actually gives the optimal completeness versus soundness tradeoff. Instead of restricting the range of the folding, we simply enlarge the domain of $f$. Specifically, let $\gamma > 0$ be any small constant and define $Q = \lceil q/\gamma \rceil$. To prove Theorem 16.1.2, we run the KKMO reduction with functions $f$ whose domain is $\mathbb{Z}_Q^L$. We still active folding with $c \in \mathbb{Z}_Q$. In other words, the test chooses $x, x'$ to be $(1-\epsilon)$-correlated strings in $\mathbb{Z}_Q^L$, chooses $c, c' \in \mathbb{Z}_Q$ uniformly and independently, and outputs the constraint $f(x \oplus_{\mathbb{Z}} (c, \ldots, c)) - c = f(x' \oplus_{\mathbb{Z}} (c', \ldots, c')) - c'$. Note that this is a $q$-Bounded-$\Gamma$-2-Lin constraint. As the ‘wrap-around” probability is $q/Q \leq \gamma$, we have completeness over $\mathbb{Z}$ of at least $1 - \epsilon - \gamma$. As for the soundness over $\mathbb{Z}_Q$, we now need to consider functions $f : \mathbb{Z}_Q^L \rightarrow \Delta_q$. If we introduce the folded function $\tilde{f} : \mathbb{Z}_Q^L \rightarrow \Delta_q$, as in (16.1), the probability $f$ passes the test over $\mathbb{Z}_Q$ is again $\operatorname{Stab}_{1-\epsilon}(\tilde{f})$, and we still have $\mathbf{E}[\tilde{f}_a] = 1/q$ by Lemma 16.4.1. Hence the soundness analysis for Theorem 16.1.2 becomes essentially identical to the soundness analysis for KKMO with active folding. The only tiny difference is that we need to apply the Majority Is Stablest Theorem with domain $\mathbb{Z}_Q^L$ rather than $\mathbb{Z}_Q^L$. But $Q$ is still a constant since $\gamma$ and $q$ are; hence we obtain the claimed $1 - \epsilon - \gamma$ completeness over $\mathbb{Z}$ and $q\Lambda_{1-\epsilon}(1/q)$ soundness over $\mathbb{Z}_Q$.

### 16.4.3 Modifying the Håstad proof

The modification to Håstad’s test needed to obtain Theorem 16.1.3 is similar. If one carries out Håstad’s proof using the Efron–Stein decomposition rather than harmonic analysis over $\mathbb{Z}_Q$, one sees that the soundness relies entirely on $\mathbf{E}[f_a] = 1/q$ for all $a \in \mathbb{Z}_Q$. Thus we only need to apply folding to $f$. Let us examine the Håstad $\Gamma$-3-Lin test on $f : \mathbb{Z}_Q^K \rightarrow \mathbb{Z}_Q$, $g : \mathbb{Z}_Q^L \rightarrow \mathbb{Z}_Q$, and $\pi : L \rightarrow K$. We will use active folding on $f$, and for simplicity of this discussion ignore the $\epsilon$-noise. The test chooses $x \sim \mathbb{Z}_Q^K$ and $y \sim \mathbb{Z}_Q^L$ uniformly and
Figure 16.1: Test $\mathcal{T}$ with parameters $\epsilon, \gamma, q$ for functions on $\mathbb{Z}_Q^K$.

- Choose $x, x' \sim \mathbb{Z}_Q^K$ to be a pair of $(1 - \epsilon)$-correlated random strings.
- Choose $c, c' \sim [q]$ independently and uniformly.
- Define $\tilde{x} = x \oplus_Q (c, c, \ldots, c)$, and define $\tilde{x}' = x' \oplus_Q (c', c', \ldots, c')$.
- Test the constraint $f(\tilde{x}) - c = f(\tilde{x}') - c'$.

Independently, defines $z \in \mathbb{Z}_q^L$ by $z = y \oplus_q (x \circ \pi)$ (again, $(x \circ \pi)_i := x_{\pi(i)}$), chooses $c \sim \mathbb{Z}_q$ uniformly, and finally checks the $\Gamma$-3-LIN constraint

$$f(x \oplus_q (c, \ldots, c)) - c + g(y) = g(z).$$

Again, if we simply use this reduction in an attempt to prove the soundness is fine but the completeness over $\mathbb{Z}$ is a problem due to wrap-around. Indeed, there are two possibilities for wrap-around here: in $x_i + c$ and in $y_j + x_{\pi(j)}$. We mitigate this with the same idea used for $\text{MAX}\Gamma$-2-LIN. Given constants $\epsilon$ and $q$ we define constants $Q = \lceil q/\epsilon \rceil$ and $Q = \lceil Q/\epsilon \rceil$. We enlarge $f$’s domain to $\mathbb{Z}_Q^K$ and $g$’s domain to $\mathbb{Z}_L^L$. We continue to fold $f$ using $c \sim \mathbb{Z}_q$. Now the two possibilities for wrap-around occur with probability at most $\epsilon$ each and hence the completeness over $\mathbb{Z}$ is $1 - O(\epsilon)$. Defining $\tilde{f} : \mathbb{Z}_Q^K \rightarrow \Delta_q$ as in [16.1], we again have $E[\tilde{f}_a] = 1/q$ for each $a \in \mathbb{Z}_q$ and can carry out the (Efron–Stein-style) Hästad soundness analysis, obtaining soundness $1/q + \kappa$ over $\mathbb{Z}_q$.

16.5 Dictator Test details

16.5.1 Dictator Test for $\text{MAX}\Gamma$-2-LIN

Given constants $\epsilon, \gamma, \kappa > 0$ and $q, K \in \mathbb{Z}^+$, let $Q = \lceil q/\gamma \rceil$. We define the Dictator Test $\mathcal{T}$ for functions $f$ with domain $\mathbb{Z}_Q^K$ as in Figure 16.1. Let $\text{val}_{\mathbb{Z}_Q}^f(f)$ be the probability that $f$ passes the test, and let $\text{val}_{\mathbb{Z}_Q}^{\Gamma}(f)$ be the probability that $f$ passes the test over $\mathbb{Z}_Q$.

**Theorem 16.5.1.** There exists $\tau, \eta > 0$ such that $\mathcal{T}$ is a $q$-BOUNDED-$\Gamma$-2-LIN test with following properties:
• (Completeness.) Each of the $K$ dictators $f : \mathbb{Z}_Q^K \rightarrow \mathbb{Z}$ has $\text{val}^T_Z(f) \geq 1 - \epsilon - \gamma$.

• (Soundness.) Let $f : \mathbb{Z}_Q^K \rightarrow \Delta_q$ and define $\tilde{f} : \mathbb{Z}_Q^K \rightarrow \Delta_q$ as in (16.1). Suppose that $\inf_{\eta}^{1-\eta} \{ \tilde{f} \} \leq \tau$ for all $i \in [K]$. Then $\text{val}^T_{\mathbb{Z}_q}(f) \leq q \Lambda_{1-\epsilon}(1/q) + \kappa$, where $\kappa = \kappa(\tau, Q, \eta) > 0$ can be made arbitrarily small by taking $\tau, \eta > 0$ sufficiently small.

Theorem 16.5.1 together with the following lemma proves Theorem 16.1.2.

**Lemma 16.5.2.** Theorem 16.5.1 implies Theorem 16.1.2.

Lemma 16.5.2 is implicit from [141], and is proved in Section 16.7.1.

**Proof of Theorem 16.5.1.** For the Completeness case, we need to analyze for a fixed $i \in [K]$ the probability that

$$(x_i \oplus_Q c) - c = (x'_i \oplus_Q c') - c'$$  \hspace{1cm} (16.3)

holds over $\mathbb{Z}$. We have $x_i = x'_i$ except with probability at most $\epsilon$, and $x_i \leq Q - q$ except with probability at most $q/Q \leq \gamma$. When both of these events occur, equation (16.3) holds. This proves the completeness.

As for the Soundness case, by Lemma 16.4.1 we have $\mu_a = E[f_a] = 1/q$ for each $a \in \mathbb{Z}_q$. By assumption we have $\inf_{\eta}^{1-\eta} \{ f_a \} \leq \inf_{\eta}^{1-\eta} \{ \tilde{f} \} \leq \tau$. Thus from Theorem 16.2.8 we obtain $\text{Stab}_{1-\epsilon}[f_a] \leq \Lambda_{1-\epsilon}(1/q) + e(\tau, Q, \eta)$ for each $a$. Summing this over $a \in \mathbb{Z}_q$ yields

$$\text{Stab}_{1-\epsilon}[\tilde{f}] \leq q \Lambda_{1-\epsilon}(1/q) + q \cdot e(\tau, Q, \eta).$$

The proof is completed by taking $\kappa = q \cdot e(\tau, Q, \eta)$, since $\text{Stab}_{1-\epsilon}[\tilde{f}] = \text{val}^T_{\mathbb{Z}_q}(f)$ by unrolling definitions.

16.5.2 Matching Dictator Test for $\text{MAX}_3-\text{LIN}$

Given constants $\epsilon, \kappa > 0$ and $q, L, K \in \mathbb{Z}$, let $Q = \lceil q/\epsilon \rceil$ and $Q = \lceil Q/\epsilon \rceil$. In Figure 16.2 we define the Matching Dictator Test $\mathcal{U}$ for function $f$ with domain $\mathbb{Z}_Q^K$, function $g$ with domain $\mathbb{Z}_Q^L$, and projection $\pi : L \rightarrow K$. Let $\text{val}^\mathcal{U}_Z(f, g)$ be the probability that $f, g$ pass the test, and let $\text{val}^\mathcal{U}_{\mathbb{Z}_q}(f, g)$ be the probability that $f, g$ pass the test over $\mathbb{Z}_q$.

**Theorem 16.5.3.** $\mathcal{U}$ is a $q$-BOUNDED-$\Gamma^3$-LIN test satisfying:
• Choose $x \sim \mathbb{Z}_Q^K$. $y \sim \mathbb{Z}_Q^L$ uniformly and independently.
• Define $z \in \mathbb{Z}_Q^L$ by $z = y \oplus_Q (x \circ \pi)$.
• Choose $c \sim \mathbb{Z}_q$ uniformly and define $\tilde{x} \in \mathbb{Z}_Q^K$ by $\tilde{x} = x \oplus_Q (c, c, \ldots, c)$.
• Let $x' \in \mathbb{Z}_Q^K$ be $(1 - \epsilon)$-correlated to $\tilde{x}$, let $y' \in \mathbb{Z}_Q^L$ be $(1 - \epsilon)$-correlated to $y$, and let $z' \in \mathbb{Z}_Q^L$ be $(1 - \epsilon)$-correlated to $z$.
• Test the constraint $f(x') - c + g(y') = g(z')$.

• (Completeness.) If $f \colon \mathbb{Z}_Q^K \to \mathbb{Z}$ and $g \colon \mathbb{Z}_Q^L \to \mathbb{Z}$ are matching dictators — i.e., $f(x) = x_{\pi(j)}$ and $g(y) = y_j$ for some $j \in [L]$ — then $\text{val}_U(f, g) \geq 1 - 5\epsilon$.

• (Soundness.) Let $f \colon \mathbb{Z}_Q^K \to \mathbb{Z}_q$, $g \colon \mathbb{Z}_Q^L \to \mathbb{Z}_q$ and define $\tilde{f} : \mathbb{Z}_Q^K \to \Delta_q$ as in (16.1). Suppose that $\text{val}_U(f, g) \geq 1/q + \kappa$, then there is a randomized “decoding procedure” $D$ which decodes $g$ to a coordinate $D(g) \in [L]$ and $f$ to a coordinate $D(f) \in [K]$ such that $\pi(D(g)) = D(f)$ with at least a constant probability $\zeta = \zeta(q, \epsilon, \kappa)$ independent of $\pi, L, K$.

Theorem 16.5.3 together with the following lemma proves Theorem 16.1.3.

Lemma 16.5.4. Theorem 16.5.3 implies Theorem 16.1.3.

Lemma 16.5.4 is proved in Section 16.7.2.

Proof of Theorem 16.5.3 Define $\tilde{f} : \mathbb{Z}_Q^K \to \Delta_q$ as in (16.1). For the completeness case, we need to analyze for a fixed $j \in [L]$ the probability that

$$x'_{\pi(j)} - c + y'_j = z'_j$$  \hspace{1cm} (16.4)$$

holds over $\mathbb{Z}$. Except with probability at most $3\epsilon$ we have all of

$$x'_{\pi(j)} = \tilde{x}_{\pi(j)} = x_{\pi(j)} \oplus_Q c,$$

$$y'_j = y_j, \quad z'_j = z_j = x_{\pi(j)} \oplus_Q y_j.$$
Except with probability at most \(q/Q \leq \epsilon\) we have \(x_{\pi(j)} \leq Q - q\), in which case \(x_{\pi(j)} \oplus_q c\) equals \(x_{\pi(j)} + c\). Except with probability at most \(Q/Q \leq \epsilon\) we have \(y_j \leq Q - q\), in which case \(x_{\pi(j)} \oplus_q y_j = x_{\pi(j)} + y_j\). Thus when all five events occur, equation (16.4) indeed holds over \(\mathbb{Z}\).

As for the soundness case, write \(f' = T_{1-\epsilon} \tilde{f}\) and \(g' = T_{1-\epsilon} g\), where we think of \(g\) as \(g : \mathbb{Z}_Q \rightarrow \Delta_q\). By unrolling definitions we have

\[
\text{val}_{\mathbb{Z}_q}^f(f, g) = \sum_{a,b \in \mathbb{Z}_q} E_{x,y,z}[f'_a(x)g'_b(y)g_{a \oplus_q b}(z)].
\]

Write \(\mu_a = E[f'_a(x)]\). Thus \(\mu_a = E[\tilde{f}_a] = 1/q\), by Lemma 16.4.1. We conclude that

\[
\text{val}_{\mathbb{Z}_q}^f(f, g) = \sum_{a,b \in \mathbb{Z}_q} E[(f'_a(x) - \mu_a)g'_b(y)g_{a \oplus_q b}(z)] + (1/q) \sum_{a,b \in \mathbb{Z}_q} E[g'_b(y)g_{a \oplus_q b}(z)].
\]

The second term above is

\[
(1/q) \sum_{a,b \in \mathbb{Z}_q} E[g'_b(y)g_{a \oplus_q b}(z)] = (1/q) E[(\sum_c g'_c(y)) \cdot (\sum_c g'_c(z))] = (1/q) E[1 \cdot 1] = 1/q,
\]

since \(g'\) is \(\Delta_q\)-valued. Thus to complete the proof it remains to show that if

\[
\sum_{a,b \in \mathbb{Z}_q} E[(f'_a(x) - \mu_a)g'_b(y)g_{a \oplus_q b}(z)]
\]

is at least \(\kappa > 0\) then we can suitably decode \(\tilde{f}\) and \(g\). Let us now apply the Efron–Stein decomposition to \(f'\) and \(g'\) with respect to the uniform distributions on their domains. Given \(S \subseteq [K]\), \(T \subseteq [L]\), for simplicity we write

\[
F_a^S = f'_a(x), \quad G_b^T = g'_b(y), \quad H_{a+b}^T = g'_{a \oplus_q b}(z).
\]

Thus

\[
\begin{align*}
\text{Eq. (16.5)} &= \sum_{a,b \in \mathbb{Z}_q} E\left[ \left( \sum_{\emptyset \neq S \subseteq [K]} F_a^S \right) \left( \sum_{T \subseteq [L]} G_b^T \right) \left( \sum_{U \subseteq [L]} H_{a+b}^U \right) \right] \\
&= \sum_{a,b \in \mathbb{Z}_q} \sum_{\emptyset \neq S \subseteq [K]} \sum_{T,U \subseteq [L]} E[F_a^S G_b^T H_{a+b}^U].
\end{align*}
\]

Let us simplify the above. We have \(E[F_a^S G_b^T H_{a+b}^U] = E[F_a^S \cdot E[G_b^T | H_{a+b}^U, x]\]. Note that even if we condition on \(x\), the marginals on \(y\) and \(z\) are uniform on \(\mathbb{Z}_q\). It follows
from the properties of the Efron–Stein decomposition that \( E[G_b^T H_{a+b}^U | x] \) is always 0 if \( T \neq U \). Thus

\[
E[16.5] = \sum_{a,b \in \mathbb{Z}_q} \sum_{\emptyset \neq S \subseteq \{K\}}^{U \subseteq \{L\}} E[F_a^S G_b^U H_{a+b}^U].
\]

Similarly, conditioned on the \( U \)-coordinates of \( y \) and \( z \), the coordinates of \( x \) outside \( \pi(U) \) are independent and uniform on \( \mathbb{Z}_r \). Hence \( E[F_a^S G_b^U H_{a+b}^U] = 0 \) if \( S \nsubseteq \pi(U) \). We conclude that

\[
E[16.5] = \sum_{a,b \in \mathbb{Z}_q} \sum_{U \neq \emptyset}^{\emptyset \neq S \subseteq \pi(U)} E[F_a^{\leq \pi(U)} G_b^U H_{a+b}^U] = \sum_{a,b \in \mathbb{Z}_q} \sum_{U \neq \emptyset} E[F_a^{\leq \pi(U)} G_b^U H_{a+b}^U],
\]

where we defined \( F_a^{\leq \pi(U)} = \sum_{\emptyset \neq S \subseteq \pi(U)} F_a^S \). Shifting the sum over \( a \) and \( b \) to the inside we obtain

\[
E[16.5] = \sum_{U \neq \emptyset} E\left[ \sum_{a,b \in \mathbb{Z}_q} F_a^{\leq \pi(U)} G_b^U H_{a+b}^U \right] \leq \sum_{U \neq \emptyset} E\left[ \sqrt{\sum_{a,b} (F_a^{\leq \pi(U)})^2 (G_b^U)^2} \sqrt{\sum_{a,b} (H_{a+b}^U)^2} \right],
\]

having used Cauchy-Schwarz. We can think of, e.g., \( (G_0^U, \ldots, G_{q-1}^U) \) as a vector in \( \mathbb{R}^q \); writing \( \|G^U\| \) for the Euclidean length of this vector (and similarly for \( F \) and \( H \)), the right side above is precisely \( \sqrt{q} \sum_{U \neq \emptyset} E[\|F^{\leq \pi(U)}\| \cdot \|G^U\| \cdot \|H^U\|] \). Thus

\[
E[16.5] \leq \sqrt{q} \sum_{U \neq \emptyset} E[\|F^{\leq \pi(U)}\| \cdot \|G^U\| \cdot \|H^U\|] \leq \sqrt{q} \sum_{U \neq \emptyset} \sqrt{E[\|F^{\leq \pi(U)}\|^2 \|G^U\|^2 \|H^U\|^2]},
\]

using Cauchy-Schwarz again. Now \( F^{\leq \pi(U)} \) depends only on \( x \) and \( G^U \) depends only on \( y \); hence they are independent. Further, since \( y \) and \( z \) have the same distribution (though they are not independent), the same is true of \( G^U \) and \( H^U \). Hence

\[
E[16.5] \leq \sqrt{q} \sum_{U \neq \emptyset} \sqrt{E[\|F^{\leq \pi(U)}\|^2] E[\|G^U\|^2]}
\]

\[
\leq \sqrt{q} \sqrt{\sum_{U \neq \emptyset} E[\|F^{\leq \pi(U)}\|^2] E[\|G^U\|^2] \sum_{U \neq \emptyset} E[\|G^U\|^2]},
\]

using Cauchy-Schwarz again. By (generalized) Parseval, \( \sum_{U \neq \emptyset} E[\|G^U\|^2] \leq \sum_U E[\|G^U\|^2] = E[\|G\|^2] \leq 1 \), since \( G \) takes values in \( \Delta_q \). Thus we finally conclude

\[
E[16.5] \leq \sqrt{q} \sqrt{\sum_{U \neq \emptyset} E[\|F^{\leq \pi(U)}\|^2] E[\|G^U\|^2]}
\]
Lemma 16.6.1. Given a \( q \)-Bounded-Max\( \Gamma \)-\( k \)-Lin instance and positive integer \( m \geq q \):
• When $k = 2$, $\text{opt}_Z(I), \text{opt}_R(I), \text{opt}_{Z_m}(I) \leq 4 \cdot \text{opt}_{Z_q}(I)$.

• When $k = 3$, $\text{opt}_Z(I), \text{opt}_R(I), \text{opt}_{Z_m}(I) \leq 8 \cdot \text{opt}_{Z_m}(I)$.

Proof. It is obvious that the $\text{opt}_Z$ is a lower bound for $\text{opt}_{Z_q}$. It suffice then to show how to convert a $\delta$-good assignment over $Z_m$ and $R$ to a $\Omega(\delta)$-good assignment over $Z$.

First we show the conversion from an assignment over $R$ to $Z$. For case of $k = 3$, as is noted in [108], suppose one has an $\delta$-good real assignment to a system of equations of the form $x_{i_1} - x_{i_2} + x_{i_3} = b$, $b \in Z$. If one randomly rounds each variable up or down to an integer, every formerly satisfied equation has probability at least $1/8$ of remaining satisfied. Hence there must exist a $\delta/8$-good integer assignment. For the case of $k = 2$, the reduction from $\text{MAX}\Gamma-2\text{-LIN}(Z)$ to $\text{MAX}\Gamma-2\text{-LIN}(R)$ is even easier and incurs no loss: given a $\delta$-good real assignment, simply dropping the fractional parts yields a $\delta$-good integer assignment.

Next we show the conversion from assignment over $Z_m$ to $Z_q$. First let us consider the case of $k = 3$. Suppose one has an $\delta$-good assignment $A : x_i \to Z_m$ to a system of equations of the form

$$x_{i_1} - x_{i_2} + x_{i_3} = b \mod m.$$

Then we know that if $A(x_{i_1}) - A(x_{i_2}) + A(x_{i_3}) = b \mod m$. Notice that $|b| \leq q \leq m$, we must have that $A(x_{i_1}) - A(x_{i_2}) + A(x_{i_3}) \in \{b, b - m, b + m\}$ when the assignment is evaluated over $Z$. If we define assignments $A_1(x_i) = A(x_i) - m$ and $A_2(x_i) = A(x_i) + m$ for every $x_i$. Then it is easy to verify that the best assignment among $A, A_1, A_2$ will give a $\delta/3$-good assignment. Essentially, every equation over $Z_m$ satisfiable by $A$ must also be satisfiable by one of $A, A_1, A_2$ over $Z$.

As for the case $k = 2$, we know that for a $\delta$-good assignment $A$ over $Z_m$, we know that if $A(x_{i_1}) - A(x_{i_2}) = b \mod m$, then $A(x_{i_1}) - A(x_{i_2}) \in \{b - m, b\}$ when evaluated over $Z$. Therefore, we can randomly set $A'(x_i)$ to be $A(x_i) - m$ or $A(x_i)$. Then we know that $A'$ is at least a $\delta/4$-good assignment over $Z$.

It is not too hard to see that the proof technique also works for $m < q$; in particular, a $\delta$-good assignment for $q\text{-BOUND}-\text{MAX}\Gamma-k\text{-LIN}$ on $Z_m$ implies a $\Omega(\frac{\delta}{(2q/m)^r})$-good assignment on $Z_q$.\[\square\]

1In the usual case when the hard instances also have “bipartite” structure, it is not hard to make the loss only a factor of 2 rather than 8.
16.7 From Dictator Tests to hardness of approximation

16.7.1 Proof of Lemma 16.5.2

We start by defining UNIQUEGAMES and the Unique Games Conjecture.

**Definition 16.7.1 (UNIQUEGAMES).** A UNIQUEGAMES instance \( \mathcal{L}(G(U, V, E), \Sigma, \{\pi_e|e \in E\}) \) is a constraint satisfaction problem defined as follows. \( G(U, V, E) \) is a bipartite graph whose vertices represent variables and edges represent constraints. The goal is to assign to each vertex a label from the set \( \Sigma \). The constraint on an edge \( e = (u, v) \in E \), where \( u \in U, v \in V \), is described by a bijection \( \pi_e : \Sigma \rightarrow \Sigma \). A labeling \( \sigma : U \cup V \rightarrow \Sigma \) satisfies the constraint on edge \( e = (u, v) \) if and only if \( \pi_e(\sigma(v)) = \sigma(u) \). Let \( \text{opt}(U) \) denote the maximum fraction of constraints that can be satisfied by any labeling:

\[
\text{opt}(U) := \max_{L : U \cup V \rightarrow \Sigma} \frac{1}{|E|} \cdot |\{e \in E | L \text{ satisfies } e\}|.
\]

**Conjecture 2** (Unique Games Conjecture [136]). For every \( \gamma, \delta > 0 \), there exists a constant \( M = M(\gamma, \delta) \), such that given a UNIQUEGAMES instance \( \mathcal{L}(G(U, V, E), \Sigma, \{\pi_e|e \in E\}) \) with \( |\Sigma| = M \), it is NP-hard to distinguish between these two cases:

- **YES Case:** \( \text{opt}(L) \geq 1 - \gamma \).
- **NO Case:** \( \text{opt}(L) \leq \delta \).

By standard reductions, we can assume the bipartite graph \( G(U, V, E) \) is left-regular in the conjecture.

Now we are ready to prove **Lemma 16.5.2**

**Proof of Lemma 16.5.2** Given a UNIQUEGAMES instance \( \mathcal{L}(G(U, V, E), \Sigma, \{\pi_e|e \in E\}) \), and a Dictator Test \( T(\epsilon, \gamma, \kappa, q, K = |\Sigma|) \) described in the lemma statement, we build a \( q \)-BOUNDED-MAX\( \Gamma \)-2-LIN instance \( \mathcal{I} \) as follows. The variable set consists of all the entries of \( g_v : [Q]^\Sigma \rightarrow \mathbb{Z}, \forall v \in V \), which are supposed \( Q \)-ary Long Codes of the labels for \( v \in V \), where \( Q = q/\gamma \) is defined in the Dictator Test. The equations are placed by the following random process, where the probability of a equation being placed corresponds to its weight.

- Pick a random vertex \( u \) and two of its random neighbors of \( v, v' \in V \), let \( \pi = \pi_{(u,v)} \) and \( \pi' = \pi_{(u,v')} \).
• Run the Dictator Test \( T \) on an imaginary function \( f \) defined on \([Q]^\Sigma\), suppose \( T \) chooses to test \( f(x) - f(y) = b \).

• Place the equation \((g_v \circ \pi_1(x)) - (g_w \circ \pi_1(y)) = b\), where \((g \circ \pi_1(x)) := g(\pi_1(x))\).

**Completeness.** Suppose \( \text{opt}(L) \geq 1 - \gamma \), and \( \sigma \) is a labeling function satisfying \( 1 - \gamma \) fraction of the constraints. Let \( g_v \) be the Long Code for \( \sigma(v) \), i.e. let \( g_v(x) = x_{\sigma(v)} \) for each \( x \). According to the random process shown above, we pick a random equation in \( L \). With probability at least \( 1 - 2\gamma \), both of the constraints on \((u, v)\) and \((u, v')\) are satisfied by \( \sigma \). In this case, both \( g_v \circ \pi \) and \( g_{v'} \circ \pi' \) are the Long Code for \( \sigma(u) \), and \( g_v \circ \pi(x) - g_{v'} \circ \pi'(y) = b \) is satisfied with probability \( 1 - \epsilon - \gamma \) by the property of \( T \). In all, at least \( 1 - \epsilon - 3\gamma \) fraction (of weight) of the equations are satisfied.

**Soundness.** Suppose there is a set of functions \( g_v : [Q]^\Sigma \rightarrow \mathbb{Z}_q \) satisfying more than \( q\Lambda_1 - \epsilon(1/q) + \kappa \) fraction (of weight) of the equations over \( \mathbb{Z}_q \). Then there are at least \( \kappa / 2 \) fraction of vertices \( u \in U \) such that conditioned on \( u \) is picked in the first step of the random process shown above, the equation is satisfied over \( \mathbb{Z}_q \) with probability more than \( q\Lambda_1 - \epsilon(1/q) + \kappa / 2 \). We call such \( u \)’s “good”. For each \( u \), we define \( f_u : [Q]^\Sigma \rightarrow \Delta_q \) to be \( f_u = \text{avg}_{v \in (u, v) \in E} \{g_v \circ \pi(u, v)\} \). Since the equations generated after picking \( u \) are indeed a Dictator Test \( T \) running on \( f_u \) for good \( u \)'s, we have \( \text{val}_{\mathbb{Z}_q}^T(f_u) > q\Lambda_1 - \epsilon(1/q) + \kappa / 2 \). Therefore, for each good \( u \), there exists \( i = i_u \in \Sigma \), such that \( \text{Inf}_{i_u}^{(1-\eta)}[\widetilde{f}_u] > \tau \). Note that

\[
\widetilde{f}_u = \text{avg}_{v \in (u, v) \in E} \{g_v \circ \pi(u, v)\}.
\]

By Proposition 16.2.6, we have

\[
\tau < \text{Inf}_{i_u}^{(1-\eta)}[\widetilde{f}_u] = \text{Inf}_{i_u}^{(1-\eta)} \left[ \text{avg}_{v \in (u, v) \in E} \{g_v \circ \pi(u, v)\} \right] \leq \text{avg}_{v \in (u, v) \in E} \left\{ \text{Inf}_{i_u}^{(1-\eta)}[g_v \circ \pi(u, v)] \right\}.
\]

Therefore, for at \( \tau / 2 \) fraction of neighbors \( v \) of \( u \), there exists \( j = \pi(u, v)(i) \), such that \( \text{Inf}_{j}^{(1-\eta)}(g_v) > \tau / 2 \).

Let \( \sigma(u) = i_u \) if \( u \) is good. For each \( v \in V \), let \( \text{Cand}(v) = \{i : \text{Inf}_{i}^{(1-\eta)}(g_v) > \tau / 2\} \). By Fact 16.2.5, we have \( |\text{Cand}(v)| < 1/(\tau \eta) \). If \( \text{Cand}(v) \neq \emptyset \), let \( \sigma(v) \) be a random element in \( \text{Cand}(v) \). Now for a good \( u \), there are \( \tau / 2 \) fraction of neighbors \( v \) of \( u \) such that \( j = \pi(u, v)(\sigma(u)) \in \text{Cand}(v) \), therefore the edge \((u, v)\) is satisfied with probability \( 1/|\text{Cand}(v)| > \tau \eta \). It follows that \( \sigma \) satisfies more than \( (\kappa / 2) (\tau / 2) \tau \eta = \kappa \eta \tau^2 / 2 \) fraction of the constraints in expectation. Therefore there is a labeling satisfying more than \( \delta' = \kappa \eta \tau^2 / 2 \) fraction of the constraints. \( \square \)
16.7.2 Proof of Lemma 16.5.4

We start by defining Label Cover Games and introducing its hardness.

Definition 16.7.2 (Label Cover Games). A Label Cover Game $\mathcal{C}(G(U, V, E), [K], [L], \{\pi_e|e \in E\})$ is a constraint satisfaction problem defined as follows. $G(U, V, E)$ is a bipartite graph whose vertices represent variables and edges represent the constraints. The goal is to assign to each vertex in $U$ a label from the set $[K]$ and to each vertex in $V$ a label from the set $[L]$. The constraint on an edge $e = (u, v) \in E$ is described by a “projection” $\pi_e : [L] \to [K]$. The projection is onto. A labeling $\sigma : U \to [K], \sigma : V \to [L]$ satisfies the constraint on edge $e = (u, v)$ if and only if $\pi_e(\sigma(v)) = \sigma(u)$. Let $\text{opt}(\mathcal{C})$ denote the maximum fraction of constraints that can be satisfied by any labeling:

$$\text{opt}(\mathcal{C}) := \max_{\sigma : U \to [K], \sigma : V \to [L]} \frac{1}{|E|} \cdot |\{e \in E | \sigma \text{ satisfies } e\}|.$$

Theorem 16.7.3 (PCP Theorem + Raz’s Parallel Repetition Theorem [22, 20, 198]). There exists an absolute constant $c$ such that for every $\delta > 0$, given $\mathcal{C}(G(U, V, E), [K], [L], \{\pi_e|e \in E\})$, $K = (1/\delta)^C$, it is $\text{NP}$-hard to distinguish between:

- **YES Case**: $\text{opt}(\mathcal{C}) = 1$.
- **NO Case**: $\text{opt}(\mathcal{C}) = \delta$.

Now we are ready to prove Lemma 16.5.4

Proof of Lemma 16.5.4 Given a Label Cover Game instance $\mathcal{C}(G(U, V, E), [K], [L], \{\pi_e|e \in E\})$, and a Matching Dictator Test $\mathcal{U}(\epsilon, \kappa, q, L, K)$ described in the lemma statement, we build a $q$-BOUNDED-MAX-$3$-LIN instance $\mathcal{I}$ as follows. The variable set consists of all the entries of $f_u : [Q]^L \to \mathbb{Z}$ and $g_v : [Q]^K \to \mathbb{Z}$ for all $u \in U, v \in V$. The equations are the gathering of the Matching Dictator Tests $\mathcal{U}$ for $f_u, g_v$ with projection $\pi_{(u,v)}$ for all $(u, v) \in E$. The weights of the equations are normalised by a factor $1/|E|$.

Completeness. Suppose $\text{opt}(\mathcal{C}) = 1$, and $\sigma$ is a labeling function satisfying all the constraints. For all $u \in U, v \in V$, let $f_u$ and $g_v$ be the Long Codes for $\sigma(u), \sigma(v)$ respectively, i.e. let $f_u(x) = x_{\sigma(u)}; g_v(y) = y_{\sigma(v)}$. For each edge $(u, v) \in E$, the Matching Dictator Test $\mathcal{U}$ passes with probability at least $1 - \epsilon$. Therefore, at least $1 - \epsilon$ fraction (of weight) of the equations are satisfied.

Soundness. Suppose there is a set of functions $f_u : [Q]^K \to \mathbb{Z}_q, g_v : [Q]^L \to \mathbb{Z}_q$ satisfying more than $1/q + \kappa$ fraction (of weight) of the equations over $\mathbb{Z}_q$. By averaging
argument, for at least \( \kappa/2 \) fraction of the edges, the corresponding Matching Dictator Test passes with probability more than \( 1/q + \kappa/2 \). Call these edges “good edges”. For all \( u \in U, v \in V \), let \( \sigma(u) = D(f_u), \sigma(v) = D(g_v) \). For good edges \( e \in E \), the probability that \( e \) is satisfied by \( \sigma \) is at least \( \zeta = \zeta(q, \epsilon, \kappa) \). It follows that \( \sigma \) satisfies more than \( \zeta \kappa/2 \) fraction of the constraints in expectation. Therefore there is a labeling satisfying more than \( \delta' = \zeta \kappa/2 \) fraction of the constraints.

\( \square \)
Chapter 17

Hardness of approximating almost satisfiable MAXHORN3-SAT

17.1 Introduction

Schaefer proved long ago that there are only three non-trivial classes of Boolean constraint satisfaction problems (CSPs) for which satisfiability is polynomial time decidable [202]. These are LIN(2) (linear equations modulo 2), 2-SAT, and HORNSAT. The maximization versions of these problems (where the goal is to find an assignment satisfying the maximum number of constraints) are NP-Hard, and in fact APX-Hard, i.e., NP-Hard to approximate within some constant factor bounded away from 1. An interesting special case of the maximization version is the following problem of “finding almost-satisfying assignments”: Given an instance which is \((1 - \epsilon)\)-satisfiable (i.e., only \(\epsilon\) fraction of constraints need to be removed to make it satisfiable for some small constant \(\epsilon\)), can one efficiently find an assignment satisfying most (say, \(1 - f(\epsilon) - o(1)\) where \(f(\epsilon) \rightarrow 0\) as \(\epsilon \rightarrow 0\)) of the constraints?\(^1\)

The problem of finding almost-satisfying assignments was first suggested and studied in a beautiful paper by Zwick [227]. This problem seems well-motivated, as even if a MAXCSP is APX-Hard in general, in certain practical situations instances might be close to being satisfiable (for example, a small fraction of constraints might have been corrupted

\(^1\)Throughout this chapter, constraints could have weights, and by a “fraction \(\alpha\) of constraints” we mean any subset of constraints whose total weight is a fraction \(\alpha\) of the sum of the weights of all constraints. For CSPs with no unary constraints, the approximability of the weighted and unweighted versions are known to be the same [71].
by noise). An algorithm that is able to satisfy most of the constraints of such an instance could be very useful.

As pointed out in [135], Schaefer’s reductions together with the PCP theorem imply that the previous goal is NP-Hard to achieve for any Boolean CSP for which the satisfiability problem is NP-complete. Indeed, all but the above three tractable cases of Boolean CSPs have a “gap at location 1,” which means that given a satisfiable instance it is NP-Hard to find an assignment satisfying $\alpha$ fraction of the constraints for some constant $\alpha < 1$. This result has been extended to CSPs over arbitrary domains recently [128].

The natural question therefore is whether for the three tractable Boolean CSPs, LIN(2), 2-SAT, and HORNSAT, one can find almost-satisfying assignments in polynomial time. Effectively, the question is whether there are “robust” satisfiability checking algorithms that can handle a small number of inconsistent constraints and still produce a near-satisfying assignment.

With respect to the feasibility of finding almost-satisfying assignments, LIN(2), 2-SAT, and HORNSAT behave rather differently from each other. For LIN(2), Håstad in his breakthrough paper [116] showed that for any $\epsilon, \delta > 0$, finding a solution satisfying $1/2 + \delta$ of the equations of a $(1 - \epsilon)$-satisfiable instance is NP-Hard. In fact, this result holds even when each equation depends on only 3 variables. Since just picking a random assignment satisfies $1/2$ the constraints in expectation, this shows, in a very strong sense, that there is no robust satisfiability algorithm for LIN(2).

In sharp contrast to this extreme hardness for linear equations, Zwick [227] proved that for 2-SAT and HORNSAT, one can find almost-satisfying assignments in polynomial time. For MAX2-SAT, Zwick gave a semidefinite programming (SDP) based algorithm that finds a $(1 - O(\epsilon^{1/3}))$-satisfying assignment (i.e., an assignment satisfying a fraction $(1 - O(\epsilon^{1/3}))$ of the constraints) given as input a $(1 - \epsilon)$-satisfiable instance. This algorithm was later improved to one that finds a $1 - O(\sqrt{\epsilon})$-satisfying assignment by Charikar, Makarychev, and Makarychev [62]. The $1 - O(\sqrt{\epsilon})$ bound is known to be best possible under the Unique Games Conjecture (UGC) [136, 141]. In fact, this hardness result for MAX2-SAT was the first application of the UGC and one of the main initial motivations for its formulation by Khot [136].

For HORNSAT, Zwick gave a linear programming (LP) based algorithm to find an assignment satisfying $(1 - O(\log \log(1/\epsilon)/\log(1/\epsilon)))$ of constraints of a $(1 - \epsilon)$-satisfiable instance. Recall that an instance of HORNSAT is a CNF formula where each clause consists of at most one unnegated literal. Equivalently, each clause is of the form $x_i, \overline{x_i}$, or

\footnote{The dual variant DUALHORNSAT is an instance of SAT where each clause has at most one negated literal and it is also polynomial time solvable.}
$x_i \land x_2 \land \ldots \land x_k \to x_{k+1}$ for variables $x_i$. For HORN3-SAT where each clause involves at most three variables, the algorithm finds a $(1 - O(1/\log(1/\epsilon)))$-satisfying assignment. Note that the fraction of unsatisfied constraints is exponentially worse for HORN3-SAT compared to MAX2-SAT.

Horn-SAT is a fundamental problem in logic and artificial intelligence. Zwick’s robust Horn satisfiability algorithm shows the feasibility of solving instances where a small number of constraints are faulty and raises the following natural question, which was also explicitly raised in [227]. Is this $1/\log(1/\epsilon)$ deficit inherent? Or could a more sophisticated algorithm, say based on an SDP relaxation instead of the LP relaxation used in [227], improve the deficit to something smaller (such as $\epsilon b$ for some constant $b$ as in the case of the SDP based algorithm for MAX2-SAT)? It is known that for some absolute constant $c < 1$, it is NP-Hard to find a $(1 - \epsilon c)$-satisfying assignment given a $(1 - \epsilon)$-satisfiable instance of HORN3-SAT [135].

In this chapter, we address the above question and resolve it (conditioned on the UGC), showing the $1/\log(1/\epsilon)$ deficit to be inherent. We describe our results in more detail below in Section 17.2.

**Remark 17.1.1.** For $(1 - \epsilon)$-satisfiable instances of MAX2-SAT, even the hardness of finding a $(1 - \omega(1/\epsilon))$-satisfying assignment is not known without assuming the UGC (and the UGC implies the optimal $1 - \Omega(\sqrt{\epsilon})$ hardness bound). For HORN3-SAT, as mentioned above, we know the NP-Hardness of finding a $(1 - \epsilon c)$-satisfying assignment for some absolute constant $c < 1$. Under the UGC, we are able to pin down the exact asymptotic dependence on $\epsilon$.

### 17.2 Our contributions and previous work

We prove the following hardness result concerning finding almost-satisfying assignments for HORN3-SAT (in fact for the arity 3 case where all clauses involve at most 3 variables). In the sequel, we use the terminology “UG-Hard” to mean at least as hard as refuting the Unique Games Conjecture.

**Theorem 17.2.1.** For some absolute constant $C > 0$, for every $\epsilon > 0$, given a $(1 - \epsilon)$-satisfiable instance of HORN3-SAT, it is UG-Hard to find an assignment satisfying more than a fraction $\left(1 - \frac{C}{\log(1/\epsilon)}\right)$ of the constraints.

Zwick gave a polynomial time algorithm that finds a $1 - O(\log k/\log(1/\epsilon))$-satisfying assignment on input a $(1 - \epsilon)$-satisfiable instance of MAXHORN$k$-SAT. Our inapproximability
bound is therefore optimal up to the constant $C$, and resolves Zwick’s question on whether his algorithm can be improved in the negative. (For arbitrary arity HORNSAT, Zwick’s algorithm has the slightly worse $1 - O(\log \log(1/\epsilon)/\log(1/\epsilon))$ performance ratio; we do not show this to be tight.)

Theorem 17.2.1 shows that HORNSAT has a very different quantitative behavior compared to MAX2-SAT with respect to approximating near-satisfiable instances: the fraction of unsatisfied clauses $\Omega(1/\log(1/\epsilon))$ is exponentially worse than the $O(\sqrt{\epsilon})$ fraction that can be achieved for MAX2-SAT.

A strong hardness result for MINHORNDELETION, the minimization version for HORNSAT, was shown in [135]. It follows from their reduction that for some absolute constant $c < 1$, it is NP-Hard to find a $(1 - \epsilon^c)$-satisfying assignment given a $(1 - \epsilon)$-satisfiable instance of HORNSAT. The constant $c$ would be extremely close to 1 in this result as it is related to the soundness in Raz’s parallel repetition theorem. While our inapproximability bound is stronger and optimal, we are only able to show UG-Hardness and not NP-Hardness.

In light of our strong hardness result for HORN3-SAT, we also consider the approximability of the arity two case. For HORN2-SAT, given a $(1 - \epsilon)$-satisfiable instance, an approximation preserving reduction from vertex cover shows that it is UG-Hard to find a $(1 - c\epsilon)$-satisfying assignment for $c < 2$. It is also shown in [135] that one can find a $(1 - 3\epsilon)$-satisfying assignment efficiently. We improve the algorithmic bound (to the matching UG-Hardness) by proving the following theorem, based on half-integrality of an LP relaxation for the problem.

Theorem 17.2.2. Given a $(1 - \epsilon)$-satisfiable instance for HORN2-SAT, it is possible to find a $(1 - 2\epsilon)$-satisfying assignment in polynomial time.

### 17.3 Proof method

We construct integrality gap instances for a certain semidefinite programming relaxation (described in Section 17.3.1), and then use Raghavendra’s theorem [189] to conclude that assuming the Unique Games Conjecture, no algorithm can achieve an approximation ratio better than the SDP integrality gap.

In contrast to previous such integrality gap constructions (eg., for MAXCUT) where the instances had a good SDP solution “by design” and the technical core was bounding the integral optimum, in our case bounding the integral optimum is the easy part and the challenge is in the construction of appropriate SDP vectors. See Section 17.3.2 for an
overview of our gap instances. It is also interesting that our SDP gaps match corresponding LP gaps. In general it seems like an intriguing question for which CSPs this is the case and therefore LPs suffice to get the optimal approximation ratio.

For our algorithmic results (see Section 17.3.3), we use a natural linear programming relaxation. The algorithm for \textsc{Horn2-SAT} proceeds by showing half-integrality of the LP.

### 17.3.1 The canonical SDP for Boolean CSPs and UG-Hardness

For Boolean CSP instances, we write $C$ as the set of constraints over variables $x_1, x_2, \ldots, x_n \in \{0, 1\}$. The SDP relaxation from \cite{189}, which we call the canonical SDP, sets up for each constraint $C \in C$ a local distribution $\pi_C$ on all the truth-assignments $\{\sigma : X_C \rightarrow \{0, 1\}\}$, where $X_C$ is the set of variables involved in the constraint $C$. This is implemented via scalar variables $\pi_C(\sigma)$ which are required to be non-negative and satisfy $\sum_{\sigma : X_C \rightarrow \{0, 1\}} \pi_C(\sigma) = 1$. For each variable $x$, two orthogonal vectors $v(x,0)$ and $v(x,1)$, corresponding to the events $x = 0$ and $x = 1$, are set up. The SDP requires for each variable $x$, $v(x,0) \cdot v(x,1) = 0$ and $v(x,0) + v(x,1) = I$ where $I$ is a global unit vector. (In the integral solution, one of the vectors $v(x,1)$, $v(x,0)$ — based on the $x$’s Boolean value — is intended to be $I$ and the other one to be 0.)

Then, as constraint (17.5), the SDP does a consistency check: for two variables $x, y$ (that need not be distinct) involved in the same constraint $C$, and for every $b_1, b_2 \in \{0, 1\}$, the SDP insists that the inner product $v(x,b_1) \cdot v(y,b_2)$ equals $\Pr_{\sigma \in \pi_C}[(\sigma(x) = b_1) \land (\sigma(y) = b_2)]$.

Maximize \[ E \sum_{C \in C} \Pr_{\sigma \in \pi_C}[C(\sigma) = 1] \] (17.1)

Subject to \[ v(x_i,0) \cdot v(x_i,1) = 0 \quad \forall i \in [n] \] (17.2)
\[ v(x_i,0) + v(x_i,1) = I \quad \forall i \in [n] \] (17.3)
\[ \|I\|^2 = 1 \] (17.4)
\[ \Pr_{\sigma \in \pi_C}[(\sigma(x_i) = b_1) \land (\sigma(x_j) = b_2)] = v(x_i,b_1) \cdot v(x_j,b_2) \quad \forall C \in C, x_i, x_j \in C, \]
\[ b_1, b_2 \in \{0, 1\} \] (17.5)

Note that if we discard all the vectors by removing constraints (17.2)–(17.4), and changing constraints (17.5) to $\Pr_{\sigma \in \pi_S}[(\sigma(x_i) = b_1) \land (\sigma(x_j) = b_2)] = X(x_i,b_1,x_j,b_2)$, the SDP becomes a lifted LP in Sherali-Adams system. We call this LP scheme the lifted LP in this chapter.

The following striking theorem (Theorem 1.1 in \cite{189}) states that once we have an integrality gap for the canonical SDP, we also get a matching UG-Hardness. Below and
elsewhere in the chapter, a \(c\) vs. \(s\) gap instance is an instance with SDP optimum at least \(c\) and integral optimum at most \(s\).

**Theorem 17.3.1.** Let \(1 > c > s > 0\). If a constraint satisfaction problem \(\Lambda\) admits a \(c\) vs. \(s\) integrality gap instance for the above canonical SDP, then for every constant \(\eta > 0\), given an instance of \(\Lambda\) that admits an assignment satisfying \((c - \eta)\) of constraints, it is UG-Hard to find an assignment satisfying more than \((s + \eta)\) of constraints.

To make our construction of integrality gaps easier, we notice the following simplification of the above SDP. Suppose we are given the global unit vector \(I\) and a vector \(v_x\) for each variable \(x\) in the CSP instance, subject to the following constraints:

\[
(I - v_x) \cdot v_x = 0 \quad \forall \text{ variables } x \tag{17.6}
\]

\[
\Pr_{\sigma \in \pi_C} [\sigma(x_i) = 1 \land \sigma(x_j) = 1] = v_{x_i} \cdot v_{x_j} \quad \forall C \in \mathcal{C}, x_i, x_j \in C. \tag{17.7}
\]

Defining \(v_{(x,1)} = v_x\) and \(v_{(x,0)} = I - v_x\), it is easy to check that all constraints of the above SDP are satisfied. For instance, for variables \(x, y\) belonging to a constraint \(C\),

\[
v_{(x,0)} \cdot v_{(y,1)} = (I - v_{(x,1)}) \cdot v_{(y,1)} = ||v_{(y,1)}||^2 - v_{(x,1)} \cdot v_{(y,1)}
\]

\[
= \Pr_{\sigma \in \pi_C} [\sigma(y) = 1] - \Pr_{\sigma \in \pi_C} [(\sigma(x) = 1) \land (\sigma(y) = 1)]
\]

\[
= \Pr_{\sigma \in \pi_C} [(\sigma(x) = 0) \land (\sigma(y) = 1)],
\]

and other constraints of (17.5) follow similarly.

Henceforth in this chapter, we will work with this streamlined canonical SDP with vector variables \(I\), \(\{v_x\}\), scalar variables corresponding to the local distributions \(\pi_C\), constraints (17.6) and (17.7), and objective function (17.1).

### 17.3.2 Overview of construction of SDP gaps

In the concluding section of [227], Zwick remarks that there is an integrality gap for the LP he uses that matches his approximation ratio. Indeed such a LP gap is not hard to construct and we start by describing one such instance. The instance begins with clause \(x_1\), and in the intermediate \((k - 1)\) clauses, the \(i\)-th clause \(x_1 \land ... \land x_i \rightarrow x_{i+1}\) makes \(x_{i+1}\) true if all the previous clauses are satisfied. Then the last clause \(x_k\) generates a contradiction. Thus the optimal integral solution is at most \((1 - 1/k)\). On the other hand, one possible fractional solution starts with \(x_1 = (1 - \epsilon)\) for some \(\epsilon > 0\). Then for \(1 \leq i < k\), by letting
\[(1 - x_{i+1}) = \sum_{j=1}^{i} (1 - x_j), \text{ all the intermediate } (k - 1) \text{ clauses are perfectly “satisfied” by the LP, while the gap } (1 - x_{i+1}) = 2^{i-1}\epsilon \text{ increases exponentially. Thus by letting } \epsilon = 1/2^{k-2}, \text{ we get } x_k = 0 \text{ and the LP solution is at least } (1 - 1/2^{\Omega(k)}) \text{. The instance gives a } (1 - 2^{-\Omega(k)}) \text{ vs. } (1 - 1/k) \text{ LP integrality gap.}

Now we convert this LP gap instance into an SDP gap instance in two steps. First, since we are going to give a gap instance for MAX HORN-3SAT, we reduce the arity of the instance from \(k\) to \(3\). Then, we find a set of vectors for the LP solution to make it an SDP solution.

For the first step, to get an instance of MAX HORN-3SAT, we introduce \(y_i\) which is intended to be \(x_1 \land \ldots \land x_{i-1}\). For \(1 \leq i < k\), we replace the intermediate clauses by \(x_i \land y_i \rightarrow x_{i+1}\), and add \(x_i \land y_i \rightarrow y_{i+1}\) to meet the intended definition of \(y_i\). We call each of these two clauses as comprising one step. It is easy to show that for this instance there is a solution of value \((1 - 1/2^{\Omega(k)})\) even for the lifted LP. (The difference between the lifted LP and Zwick’s LP is that the lift LP introduces local distributions over clauses which are consistent in the first and second moments, while Zwick’s LP only has variables for singletons.)

Finding vectors for the SDP turns out to be more challenging. Note that if we want to perfectly satisfy all the intermediate clauses in SDP, we need to obey \(v_{x_i} \cdot v_{y_i} \leq \|v_{x_{i+1}}\|^2\) and \(v_{x_i} \cdot v_{y_i} \leq \|v_{y_{i+1}}\|^2\) for \(1 \leq i < k\). Thus to make the norms \(\|v_{x_{i+1}}\|^2\) and \(\|v_{y_{i+1}}\|^2\) decrease fast (since we want \(\|v_{x_i}\|^2 = \|v_{y_i}\|^2 = 0\)), we need to make the inner product \(v_{x_i} \cdot v_{y_i}\) decrease as well. But technically it is hard to make both kinds of quantities decrease at a high rate for all intermediate clauses. Our solution is to decrease the norms and inner products alternately. More specifically, we divide the intermediate clauses into blocks, each of which contains two consecutive steps. In the first step of each block, we need that the inner product is much smaller than the norms so that we can decrease the norms quickly, but we preserve the value of inner product. Thus we cannot do this step repeatedly, and we need the second step, where we decrease the inner product (while preserving the norms) in preparation to start the first step of the next block.

### 17.3.3 Overview of algorithmic results

Our algorithmic result for HORN2-SAT (Theorem 17.2.2) is obtained by rounding fractional solutions of appropriate linear programming (LP) relaxations. The algorithm is indeed a 2-approximation algorithm for MINHORNDELETION problem (refer to Section 17.4.2 for the definition of MINHORNDELETION). We prove a half-integrality property of the optimal solution to the natural LP relaxation of the problem, which can be
viewed as a generalization of half-integrality property of (the natural LP for) Vertex Cover. We take the optimal solution of the natural LP relaxation, iteratively make every variable move towards half-integral values (0, 1, and 1/2), while never increasing the value of the solution. This yields an optimal half-integral solution which can then be trivially rounded to obtain an integral solution that gives a factor 2 approximation.

17.4 Approximability of HORN3-SAT

17.4.1 SDP gap and UG hardness for HORN3-SAT

17.4.1.1 Instance

We consider the following HORN3-SAT instance $I^\text{Horn}_k$ parameterized by $k \geq 1$. (This construction is essentially the same as the one described in Section 17.3.2.)

Start point: $x_0, y_0$

Block $i$ ($0 \leq i \leq k - 1$)

Step i.1: $x_{2i} \land y_{2i} \rightarrow x_{2i+1}$, $x_{2i} \land y_{2i} \rightarrow y_{2i+1}$

Step i.2: $x_{2i+1} \land y_{2i+1} \rightarrow x_{2i+2}$, $x_{2i+1} \land y_{2i+1} \rightarrow y_{2i+2}$

End point: $x_{2k} \land y_{2k} \rightarrow x_{2k+1}$, $x_{2k} \land y_{2k} \rightarrow y_{2k+1}$

It is easy to see this instance contains $(4k + 6)$ clauses, and cannot be completely satisfied. Thus we have:

**Lemma 17.4.1.** Every Boolean assignment satisfies at most a fraction $1 - 1/(4k + 6)$ of the clauses of $I^\text{Horn}_k$.

17.4.1.2 Construction of a good SDP solution

We will work with the SDP in simplified form described at the end of Section 17.3.1. Recall that the SDP requires a local distribution for each clause, and uses vectors to check the consistency on every pair of variables that belong to the clause. To construct a good solution for the SDP, we want to first find a good solution in the scalar part (i.e., local distributions), and then construct vectors which meet the consistency requirement. But it is difficult to construct a lot of vectors which meet all the requirements simultaneously. Thus, we break down the whole construction task into small pieces, each of which is easy to deal with. As long as there are solutions to these small pieces, and the solutions agree
with each other on some interfaces, we can coalesce the small solutions together and come up with a global solution. The following definition and claim formally help us bring down the difficulty, and focus on one local block of variables at a time.

**Definition 17.4.2 (partial solution).** Let \( C' \subseteq C \) be a subset of clauses. \( f = \{ \pi_C = \pi_C(f), v_x = v_x(f), I = I(f) \mid \forall C \in C', x \in C' \} \) is said to be a partial solution on \( C' \), if all constraints of the SDP restricted to the subset of variables defined in \( f \) are satisfied.

**Claim 17.4.3.** Let \( C_1, C_2 \subseteq C \) be two disjoint set of clauses. Let \( f \) and \( g \) be partial solutions on \( C_1, C_2 \) respectively. If for all \( v_1, v_2 \) (not necessarily distinct) defined in both \( f \) and \( g \), \( v_1(f) \cdot v_2(f) = v_1(g) \cdot v_2(g) \), then there exists a partial solution, namely \( h \), for \( C_1 \cup C_2 \), such that \( \forall C_1 \in C_1, C_2 \in C_2, \pi_{C_1}(h) = \pi_{C_1}(f), \pi_{C_2}(h) = \pi_{C_2}(g) \).

**Proof.** Let \( X \) be the set of variables \( x \) for which \( v_x(f) \) and \( v_x(g) \) are both defined. Denote \( V_f = \{ v_x(f) \mid x \in X \} \cup \{ I(f) \} \) and \( V_g = \{ v_x(g) \mid x \in X \} \cup \{ I(g) \} \). Since the dot products of every pair of vectors in \( V_f \) exactly equals the dot product between the corresponding pair in \( V_g \), there is a rotation (orthogonal transformation) \( T \) such that \( I(f) = TI(g) \) and for all \( x \in X \), \( v_x(f) = Tv_x(g) \).

Now define the partial solution \( g' \) as \( \pi_C(g') = \pi_C(g) \) for all \( C \in C_2 \) and \( v_x(g') = Tv_x(g) \), \( I(g') = TI(g) \) for all \( x \in C \in C_2 \). Obviously \( f \) and \( g' \) agree on all the scalar and vector variables that are defined in both \( f \) and \( g' \). Letting

\[
v_x(h) = \begin{cases} v_x(f), & x \in C \in C_1 \\ v_x(g'), & x \in C \in C_2 \end{cases}, \pi_C(h) = \begin{cases} \pi_C(f), & C \in C_1 \\ \pi_C(g'), & C \in C_2 \end{cases},
\]

it is easy to see \( h \) is a partial solution on \( C_1 \cup C_2 \). \( \square \)

By the above lemma, if we establish the following lemma which constructs a good partial solution on each block (the proof of which is deferred to Section 17.4.1.3), it is then easy to get a good global solution.

**Lemma 17.4.4.** For each Block \( i \) \((0 \leq i \leq k - 1)\), each \( 0 < c \leq 0.2 \), let \( r_c = 1.5(1 + c)/(1.5 + c) > 1 \), and for each \( 0 < p \leq \frac{1}{(1+c)r_c} \), there is a partial solution \( f \) which completely satisfies all the clauses in Block \( i \) (by local distributions), and with following properties,

\[
\|v_{x_{2i}}(f)\|^2 = \|v_{y_{2i}}(f)\|^2 = 1 - p
\]
\[
v_{x_{2i}}(f) \cdot v_{y_{2i}}(f) = 1 - (1 + c)p
\]
\[
\|v_{x_{2i+2}}(f)\|^2 = \|v_{y_{2i+2}}(f)\|^2 = 1 - r_c p
\]
\[
v_{x_{2i+2}}(f) \cdot v_{y_{2i+2}}(f) = 1 - (1 + c)r_c p.
\]

287
Proof. By Corollary 17.4.5, for any Lemma 17.4.6.
The optimal SDP solution for the instance \( \{0,2\} \) on pairwise inner-products over the definition of by induction hypothesis there is a partial solution \( g \) solution of Blocks 0.

With the above pieces in place, we now come to the final SDP solution.

Lemma 17.4.6. The optimal SDP solution for the instance \( I_k^{\text{Horn}} \) has value at least 
\[
1 - \frac{1}{(2k+3)1.05^r}.
\]

Proof. By Corollary 17.4.5, for any \( 0 < c \leq 0.2 \), by setting \( p = \frac{1}{(1+c)r_c} \), there is a partial solution \( g \) completely satisfying all the clauses of all the blocks, with
\[
\|v_{x_0}(g)\|^2 = \|v_{y_0}(g)\|^2 = 1 - \frac{1}{(1+c)r_c},
\]
\[
\|v_{x_{2k'}}(g)\|^2 = \|v_{y_{2k'}}(g)\|^2 = c/(1+c)
\]
\[
v_{x_{2k'}}(g) \cdot v_{y_{2k'}}(g) = 0.
\]
Based on $g$, we define a local distribution on two “Start point” clauses by making $x_0$ (or $y_0$) equal 1 with probability $1 - p$. At “End point”, we define the local distribution on clause $x_{2k} \land y_{2k} \rightarrow x_{2k+1}$ as
\[
\Pr_{\pi} [x_{2k} = 1 \land y_{2k} = 0 \land x_{2k+1} = 0] = c/(1 + c)
\]
\[
\Pr_{\pi} [x_{2k} = 0 \land y_{2k} = 1 \land x_{2k+1} = 0] = c/(1 + c)
\]
\[
\Pr_{\pi} [x_{2k} = 0 \land y_{2k} = 0 \land x_{2k+1} = 0] = (1 - c)/(1 + c).
\]
And a similar distribution for the clause $x_{2k} \land y_{2k} \rightarrow y_{2k+1}$ can be defined (by replacing $x_{2k+1}$ by $y_{2k+1}$ in the equations above). The distribution on clauses $\overline{x}_{2k+1}$ and $\overline{y}_{2k+1}$ never picks the corresponding variable to be 1. By defining $v_{x_{2k+1}}$ and $v_{y_{2k+1}}$ to be zero vectors, we note that the distributions are consistent with vectors. Thus the solution we construct is valid.

On the other hand, note that all the distributions locally satisfy the clauses, except for the distributions at “Start point” satisfy the corresponding clause with probability $1 - \frac{1}{(1+c)\varepsilon^2}$. Thus the SDP solution has value $1 - \frac{2}{(4k+6)(1+c)\varepsilon^2} = 1 \geq 1 - \frac{1}{(2k+3)\varepsilon^2}$. By setting $c = 0.2$, we get $r_c \geq 1.05$. Thus the best SDP solution has value better than $1 - \frac{1}{(2k+3)\varepsilon^2}$.

Combining Lemma 17.4.1 and Lemma 17.4.6 we get the following theorem.

**Theorem 17.4.7.** $T^\text{Horn}_k$ is a $(1-\varepsilon)$ vs. $(1-\Omega(1/\log(1/\varepsilon)))$ gap instance of HORN3-SAT for the canonical SDP relaxation.

Together with Theorem 17.3.1 Theorem 17.4.7 implies our main result, Theorem 17.2.1, on HORN SAT.

17.4.1.3 Proof of the Key Lemma 17.4.4

For Block $i$, denote the clauses in Step $i.1$ by $C_{1x}$ and $C_{1y}$, and the clauses in Step $i.2$ by $C_{2x}$ and $C_{2y}$. We first construct partial solutions on Step $i.1$ and Step $i.2$ separately, as follows.

**Partial solution on Step** $i.1$ We first define a local distribution on satisfying assignments for $C_{1x}$ as follows, and $C_{1y}$ in a similar way (by replacing $x_{2i+1}$ by $y_{2i+1}$ in following equations).
\[
\Pr_{\pi_{C_{1x}}} [x_{2i} = 1 \land y_{2i} = 1 \land x_{2i+1} = 1] = 1 - (1 + c)p
\]
Recall $r_c$ consistent with the local distributions on satisfying assignments for $c$ negative values by the range of $i$.

Partial solution on Step $C_1$.

Therefore there is a set of vectors consistent with our local distributions, i.e., we get a partial solution on Step $i.1$.

By Claim 17.4.8 (at the end of this section) we know that $A$ is positive semidefinite, and therefore there is a set of vectors consistent with our local distributions, i.e., we get a partial solution on Step $i.1$.

Partial solution on Step $i.2$ We define the local distribution on satisfying assignments for $C_{2x}$ as follows. The distribution for $C_{2y}$ is defined in a similar way (by replacing $x_{2i+2}$ with $y_{2i+2}$ in the following equations). Let $q = r_c p$ and $\epsilon = c/1.5$.

\[
\Pr_{\pi_{C_{2x}}} [x_{2i+1} = 1 \land y_{2i+1} = 1 \land x_{2i+2} = 1] = 1 - (1 + \epsilon)q
\]

\[
\Pr_{\pi_{C_{2x}}} [x_{2i+1} = 1 \land y_{2i+1} = 0 \land x_{2i+2} = 0] = \epsilon q
\]

\[
\Pr_{\pi_{C_{2x}}} [x_{2i+1} = 0 \land y_{2i+1} = 1 \land x_{2i+2} = 0] = \epsilon q
\]

\[
\Pr_{\pi_{C_{2x}}} [x_{2i+1} = 0 \land y_{2i+1} = 0 \land x_{2i+2} = 1] = \epsilon q
\]

\[
\Pr_{\pi_{C_{2x}}} [x_{2i+1} = 0 \land y_{2i+1} = 0 \land x_{2i+2} = 0] = (1 - 2\epsilon)q.
\]

Note that all the probabilities are defined to be non-negative values by the range of $c$ and $p$, and they sum up to 1.

290
Then note that the following inner-product matrix $B$ over $I, v_{x_{2i+1}}, v_{y_{2i+1}}, v_{x_{2i+2}}, v_{y_{2i+2}}$ is consistent with the local distribution.

$$
B = \begin{bmatrix}
1 & 1 - q & 1 - q & 1 - q & 1 - q \\
1 - q & 1 - q & 1 - (1 + \epsilon)q & 1 - (1 + \epsilon)q & 1 - (1 + \epsilon)q \\
1 - q & 1 - (1 + \epsilon)q & 1 - q & 1 - (1 + \epsilon)q & 1 - (1 + \epsilon)q \\
1 - q & 1 - (1 + \epsilon)q & 1 - (1 + \epsilon)q & 1 - q & 1 - (1 + 1.5\epsilon)q \\
1 - q & 1 - (1 + \epsilon)q & 1 - (1 + \epsilon)q & 1 - (1 + 1.5\epsilon)q & 1 - q
\end{bmatrix}
$$

Again by Claim 17.4.8 $B$ is positive semidefinite, and therefore there is a set of vectors consistent with local distributions – we have constructed a partial solution on Step $i.2$.

**Combining the two partial solutions.** We first check that under our parameter setting, partial solutions on Step $i.1$ and Step $i.2$ agree on pairwise inner-products between their shared vectors $I, v_{x_{2i+1}}, v_{y_{2i+1}}$.

- For $\langle I, I \rangle$, we have $1 = 1$.
- For $\langle I, v_{x_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle I, v_{y_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle v_{x_{2i+1}}, v_{x_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.
- For $\langle v_{x_{2i+1}}, v_{y_{2i+1}} \rangle$, we have $1 - (1 + c)p = 1 - (1 + c/1.5)r_c p = 1 - (1 + \epsilon)q$.
- For $\langle v_{y_{2i+1}}, v_{y_{2i+1}} \rangle$, we have $1 - r_c p = 1 - q$.

Thus, there is a partial solution on Block $i$, with

$$
\|v_{x_{2i}}(f)\|^2 = \|v_{y_{2i}}(f)\|^2 = 1 - p \\
v_{x_{2i}}(f) \cdot v_{y_{2i}}(f) = 1 - (1 + c)p \\
\|v_{x_{2i+2}}(f)\|^2 = \|v_{y_{2i+2}}(f)\|^2 = 1 - q = 1 - r_c p \\
v_{x_{2i+2}}(f) \cdot v_{y_{2i+2}}(f) = 1 - (1 + 1.5\epsilon)q = 1 - (1 + c)r_c p. \quad \square
$$

Finally, we establish the two positive semidefinite matrices used in the proof above.

**Claim 17.4.8.** Given $0 < c \leq 0.2$, $0 < p \leq \frac{1}{1+c} r_c$, $q = r_c p$, $\epsilon = c/1.5$, the following two matrices are positive semidefinite.

$$
A = \begin{bmatrix}
1 & 1 - p & 1 - p & 1 - r_c p & 1 - r_c p \\
1 - p & 1 - p & 1 - (1 + c)p & 1 - (1 + c)p & 1 - (1 + c)p \\
1 - p & 1 - (1 + c)p & 1 - p & 1 - (1 + c)p & 1 - (1 + c)p \\
1 - r_c p & 1 - (1 + c)p & 1 - (1 + c)p & 1 - r_c p & 1 - (1 + c)p \\
1 - r_c p & 1 - (1 + c)p & 1 - (1 + c)p & 1 - r_c p & 1 - (1 + c)p
\end{bmatrix}
$$

291
\[
B = \begin{bmatrix}
1 & 1-q & 1-q & 1-q & 1-q \\
1-q & 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-(1+\epsilon)q \\
1-q & 1-(1+\epsilon)q & 1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q \\
1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-q & 1-(1+1.5\epsilon)q \\
1-q & 1-(1+\epsilon)q & 1-(1+\epsilon)q & 1-(1+1.5\epsilon)q & 1-q
\end{bmatrix}.
\]

**Proof.** Let \( J \) be the all 1 matrix, \( E_1 \) be the matrix with 1 in entry (1, 1) as the only one non-zero entry. We also define \( E_{i,j}, F_{i,j} \) and \( G_{i,j} \) as matrices with only four non-zero entries located in the intersections of Column \( i, j \) and Row \( i, j \). The sub-matrices of \( E_{i,j}, F_{i,j} \) and \( G_{i,j} \) on Column \( i, j \) and Row \( i, j \) are defined as

(for \( E_{i,j} \)) \[ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \]

(for \( F_{i,j} \)) \[ \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \]

and (for \( G_{i,j} \)) \[ \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \]

Clearly, all of \( J, E_1, E_{i,j}, F_{i,j} \) and \( G_{i,j} \) are positive semidefinite matrices.

Then we can write \( A \) as

\[
A = (1 - (1 + c)p)J + cp(E_{1,2} + E_{1,3}) + (1 + c - r_c)p(E_{1,4} + E_{1,5}) + (2r_c - 1 - 3c)pE_1
\]

\[
= (1 - (1 + c)p)J + cp(E_{1,2} + E_{1,3}) + \frac{(1 + c)c}{1.5 + c} \cdot p(E_{1,4} + E_{1,5}) + \frac{1.5 - 2.5c - 3c^2}{1.5 + c} \cdot pE_1,
\]

Note that all the coefficient before matrices are non-negative within the range of \( c \). Since \( A \) can be written as the sum of several positive semidefinite matrices, \( A \) is positive semidefinite.

For matrix \( B \), note that

\[
B = (1 - (1 + \epsilon)q)J + \epsilon q(E_{1,2} + E_{1,3} + F_{1,4} + F_{1,5}) + 0.5\epsilon qG_{4,5} + (1 - 5\epsilon)qE_1,
\]

Clearly, as long as \( 5\epsilon = 5c/1.5 < 1 \), \( B \) can be expressed as sum of positive semidefinite matrices, and hence \( B \) is positive semidefinite.

\[\Box\]

### 17.4.2 Algorithm for MINHORNDELETION and MAXHORN2-SAT

In the MINHORNDELETION problem, we are given a HORN2-SAT instance, and the goal is to find a subset of clauses of minimum total weight whose deletion makes the instance satisfiable. A factor 3 approximation algorithm for MINHORNDELETION is given in [135]. Here we improve the approximation ratio to 2. Note that following reduction from vertex cover is approximation preserving: given a graph with \( n \) vertices, for each
Our motivation to study MINHORNDELETION in the context of this chapter is to pin down the fraction of clauses one can satisfy in a \((1 - \epsilon)\)-satisfiable instance of HORN2-SAT: we can satisfy a fraction \((1 - 2\epsilon)\) of clauses (even in the weighted case), and satisfying a \((1 - c\epsilon)\) fraction is hard for \(c < 2\) assuming that vertex cover does not admit a \(c\)-approximation for any constant \(c < 2\).

In this section, we prove the following theorem by showing half-integrality of a natural LP relaxation for the problem.

**Theorem 17.4.9.** There is a polynomial-time 2-approximation algorithm for MINHORNDELETION problem.

A direct corollary of Theorem 17.4.9 is the following result for approximating near-satisfiable instances of HORN2-SAT.

**Theorem 17.2.2 (restated).** Given a \((1 - \epsilon)\)-satisfiable instance for MAXHORN2-SAT, it is possible to find a \((1 - 2\epsilon)\)-satisfying assignment efficiently.

### 17.4.2.1 LP Formulation

We find it slightly more convenient to present the algorithm for DUALHORN2-SAT where each clause has at most one negated literal. (So the clauses are of the form \(x, \bar{x}, x \lor y, \) or \(x \rightarrow y,\) for variables \(x, y.\)) Let \(w_{ij}^{(D)} > 0\) be the weight imposed on the disjunction constraint \(x_i \lor x_j\) (for each pair of \(i, j\) such that \(i < j\)), and \(w_{ij}^{(I)} > 0\) be the weight imposed on the implication constraint \(x_i \rightarrow x_j\) (for each pair of \(i, j\) such that \(i \neq j\)). For each variable \(x_i\), let \(w_i^{(T)}\) be the weight on \(x_i\) being true (i.e. \(x_i = 1\)), and \(w_i^{(F)}\) be the weight on \(x_i\) being false (i.e. \(x_i = 0\)). Then we write the following LP relaxation, where each real variable \(y_i\) corresponds to the integer variable \(x_i\).
Minimize \[
\sum_{i \in V} w_i^{(T)} (1 - y_i) + \sum_{i \in V} w_i^{(F)} y_i + \sum_{i < j} w_{ij}^{(D)} z_{ij}^{(D)} + \sum_{i \neq j} w_{ij}^{(T)} z_{ij}^{(T)}
\]
Subject to
\[
\begin{align*}
& z_{ij}^{(D)} \geq 1 - y_i - y_j \quad \forall i < j \\
& z_{ij}^{(T)} \geq y_i - y_j \quad \forall i \neq j \\
& z_{ij}^{(D)} \geq 0 \quad \forall i < j \\
& z_{ij}^{(T)} \geq 0 \quad \forall i \neq j \\
& y_i \in [0, 1] \quad \forall i \in V
\end{align*}
\]

Let OPT be the optimal value of the integral solution, and OPT\text{LP} be the optimal value of the LP solution. We have OPT\text{LP} \leq OPT.

### 17.4.2.2 Half-integrality and rounding

Given a LP solution \( f = \{z_{ij}^{(D)}, z_{ij}^{(T)}, y_i\} \), we can assume \( z_{ij}^{(D)} = \max\{1 - y_i - y_j, 0\} \) and \( z_{ij}^{(T)} = \max\{y_i - y_j, 0\} \) to minimize \( \text{val}(f) \). Thus, we only need \( f = \{y_i\} \) to characterize a solution, and we have

\[
\text{val}(f) = \\
\sum_{i \in V} w_i^{(T)} (1 - y_i) + \sum_{i \in V} w_i^{(F)} y_i + \sum_{i < j} w_{ij}^{(D)} \max\{1 - y_i - y_j, 0\} + \sum_{i \neq j} w_{ij}^{(T)} \max\{y_i - y_j, 0\}.
\]

**Lemma 17.4.10.** There is a polynomial-time algorithm that, given a solution \( f = \{y_i\} \) to the above LP, converts \( f \) into another solution \( f^* = \{y_i^*\} \) such that each \( y_i^* \) is half-integral, i.e., \( y_i^* \in \{0, 1, 1/2\} \), and \( \text{val}(f^*) \leq \text{val}(f) \).

**Proof.** We run Algorithm 3 whose input is the LP formulation and one of the solutions \( f = \{y_i\} \), and whose output is the desired \( f^* \). At a high level, the algorithm iteratively moves the LP variables that are not half integral to half integral values (according to some strategy), and we need to prove that at each step of the iteration, the algorithm creates a new valid LP solution whose objective value is no greater than the previous one.

It’s easy to see that Algorithm 3 always maintains a valid solution \( f \) to the LP (i.e., all variables \( y_i \)’s are within the \([0, 1]\) range). Then we only need to prove the following two things to show the correctness of Algorithm 3: (1) the while loop terminates (in a linear number of steps), (2) in each loop, \( \min\{\text{val}(f^{(a)}), \text{val}(f^{(b)})\} \leq \text{val}(f) \), so that \( \text{val}(f) \) never increases in the whole algorithm.
Algorithm 3 Round any LP solution \( f = \{y_i\} \) to a half-integral solution \( f^* \) such that \( \text{val}(f^*) \leq \text{val}(f) \)

1: while \( \exists i \in V : y_i \not\in \{0, 1, 1/2\} \) do
2: \hspace{1em} choose \( k \in V \), such that \( y_k \not\in \{0, 1, 1/2\} \) (arbitrarily)
3: \hspace{1em} if \( y_k < 1/2 \) then
4: \hspace{1em} \hspace{1em} \( p \leftarrow y_k \)
5: \hspace{1em} else
6: \hspace{1em} \hspace{1em} \( p \leftarrow 1 - y_k \)
7: \hspace{1em} \hspace{1em} \( S \leftarrow \{i : y_i = p\}, S' \leftarrow \{i : y_i = 1 - p\} \)
8: \hspace{1em} \hspace{1em} \( a \leftarrow \max\{y_i : y_i < p, 1 - y_i : y_i > 1 - p, 0\}, b \leftarrow \min\{y_i : y_i > p, 1 - y_i : y_i < 1 - p, 1/2\} \)
9: \hspace{1em} \hspace{1em} \( f^{(a)} \leftarrow \{y_i^{(a)} = a\}_{i \in S} \cup \{y_i^{(a)} = 1 - a\}_{i \in S'} \cup \{y_i^{(a)} = y_i\}_{i \in V \setminus (S \cup S')} \)
10: \hspace{1em} \hspace{1em} \( f^{(b)} \leftarrow \{y_i^{(b)} = b\}_{i \in S} \cup \{y_i^{(b)} = 1 - b\}_{i \in S'} \cup \{y_i^{(b)} = y_i\}_{i \in V \setminus (S \cup S')} \)
11: \hspace{1em} \hspace{1em} if \( \text{val}(f^{(a)}) \leq \text{val}(f^{(b)}) \) then
12: \hspace{1em} \hspace{1em} \hspace{1em} \( f \leftarrow f^{(a)} \)
13: \hspace{1em} \hspace{1em} else
14: \hspace{1em} \hspace{1em} \hspace{1em} \( f \leftarrow f^{(b)} \)
15: \hspace{1em} return \( f \) (as \( f^* \))

To prove the first point, we consider the set \( W_f = \{0 < y < 1/2 : \exists i \in V, \text{s.t.} y = y_i \lor y = 1 - y_i\} \). In each loop, the algorithm picks a \( p \) from \( W_f \). At the end of the loop, we see that \( p \) is wiped from \( W_f \) while no new elements are added. Thus, after linear steps of the loop, \( W_f \) becomes \( \emptyset \) and the loop terminates.

For the second point, we define \( f^{(t)} = \{y_i^{(t)} = t\}_{i \in S} \cup \{y_i^{(t)} = 1 - t\}_{i \in S'} \cup \{y_i^{(t)} = y_i\}_{i \in V \setminus (S \cup S')} \) for \( t \in [a, b] \) at \ref{line:8} in the algorithm. Then if we can show \( \text{val}(f^{(t)}) \) is a linear function within \( t \in [a, b] \), together with the fact \( p \in [a, b] \), we shall conclude that \( \min\{\text{val}(f^{(a)}), \text{val}(f^{(b)})\} \leq \text{val}(f^{(p)}) = \text{val}(f) \). To prove the linearity of \( \text{val}(f^{(t)}) \), we only need to show that \( g_1(t) = \max\{1 - y_i^{(t)} - y_j^{(t)}, 0\} \) and \( g_2(t) = \max\{y_i^{(t)} - y_j^{(t)}, 0\} \) are linear with respect to \( t \in [a, b] \), for any possible \( i, j \). Thus we discuss the following five cases.

- \( i, j \in V \setminus (S \cup S') \). In this case, \( g_1 \) and \( g_2 \) are constant functions.
- \( i \in V \setminus (S \cup S'), j \in S \cup S' \). In this case, the only “non-linear point” is at \( t = 1 - y_i \) for \( g_1 \) and \( t = y_i \) for \( g_2 \). But these two points are away from \( [a, b] \).
- \( i \in S \cup S', j \in V \setminus (S \cup S') \). Similar argument works as the previous case.

295
• $i \in S$, $j \in S'$ (or $i \in S'$, $j \in S$). In this case, $1 - y_i^{(t)} - y_j^{(t)} = 0$ always holds for $t \in [a, b]$ and therefore $g_1$ is constant function. On the other hand, since $y_i^{(t)} \leq y_j^{(t)}$ (or $y_i^{(t)} \geq y_j^{(t)}$), we also have $g_2(t) = 0$ (or $g_2(t) = y_i^{(t)} - y_j^{(t)} = 1 - 2t$) being linear.

• $i, j \in S$ (or $i, j \in S'$). In this case, $y_i^{(t)} = y_j^{(t)}$ always holds for $t \in [a, b]$ and therefore $g_2$ is constant function. On the other hand, since $y_i^{(t)} + y_j^{(t)} \leq 1$ (or $y_i^{(t)} + y_j^{(t)} \geq 1$), we also have $g_1(t) = 1 - y_i^{(t)} - y_j^{(t)} = 1 - 2t$ (or $g_1(t) = 0$) being linear.

\[ \square \]

A direct corollary of Lemma 17.4.10 is the following.

**Corollary 17.4.11.** There is a polynomial-time algorithm to get a solution $f$ such that $\text{val}(f) = \text{OPT}_{\text{LP}}$ and the variables in $f$ are half-integral (i.e. being one of 0, 1, and 1/2).

Now we are ready for the proof of Theorem 17.4.9.

**Proof of Theorem 17.4.9.** Apply Corollary 17.4.11 to get an optimal LP solution $f = \{y_i\}$ which has half-integral values. Then define $f_{\text{int}} = \{x_i\}$ as follows. For each $i \in V$, let $x_i = 1$ when $y_i \geq 1/2$, and $x_i = 0$ when $y_i = 0$. We observe that

- $x_i \leq 2y_i$ and $1 - x_i \leq 1 - y_i$ for each $i \in V$.

- For each $i < j$, we have $\max\{1 - x_i - x_j, 0\} \leq \max\{1 - y_i - y_j, 0\}$ since $x_i \geq y_i$ and $x_j \geq y_j$.

- For each $i \neq j$, we see that when $\max\{y_i - y_j, 0\} = 0$ (in which case we have $y_i \leq y_j$), we always have $x_i \leq x_j$, therefore $\max\{x_i - x_j, 0\} = 0$. On the other hand, when $\max\{y_i - y_j, 0\}$ (in which case by half-integrality we have $\max\{y_i - y_j, 0\} \geq 1/2$), we have $\max\{x_i - x_j, 0\} \leq 1 \leq 2 \max\{y_i - y_j, 0\}$.

Altogether, we have

\[
\text{val}(f_{\text{int}}) = \sum_{i \in V} w_i^{(T)}(1 - x_i) + \sum_{i \in V} w_i^{(F)} x_i + \sum_{i < j} w_{ij}^{(D)} \max\{1 - x_i - x_j, 0\} + \sum_{i \neq j} w_{ij}^{(I)} \max\{x_i - x_j, 0\}
\]

\[
\leq \sum_{i \in V} w_i^{(T)}(1 - y_i) + \sum_{i \in V} w_i^{(F)} 2y_i + \sum_{i < j} w_{ij}^{(D)} \max\{1 - y_i - y_j, 0\} + \sum_{i \neq j} w_{ij}^{(I)} 2 \max\{y_i - y_j, 0\}
\]

\[
\leq 2 \text{val}(f) = 2 \text{OPT}_{\text{LP}} \leq 2 \text{OPT}.
\]

\[ \square \]
Part V

Future directions
Chapter 18

Open problems

There are numerous questions unsolved in approximation algorithms and hardness of approximation. The biggest open question related to this thesis is whether constant-degree Parrilo–Lasserre SDP solves UNIQUEGAMES and/or gives an approximation guarantee better than that of the Goemans-Williamson algorithm; and how this type of results would help us understand the real approximability of UNIQUEGAMES and MAXCUT (and more problems such as UNIFORMSPARSESTCUT and BALANCEDSEPARATOR).

Given this goal, it is highly worthwhile to extend our current understanding of the limitations of the Parrilo–Lasserre hierarchy and prove more lower bounds for the Parrilo-Lasserre hierarchy. One concrete question is whether degree-4 Parrilo-Lasserre SDP, i.e. the first SDP in the hierarchy that is stronger than the basic SDP, already solves UNIQUEGAMES or beats the Goemans-Williamson algorithm on MAXCUT. Another concrete open question here is whether it is possible improve the current best lower bound for DENSEkSUBGRAPH (i.e. Theorem 3.1.4) and possibly match the best approximation algorithm by [41].

Apart from the limitations of the Parrilo-Lasserre hierarchy, here we list a few other interesting research directions on convex relaxation hierarchies and approximation algorithms.

**Faster hierarchy-based algorithms.** Convex programming relaxation hierarchies are powerful but have additional variables/constraints and therefore need more computation time. Authors in [111] showed a way to speed up the algorithms for some problems by only partially solving the convex program. It is desirable to extend their results to more
problems. One particular problem is \textsc{RobustMaxBisection}. As discussed in Section 13.1.2, Raghavendra and Tan [195] gave an $\left(1 - \epsilon\right)$ vs. $\left(1 - O(\sqrt{\epsilon})\right)$ algorithm for \textsc{RobustMaxBisection} (which is optimal assuming the UGC). However, their algorithm requires $n^{1/\epsilon^{\Theta(1)}}$, which is not completely polynomial in $n$ when $\epsilon$ is subconstant (say, $\frac{1}{\text{polylog}(n)}$). The concrete open problem is to speed up the algorithm by Raghavendra and Tan (possibly using the techniques in [111]) and get the same approximation guarantee in $\text{poly}(n)$ time for every $\epsilon$.

\textbf{The Parrilo-Lasserre hierarchy for average-case problems.} For problems that are hard in the worst case, it is still possible to solve them in practice if these problems are easy on average (i.e. when a random instance is solvable with high probability). One problem I am particularly interested in is estimating hypercontractive norms. This problem is highly related to the certification of the matrix restricted isometry property (RIP) and has applications to \textit{compressed sensing} and \textit{sparse recovery}. As our initial result in Chapter 15 showed that, in the average case, the $2 \rightarrow 4$ operator norm (a special form of the hypercontractive norms) can be efficiently estimated by the Parrilo-Lasserre hierarchy. Applying the Parrilo-Lasserre hierarchy to other forms of the hypercontractive norms might lead to more important results and applications.

\textbf{Robust isomorphism algorithms.} It is very interesting to extend our robust isomorphism algorithm for trees (Theorem 14.1.4) to more subclasses of graphs such as planar graphs and graphs of bounded treewidth. There is a natural linear programming relaxation-based candidate algorithm and it is worthwhile to try and prove its correctness for several subclasses of graphs.
Bibliography


301


308


310


311


[137] Subhash Khot. Ruling out PTAS for Graph Min-Bisection, Dense $k$-Subgraph, and Bipartite Clique. SIAM Journal of Computing, 36:1025–1071, 2006. 3.1.2 5.1 6.1.2


8.1.2


[143] Subhash Khot and Oded Regev. Vertex Cover might be hard to approximate to within $2 - \epsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008. 17.4.2

[144] Subhash Khot and Nisheeth Vishnoi. The Unique Games Conjecture, integrality gap for cut problems and embeddability of negative type metrics into $\ell_1$. In *Proc. 46th IEEE Symposium on Foundations of Computer Science*, pages 53–62, 2005. 2.3 5.2.1 5.2.1 6.1 8.1 8.1.2 8.1.3 8.1.2 8.1.2 8.1.2 8.1.2 9.0.8 11.3


314


[169] Vladimir Andreevich Markov. On functions of least deviation from zero in a given interval. 1892. 11.4


318


319


