Approximating NP-hard Problems
Efficient Algorithms and their Limits

Prasad Raghavendra

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2009

Program Authorized to Offer Degree: Computer Science and Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Prasad Raghavendra

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

_____

Venkatesan Guruswami

Reading Committee:

_____

Venkatesan Guruswami

_____

Paul Beame

_____

James R Lee

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Approximating NP-hard Problems
Efficient Algorithms and their Limits

Prasad Raghavendra

Chair of the Supervisory Committee:
Associate Professor Venkatesan Guruswami
Computer Science and Engineering

Most combinatorial optimization problems are **NP**-hard to solve optimally. A natural approach to cope with this intractability is to design an "approximation algorithm" – an efficient algorithm that is guaranteed to produce a good approximation to the optimum solution. The last two decades has witnessed tremendous developments in the design of approximation algorithms mostly fueled by convex optimization techniques such as linear or semidefinite programming.

In this thesis, we present algorithmic and lower bound results that characterize the power and limitations of these techniques on large classes of combinatorial optimization problems. The thesis identifies a specific simple semidefinite program and demonstrates the following:

– This semidefinite program yields the optimal approximation to every problem in one of the large classes such as constraint satisfaction problems (CSP), metric labeling problems and ordering constraint satisfaction problems under the Unique Games Conjecture (**UGC**). To show this, we exhibit a general black-box reduction from hard instances to a linear/semidefinite program to corresponding hardness results based on the **UGC**. Not only does this confirm a widely suspected connection, it settles the approximability of classic optimization problems such as CSPs, Multiway Cut and Maximum Acyclic Subgraph under **UGC**.

– The thesis presents a generic algorithm for constraint satisfaction problems (CSP) based on this semidefinite program. Irrespective of the truth of **UGC**, this generic algorithm is guaranteed to obtain an approximation at least as good as all known algorithms for specific CSPs.

– Independent of the truth of **UGC**, the approximation obtained by this semidefinite program cannot be improved by any convex relaxation that is obtained by including any valid constraints on at most $O(2^{(\log \log N)^{\frac{1}{4}}})$ vectors.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

It was after extensive deliberation, and with still lingering doubts that I flew a few thousand miles away from home to start graduate school. With the four amazing years that followed, this decision is no doubt the best I ever made. As this wonderful journey draws to a close with this thesis, I take this opportunity to thank all the people that made it possible.

I am greatly indebted to my advisor Venkat Guruswami, I couldn't have asked for a better advisor! Along with his brilliance, technical mastery, and an amazingly wide research program, Venkat always seemed to have infinite amounts of time to patiently listen to every one of my ridiculous ideas and accompany me on those thousand dead-ends before a successful proof. Over these years, I have immensely enjoyed those conversations through which I learnt a great deal both technically and otherwise. For all this and lot more left out – Thank you Venkat!

My thanks to Paul Beame and James Lee for serving on the thesis committee. Special thanks to Paul Beame for his meticulous proof reading of the thesis. All credit goes to him for parts of the thesis that look immaculate, and any errors that have crept in are solely due to me not giving him sufficient time to proof-read.

Many thanks to Rajsekar Manokaran, Seffi Naor and David Stuerer for their collaborations that led to several of the results in this thesis. Special thanks are due to David Steurer whose writing has influenced much of the notation in this thesis.

I had the rare opportunity to spend about half my time during graduate school visiting a number of places. While still learning the ropes in my first year, I had the pleasure of interning with Ramarathnam Venkatesan and Peter Montgomery at Microsoft Research Crypto Group. Following year, I enjoyed the summer collaborating with Yossi Azar, Uriel Feige, Seffi Naor and Mohit Singh at Microsoft Research Theory Group during which time I started working on what became the core of this thesis.

A major portion of the work in this dissertation was carried out, in the invigorating and vibrant environment of Princeton University between Sept 2007-June 2008. I am greatly indebted to Sanjeev Arora, Boaz Barak, Moses Charikar and the theory students at Princeton University for the highly productive and wonderful time I spent there. My special thanks to Parikshit Gopalan for making what was already a great time at Princeton even more memorable.

In summer 2008, I had a wonderful time working with Vitaly Feldman, Swastik Kopparty and T.S.Jayram at IBM Almaden Research Center. Finally, my last semester was spent visiting Carnegie Mellon University in 2009 where I enjoyed talking to Yi Wu and Ryan 'O Donnell.

I am greatly indebted to all these institutions I have visited and the the numerous people involved for hosting me. These visits have widened my research interests, built valuable

collaborations and greatly enriched my experience in graduate school. A sincere thanks to the support from NSF grants CCF-0343672, CCF-0835814, and a David and Lucile Packard Fellowship during the last four years.

Special thanks to my great roommates John and Vibhor for making the relatively short period of two years that I spend at Seattle memorable. I thoroughly enjoyed the company of friends/collaborators at UW: Atri, Ning, Gyanit, Thach, Punya, Alex, Abhay, Nilesh, Sumit, Deepak and Sashi at UW. My thanks to the graduate students Aditya, Aravindan, Rajsekar, David, Aniruddh, Moritz and Eddy, and my roommate Yogesh for all the good times at Princeton. My heartfelt thanks to roommates Nitya and Hari for hosting me for an entire semester at Pittsburgh.

I wish to thank C. Pandu Rangan, C.R. Pranesachar, B.J. Venkatachala, C.S. Yogananda and all the other teachers who have inspired me to pursue the mathematics of computer science over the years. Finally, it would be difficult to express the debt of gratitude that I owe my family: my parents Nagaraj and Jayashree, my sister Chaitra and Murali, for their love, relentless support and encouragement.

DEDICATION

To my Parents

Chapter 1

INTRODUCTION

Combinatorial optimization is one of the most commonly arising computational tasks. In a combinatorial optimization problem, the goal is to find a solution that maximizes or minimizes a certain objective value among a discrete set of feasible solutions. To demonstrate the ubiquity of combinatorial optimization problems, and to give the reader a flavor of the problems studied in this dissertation, we present a few examples below.

Problem 1. A 3-CNF boolean formula consists of a set of variables and clauses of the form $x_i \vee x_j \vee x_k$. Find an assignment to the variables that satisfies the *maximum* number of clauses.

This problem belongs to the class of Constraint Satisfaction Problems (CSP) that have numerous applications, from artificial intelligence and planning to VLSI chip design.

Problem 2. Given a graph and three designated nodes called terminals, decompose the graph into three parts containing one terminal each, while *minimizing* the number of edges between the partitions.

This is an example of a Metric Labelling Problem that arises naturally in several contexts such as image segmentation.

Problem 3. Given the results of all chess games played in major tournaments over a year, let us suppose we wish to rank the players. Specifically, we would like to find a ranking that agrees with the results of *maximum number* of games. Here the result of a game agrees with the ranking if the higher ranked among the two players wins.

Here we are trying to find an ordering/permutation/ranking that satisfies the maximum number of given constraints. This problem is a classic example of an Ordering CSP (OCSP).

Problem 4. Given a system of sparse linear equations over real numbers that is completely consistent, it is easy to find a solution using Gaussian elimination. Suppose that the system is not completely consistent in that some of the equations are erroneous. In this case, the natural optimization problem to consider is to find a solution that satisfies the *maximum number* of equations.

While combinatorial optimization problems occur very naturally in practice, many seemingly different tasks such as pattern classification, clustering, and learning can also be posed as combinatorial optimization. For instance, consider the following problem,

Problem 5. Suppose that we are given a set of points labelled $+$ or $-$ in a high dimensional space. For instance, this could be a representation of a set of emails with the labellings indicating whether they are spam or non-spam. Find a halfspace (also referred to as a perceptron) that correctly classifies the *maximum* number of the given set of points.

Unfortunately, for an overwhelming majority of combinatorial optimization problems, finding the optimal solution turns out to be **NP**-hard. Therefore, unless $\mathbf{P} = \mathbf{NP}$, there are no efficient algorithms to solve any of the above problems optimally.

To cope with this intractability, one settles for solutions that are approximately optimal. For instance, can one design an efficient algorithm that always outputs a solution that is at least half as as good as the optimum? Formally, an algorithm is an α-approximation to a problem, if on every instance the solution output by the algorithm has value within an α-factor of the optimal solution. In this dissertation, in contrast to most previous work we will consider not just a single approximation ratio, but the entire spectrum of approximations that depend on the quality of the optimal solutions.

The following question stems naturally from the notion of approximation algorithms:

> For a given combinatorial optimization problem, what is the best approximation to the optimum that can be efficiently computed?

There are two important facets to answering the above question, designing approximation algorithms and showing that no efficient algorithm can approximate better (hardness result).

In this dissertation, we present both optimal approximation algorithms and hardness results that apply in general to entire classes of problems such as the constraint satisfaction problems. More precisely, we exhibit a generic algorithm for every constraint satisfaction problem whose approximation behaviour is at least as good as that of all known algorithms. The generic algorithm relies on a convex optimization subroutine, specifically a simple semidefinite program. Furthermore, we show that under the well-known Unique Games Conjecture, this simple semidefinite program yields the best approximation for every problem in one of the classes: constraint satisfaction problem (CSP), ordering CSP and metric labelling problem. Specifically, this demonstrates that existing algorithmic techniques have reached a common barrier on every problem in these classes, a barrier that is achieved by the generic algorithm and encapsulated by the Unique Games Conjecture.

In the remainder of the introduction, we lay out the motivation and context for our work by surveying the relevant milestones in the study of approximation algorithms and highlighting the main contributions of this dissertation.

1.1 Motivation

It is beyond any doubt that **NP**-hard combinatorial optimization problems are indispensable in many practical applications. Therefore, it is obligatory to develop *heuristics* to cope with their intractability. Any systematic study of heuristics should provide ways to measure, analyze, compare, and improve their performance. It is our viewpoint that the study of approximation algorithms is an attempt in this direction, in that it is a *systematic, theoretical* study of heuristics. In what follows, we make an attempt to justify this position.

Why worst-case analysis? By definition, an **NP**-hardness result concerns the performance of any efficient algorithm over the worst-case choice of input. Therefore it is conceivable that efficient algorithms could be designed for subclasses of inputs that occur in practice. A large body of work has emerged motivated along these lines, some of whose highlights include:

– Efficient algorithms for problems over special classes of graphs such as sparse graphs, dense graphs, planar or low-genus graphs, and perfect graphs.

– Algorithms that are guaranteed to perform well with high probability when the input is generated from a certain distribution.

– Parametrized Complexity: Here the running time of an algorithm is not only measured in terms of the input size, but also an additional parameter associated with the input. In particular, the goal is to design algorithms that are efficient on inputs where the associated parameter is small.

In many real-life settings, the inputs are generated from complex processes that makes discovering additional structure in them a daunting task. Furthermore, some optimization problems remain hard even on special classes of inputs such as sparse graphs or those with some small parameter.

There is little reason to believe that the general distributions that have been analyzed truly reflect problems in real-life settings. The worst-case guarantees that approximation algorithms provide make them applicable even in contexts where there is little or no additional information available about the inputs. In this dissertation, we restrict ourselves to approximation algorithms that have provable guarantees over all inputs.

Why these problems? There are a few classic problems such as Max Cut and Max 3-Sat that have received considerable amount of attention in approximation algorithms literature. One possible intent of this extensive study is to determine the best possible approximation ratios/curves for these specific problems. However, the exact value of approximation ratio for, say, Max Cut is probably of little value in practice. More often than not, a combinatorial optimization problem that arises in practice is a variant or entirely different from any problem appearing in approximation algorithms literature.

The real motivation to extensively study problems like Max Cut is that they serve as simple examples that help us understand the power and limits of existing algorithmic techniques. An ambitious goal is to translate insights from these classic examples to general results for arbitrary combinatorial optimization problems, thereby making the results applicable in practical scenarios. This dissertation makes progress towards this goal by exhibiting very generic algorithms and hardness results that hold for every problem in large classes such as constraint satisfaction problems and might be more likely effective in practice.

Approximation ratios and approximation curves Recall that an α-approximation algorithm is one that always outputs a solution whose value is within α factor of the optimum. This is just one possible measure for the quality of a heuristic. In some cases, the correct measure would be an additive approximation where the algorithm's output is within α additive error of the optimum. A more refined measure is to plot the the output of the algorithm as a function of the optimum value of the instance. Formally, the *approximation curve* $\alpha(c)$ of an algorithm $\mathcal{A}$ is the plot of the value of the solution returned by the algorithm $\mathcal{A}$ on instances where the optimum value is c. The *approximation curve* encodes a lot of information about the performance of the heuristic algorithm $\mathcal{A}$.

The study of approximation algorithms encompasses all these measures of performance for heuristics. With this broad agenda, this dissertation studies the performance and limits of approximation algorithms by attempting to map out their entire approximation curves instead of approximation ratios alone.

1.2 Relaxation and Rounding Methodology

A vast majority of approximation algorithms follow a two step approach consisting of relaxation and rounding. To describe the context and contributions of the dissertation, a rudimentary understanding of this popular paradigm would be useful.

Relaxation By virtue of the fact that the space of feasible solutions is discrete, every combinatorial optimization problem can be reformulated as an optimization problem with integer-valued variables, i.e., an integer program. In other words, given an instance of a combinatorial optimization problem, we can encode it as maximizing or minimizing a function of a set of variables (say $\{x_1, \ldots, x_n\}$) that take certain integer values (say $\{0, 1\}$) and are required to satisfy a set of constraints.

Being a reformulation, the resulting integer program is also **NP**-hard. Technically, the intractability of the integer program stems from the *non-convexity* of the space of solutions. The main idea is to relax the constraints of the integer program so as to make it tractable. Specifically, we will relax the condition that the variables are to be assigned values 0 or 1 only, and permit them to be assigned real numbers, or even vectors.

By a suitable relaxation, the intractable integer program is converted to a convex optimization problem that can be solved in polynomial time.

Clearly, the relaxation permits more solutions than the original integer program does. Hence, it immediately follows that the optimum of the relaxation is at least as good as the optimum of the integer program. Formally, let $\Im$ be an instance of a minimization problem. If $\mathrm{opt}(\Im)$ denotes the value of the optimum solution to the instance $\Im$, and $\mathsf{Conv}(\Im)$ denotes the optimal value of the corresponding relaxation, then $\mathsf{Conv}(\Im) \leqslant \mathrm{opt}(\Im)$.

The value of $\mathsf{Conv}(\Im)$ serves as an efficiently computable estimate for $\mathrm{opt}(\Im)$. Of course, it is completely unclear at this point how good an estimate $\mathsf{Conv}(\Im)$ is. The "integrality gap" is a coarse measure of the quality of the estimate $\mathsf{Conv}(\Im)$. In particular, the integrality gap is the worst-case ratio between $\mathrm{opt}(\Im)$ and its estimate $\mathsf{Conv}(\Im)$ over all instances $\Im$. The "hard" instances for the relaxation, where the worst-case ratio is achieved are referred to as *integrality gap instances*.

Rounding The optimal solution to the relaxation will probably consist of an assignment of real or possibly vector values to the variables. However, only integer-valued assignments to the variables will correspond to a possible solution to the original combinatorial optimization problem $\Im$. In the next step, a procedure (*a rounding scheme*) is devised to "round" the real or vector valued assignment to the variables into an integer assignment. The challenge is to devise a *rounding scheme* that incurs only a small loss in the objective value.

Formally, a *rounding scheme* of an α-approximation algorithm is an algorithm that takes as input the optimal solution to the relaxation Conv consisting of real or vector values and

rounds the solution to an integer/discrete solution by losing at most an α-factor in the value.

Since $\mathsf{Conv}(\Im) \leqslant \mathrm{opt}(\Im)$, the value of the rounded solution is at most α-times the value of the optimal solution. Therefore, the convex relaxation along with the above rounding scheme yields a α-approximation to the problem.

1.3 Relaxation Techniques

An overwhelming majority of approximation algorithms use a specific type of convex relaxation - linear programming. A linear program consists of either maximizing or minimizing a linear function over real-valued variables while satisfying certain linear constraints among them. There are numerous techniques to solve linear programs, while the simplex method and its variants are used extensively in practice, others such as interior point methods (see Alizadeh [2], Nemirovsky et al. [127]) are provably efficient.

Not only have linear programming been instrumental in the design of approximation algorithms for a vast array of combinatorial optimization problems, it is a technique that is extensively used in practice. We refer the reader to the book by Vazirani [158] for a glimpse of the pervasiveness of linear programs in the field of approximation algorithms.

Apart from linear programs, the other relaxation technique that has proved useful in this context are semidefinite programs. A semidefinite program consists of vector-valued variables, with linear constraints on their inner products. The objective being optimized is a linear function of the inner products of the variables. Semidefinite programs can be solved in polynomial time using the ellipsoid method (see Grotschel et al. [70]) or interior point methods (see Alizadeh [2], Nemirovsky et al. [127]). More precisely, semidefinite programs can be optimized to within an error ε in time polynomial in $\ln \frac{1}{\varepsilon}$ and the size of the program.

Semidefinite programs (SDP) made their appearance in combinatorial optimization as early as 1979 with the classic work of Lovász [117]. The Lovász Theta function, as it is referred to today, is a semidefinite programming relaxation for the Maximum Independent Set problem. In [117], this relaxation was used to show that the Maximum Independent Set problem, and the Minimum Vertex Cover problem can be solved efficiently on perfect graphs.

After what seems to be a hiatus, semidefinite programming (SDP) made a comeback in approximation algorithms with the seminal work of Goemans and Williamson [65] on the Max Cut problem in 1994 . At the time, it was clear by the work of Poljak et al. [135] that linear programming techniques cannot yield an approximation better than $\frac{1}{2}$ for the Max Cut problem. Using a simple semidefinite programming relaxation and an elegant rounding scheme, Goemans and Williamson obtained a 0.878 factor approximation for Max Cut. Since the work of Goemans and Williamson, SDPs have fueled some of the major advances in approximation algorithms. They have found application in problems ranging from Constraint Satisfaction Problems (Charikar et al. [32], Lewin et al. [114], Karloff et al. [93], Hast [79], Charikar et al. [35], Chlamtac et al. [41], Charikar and Wirth [36], Frieze and Jerrum [62], Halperin and Zwick [78], Matuura et al. [122], Zwick [162], Goemans and Williamson [66], Feige and Goemans [54], Zwick [161, 164]) to Vertex Coloring (Karger et al. [90], Chlamtac [40], Arora and Chlamtac [12], Chlamtac and Singh [42]), Vertex Ordering

(Charikar et al. [33], Chor [43]) to Graph decomposition (Frieze and Jerrum [62] Arora et al. [16]), and Discrete optimization (Alon and Naor [4], Charikar and Wirth [36], Khot and Naor [100], Nesterov [128]).

Among the two techniques, linear programming is a special case of its semidefinite counterpart. Therefore, semidefinite programming is arguably one of the most powerful tools in the design of approximation algorithms.

Stronger Relaxations In general, there is no single canonical choice for a convex relaxation of a given optimization problem. There are multiple ways to formulate a combinatorial optimization problem as an integer program and, more importantly, many possible convex relaxations of the integer program.

Furthermore, given a convex relaxation of an integer program, it could be strengthened by introducing additional constraints. As one includes more constraints into a convex relaxation Conv, the optimum $\mathsf{Conv}(\mathcal{I})$ of the relaxation may be a much better estimate of the optimum of the instance. Here again, there are numerous choices of additional constraints that can be included to strengthen a given convex relaxation. In fact, there are several hierarchies of increasingly stronger relaxations called the Lovász-Schriver [118], the Lasserre [110] and the Sherali-Adams hierarchies [150]. On the flipside, including additional constraints increases the size of the relaxation thereby increasing the complexity of optimizing it. For instance, optimizing the relaxation in the k^{th} level of the above mentioned hierarchies takes roughly $n^{\Omega(k)}$ time.

Nearly every algorithm based on semidefinite programming actually makes use of a very simple semidefinite relaxation. A notable exception is the breakthrough result of Arora, Rao and Vazirani [16] that used a stronger semidefinite program with triangle inequalities to obtain an $O(\sqrt{\log n})$ approximation for the Sparsest Cut problem. This has sparked a lot of interest in using stronger semidefinite programs to obtain better approximation algorithms (Chlamtac [40], Arora and Chlamtac [12], Chlamtac and Singh [42]). In particular, hierarchies of stronger SDP relaxations such as Lovász-Schriver [118], Lasserre [110] and Sherali-Adams hierarchies [149] (see Laurent [112] for a comparison) have been touted as tools to push the limits of approximability.

On one hand, this sounds like an opportunity, since stronger semidefinite programs might yield much better approximation ratios. On the other hand, the lack of work in this direction points to the difficulty of using stronger semidefinite programs in algorithms. In the case of linear programming, there has been considerable progress in understanding the limits of strong linear programming relaxations obtained via these hierarchies. However, for the more powerful technique of semidefinite programming, the limits of stronger relaxations are poorly understood.

1.4 Understanding the Limits

With every approximation algorithm devised, the question arises as to whether we could find better approximations? Thus, the design of approximation algorithms is to be complemented with the study of limits of approximability.

The first breakthrough in this direction came in the early 1990's with the celebrated PCP theorem by Arora et al. [15]. The PCP theorem is equivalent to the following hardness result,

Theorem 1.1 (PCP Theorem)**.** *For some constraint satisfaction problem (3-SAT), it is not only **NP**-hard to find an assignment satisfying all constraints, even satisfying more than 0.99999999-fraction of the constraints is **NP**-hard.*

Thus, the PCP theorem is a hardness of approximation result for the problem of satisfying the *maximum* number of clauses in a 3-SAT formula. Over the last decade, using fairly involved reductions starting from the PCP theorem, the approximability of several important computational problems like Minimum Set Cover, Maximum Clique and Coloring have been resolved. Table 1.1 lists a few of the **NP**-hardness results obtained from the PCP theorem.

The pursuit of hardness of approximation results has been an extremely fruitful endeavor. In some cases like Minimum Set Cover, the lower bounds confirmed that a simple heuristic (the greedy algorithm) is the best one can do. While in a few cases like the work on Euclidean Traveling Salesman Problem by Arora [8], it led to new algorithms with better guarantees. From a theoretical standpoint, this pursuit enriched the area with connections to testing, computational learning, and techniques from discrete geometry and Fourier analysis.

Despite considerable success in showing hardness of approximation results, the approximability of many basic problems such as Minimum Vertex Cover still remain open. For instance, the best known approximation algorithm for Minimum Vertex Cover achieves a factor 2 approximation, while the best known lower bound on the approximation possible is 1.36 shown by Dinur and Safra [50]. Hence, either there exists a better approximation algorithm for Minimum Vertex Cover or the hardness result can be improved to a factor 2.

In 2002, Subhash Khot [97] introduced the Unique Games Conjecture, and observed that it could imply tight hardness results. The Unique Games Conjecture is a hardness assumption stating that a certain combinatorial optimization problem is hard to approximate in a strong sense. For the sake of exposition, we present an equivalent formulation of the Unique Games Conjecture (**UGC**) due to Khot et al. [99]. To this end, we first define a special case of the Unique Games problem referred to as the Γ-Max Lin.

Problem 6 (Γ-Max Lin (p))**.** For a natural number p, given a set of linear equations over integers of the form $x_i - x_j \equiv c_{ij} \pmod{p}$, find an assignment satisfying the maximum number of equations.

(**Unique Games Conjecture** [97])**.** For any $\varepsilon, \delta > 0$, there is a large enough number p such that: given a Γ-Max Lin (p) instance for which there is a solution satisfying a $1-\varepsilon$-fraction of equations, it is **NP**-hard to find one that satisfies even a δ-fraction of the equations.

Assuming the **UGC**, hardness of approximation results have been obtained for several fundamental optimization problems. Specifically, Unique Games and, in turn, Γ-Max

Problem	Best Algorithm	**NP**-hardness	UG-Hardness
Max 3-Sat	$\frac{7}{8}$	$\frac{7}{8}$ [86]	
Set Cover	$\ln n$	$\ln n$[56]	
Independent Set	$\frac{n(\log\log n)^2}{\log^2 n}$ [26]	$n^{1-\varepsilon}$ [85]	
Coloring	$\frac{n}{\log^2 n}$ [77]	$n^{1-\varepsilon}$ [57]	
Minimum Vertex Cover	2	1.36 [50]	2 [103]
Max Cut	0.878 [65]	0.941 [86]	0.878 [99]
Max 2-Sat	0.9401 [114]	0.9546[86]	0.9401 [17]
Sparsest Cut	$\sqrt{\log n}$[16]	$1+\varepsilon$	Every constant[38, 104]
Max k-CSP	$\Omega(\frac{k}{2^k})$ [32]	$O(\frac{2^{\sqrt{k}}}{2^k})$[143]	$O(\frac{k}{2^k})$[144, 19, 75]

Table 1.1: Results on Approximability

Lin (p) problems have been reduced to the task of approximating several problems like Minimum Vertex Cover and Max Cut up to a certain ratio. Therefore, if the **UGC** is true (the Unique Games problem is **NP**-hard), then it would imply that approximating Minimum Vertex Cover and Max Cut are **NP**-hard beyond a specific threshold.

For the sake of clarity, we will say a problem is Unique Games-hard or just UG-hard, if there is a reduction from Unique Games to the problem. For instance, by the work of Khot and Regev [103], the problem of approximating Minimum Vertex Cover better than factor 2 is UG-hard. If the **UGC** is true, then a problem that is UG-hard is indeed **NP**-hard.

We list a few of the UG-hardness results in Table 1.1. Observe that UG-hardness results exactly match the best algorithms known for classic problems like Minimum Vertex Cover, Max Cut and Max 2-Sat.

The Unique Games Conjecture (**UGC**) remains a notorious open problem today. Not only is the conjecture unresolved, but there is no consensus among theorists about its truth. There have been several attempts at disproving the conjecture (Trevisan [155], Gupta and Talwar [72], Chlamtac et al. [41], Charikar et al. [35] and Arora et al. [14]) and much lesser progress towards proving it (Feige and Reichman [60]).

While there is no consensus on its truth, studying the **UGC** and its implications has been extremely fruitful. Several unconditional results (results that hold irrespective of the truth of the **UGC**) have been obtained via UG-hardness reductions. Many of these results have vastly improved our understanding of the power and limitations of semidefinite programming - arguably the most powerful technique in approximation algorithms. For instance, the well-known Goemans-Linial conjecture asserted that a semidefnite program yields a constant factor approximation for a fundamental graph partitioning problem, namely the Sparsest Cut problem. In a breakthrough result, Khot and Vishnoi [104] used UG-hardness reductions to disprove the conjecture. Starting with this work, UG-hardness reductions have exposed the limits of semidefinite programs on numerous combinatorial optimization problems.

Finally, the study of the **UGC** and its implications could shed light on an approach to its eventual resolution. A confirmation of the **UGC** will resolve the approximability of fundamental combinatorial optimization problems, while a refutation is likely to lead to new and powerful algorithmic techniques.

1.5 Integrality Gaps vs Hardness Results

The Minimum Set Cover problem is a classic combinatorial optimization problem. There is a simple natural linear programming relaxation for the problem that has an integrality gap of $\ln n$. It was shown by Feige [56] that the best possible approximation ratio for the problem is exactly $\ln n$. It is rather surprising that the integrality gap of the relaxation matches the best possible approximation ratio. Indeed, this is just one example of a long and widely observed phenomena in approximation algorithms.

For a convex relaxation, recall that the integrality gap is the worst possible ratio between the optimum of the relaxation and the optimal solution to the instance. An integrality gap instance for a relaxation Conv is a *hard instance* for a particular algorithm given by the relaxation Conv. Specifically, the integrality gap instance could possibly be an easy instance (better approximated) for a different convex relaxation or algorithm.

For a combinatorial optimization problem, let the *hardness threshold* refer to the best possible approximation computable in polynomial time. Therefore, the *hardness threshold* measures the limits of all polynomial time algorithms, not just a specific convex relaxation.

Apriori, there is little reason to suspect that the integrality gap of a particular convex relaxation to a problem would equal its *hardness threshold*. After all, the integrality gap is the limit of a specific convex relaxation, while the hardness threshold measures the limit of all polynomial time algorithms. Yet, it has often transpired that the integrality gap of a simple convex relaxation exactly matches the *hardness threshold*. In other words, for many classic combinatorial optimization problems, it has so transpired that certain simple convex relaxations yield the optimal approximation computable in polynomial time.

This connection between integrality gaps and hardness thresholds have grown stronger with the advent of the Unique Games Conjecture. For the Max Cut problem, the semidefinite programming based algorithm of Goemans and Williamson yields an approximation ratio which is an irrational number close to 0.878. Surprisingly, the UG-hardness result for Max Cut by Khot et al. [99] exactly matches the approximation factor! Even in the case of Max 2-Sat, the UG-hardness result by Austrin [17] exactly matches the approximation obtained using a semidefinite program due to Lewin et al. [114].

In earlier cases like Minimum Set Cover, the fact that integrality gap equals the hardness threshold seemed more of a coincidence. However, with UG-hardness results as in the case of Max Cut, the integrality gap instances appeared to play a role in the proof of the hardness result. Furthermore, in the reverse direction, UG-hardness results paved the way to new SDP integrality gaps for Sparsest Cut and Max Cut (Khot and Vishnoi [104]).

Summarizing, the somewhat mysterious and long-observed relation between integrality gaps and hardness thresholds has grown stronger with the advent of the Unique Games Conjecture. While many earlier works such as Khot et al. [99], Austrin [17] and, Khot

and O'Donnell [101] hinted at its existence, the work of Austrin [18] established a partial connection between SDP integrality gaps and UG-hardness results.

1.6 Brief Summary of Contributions

The dissertation obtains both approximation algorithms and lower bound results, many of which are based on the Unique Games Conjecture. Here, we present a brief summary of the main contributions of this dissertation. By virtue of its brevity, the summary is necessarily incomplete and imprecise. We refer the reader to Section 2.6 for a detailed chapter-by-chapter description of the contributions of the dissertation.

The rest of the dissertation is organized into three parts: algorithmic techniques, the UG Barrier and unconditional lower bounds.

Algorithmic Techniques We present a generic approximation algorithm that applies to every constraint satisfaction problem (CSP). This generic algorithm obtains an approximation that is at least as good as all known algorithms designed for specific CSPs. In turn, it unifies a large body of existing work on semidefinite programming based algorithms for constraint satisfaction problems. The SDP relaxation underlying the generic algorithm which we term the LC relaxation, is also applicable to the classes of metric labelling problems and ordering CSPs.

The Unique Games Barrier Among the main contributions of the dissertation is the direct reduction from integrality gaps for semidefinite programs to UG-hardness results. This confirms the long-suspected connection between integrality gaps and hardness of approximation.

Harnessing this connection, we show that the LC semidefinite program yields the optimal approximation for every constraint satisfaction problem under the **UGC**. While this unifies several earlier UG-hardness results for specific CSPs like Max Cut, it also asserts that the generic algorithm presented in our algorithmic techniques section is optimal for every CSP under the **UGC**.

Extending this connection further, we obtain optimal hardness results for every metric labelling problem. More precisely, we show that a simple linear program that is referred to in the literature as the "earthmover relaxation" yields the best approximation for every metric labelling problem under the **UGC**. Specifically, this settles the approximability of the classic problem of Multiway Cut that belongs to this class.

Developing technical machinery to work with orderings, we show that the fundamental ordering problem of Maximum Acyclic Subgraph is UG-hard to approximate to a factor better than $\frac{1}{2}$. More generally, the LC relaxation yields the optimal approximation for every ordering CSP under **UGC**.

We also outline some other interesting consequences of the direct connection we establish between integrality gaps and UG-hardness results. Specifically, we devise an algorithm to compute the Grothendieck constant - an important mathematical constant determining whose value remains a long-standing open question.

Summarizing, we demonstrate that the Unique Games is a common barrier that existing algorithmic techniques seem to have reached on every problem in large classes like CSPs, metric labelling problems and ordering CSPs. Furthermore, under **UGC**, the best approximation for all these problems is given by one of the simplest semidefinite programs – the LC relaxation.

Unconditional Lower Bounds The LC relaxation is simple enough that it can be solved in near linear time up to any constant additive error (see Steurer [151]). This suggests that a stronger relaxation could yield better approximations to these combinatorial optimization problems, thus disproving the **UGC**. In Chapter 12, we obtain preliminary results towards ruling out this possibility. Specifically, we show that for any constraint satisfaction, metric labelling or ordering CSP, the SDP integrality gap does not improve on including *all* valid constraints on $O(2^{(\log \log N)^{1/4}})$ vectors to the LC relaxation (here N denotes the number of variables in the combinatorial optimization problem). We also obtain hard instances for the Balanced Separator problem thereby constructing metric spaces that are locally L_1 embeddable yet require large distortiion to embedd in to L_1 globally.

This result could be considered as very preliminary evidence for the intractability of Unique Games. In the final chapter, generalizing the work of Hastad [86], we present strong **NP**-hardness result for approximating sparse linear equations over reals.

1.6.1 The Approximation Landscape

Here we succinctly summarize the rough picture of the landscape of approximability that has emerged in the course of this dissertation.

- Under the **UGC**, the LC semidefinite program which is arguably one of the simplest SDP relaxations yields the optimal approximation for every constraint satisfaction problem, metric labelling or an ordering CSP. All known algorithms for these classes of problems obtain an approximation weaker than the LC relaxation.
- For the LC relaxation applied to CSPs, there are generic algorithms to round the fractional solution optimally (to its integrality gap), and also compute the value of the optimal approximation ratio (integrality gap).
- While the LC relaxation is simple, there is evidence showing that certain more complicated SDP relaxations do not yield any better approximation. Specifically, irrespective of the **UGC**, including any additional constraints on up to $O(2^{(\log \log N)^{1/4}})$ vectors in the LC relaxation does not improve the approximation ratio.

Chapter 2

PRELIMINARIES AND ORGANIZATION OF THESIS

2.1 Relaxation and Rounding: Examples

In this section, we will describe the relaxation-rounding paradigm using two examples Minimum Vertex Cover and Max Cut both of which have been fundamental and influential combinatorial optimization problems.

2.1.1 Minimum Vertex Cover

The Minimum Vertex Cover problem is defined as follows,

Problem 7 (Minimum Vertex Cover)**.** An instance of Minimum Vertex Cover problem consists of a graph $G = (V, E)$ over a set of vertices V and edges E. A vertex cover is a set of vertices S, such that every edge in the graph, has one of its endpoints in the set S. The goal is to find a vertex cover S with the minimum number of vertices.

Relaxation By virtue of the fact that space of feasible solutions is discrete, every combinatorial optimization problem can be reformulated as a constrained optimization with integer valued variables, i.e., an integer program. In other words, given an instance of a combinatorial optimization problem, we can encode it as maximizing or minimizing a function of a set of variables that takes certain integer values (say $\{0, 1\}$) and are required to satisfy a set of constraints.

Let $G = (V, E)$ be an instance of the Minimum Vertex Cover problem. To formulate it as an integer program, introduce a variable X_v for every vertex v in the graph G. The variable X_v indicates whether the vertex v belongs to the vertex cover or not. Specifically, X_v is a $\{0, 1\}$-variable defined as follows:

$$X_v = \begin{cases} 1 & \text{if } v \text{ is in vertex cover}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Consider an edge (u, v) in the graph G. In a valid vertex cover, at least one endpoint of the edge (u, v) must belong to the vertex cover. Hence, the variables X_u, X_v corresponding to u and v must satisfy $X_u + X_v \geqslant 1$. The integer program in variables X_v is given by,

Integer Program for Minimum Vertex Cover		
Minimize	$\sum_{v \in V} X_v$	(size of the vertex cover)
Subject to	$X_u + X_v \geqslant 1$	for each edge (u, v) in the graph
	$X_u \in \{0, 1\}$	for every vertex u.

Being a reformulation, the resulting integer program is also **NP**-hard. Technically, the intractability of the integer program stems from the *non-convexity* of the space of solutions. The main idea is to relax the constraints of the integer program so as to make it tractable. Specifically, we will relax the condition that the variables are to be assigned values 0 or 1 only. We will let the variables X_u to take real values in the range $[0, 1]$. The resulting relaxation is what is referred to as a *linear program*, and can be solved efficiently.

Linear Programming Relaxation for Minimum Vertex Cover		
Minimize	$\sum_{v \in V} X_v$	(size of the vertex cover)
Subject to	$X_u + X_v \geqslant 1$	for each edge (u, v) in the graph
	$0 \leqslant X_u \leqslant 1$	for every vertex u.

Clearly, any solution to the integer program is also a valid solution to the linear programming relaxation. In other words, the relaxation permits more solutions than the original integer program. Hence, it immediately follows that the optimum of the linear program is at most the optimum of the integer program. Formally, if $\mathrm{opt}(G)$ denotes the size of the minimum vertex cover of graph G, and $\mathsf{Conv}(G)$ denotes the optimal value of the linear program, then $\mathsf{Conv}(G) \leqslant \mathrm{opt}(G)$.

The value of $\mathsf{Conv}(G)$ serves as an efficiently computable estimate for $\mathrm{opt}(G)$. Of course, it is completely unclear at this point how good an estimate $\mathsf{Conv}(G)$ is. The worst case ratio between $\mathrm{opt}(G)$ and its estimate $\mathsf{Conv}(G)$ over all graphs G is referred to as the *integrality gap* of the relaxation. The "hard" graphs for the relaxation, where the worst case ratio is achieved are referred to as em integrality gap instances. Integrality gap serves as a coarse measure of the quality of the estimate $\mathrm{opt}(G)$ and more refined measures will be used whenever possible.

Consider the complete graph on 5 vertices, denoted by K_5. It is easy to check that $X_v = 1/2$ is a feasible solution to the linear program. Thus, $\mathrm{conv}(K_5) \leqslant 5 \times 1/2 = 2.5$ while the size of every vertex cover is at least 4. Consequently, the integrality gap of the relaxation is at least $4/2.5 = 1.6$. More generally, the gap between the two quantities approaches 2 on complete graphs with larger and larger size.

Rounding The value $\mathrm{conv}(G)$ is only an estimate for the value of the minimum vertex cover ($\mathrm{opt}(G)$). However, recall that our initial goal was to find a vertex cover of as small a size as possible. Furthermore, as yet there is no guarantee on the quality of the estimate $\mathrm{conv}(G)$.

On optimizing the linear program, we obtain an assignment of values in the range $[0, 1]$ to the variables $\{X_v\}_{v \in V}$. The value of the variable X_v was intended to indicate whether vertex v belonged to the vertex cover or not. Thus, assigning X_v to some real value (say 0.9) seems to have no apparent meaning in the context of the graph G.

In this light, we will "round" the real valued solution X_v into an integral assignment X_v^* taking $\{0, 1\}$ values, while losing a small factor on the size of the vertex cover. Specifically, for each vertex v, if $X_v \geqslant 1/2$ then set X_v^* to 1, else assign X_v^* to 0. Notice that for any edge (u, v), the constraint $X_u + X_v \geqslant 1$ ensures that at least one X_u and X_v is $\geqslant 1/2$. Consequently, at least one of X_u^* and X_v^* is set to 1 for every edge (u, v). Hence, X_u^* is a valid solution to the integral program, and the above procedure is a correct *rounding scheme*.

Now we will analyze the performance of the above *rounding scheme*. Observe that, for each vertex v, X_v^* is always at most $2 \times X_v$ by definition. Therefore, the size of the vertex cover is bounded by

$$\sum_{v \in V} X_v^* \leqslant \sum_{v \in V} 2X_v = 2 \cdot \mathsf{Conv}(G)$$

There are two important conclusions to be derived from the above inequality. First, since $\mathsf{Conv}(G) \leqslant \mathrm{opt}(G)$, the value of the integral solution X_v^* is at most twice the size of the optimal vertex cover. Therefore, the linear programming relaxation along with the above rounding scheme yields a 2-approximation to the Minimum Vertex Cover problem. Furthermore, by definition of $\mathrm{opt}(G)$, any vertex cover of G has value at least $\mathrm{opt}(G)$. In particular, this implies that

$$\mathrm{opt}(G) \leqslant \sum_{v \in V} X_v^* \leqslant 2 \cdot \mathsf{Conv}(G)\,.$$

Thus the rounding scheme serves as a proof that the worst case ratio between $\mathrm{opt}(G)$ and $\mathsf{Conv}(G)$ (the *integrality gap*) is at most 2.

In general, given an instance $\Im$ of a combinatorial optimization problem, it is reformulated as an integer program, and then a convex relaxation $\mathsf{Conv}(\Im)$ of the integer program is constructed. The value of the optimum $\mathrm{opt}(\mathsf{Conv}(\Im))$ for the relaxation is an "estimate" for the actual optimum value $\mathrm{opt}(\Im)$ (which is **NP**-hard to compute). The *integrality gap* of the relaxation is a measure of how good an estimate $\mathrm{opt}(\mathsf{Conv}(\Im))$ is for the actual optimum $\mathrm{opt}(\Im)$.

The optimal solution to the relaxation will probably consist of an assignment of real or possibly vector values to the variables. In the next step, a procedure (*rounding scheme*) is devised to "round" the real or vector valued assignment to the variables, into an integer assignment, with a small loss in the objective value. The relaxation and rounding scheme together yield an approximation algorithm.

2.1.2 The Max Cut *Example*

Problem 8 (Max Cut)**.** Given a graph $G = (V, E)$ with vertices $V = \{v_1, \cdots, v_n\}$ and edges E, find a partition $S \cup \bar{S} = V$ of the set of vertices that maximizes the number of edges *cut* by the partition. An edge $e = (v_i, v_j)$ is cut, if $v_i \in S$ and $v_j \in \bar{S}$ or vice versa.

In their seminal work, Goemans and Williamson [65] used a semidefinite programming relaxation to obtain a 0.878-factor approximation for the problem.

Relaxation First, we will formulate the MAX CUT problem as a quadratic program. For each vertex $v_i \in V$, introduce a variable x_i that takes one of two values $+1$ or -1. The value of x_i indicates which set in the partition $S \cup \bar{S}$, the vertex v_i lies in. Formally, let

$$x_i = \begin{cases} +1 & \text{if } v_i \in S \\ -1 & \text{if } v_i \in \bar{S} \end{cases}$$

In this encoding, an edge $e = (v_i, v_j)$ is cut if and only if $x_i \neq x_j$. In fact, the following holds

$$\frac{(x_i - x_j)^2}{4} = \begin{cases} 1 & \text{if the edge}(v_i, v_j) \text{ is cut} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the MAX CUT problem on the graph G can be expressed as the following quadratic program in the variables $\{x_i\}_{i=1}^n$,

Maximize	$\frac{1}{4} \sum_{(v_i,v_j)\in E} (x_i - x_j)^2$			(Number of edges cut)
Subject to		$x_i^2 = 1$	$\forall i, 1 \leqslant i \leqslant n$	(x_i is either $+1$ or -1)

Being an exact reformulation of the MAX CUT problem, the above quadratic program is **NP**-hard to optimize exactly. Consequently, we will consider a convex or more precisely semidefinite relaxation of the above program.

Recall that the variables x_i are equal to ± 1, or equivalently each x_i is a one-dimensional vector of length 1. Relaxing this constraint, we will require the variables x_i to be unit vectors in a high dimensional space. More precisely, we will now associate a n-dimensional unit vector $\boldsymbol{v}_i$ to each vertex v_i. This yields the following semidefinite programming relaxation.

	$\mathsf{GW}(G)$ Relaxation			(GW)
Maximize	$\frac{1}{4} \sum_{(v_i,v_j)\in E} \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$			(Total Squared Length of Edges)
Subject to		$\|\boldsymbol{v}_i\|_2^2 = 1$	$\forall i, 1 \leqslant i \leqslant n$	(all vectors $\boldsymbol{v}_i$ are unit vectors)

Every feasible solution $\{x_i\}_{i=1}^n$ to the original quadratic program is also a feasible solution to the GW SDP relaxation, since the variables x_i can be thought of as a n-dimensional vectors with $n-1$ coordinates equal to 0. Hence, the GW SDP relaxation is indeed a relaxation of the quadratic program.

It is well known that the above convex relaxation can be optimized efficiently [65], i.e., for any $\varepsilon > 0$, the optimum can be approximated to within ε in time polynomial in $\log 1/\varepsilon$ and the size of the graph G.

Being a relaxation, it is immediate that

$$\mathsf{GW}(G) \geqslant \mathrm{opt}(G) \tag{2.1}$$

where $\mathrm{opt}(G)$ denotes the value of the maximum cut of G. The integrality gap of the relaxation GW is the maximum ratio between the two quantities over all graphs G, i.e.,

$$\mathsf{Gap}_{\mathsf{GW}} = \min_{\text{graph } G} \frac{\mathrm{opt}(G)}{\mathsf{GW}(G)}\,.$$

Often, the term "integrality gap" is used to refer to the graph G for which the above ratio is minimized.

Rounding On solving the GW SDP relaxation, we obtain a set of unit vectors $\{\boldsymbol{v}_i\}_{i=1}^n$ on the n-dimensional space $\mathbb{R}^n$. Recall that the vector $\boldsymbol{v}_i$ corresponds to the vertex v_i in the graph G. Hence, the optimum solution yields an embedding of the graph G on to the n-dimensional unit sphere.

The following *rounding scheme* is a randomized procedure that takes as input a feasible solution to the GW SDP and obtains a cut in the original graph G.

Halfspace Rounding Scheme ($\mathsf{Round}_{\mathcal{H}}$)
Input: A feasible solution $\{\boldsymbol{v}_i\}_{i=1}^n$ for the GW SDP for a graph $G = (V, E)$.

- Sample a random hyperplane $\mathcal{H}$ passing through the origin.

  The hyperplane $\mathcal{H}$ induces a partition of the n-dimensional unit sphere $\mathrm{S_n}$ into two parts of equal hemispheres (say $\mathrm{S_n^+}$ and $\mathrm{S_n^-}$).

- Output the cut induced by the hyperplane $\mathcal{H}$ on the graph G. Formally, output the cut $A \cup \bar{A}$ where $A = \mathrm{S_n^+} \cap \mathrm{V}$ and $\bar{A} = \mathrm{S_n^-} \cap \mathrm{V}$.

Now we will estimate the expected value of the cut output by the above procedure. Consider an edge $e = (v_i, v_j)$ in the graph G. Let θ be the angle between the vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ given by $\theta = \arccos(\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle)$. Note that a random hyperplane $\mathcal{H}$ projects to a random line passing through the origin in the plane containing $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. Therefore, we see that,

$$\Pr_{\mathcal{H}}[e = (v_i, v_j) \text{ is cut}] = \frac{\arccos(\boldsymbol{v}_i \cdot \boldsymbol{v}_j)}{\pi}$$

Let us suppose we execute the halfspace rounding scheme on the optimum solution $\{\boldsymbol{v}_i\}_{i=1}^n$ to the SDP. Let $\mathsf{Round}_{\mathcal{H}}(G)$ denote the expected value of the cut output by the

above rounding procedure. Hence,

$$\mathsf{Round}_{\mathcal{H}}(G) = \mathop{\mathbb{E}}_{\mathcal{H}}[\text{Number of edges cut by } \mathcal{H}] = \sum_{(v_i,v_j)\in E} \frac{\arccos(\boldsymbol{v}_i \cdot \boldsymbol{v}_j)}{\pi}$$

To bound the approximation ratio, the above quantity is to be compared with the optimum cut of the graph G. Instead, by Equation 2.1, we can compare the above quantity with the SDP value of the solution $\{\boldsymbol{v}_i\}_{i=1}^n$,

$$\mathrm{val}(\{\boldsymbol{v}_i\}_{i=1}^n) = \frac{1}{4} \sum_{(v_i,v_j)\in E} \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$$

We will use a "local analysis" (one edge at a time) to compare the two quantities. Specifically, an edge (v_i, v_j) contributes $\frac{1}{4}\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 = (1-v_i\cdot v_j)/2$ to $\mathsf{GW}(G)$, while it adds $\frac{\arccos(\boldsymbol{v}_i\cdot\boldsymbol{v}_j)}{\pi}$ to $\mathsf{Round}_{\mathcal{H}}(G)$. Now we will appeal to the following easy fact:

Fact 2.1.1. *For $x \in [-1,1]$, $\frac{\arccos(x)}{2\pi} \geqslant \alpha_{GW} \times \frac{(1-x)}{2}$ where α_{GW} is an absolute constant greater than* 0.878*.*

Consequently, for each edge (v_i, v_j) we have

$$\frac{\arccos(\boldsymbol{v}_i \cdot \boldsymbol{v}_j)}{2\pi} \geqslant \alpha_{\mathsf{GW}} \times \frac{\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2}{4}$$

Summing the above inequality over all edges we get $\mathsf{Round}_{\mathcal{H}}(G) \geqslant \alpha_{\mathsf{GW}} \times \mathsf{GW}(G)$. As $\mathsf{GW}(G)$ is at least the value of the maximum cut of G ($\mathrm{opt}(G)$), we have $\mathsf{Round}_{\mathcal{H}}(G) \geqslant \alpha_{\mathsf{GW}} \times \mathrm{opt}(G)$, i.e., $\mathsf{Round}_{\mathcal{H}}$ along with the GW SDP relaxation gives a α_{GW}-factor approximation algorithm for Max Cut.

Moreover, the rounding scheme $\mathsf{Round}_{\mathcal{H}}$ serves as a constructive proof that the integrality gap of the GW SDP is at most α_{GW}. Specifically, by definition, the value of the cut returned by $\mathsf{Round}_{\mathcal{H}}$, is at most the optimum cut of G, i.e., $\mathrm{opt}(G) \geqslant \mathsf{Round}_{\mathcal{H}}(G)$. As an immediate conclusion, for all graphs G we have,

$$\mathrm{opt}(G) \geqslant \mathsf{Round}_{\mathcal{H}}(G) \geqslant \alpha_{\mathsf{GW}} \times \mathsf{GW}(G)\,.$$

2.2 Definitions and Terminology

To begin with, we present the formal definition of an approximation algorithm.

Definition 2.2.1. A randomized algorithm $\mathcal{A}$ is said to be an α-approximation algorithm for the maximization problem Λ, if for every instance $\mathfrak{I}$ of the problem:

$$\alpha = \inf_{\mathfrak{I}\in\Lambda} \frac{\mathcal{A}(\mathfrak{I})}{\mathrm{opt}(\mathfrak{I})}$$

Here $\mathcal{A}(\mathfrak{I})$ denotes the expected value of the solution output by the algorithm $\mathcal{A}$, while $\mathrm{opt}(\mathfrak{I})$ denotes the optimum solution to instance $\mathfrak{I}$. The infimum is taken over all in-

stances $\Im$ of the maximization problem Λ. For minimization problems, an α-approximation algorithm should satisfy $\alpha = \sup_{\Im \in \Lambda} \frac{\mathcal{A}(\Im)}{\text{opt}(\Im)}$.

Thus, the approximation ratio achieved by an algorithm $\mathcal{A}$ for a maximization problem Λ, is given by $\inf_{\Im \in \Lambda} \frac{\mathcal{A}(\Im)}{\text{opt}(\Im)}$.

In many cases, the approximation factor is a function of the size of the input instance $\Im$. Then, the approximation factor $\alpha(n)$ as a function of the input size n can be defined as

$$\alpha(n) = \inf_{\substack{\Im \in \Lambda \\ |\Im| = n}} \frac{\mathcal{A}(\Im)}{\text{opt}(\Im)}$$

The approximation ratio is a somewhat crude measure of the performance of an approximation algorithm. For several combinatorial optimization problems such as Max Cut, the difficulty of approximating an instance $\Im$, varies considerably with the value of the optimum $\text{opt}(\Im)$. For instance, a Max Cut instance $\Im$ that is completely satisfiable (all edges can be cut by a partition) can be solved optimally in polynomial time. On the other hand, the Max Cut problem is seemingly hardest to approximate on instances where the optimal cut separates roughly 75% of the edges.

To take advantage of varying difficulty of approximation with the value of the optimum, it will be useful to consider a refined measure of quality of approximation. Specifically, we define the *approximation curve* of an algorithm is defined as follows:

Definition 2.2.2. Let $\mathcal{A}$ be a randomized algorithm for a maximization problem Λ. The *approximation curve* $\alpha(c)$ of an algorithm $\mathcal{A}$, parametrized by the value of the optimum c, is the smallest value of the solution returned by $\mathcal{A}$ on instances with optimum value c, i.e.,

$$\alpha(c) = \inf_{\substack{\Im \in \Lambda \\ \text{opt}(\Im) = c}} \mathcal{A}(\Im)$$

For minimization problems, the infimum in the above definition is replaced by a supremum.

Remark 2.2.1. The approximation ratio can be inferred from the approximation curve as $\alpha = \inf_c {}^{\alpha(c)}/_{c}$.

Remark 2.2.2. In the definition of the approximation curve, the range of values for the parameter c is unspecified. Yet in most cases, after a suitable normalization, the optimum can be assumed to lie in the range $[-1, 1]$.

Relaxations, Integrality Gaps and Rounding Schemes Let Conv denote a convex relaxation either a linear or a semidefinite program for a maximization problem Λ. For an instance $\Im$, let $\mathsf{Conv}(\Im)$ denote the value of the optimum solution to the relaxation Conv on the instance $\Im$.

The integrality gap is a coarse measure of the quality of the estimate obtained by the convex relaxation Conv for the optimization problem Λ. Formally, the integrality gap of Conv is defined as

$$\mathsf{GapRatio}_{\Lambda,\mathsf{Conv}} = \inf_{\Im \in \Lambda} \frac{\text{opt}(\Im)}{\mathsf{Conv}(\Im)}$$

Again, it is useful to measure the integrality gap as a function of the optimum value of the instance. In this light, we define the *integrality gap curve* or simply *gap curve*.

Definition 2.2.3. For a convex relaxation Conv of a maximization problem Λ, the integrality gap curve $\mathsf{Gap}_{\mathsf{Conv}}(c)$ is defined as

$$\mathsf{Gap}_{\Lambda,\mathsf{Conv}}(c) = \inf_{\substack{\Im \in \Lambda \\ \mathsf{Conv}(\Im)=c}} \mathrm{opt}(\Im)$$

Let $(\boldsymbol{V}, \boldsymbol{\mu})$ be a feasible solution to the convex relaxation Conv of the instance $\Im$. The solution $(\boldsymbol{V}, \boldsymbol{\mu})$ could consist of real or possibly vector valued assignment to variables in the convex relaxation Conv. Let $\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu})$ denote the objective value of the solution $(\boldsymbol{V}, \boldsymbol{\mu})$.

To obtain an integral solution, the assignment $(\boldsymbol{V}, \boldsymbol{\mu})$ is to be *rounded* into an integral assignment. Formally, the Round procedure takes as input a feasible solution to the relaxation Conv, and outputs a solution to the original instance $\Im$. Let $\mathsf{Round}(\boldsymbol{V}, \boldsymbol{\mu})$ denote the expected value of solution returned by the randomized procedure Round.

Definition 2.2.4. For a maximization problem Λ, the *rounding ratio* of a randomized procedure Round is the following worst case ratio

$$\mathsf{RoundRatio}_{\Lambda,\mathsf{Conv}} = \inf_{\Im \in \Lambda} \frac{\mathsf{Round}(\boldsymbol{V}, \boldsymbol{\mu})}{\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu})}$$

A rounding scheme is said to *achieve the integrality gap*, if the *rounding ratio* is equal to the integrality gap ratio for the relaxation.

Here again, we define the *rounding curve* $\mathsf{Round}_{\mathsf{Conv}}(c)$ along the lines of *approximation* and *integrality gap* curves.

$$\mathsf{Round}_{\Lambda,\mathsf{Conv}}(c) = \inf_{\mathrm{val}(\boldsymbol{V},\boldsymbol{\mu})=c} \mathsf{Round}(\boldsymbol{V}, \boldsymbol{\mu})$$

For the sake of brevity, we will drop either Λ or conv from the subscripts, when it is clear from the context.

Remark 2.2.3. Set $(\boldsymbol{V}, \boldsymbol{\mu})$ to be the optimal solution to the relaxation Conv, i.e., $\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathrm{conv}(\Im)$. By definition of $\mathrm{opt}(\Im)$, we have $\mathsf{Round}(\boldsymbol{V}, \boldsymbol{\mu}) \leqslant \mathrm{opt}(\Im)$. Therefore, the rounding curve $\mathsf{Round}_{\mathsf{Conv}}(c)$ is always at most the integrality gap curve $\mathsf{Gap}_{\mathrm{conv}}(c)$ of the relaxation, for a maximization problem.

2.3 Problem Classes

The major problem classes considered in this thesis are constraint satisfaction, graph labelling and ordering constraint satisfaction problems. Definitions of specific problems considered in the thesis have been collected in Appendix A.

Constraint Satisfaction Problems A large number of fundamental combinatorial optimization problems like MAX CUT and MAX 3-SAT fall under the category of constraint satisfaction problems (CSP). The input to a CSP consists of a set of variables that can be assigned values from a finite domain (say $\{0, 1\}$), and a set of constraints among them. The objective is to find an assignment that satisfies the maximum number of constraints. By restricting the type of constraints to different sets of predicates, one obtains different CSPs. For instance, MAX 3-SAT is a CSP over boolean variables where all the constraints are of the form $\ell_i \vee \ell_j \vee \ell_k = \mathsf{TRUE}$, where each ℓ_i, ℓ_j, ℓ_k are either variables or their negations.

Constraint satisfaction problems arise naturally in a vast variety of applications in artificial intelligence, planning and other areas. The study of approximability of constraint satisfaction problems is a rich and influential area, with problems such as MAX CUT, MAX 3-SAT that have been testing grounds for new algorithmic and hardness ideas, a large number of algorithms for specific CSPs based on semidefinite programming, and finally the development of techniques such as long-code testing [86].

Graph Labelling Problems The simplest and probably most familiar problem in this class is the minimum (s, t) cut problem. Given two terminals s and t in a graph G, the goal is to split the graph into two parts separating s and t, while cutting the minimum number of edges. This problem can be solved precisely in polynomial time following the classic work of Ford and Fulkerson.

The 3-WAY CUT problem is a natural generalization of the minimum (s, t) cut problem where there are three terminals that need to be separated from each other. This close variant of minimum (s, t) cut problem already turns turns out to be **NP**-hard. The best known algorithm for the problem achieves an approximation ratio of $\frac{12}{11}$ [46]. Separating not three, but an arbitrary number of terminals is the objective in the MULTIWAY CUT problem. Generalizing this problem further, one obtains the class of 0-EXTENSION and METRIC LABELING problems. Many problems in this class arise naturally in applications to computer vision [107], metric embeddings [108] and analysis [113].

Ordering Constraint Satisfaction Problems Given the results of games between several football teams, let us suppose we wish to rank the teams. Specifically, we would like to find a ranking that agrees with the results of maximum number of games. Here the result of a game agrees with the ranking if the higher ranked among the two teams wins. Formally, the problem can be restated as follows: Given a directed graph find an ordering of its vertices that maximizes the number of edges in the *forward* direction. This problem called the MAXIMUM ACYCLIC SUBGRAPH problem is perhaps the most well-known problem in this class. In particular, the MAXIMUM ACYCLIC SUBGRAPH figured in Karp's early list of **NP**-hard problems [94].

More generally, in an ordering constraint satisfaction problem, there are n objects that are to be ordered. There are constraints such as "A is *before* B", "B is between A and C" on the ordering. The goal is to find an ordering that satisfies the maximum number of clusters. By restricting the type of constraints between objects, one obtains various ordering CSPs.

2.4 Generalized Constraint Satisfaction Problems

In this thesis, we define a class of combinatorial optimization problems referred to as the Generalized Constraint Satisfaction Problems (GCSP). This class is a generalization of the traditional constraint satisfaction problem, which encompasses both maximization and minimization problems.

In a Constraint Satisfaction Problem (CSP), the goal is to satisfy the maximum number of a set of constraints. Formally, a CSP Λ is specified by a family of predicates over a finite domain $[q] = \{1, 2, \ldots, q\}$. Every instance of the CSP Λ consists of a set of variables $\mathcal{V}$, along with a set of constraints $\mathcal{P}$ on them. Each constraint in $\mathcal{P}$ consists of a predicate from the family Λ applied to a subset of variables. The objective is to find an assignment to the variables that satisfies the maximum number of constraints. The arity k of the CSP Λ is the maximum number of inputs to a predicate in the family Λ.

A constraint can be thought of as a payoff that returns either 0 or 1 depending on whether it is satisfied. Roughly speaking, a GCSP is the natural generalization where the constraints are replaced by bounded real-valued payoff functions, and the goal is to maximize the total payoff. Formally,

Definition 2.4.1. A Generalized Constraint Satisfaction Problem (GCSP) Λ is specified by a family of *payoff functions* $\Lambda = \{P | P : [q]^k \to [-1, 1]\}$. The integer k is referred to as the arity of the GCSP Λ, while q denotes the domain size.

A payoff function is said to be of *type* Λ if it belongs to the family Λ.

Definition 2.4.2 (Λ-GenerlizedConstraintSatisfactionProblem (GCSP))**.** An instance $\Im$ of Generalized Constraint Satisfaction Problem Λ is given by $\Im = (\mathcal{V}, \mathcal{P})$ where

- $\mathcal{V} = \{y_1, \ldots, y_m\}$ is the set of variables that are to be assigned values in $[q]$.
- A function $P' : [q]^{\mathcal{V}} \to [-1, 1]$ is said to be of type Λ, if $P'(y) = P(y_{i_1}, \cdots, y_{i_k})$ for some $P \in \Lambda$ and some $y_{i_1}, y_{i_2}, \ldots, y_{i_k} \in \mathcal{V}$. $\mathcal{P}$ is a probability distribution over a payoffs of type Λ.

The objective is to find an assignment $y \in [q]^{\mathcal{V}}$ to the variables that maximizes the expected payoff denoted by $\mathrm{val}(y)$, i.e.,

$$\mathrm{val}(y) = \mathop{\mathbb{E}}_{P \sim \mathcal{P}} \Big[P(y) \Big] .$$

We define the value $\mathrm{opt}(\Im)$ as

$$\mathrm{opt}(\Im) \stackrel{\text{def}}{=} \max_{y \in [q]^{\mathcal{V}}} \mathrm{val}(y) .$$

For a payoff P' of type Λ, let $V(P') \subseteq \mathcal{V}$ denote the set of variables on which P' depends on. Further, the arity of the GCSP Λ will be denoted by k. Now let us see how the classic CSP Max Cut can be posed as a GCSP.

Example 2.4.1. The MAX CUT problem is a GCSP over the domain $[2] = \{0, 1\}$, with a single payoff function P given by:

$$P(0,0) = P(1,1) = 0 \qquad\qquad P(0,1) = P(1,0) = 1$$

The 4-cycle graph is an instance of MAX CUT problem. It would be specified as

$$\Im = \Big\{\mathcal{V} = \{y_1, y_2, y_3, y_4\}, \mathcal{P} = \text{uniform distribution over } \{P(y_1, y_2), P(y_2, y_3), P(y_3, y_4), P(y_4, y_1)\}\Big\}$$

Notice that the definition of GCSP does not restrict the Pay-Off functions to be positive or negative. By permitting negative payoffs, the framework encompasses certain minimization problems too. The idea is that, if we wish to minimize a positive function say f, then it is equivalent to maximizing the negative function $-f$.

Example 2.4.2 (3-WAY CUT)**.** 3-WAY CUT problem is a minimization problem in the class of graph labelling problems mentioned earlier. In 3-WAY CUT, the input consists of a graph with three designated terminals. The goal is to partition the graph, so as to separate the three terminals, while minimizing the number of edges cut. We can

The 3-WAY CUT can be formulated as GCSP over the domain $[3] = \{0, 1, 2\}$, with four payoff functions $\{P_e, P_1, P_2, P_3\}$ given by

$$P_e(x,y) = \begin{cases} -1 & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases} \qquad\qquad P_i(x) = \begin{cases} -1 & \text{if } x \neq i \\ 0 & \text{otherwise} \end{cases}$$

Given a graph $G = (V, E)$ and three terminals t_1, t_2, t_3, we can write: $\mathcal{V} = \{y_v | v \in V - \{t_1, t_2, t_3\}\}$, and for each edge e,

- If $e = (u, v)$ where neither u or v is a terminal, then introduce the payoff $P_e(y_u, y_v)$.
- For an edge $e = (u, t_i)$, introduce the payoff $P_i(u)$.

It is easy to check that the total payoff is exactly the negative of the number of edges cut, by the partition induced by the assignment to variables y_v.

More generally, all constraint satisfaction problems and graph labelling problems over a constant number of labels, can be formulated as GCSP. Thus, the framework includes maximization problems such as MAX CUT, MAX 2-SAT, UNIQUE GAMES and Minimization problems such as MULTIWAY CUT , METRIC LABELING and MIN-SAT.

Remark 2.4.1. The GCSP definition requires a normalization, so as to ensure that the $\mathcal{P}$ is a probability distribution over payoffs. This ensures that the value of any solution to a GCSP instance is in the range $[-1, 1]$. In particular, the integrality gap curve and the approximation curve are defined in the range $[-1, 1]$.

Remark 2.4.2. Unless otherwise mentioned, the domain size q of a GCSP is assumed to be an absolute constant. At times, it will be useful to consider GCSP with a domain size growing with the input (say n). Specifically, in constructing relaxations for ordering CSPs, it is convenient to think of them as a GCSP over a growing domain.

2.5 *Label Cover and* Unique Games

The Label Cover problem which serves as a starting point for numerous reductions in hardness of approximation is defined as follows:

Definition 2.5.1. An instance of Label Cover (c, s) is given by $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, [R], \Pi)$ consists of a bipartite graph over node sets $\mathcal{V}_\Phi$,$\mathcal{W}_\Phi$ with the edges E between them, such that all nodes in $\mathcal{V}_\Phi$ are of the same degree. Also part of the instance is a set of labels $[R]$, and a set of mappings $\Pi = \{\pi_{v \leftarrow w} : [R] \to [R]\}$ for each edge $e = (w, v) \in E$. An assignment A of labels to vertices is said to satisfy an edge $e = (w, v)$, if $\pi_{v \leftarrow w}(A(w)) = A(v)$. The problem is to distinguish between the following two cases:

- There exists an assignment A that satisfies at least a fraction c of the edge constraints Π.
- Every assignment satisfies less than a fraction s of the constraints in Π.

The following strong hardness result for Label Cover has been the starting point for numerous reductions in hardness of approximation.

Theorem 2.1. *[140, 15] There exists an absolute constant $\gamma > 0$ such that for all large enough integer constants n, the gap problem* Label Cover $(1, \frac{1}{R^\gamma})$ *is NP-hard, even when the input is restricted to label cover instances with the size of the alphabet n.*

From the PCP theorem [15], it is easy to show that there exists an absolute constant ε such that Label Cover $(1, 1 - \varepsilon)$ is NP-hard on instances where the size of alphabet is restricted to a small absolute constant (say 7). With this as the starting point, one applies the Parallel Repetition theorem [140] to obtain hardness of label cover instances over larger alphabet. On applying k-wise parallel repetition, the 1 vs $1 - \varepsilon$ gap is amplified to 1 vs c^k for some absolute constant c, while the alphabet size also grows exponentially in k. This yields the above inapproximability result with the required polynomial dependence between the alphabet size n and the soundness $\frac{1}{R^\gamma}$.

2.5.1 Unique Games Conjecture

Definition 2.5.2. An instance of Unique Games represented as $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi, [R])$, consists of a bipartite graph over node sets $\mathcal{W}_\Phi$,$\mathcal{V}_\Phi$ with the edges E between them. Also part of the instance is a set of labels $[R] = \{1, \ldots, R\}$, and a set of permutations $\pi_{v \leftarrow w} : [R] \to [R]$ for each edge $e = (w, v) \in E$. An assignment A of labels to vertices is said to satisfy an edge $e = (w, v)$, if $\pi_{v \leftarrow w}(A(w)) = A(v)$. The objective is to find an assignment A of labels that satisfies the maximum number of edges.

For sake of convenience, we shall use a version of the Unique Games Conjecture which was shown to be equivalent to the original conjecture [103]. To this end, we define the notion of *strong satisfiability* below.

Definition 2.5.3. An assignment $\mathcal{A}$ to a UG instance Φ is said to *strongly satisfy* a vertex $w \in \mathcal{W}_\Phi$ if it satisfies all the edges (v, w) incident at w. The instance Φ is said to be $(1-\gamma)$-*strongly satisfiable* if there exists an assignment $\mathcal{A}$ that *strongly satisfies* $(1-\gamma)$-fraction of the vertices $w \in \mathcal{W}_\Phi$.

First, we define the following decision version of the Unique Games problem.

Problem 9 (Unique Games $(R, 1-\gamma, \delta)$)**.** Given a bipartite Unique Games instance $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi = \{\pi_{v \leftarrow w} : [R] \to [R] \mid e = (w, v) \in E\}, [R])$ with number of labels R, distinguish between the following two cases:

- $(1-\gamma)$-strongly satisfiable instances: There exists an assignment A of labels such that for $1-\delta$ fraction of vertices $w \in \mathcal{W}_\Phi$ are strongly satisfied, i.e., *all* the edges (w, v) are satisfied.
- Instances that are not δ-satisfiable: No assignment satisfies more than a δ-fraction of the edges E.

The Unique Games Conjecture asserts that the above decision problem is **NP**-hard when the number of labels is large enough. Formally,

Conjecture 2.5.1 (Unique Games Conjecture [103])**.** *For all constants $\gamma, \delta > 0$, there exists large enough constant R such that* Unique Games $(R, 1-\gamma, \delta)$ *is* ***NP****-hard.*

For conceptual clarity, we will state our results in terms of the notion of UG-hardness.

Definition 2.5.4 (UG-hardness)**.** A decision problem Λ is UG-hard, if for all $\gamma, \delta > 0$, there exists a polynomial-time reduction from Unique Games $(R, 1-\gamma, \delta)$ to Λ.

Assuming the **UGC**, a decision problem Λ which is UG-hard, is in fact **NP**-hard.

2.6 Results and Organization of thesis

In Chapter 2, we present some basic definitions, set up notation and recall some mathematical preliminaries. The rest of the thesis is divided into three parts, the algorithmic techniques, the Unique Games barrier and unconditional lower bounds.

2.6.1 Algorithmic Techniques

Chapter 4: Linear and Semidefinite Relaxations In this chapter, we present generic linear and semidefinite programming relaxations that are applicable to the class of generalized constraint satisfaction problems (GCSP). The generic SDP relaxation LC unifies many well known semidefinite programs for constraint satisfaction problems in literature. In fact, we will demonstrate that the SDP relaxation LC is stronger than any relaxation used in literature for constraint satisfaction problems. Yet the SDP relaxation LC is surprisingly simple both conceptually, and in terms of complexity. Specifically, for CSPs, the LC relaxation can be solved in near-linear time by the techniques of [13]. At the end of the chapter, we show that the generic SDP relaxation LC satisfies a certain robustness property

by which near-feasible solutions to LC can be "corrected" to make them completely feasible. This robustness property is surprisingly useful, both in solving the program efficiently, and rounding it.

Chapter 5: Generic Rounding Scheme We exhibit a generic rounding scheme that unconditionally achieves the integrality gap of the LC relaxation for every CSP. More generally, the *rounding curve* of the generic scheme for every GCSP Λ, is within an additive error η of the integrality gap curve of the relaxation LC. The error η is a parameter of the algorithm which can be chosen to be any small constant.

As the LC SDP is the stronger than any relaxation used in literature for a CSP, the generic rounding scheme yields a single algorithm for all CSPs, which is at least as good as all the known algorithms [65, 32, 114, 93, 79, 35, 41, 36, 62, 78, 122, 162, 66, 54, 156, 161, 164] for various CSPs. Furthermore, we will see in Chapter 7 that it is UG-hard to obtain an approximation better than that obtained by the generic rounding scheme.

The generic rounding scheme can be succinctly summarized as follows: Reduce the dimension of the SDP solution by randomly projecting it into a constant dimensional space, identify all variables whose projected vectors are close to each other, and solve the resulting instance by brute force!

2.6.2 The Unique Games *Barrier*

In this part, we demonstrate that UG-hardness is a common barrier that approximation algorithms have reached, on a surprising variety of combinatorial optimization problems.

Chapter 6: Dictatorship Tests, Integrality Gaps and UG-hardness results The generality of the UG-hardness results shown in the thesis, stems from the formal connection it establishes between integrality gaps and UG-hardness results. In fact, the direct reduction between integrality gaps and UG-hardness results is one of the main contributions of the thesis.

The direct conversion from integrality gaps to UG-hardness results is to be seen in the context of other interesting connections that have emerged in recent years. In particular, the notions of SDP integrality gaps, UG-hardness results and "Dictatorship tests" - which are constructs useful in hardness reductions, have been intimately tied to each other in many recent works [99, 104] including the thesis. While a conversion from dictatorship tests to UG-hardness results was more or less demonstrated in [99], the work of Khot et al. [104] exhibited a reduction from UG-hardness results to SDP integrality gaps. This work completes the cycle of reductions by exhibiting a generic conversion from SDP integrality gaps to dictatorship tests.

In Chapter 6, we survey all these emerging connections with Max Cut as a running example. In subsequent chapters, we will harness these connections to obtain UG-hardness results, rounding schemes and SDP integrality gaps for several classes of problems.

There is another interesting facet to the direct reduction between SDP integrality gaps to Dictatorship tests exhibited by the thesis. The analysis of the reduction yields an efficient rounding scheme for certain semidefinite programs! This exposes the intriguing connection

between rounding schemes for semidefinite programs - an algorithmic construct, and dictatorship tests - a notion useful in hardness reductions. Analogous connection was discovered earlier in the context of a certain dichotomy conjecture for constraint satisfaction problems. In the final section of Chapter 6, we survey this intriguing connection and the implications of the thesis on the dichotomy conjecture in detail.

Chapter 7: Generalized Constraint Satisfaction Problems In this chapter, we exhibit the generality of the connections surveyed in Chapter 6, by exhibiting the connections for every generalized constraint satisfaction problem. For instance, we show that for every GCSP Λ, it is UG-hard to approximate Λ to an approximation better than the integrality gap of the LC relaxation. Not only does this unify several well known UG-hardness results, it settles the approximability of every CSP, if the **UGC** is true.

Harnessing the connection between dictatorship tests and rounding schemes, we also obtain a generic rounding scheme for the LC relaxation of every GCSP, that is guaranteed to achieve the integrality gap.

Surprisingly, the hardness reduction also yields a generic rounding scheme that achieves the integrality gap of the LC relaxation. In particular, the hardness reduction exposes an intriguing connection between rounding schemes for semidefinite programs - an algorithmic construct, with *dictatorship tests* - a notion useful for showing hardness results. Furthermore, we exhibit an algorithm to compute the integrality gap curve of the LC relaxation to any desired accuracy.

Chapter 8: Graph Labeling Problems For every problem in this class, the best known approximation ratios [46, 30, 53, 107, 71, 39, 6] are achieved using linear programming. More precisely, all known algorithms use a linear program that is either equivalent or strictly weaker than the so called "earth-mover relaxation". However, the hardness results [92, 47] known for the problems in this class, did not match the best known approximation algorithms.

In Chapter 8, we show that for every graph labeling problem Λ, it is UG-hard to approximate the problem to a factor better than the integrality gap of the earth mover relaxation (EM-LP). Recall that the UG-hardness results in Chapter 6 and Chapter 7, matched the integrality gap of a semidefinite program. For the class of graph labelling problems, the UG-hardness we obtain matches the integrality gap of the linear program. Note that this is a stronger conclusion since linear programs are in general weaker than their semidefinite counterparts. To obtain this stronger UG-hardness result, we present a different reduction from integrality gaps to UG-hardness, than the one in Chapters 6 and 7.

Chapter 9: Ordering Constraint Satisfaction Problems The approximability of Maximum Acyclic Subgraph is one of the long-standing open questions in the area of approximation algorithms. While the best known algorithm is the naive algorithm that yields a $1/2$-approximation, obtaining a 0.99-approximation to the problem has not been ruled out.

In Chapter 9, we show that it is UG-hard to approximate the Maximum Acyclic Subgraph problem to a factor greater than $\frac{1}{2}$. This UG-hardness result is the first tight

inapproximability result for an ordering problem. While the central theme is still converting SDP integrality gaps to UG-hardness results, implementing the program for ordering CSPs poses considerable difficulties. In overcoming these technical hurdles, we develop technical machinery such as the notion of influences for orderings.

The techniques developed in the case of Maximum Acyclic Subgraph readily generalize to all ordering CSPs. Specifically, we show that for every Ordering CSP, it is UG-hard to obtain an approximation ratio better than the integrality gap of the explicit semidefinite program similar to the LC relaxation.

Chapter 10: Grothendieck Problem Formally, the input to the Grothendieck problem is a matrix $A = (a_{ij})_{i,j}$ and the goal is to solve the following optimization over $\{-1, 1\}$ values:

$$\text{Maximize} \qquad \sum_{i,j} a_{ij} x_i y_j \qquad \text{for } x_i, y_j \in \{-1, 1\}\,.$$

A natural SDP relaxation yields a constant factor approximation for the Grothendieck problem. In fact, the well known Grothendieck inequality from functional analysis, is equivalent to the fact that the integrality gap of the SDP relaxation is a constant. The value of the integrality gap known as the Grothendieck constant (K_G) is a fundamental mathematical constant determining whose value remains a long standing open problem.

In Chapter 10, we utilize the connections outlined in Chapter 6 to obtain hardness results and algorithms for the Grothendieck problem. Naively translating the techniques from Chapter 6 runs into certain technical difficulties, since the additive error incurred in the reductions presented in Chapter 6, could completely alter the approximation factor. Using a bootstrapping argument, similar to the one used in the proof of Grothendieck inequality, we translate the connections of Chapter 6 into this setting.

Specifically, using black box reductions from integrality gaps to UG-hardness results, we show that it is UG-hard to approximate the Grothendieck problem to a factor better than the Grothendieck constant K_G. By virtue of the connection between dictatorship tests and rounding schemes, this also yields a rounding scheme for the SDP relaxation for Grothendieck problem, that achieves the integrality gap (an approximation factor equal to K_G).

More importantly, as the Grothendieck constant K_G is the integrality gap of a SDP, we obtain an algorithm to compute it to any desired accuracy, thereby taking a step towards determining this fundamental mathematical constant.

Chapter 11: Hardest CSP A natural question regarding the approximability of CSPs is,

> Among all CSPs over a domain $\{0, 1, 2, \ldots, q-1\}$, with each constraint on at most k variables, which CSP is the hardest to approximate?

Using techniques from additive combinatorics, Samorodnitsky and Trevisan [144] exhibited a boolean CSP that is UG-hard to approximate by a factor better than $O(\frac{k}{2^k})$. Indeed, an algorithm of Charikar et al. [32] achieves an approximation of $O(\frac{k}{2^k})$ for every boolean

CSP of arity k, thus making this CSP, the hardest to approximate among boolean CSPs. In Chapter 11, we extend this result to CSPs over larger domains $[q] = \{1, 2, \ldots, q\}$, again by appealing to techniques from additive combinatorics. Specifically, for every prime q, we exhibit a CSP Λ over the domain $\{0, 1, \ldots, q-1\}$ such that it is UG-hard to approximate Λ to a factor better than $\frac{q^2 k}{q^k}$.

2.6.3 Unconditional Lower Bounds

In the final part of the thesis, we obtain hardness of approximation results that hold independent of **UGC**. While some of these lower bounds are **NP**-hardness results, others are against the special class of semidefinite programming based algorithms. On one hand, the lower bounds against semidefinite programs support the validity of the Unique Games barrier, while the **NP**-hardness results find ways to bypass the need for the notion of UG-hardness in specific circumstances.

Chapter 12: Limits of Semidefinite Programming In view of the UG-hardness results, a natural question arises as to whether stronger semidefinite programming relaxations are sufficient to breach this barrier and disprove the UGC. Or does disproving UGC warrant the use of a new technique different from semidefinite programming?

Unfortunately, progress towards answering this compelling question has been slow and difficult. The following possibility is entirely consistent with the existing literature: Even for Max Cut, which is a fairly well-studied problem, including additional valid inequalities on every set of five variables in the standard SDP relaxation yields a better approximation than $\alpha_{\text{GW}} \approx 0.878$, thus disproving UGC. In Chapter 12, we show that the Unique Games Conjecture cannot be disproved by a strong SDP relaxation consisting of the vectors along with *all* valid inequalities on the inner products of $O(2^{(\log \log N)^{1/4}})$ vectors. Specifically, for all problems for which a tight Unique Games hardness is known, a stronger SDP that includes *all* valid constraints on $O(2^{(\log \log N)^{1/4}})$ vectors does not yield a better approximation than the UG-hardness. We also obtain optimal integrality gaps for the basic SDP relaxation strengthened by $O((\log \log N)^{1/4})$ rounds of Sherali–Adams lift-and-project.

We wish to point out that for certain problems like Maximum Acyclic Subgraph and Multiway Cut , integrality gaps for even the simple SDP were unknown.

Furthermore, we show strong SDP integrality gaps for Balanced Separator, and exhibit an N-point negative-type metric such that every subset of size $O(2^{(\log \log N)^{\delta}})$ embeds isometrically into L_1, while the whole metric requires distortion $\Omega((\log \log N)^{\alpha})$ to embedd into L_1 for some absolute constants $\alpha, \delta > 0$.

Chapter 13: Sparse Linear Systems By the work of Khot et al. [99], the Unique Games Conjecture is equivalent to the following hardness for solving sparse linear systems: for every $\varepsilon, \delta > 0$, given a system of sparse linear equations of the form $x_i - x_j = c_{ij} (\mod p)$, modulo a number p such that $1 - \varepsilon$ of the equations can be simultaneously satisfied, it is **NP**-hard to find an assignment satisfying more than δ-fraction of the equations.

It is natural to ask whether the **UGC** is equivalent to a similar hardness of solving sparse linear systems of integers or real numbers. Unfortunately, this question remains

open. Building on the work of Hastad [86], we show that for every $\varepsilon, \delta > 0$, given a system of sparse (3-variables per equation) linear equations over reals such that $1-\varepsilon$ fraction of the equations can be simultaneously satisfied, it is **NP**-hard to find an assignment satisfying more than δ-fraction of the equations.

While the above result is the natural generalization of the celebrated work of Hastad [86], the proof of the result is interesting from an alternate perspective. In proving the result, we obtain an extreme derandomization of Hastad's 3-query dictatorship test. An extreme derandomization of this nature could lead to smaller gadgets (dictatorship tests) for reductions, in turn yielding stronger hardness of approximation results.

Chapter 3

MATHEMATICAL TOOLS

The goal of this chapter is to develop notation and introduce some of the mathematical tools used in this dissertation. Each chapter in the dissertation includes a list of dependencies to sections in this chapter.

3.1 Notation

We list below the notations for various sets that appear in the dissertation.

Notation	Set
$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers
$[q]$	$\{1, 2, \ldots, q\}$
$\mathbb{F}_q$	finite field with q elements
$\mathbb{Z}_q$	$\{0, 1, 2, \ldots, q-1\}$
Δ_q	set of standard basis vectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_q\}$ in $\mathbb{R}^q$
$\blacktriangle_q$	convex hull of Δ_q in $\mathbb{R}^q$
$\blacktriangle(S)$	set of probability distributions over the set S.

For two sets A, B, let A^B denote the set of functions from A to B. For notational convenience, if $B = [n]$ then we will write A^n instead of $A^{[n]}$. An element $\boldsymbol{x} \in A^n$ consists of $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(n)})$ where each $x^{(i)} \in A$. Unless otherwise specified we will use superscripts to index the entries of an element $\boldsymbol{x}$ in a product space A^n.

For the sake of clarity, we will use different typefaces to indicate the type of objects. As a rule, we will use boldface to denote multidimensional objects.

Typeface	Object
$\mathcal{G}, \mathcal{L}, \mathcal{X}$	ensembles of random variables
$\boldsymbol{\zeta}, \boldsymbol{g}, \boldsymbol{\ell}, \boldsymbol{z}, \boldsymbol{x}$	multidimensional random variables
$\mathcal{F}, \mathcal{H}$	one-dimensional real/complex valued functions
$\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{H}}$	multidimensional real/complex valued functions
F, H	one-dimensional multivariate polynomials
$\boldsymbol{F}, \boldsymbol{H}$	multidimensional multivariate polynomials
F, H	one-dimensional functions with range $[q]$
$\boldsymbol{b}, \boldsymbol{v}$	SDP vectors

3.2 Probability Spaces and Random Variables

A probability space $\Omega = (\mathcal{E}, \mu)$ consists of a set of atoms $\mathcal{E}$ and a probability distribution μ over them. The words "event" and "atom" will be used interchangeably. The notation $E \in \Omega$ will denote that atom/event E sampled from the distribution μ.

A real-valued random variable ℓ on the probability space Ω is a function $\ell : \mathcal{E} \to \mathbb{R}$ from events to real numbers. Let $L^2(\Omega)$ denote the set of square-integrable functions over Ω, i.e. $\int_{\mathcal{E}} f^2 d\Omega \leqslant \infty$. For the most part, we will be interested in finite probability spaces. For a finite probability space Ω, $L^2(\Omega)$ is just the set of all functions from $\mathcal{E}$ to $\mathbb{R}$. Alternatively, the vector space $L^2(\Omega)$ consists of all real-valued random variables over the probability space Ω. The following natural inner product is defined over the space $L^2(\Omega)$.

$$\langle \ell_1, \ell_2 \rangle = \mathop{\mathbb{E}}_{E \in \Omega}[\ell_1(E) \cdot \ell_2(E)]$$

Definition 3.2.1. For a random variable x, define $\|x\| = (\mathbb{E}[x^2])^{\frac{1}{2}}$. Similarly, for a function $\mathcal{F} \in L_2(\Omega)$, define

$$\|\mathcal{F}\| = (\mathop{\mathbb{E}}_{\boldsymbol{z} \in \Omega}[\mathcal{F}(\boldsymbol{z})^2])^{\frac{1}{2}} \qquad \|\mathcal{F}\|_\infty = \max_{z \in \Omega} |\mathcal{F}(\boldsymbol{z})|$$

Fact 3.2.1. *For every random variable $\boldsymbol{z}$, we have $\|\boldsymbol{z}\| \leqslant \|\boldsymbol{z}\|_\infty$*

3.3 Harmonic Analysis of Boolean Functions

In numerous applications, the central object of study is a boolean function $\mathsf{F} : \{0,1\}^R \to \{0,1\}$. For reasons that will be clear later, it is convenient to represent the domain of a boolean function as $\{\pm 1\}^R$ instead of $\{0,1\}^R$. Specifically, encode $\{0,1\}$ into $\{\pm 1\}$ as follows:

$$0 \to 1 \qquad\qquad 1 \to -1\,.$$

Using this encoding for the input and the output, we can rewrite $\mathsf{F} : \{0,1\}^R \to \{0,1\}$ as a function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$.

Let us associate the uniform measure over the set $\{\pm 1\}$. The space $L^2(\{\pm 1\}^R)$ consists of functions over $\{\pm 1\}^R$, with the natural inner product defined by

$$\langle \mathcal{F}, \mathcal{F}' \rangle = \mathop{\mathbb{E}}_{\boldsymbol{x} \in \mathbb{Z}_2^R}[\mathcal{F}(\boldsymbol{x})\mathcal{F}'(\boldsymbol{x})]\,.$$

For a subset $S \subseteq [R]$, define the *character* function $\chi_S : \{\pm 1\}^R \to \{\pm 1\}$ as

$$\chi_S(\boldsymbol{x}) = \prod_{i \in S} x^{(i)}\,.$$

The following fact is an immediate consequence of the fact that $(x^{(i)})^2 = 1$ for all $i \in [R]$.

Proposition 3.0.1. *For $S, T \subseteq [R]$, $\chi_S(\boldsymbol{x})\chi_T(\boldsymbol{x}) = \chi_{S \Delta T}(\boldsymbol{x})$.*

Proposition 3.0.2. *The characters $\{\chi_S | S \subseteq [R]\}$ form an orthonormal basis for $L^2(\{\pm 1\}^R)$.*

Proof. For any $S, T \subseteq [R]$,

$$\langle \chi_S, \chi_T \rangle = \mathop{\mathbb{E}}_{\boldsymbol{x}}[\chi_S(\boldsymbol{x})\chi_T(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\boldsymbol{x}}[\chi_{S\Delta T}(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\boldsymbol{x}}\Big[\prod_{i \in S\Delta T} x^{(i)}\Big]\,.$$

Using independence of the coordinates $x^{(i)}$, we can rewrite the above expression as,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}}\Big[\prod_{i\in S\Delta T} x^{(i)}\Big] = \prod_{i\in S\Delta T} \mathop{\mathbb{E}}_{\boldsymbol{x}}[x^{(i)}] = \begin{cases} 1 & \text{if } S\Delta T = \phi \\ 0 & \text{otherwise.} \end{cases}$$

■

By virtue of Proposition 3.0.2, any function $\mathcal{F} : \{\pm 1\}^R \to \mathbb{R}$ can be expressed as a linear combination of the character functions. In particular, one can write $\mathcal{F}$ as,

$$\mathcal{F}(\boldsymbol{x}) = \sum_{S\subseteq[R]} \hat{\mathcal{F}}_S \; \chi_S(\boldsymbol{x})\,,$$

where $\hat{\mathcal{F}}_S = \langle \mathcal{F}, \chi_S\rangle = \mathbb{E}_{\boldsymbol{x}}[\mathcal{F}(\boldsymbol{x})\chi_S(\boldsymbol{x})]$. The quantities $\{\hat{\mathcal{F}}_S\}_{S\subseteq[R]}$ are referred to as the Fourier coefficients or the Fourier spectrum of $\mathcal{F}$.

Proposition 3.0.3 (Plancherel's identity). *For any two functions $\mathcal{F}, \mathcal{F}' : \{\pm 1\}^R \to \mathbb{R}$,*

$$\langle \mathcal{F}, \mathcal{F}'\rangle = \sum_{S\subseteq[R]} \hat{\mathcal{F}}_S \hat{\mathcal{F}'}_S\,.$$

On setting $\mathcal{F} = \mathcal{F}'$ we get the Parseval's identity.

(Parseval's Identity) $$\mathop{\mathbb{E}}_{\boldsymbol{x}}[\mathcal{F}^2(\boldsymbol{x})] = \sum_{S\subseteq[R]} \hat{\mathcal{F}}_S^2$$

Proof.

$$\begin{aligned} \langle \mathcal{F}, \mathcal{F}'\rangle &= \langle (\sum_{S\subseteq[R]} \hat{\mathcal{F}}_S\chi_S), (\sum_{T\subseteq[R]} \hat{\mathcal{F}'}_T\chi_T)\rangle \\ &= \sum_{S,T\subseteq[R]} \hat{\mathcal{F}}_S\hat{\mathcal{F}'}_T\langle\chi_S,\chi_T\rangle \\ &= \sum_{S\subseteq[R]} \hat{\mathcal{F}}_S\hat{\mathcal{F}'}_S \qquad (\because \text{ orthonormality of characters.}) \end{aligned}$$

■

We will use $\mathbf{Var}[\mathcal{F}]$ to denote the variance of the random variable $\mathcal{F}(\boldsymbol{x})$ over a uniformly random input $\boldsymbol{x}$.

Proposition 3.0.4.

$$\mathbf{Var}[\mathcal{F}] = \sum_{\substack{S\subseteq[R],\\ S\neq\emptyset}} \hat{\mathcal{F}}_S^2\,.$$

Proof.

$$\begin{aligned}\mathbf{Var}[\mathcal{F}] &= \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\mathcal{F}^2(\boldsymbol{x})\right] - \left(\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\mathcal{F}(\boldsymbol{x})\right]\right)^2 \\ &= \left(\sum_{S\subseteq[R]} \hat{\mathcal{F}}_S^2\right) - \hat{\mathcal{F}}_\emptyset{}^2 \qquad \text{(using Parseval's identity and definition of } \hat{\mathcal{F}}_\emptyset) \\ &= \sum_{\substack{S\subseteq[R],\\ S\neq\emptyset}} \hat{\mathcal{F}}_S^2\end{aligned}$$

■

3.3.1 Influences

The influence of a coordinate on a boolean function is a measure of how often a change in that coordinate affects the value of the function. To define influences formally, let us denote by $\boldsymbol{x}^\ell \in \{\pm1\}^R$ the vector obtained by flipping the ℓ^{th} coordinate of $\boldsymbol{x} \in \{\pm1\}^R$. The influence of the ℓ^{th} coordinate on a boolean function $\mathcal{F} : \{\pm1\}^R \to \{\pm1\}$ is defined as,

$$\mathrm{Inf}_\ell(\mathcal{F}) = \mathop{\mathbb{P}\mathrm{r}}_{\boldsymbol{x}}[\mathcal{F}(\boldsymbol{x}) \neq \mathcal{F}(\boldsymbol{x}^\ell)]\,.$$

Proposition 3.0.5. *For a function* $\mathcal{F} : \{\pm1\}^R \to \{\pm1\}$, $\mathrm{Inf}_\ell(\mathcal{F}) = \sum_{S\ni\ell} \hat{\mathcal{F}}_S^2$.

Proof. Since the function $\mathcal{F}$ takes only $\{\pm1\}$ values,

$$\begin{aligned}\mathrm{Inf}_\ell(\mathcal{F}) &= \mathop{\mathbb{P}\mathrm{r}}_{\boldsymbol{x}}[\mathcal{F}(\boldsymbol{x}) \neq \mathcal{F}(\boldsymbol{x}^\ell)] \\ &= \frac{1}{4}\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[(\mathcal{F}(\boldsymbol{x}) - \mathcal{F}(\boldsymbol{x}^\ell))^2\right] \\ &= \frac{1}{4}\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\left(\sum_{S\subseteq[R]} \hat{\mathcal{F}}_S\chi_S(\boldsymbol{x}) - \sum_{S\subseteq[R]} \hat{\mathcal{F}}_S\chi_S(\boldsymbol{x}^\ell)\right)^2\right]\end{aligned}$$

Since $|\chi_S(\boldsymbol{x}) - \chi_S(\boldsymbol{x}^\ell)| = 2$ if $\ell \in S$ and is 0 otherwise.

$$\mathrm{Inf}_\ell(\mathcal{F}) = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\left(\sum_{S\ni\ell} \hat{\mathcal{F}}_S\ \chi_S(\boldsymbol{x})\right)^2\right] = \sum_{S\ni\ell} \hat{\mathcal{F}}_S^2 \qquad (\because \text{ Parseval's identity })$$

■

The notion of influences can be generalized to real valued functions $\mathcal{F} : \{\pm1\}^R \to \{\pm1\}$. To this end, denote let $\boldsymbol{x}^{(-\ell)}$ denote the vector consisting of all but the ℓ^{th} coordinate of $\boldsymbol{x}$.

Definition 3.3.1 (Influences)**.** The *influence* of the ℓ^{th} coordinate on a function $\mathcal{F} : \{\pm1\}^R \to \mathbb{R}$ is given by $\mathrm{Inf}_\ell(\mathcal{F}) = \mathbb{E}_{\boldsymbol{x}^{(-\ell)}}\left[\mathbf{Var}_{x^{(\ell)}}[\mathcal{F}(\boldsymbol{x})]\right] = \sum_{S\ni\ell} \hat{\mathcal{F}}_S^2$.

The following is an immediate consequence of the definition.

Proposition 3.0.6. *For a function $\mathcal{F} : \{\pm 1\}^R \to \mathbb{R}$ and $\ell \in [R]$, $\mathrm{Inf}_\ell(\mathcal{F}) \leqslant \mathbf{Var}[\mathcal{F}(\boldsymbol{x})]$.*

Proposition 3.0.7 (Convexity of Influences)**.** *Let $\mathcal{F}$ be a distribution over functions from $\{\pm 1\}^R$ to $\mathbb{R}$ and let $\mathcal{H} = \mathbb{E}[\mathcal{F}]$ denote the average function. Then,*

$$\mathbb{E}[\mathrm{Inf}_\ell(\mathcal{F})] \geqslant \mathrm{Inf}_\ell(\mathbb{E}[\mathcal{F}]) = \mathrm{Inf}_\ell(\mathcal{H})\,.$$

Proof. By definition and linearity of expectation, we have $\hat{\mathcal{H}}_S = \mathbb{E}[\hat{\mathcal{F}}_S]$ for all $S \in [R]$. Writing out the expression for $\mathrm{Inf}_\ell(\mathcal{H})$ we get,

$$\mathrm{Inf}_\ell(\mathcal{H}) = \sum_{S \ni \ell} \hat{\mathcal{H}}_S^2 = \sum_{S \ni \ell} \Big(\mathbb{E}[\hat{\mathcal{F}}_S]\Big)^2 \leqslant \sum_{S \ni \ell} \mathbb{E}[\hat{\mathcal{F}}_S^2] = \mathrm{Inf}_\ell(\mathcal{F})$$

by concavity of the function $h(x) = x^2$. ∎

3.3.2 Noise Stability

The notion of noise stability of boolean functions has numerous applications in computer science ranging from hardness amplification, to computational learning, property testing to hardness of approximation.

Fix a point $\boldsymbol{x} \in \{\pm 1\}^R$. The notation $\boldsymbol{y} \sim_\rho \boldsymbol{x}$ indicates that the random vector $\boldsymbol{y}$ is generated by flipping each bit independently with probability $(1-\rho)$. Formally $\boldsymbol{y} = (y^{(1)}, \ldots, y^{(R)})$ where,

$$y^{(\ell)} = \begin{cases} x^{(\ell)} & \text{with probability } \rho \\ \text{random bit} & \text{with probability } 1-\rho \end{cases}$$

Definition 3.3.2. For $\rho \in [-1, 1]$, the *noise stability* $\mathbb{S}_\rho(\mathcal{F})$ of a boolean function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$ is given by

$$\mathbb{S}_\rho(\mathcal{F}) = 2 \Pr_{\boldsymbol{x}, \boldsymbol{y} \sim_\rho \boldsymbol{x}} \big[\mathcal{F}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{y})\big] - 1\,.$$

Definition 3.3.3. For $\rho \in [-1, 1]$, define the operator T_ρ on the space of functions $L^2(\{\pm 1\}^R)$ as,

$$\mathrm{T}_\rho \mathcal{F}(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{y} \sim_\rho \boldsymbol{x}} [\mathcal{F}(\boldsymbol{y})]\,.$$

Proposition 3.0.8. *For a function $\mathcal{F} \in L^2(\{\pm 1\}^R)$,*

$$\mathrm{T}_\rho \mathcal{F}(\boldsymbol{x}) = \sum_{S \in [R]} \rho^{|S|} \chi_S(\boldsymbol{x})\,.$$

Proof. First, observe that $\mathbb{E}_{y^{(\ell)}\sim_\rho x^{(\ell)}}[y^{(\ell)}] = \rho \cdot x^{(\ell)} + (1-\rho)\cdot 0 = \rho x^{(\ell)}$. Therefore,

$$\begin{aligned}
\mathrm{T}_\rho \mathcal{F}(\boldsymbol{x}) &= \mathop{\mathbb{E}}_{\boldsymbol{y}\sim_\rho \boldsymbol{x}}[\mathcal{F}(\boldsymbol{y})] \\
&= \sum_{S\in[R]} \hat{\mathcal{F}}_S \mathop{\mathbb{E}}_{\boldsymbol{y}\sim_\rho \boldsymbol{x}}[\chi_S(\boldsymbol{y})] \\
&= \sum_{S\in[R]} \hat{\mathcal{F}}_S \prod_{\ell\in S} \mathop{\mathbb{E}}_{y^{(\ell)}\sim_\rho x^{(\ell)}}[y^{(\ell)}] = \sum_{S\in[R]} \prod_{\ell\in S} \rho x^{(\ell)} = \sum_{S\in[R]} \rho^{|S|}\chi_S(\boldsymbol{x})\,.
\end{aligned}$$

■

Fact 3.3.1. *For a boolean function $\mathcal{F}:\{\pm1\}^R \to \{\pm1\}$, $\mathbb{S}_\rho(\mathcal{F}) = \langle \mathcal{F}, \mathrm{T}_\rho \mathcal{F}\rangle = \sum_{S\in[R]} \rho^{|S|}\hat{\mathcal{F}}_S^2$.*

Proof.

$$\begin{aligned}
\mathop{\mathbb{P}\mathrm{r}}_{\boldsymbol{x},\boldsymbol{y}\sim_\rho \boldsymbol{x}}[\mathcal{F}(\boldsymbol{y}) = \mathcal{F}(\boldsymbol{x})] &= \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}\sim_\rho \boldsymbol{x}}\left[\frac{1+\mathcal{F}(\boldsymbol{x})\mathcal{F}(\boldsymbol{y})}{2}\right], \\
&= \frac{1}{2} + \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\mathcal{F}(\boldsymbol{x}) \mathop{\mathbb{E}}_{\boldsymbol{y}\sim_\rho \boldsymbol{x}}[\mathcal{F}(\boldsymbol{y})]\right], \\
&= \frac{1}{2} + \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{x}}[\mathcal{F}(\boldsymbol{x})\mathrm{T}_\rho\mathcal{F}(\boldsymbol{x})] = \frac{1}{2} + \frac{\langle \mathcal{F}, \mathrm{T}_\rho\mathcal{F}\rangle}{2}\,.
\end{aligned}$$

The conclusion follows by using Plancherel's identity (Fact 3.0.3). ■

3.3.3 Attenuated Influences

Definition 3.3.4 (Attenuated Influences)**.** For a function $\mathcal{F}:\{\pm1\}^R \to \mathbb{R}$, the ε-attenuated influence of a coordinate ℓ is given by $\mathrm{Inf}_\ell(\mathrm{T}_{1-\varepsilon}\mathcal{F})$.

We defer the proof of the following fact to Section 3.4.3 where we prove it in a more general setting.

Lemma 3.0.1 (Sum of Influences Lemma)**.** *For $\varepsilon > 0$ and a function $\mathcal{F}:\{\pm1\}^R \to \mathbb{R}$, $\sum_{\ell\in[R]} \mathrm{Inf}_\ell(\mathrm{T}_{1-\varepsilon}\mathcal{F}) \leqslant \mathbf{Var}[\mathcal{F}]/\varepsilon$.*

3.4 Functions on Product Spaces

Notions of influences and noise stability can be generalized to functions on arbitrary product spaces. In this section, we will see how a function over a product space can be written as a multilinear polynomial via the Fourier expansion, and how this leads to more general notions of influences and noise stability.

Fix a probability space $\Omega = (\mathcal{E}, \mu)$. The product probability space Ω^R is the probability space consisting of R-tuples of events from $\mathcal{E}$, with each coordinate chosen independently from Ω. The notions of influences and noise operators can be extended to the general setting of functions on Ω^R.

3.4.1 Multi-linear polynomials

First, we will see how a real valued function on an arbitrary product space Ω^R can be expressed as a multilinear polynomial.

Definition 3.4.1. A collection of finitely many real random variables $\mathcal{L} = \{\ell_0, \ldots, \ell_{q-1}\}$ will be referred to an *ensemble*. An ensemble $\mathcal{L} = \{\ell_0, \ldots, \ell_{q-1}\}$ under inner product $\langle f, g\rangle = \mathbb{E}[f \cdot g]$, is said to be an *Orthonormal* ensemble if the random variables in $\mathcal{L}$ are orthonormal, and ℓ_0 is the constant random variable 1.

For an ensemble $\mathcal{L} = \{\ell_0, \ldots, \ell_{q-1}\}$ of random variables, we shall use $\mathcal{L}^R$ to denote the ensemble obtained by taking R independent copies of $\mathcal{L}$. Further $\mathcal{L}^{(i)} = \{\ell_0^{(i)}, \ldots, \ell_{q-1}^{(i)}\}$ will denote the i^{th} independent copy in $\mathcal{L}^R$.

Consider an ensemble $\mathcal{L} = \{\ell_0, \ldots, \ell_{q-1}\}$ that forms a basis for the vector space $L^2(\Omega)$. Thus every function $\mathcal{F} : \Omega \to \mathbb{R}$ can be expressed as a linear combination of the ensemble $\mathcal{L}$. Given such a basis $\mathcal{L}$ for $L^2(\Omega)$, it induces a basis for the space $L^2(\Omega^R)$. Specifically, the basis for $L^2(\Omega^R)$ is given by the following set of random variables:

$$\left\{ \prod_{i=1}^{R} \ell_{\sigma_i}^{(i)} | \sigma \in [q]^R \right\}.$$

Hence every function $\mathcal{F} : \Omega^R \to \mathbb{R}$ can be expressed as a multilinear polynomial in $\mathcal{L}^R$ where $\mathcal{L}$ is an arbitrary basis for $L_2(\Omega)$. To illustrate this further, we will present a concrete example of a probability space below. However we first define a concise notation for multilinear products borrowed from [124]

Definition 3.4.2. A multi-index σ is a vector $(\sigma_1, \ldots, \sigma_R) \in \mathbb{N}^R$; the *degree* of σ denoted by $|\sigma|$, is $|\{i \in \{1, \ldots, R\} | \sigma_i \neq 0\}|$. Given a doubly-indexed set of indeterminates (formal variables) $\mathcal{X}^R = \{x_i^{(j)} | i \in \mathbb{N}, 1 \leqslant j \leqslant R\}$ and a multi-index σ, define the monomial x_σ as

$$x_\sigma = \prod_{i=1}^{R} x_{\sigma_i}^{(i)}.$$

The degree of the monomial x_σ is given by $|\sigma|$. A multilinear polynomial over such a set of indeterminates is given by

$$F(x) = \sum_{\sigma} \hat{F}_\sigma x_\sigma$$

where $\hat{F}_\sigma$ are real constants, all but finitely many of which are non-zero. The degree of $F(x)$ is $\max\{|\sigma| : \hat{F}_\sigma \neq 0\}$.

Example 3.4.1. Let Ω be a probability space with two atoms $\{0, 1\}$ occurring with probability $\frac{1}{2}$ each. An example of an orthonormal ensemble in $L_2(\Omega)$ is given by $\{\ell_0 = \mathbf{1}, \ell_1\}$, where

$$\begin{aligned} \ell_0(0) &= 1 & \ell_0(1) &= 1 \\ \ell_1(0) &= 1 & \ell_1(1) &= -1\,. \end{aligned}$$

Consider the product probability space Ω^R. An orthonormal basis for $L_2(\Omega^R)$ is given by the ensemble $\{\ell_\sigma : \Omega^R \to \mathbb{R} | \sigma \in \{0,1\}^R\}$ where

$$\ell_\sigma = \prod_{i=1}^{R} \ell_{\sigma_i}$$

Thus any random variable $F \in L^2(\Omega)^R$ can be written as a multilinear polynomial $\sum_\sigma \hat{F}_\sigma \ell_\sigma$ in the ensemble $\mathcal{L}^R$. In terms of functions, this means that any function $\mathcal{F} : \Omega^R \to \mathbb{R}$ has a multilinear expansion

$$\mathcal{F}(\boldsymbol{z}) = \sum_\sigma \hat{\mathcal{F}}_\sigma \ell_\sigma(\boldsymbol{z})$$

where $\ell_\sigma(\boldsymbol{z}) = \prod_{i=1}^{R} \ell_{\sigma_i}(z_i)$. We point out that the above expansion is exactly the Fourier expansion of $\mathcal{F}$ and the functions ℓ_σ are the character functions.

A different ensemble in $L_2(\Omega)$ is given by the indicator random variables $\mathcal{L} = \{\ell_0, \ell_1\}$ defined as follows:

$$\begin{aligned} \ell_0(0) &= 1 & \ell_0(1) &= 0 \\ \ell_1(0) &= 0 & \ell_1(1) &= 1\,. \end{aligned}$$

In this case, the multilinear expansion for a function $\mathcal{F} : \Omega^R \to \mathbb{R}$ is just

$$\mathcal{F}(\boldsymbol{z}) = \sum_{\sigma \in \{0,1\}^R} \mathcal{F}(\sigma) \ell_\sigma(\boldsymbol{z})\,.$$

We will always use the symbols $\mathcal{F}, \mathcal{H}$ to denote real-valued functions on a product probability space Ω^R. Thus $\mathcal{F}, \mathcal{H}$ take an atom in Ω^R as input and output a real number. $\mathcal{L} = \{\ell_0, \ell_1, \ldots, \ell_t\}$ will denote an ensemble forming a basis for $L^2(\Omega)$. Further $F(\boldsymbol{x}), H(\boldsymbol{x})$ will denote the formal multilinear polynomials corresponding to $\mathcal{F}, \mathcal{H}$. Hence $F(\mathcal{L}^R)$ is a random variable obtained by substituting the random variables $\mathcal{L}^R$ in place of $\boldsymbol{x}$. For instance, the following equation holds in this notation:

$$\mathop{\mathbb{E}}_{z \in \Omega^R}[\mathcal{F}(z)] = \mathop{\mathbb{E}}_{\boldsymbol{x} \in \mathcal{L}^R}[F(\boldsymbol{x})] = \mathbb{E}[F(\mathcal{L}^R)]\,.$$

For the most part, the probability spaces Ω will have $[q] = \{0, \ldots, q-1\}$ as the atoms. A natural basis for $L_2(\Omega)$ consists of the ensemble of indicator variables $\mathcal{L} = \{\ell_0, \ldots, \ell_{q-1}\}$. We shall refer to this basis as the *Standard Ensemble/Basis*. Specifically, the random variable ℓ_i is defined as $\ell_i(i) = 1, \ell_i(j) = 0 \forall j \neq i$. Any function $\mathcal{F} : \Omega^R \to \mathbb{R}$ can be expressed as a multilinear polynomial as

$$\mathcal{F}(\boldsymbol{z}) = \sum_{\sigma \in [q]^R} \mathcal{F}(\sigma) \ell_\sigma(\boldsymbol{z})\,.$$

Thus the corresponding formal multilinear polynomial is given by $F(\mathbf{x}) = \sum_{\sigma \in [q]^R} \hat{F}_\sigma x_\sigma$ where $\hat{F}_\sigma = \mathcal{F}(\sigma)$.

The following identity is an easy generalization of the Plancherel's identity in harmonic analysis.

Proposition 3.0.9. *For a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, if $\mathcal{F}(\boldsymbol{z}) = \sum_\sigma \hat{\mathcal{F}}_\sigma \ell_\sigma(\boldsymbol{z})$ with respect to an orthonormal ensemble $\mathcal{L}$ then $\mathbb{E}_{\boldsymbol{z}\in\Omega^R}[\mathcal{F}(\boldsymbol{z})] = \hat{\mathcal{F}}_{\mathbf{0}}$ and $\mathbf{Var}[\mathcal{F}] = \sum_{\sigma\neq\mathbf{0}} \hat{\mathcal{F}}^2_\sigma$.*

Proposition 3.0.10. *Let $\mathcal{F}, \mathcal{F}' : \Omega^R \to \mathbb{R}$ be functions on a product space Ω^R. Let $\mathcal{L}$ be an orthonormal basis for $L_2(\Omega)$. If $\mathcal{F}(\boldsymbol{z}) = \sum_\sigma \hat{\mathcal{F}}_\sigma \ell_\sigma(\boldsymbol{z})$, $\mathcal{F}'(\boldsymbol{z}) = \sum_\sigma \hat{\mathcal{F}'}_\sigma \ell_\sigma(\boldsymbol{z})$ are multilinear expansions obtained using the basis $\mathcal{L}$, then*

$$\langle \mathcal{F}, \mathcal{F}' \rangle = \sum_\sigma \hat{\mathcal{F}}_\sigma \hat{\mathcal{F}'}_\sigma \,.$$

On setting $\mathcal{F} = \mathcal{F}'$ we get the analogue of Parseval's identity.

$$\textit{(Analogue of Parseval's Identity)} \qquad \mathop{\mathbb{E}}_{\boldsymbol{z}\in\Omega^R}[\mathcal{F}^2(\boldsymbol{z})] = \sum_\sigma \hat{\mathcal{F}}^2_\sigma \,.$$

3.4.2 Noise Operator

To begin with, we shall define the noise operator $\mathrm{T}_{1-\varepsilon}$ on set of functions over a product probability space Ω^R.

Definition 3.4.3. For $0 \leqslant \varepsilon \leqslant 1$, define the operator $\mathrm{T}_{1-\varepsilon}$ on $L_2(\Omega^R)$ as,

$$\mathrm{T}_{1-\varepsilon}\mathcal{F}(\boldsymbol{z}) = \mathbb{E}[\mathcal{F}(\tilde{\boldsymbol{z}}) \mid \boldsymbol{z}]$$

where each coordinate $\tilde{z}^{(i)}$ of $\tilde{\boldsymbol{z}}$ is equal to $z^{(i)}$ with probability $1-\varepsilon$ and a random element from Ω with probability ε.

Let us suppose that $F(\boldsymbol{x}) = \sum_\sigma \hat{F}_\sigma x_\sigma$ is the multilinear expansion of the function $\mathcal{F}$ with respect to an orthonormal basis $\mathcal{L}$. The multilinear polynomial corresponding to $\mathrm{T}_{1-\varepsilon}\mathcal{F}$ is given by

$$\mathrm{T}_{1-\varepsilon}F(\boldsymbol{x}) = \sum_{\sigma\in[k]^R} (1-\varepsilon)^{|\sigma|} \hat{F}_\sigma x_\sigma \,.$$

We stress here that the above expression holds only for an orthonormal basis $\mathcal{L}$. For a basis that is not necessarily orthonormal, the following fact yields the required expansion.

Proposition 3.0.11. *For a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, the multilinear expansion corresponding to $\mathrm{T}_{1-\varepsilon}\mathcal{F}$ is given by,*

$$\mathrm{T}_{1-\varepsilon}F(\boldsymbol{x}) = F\Big((1-\varepsilon)\cdot\boldsymbol{x} + \varepsilon\cdot\mathbb{E}[\mathcal{L}^R]\Big)\,,$$

where $F(\boldsymbol{x})$ is the multilinear expansion of $\mathcal{F}$ with respect to $\mathcal{L}$. In other words, the formal multilinear polynomial $\mathrm{T}_{1-\varepsilon}F(\boldsymbol{x})$ is just obtained by replacing each variable $x_j^{(i)}$ in $F(\boldsymbol{x})$ with $(1-\varepsilon)\cdot x_j^{(i)} + \mathbb{E}[\ell_j]$.

Proof. Recall that by the multilinear expansion of $\mathcal{F}$, we have

$$\mathcal{F}(\boldsymbol{z}) = \sum_{\sigma} \hat{F}_\sigma \prod_{i=1}^{R} \ell_{\sigma_i}(z^{(i)})\,.$$

By definition of $\mathrm{T}_{1-\varepsilon}\mathcal{F}$,

$$\mathrm{T}_{1-\varepsilon}\mathcal{F}(\boldsymbol{z}) = \mathbb{E}[\mathcal{F}(\tilde{\boldsymbol{z}})|\boldsymbol{z}] = \sum_{\sigma} \hat{F}_\sigma\, \mathbb{E}\left[\prod_{i} \ell_{\sigma_i}(\tilde{z}^{(i)})|\boldsymbol{z}\right].$$

Conditioned on the value of $\boldsymbol{z}$, the coordinates $\boldsymbol{z}^{(i)}$ are independent of each other. Thus,

$$\mathrm{T}_{1-\varepsilon}\mathcal{F}(\boldsymbol{z}) = \sum_{\sigma} \hat{F}_\sigma \prod_{i=1}^{R} \mathbb{E}\left[\ell_{\sigma_i}(\tilde{z}^{(i)})|\boldsymbol{z}\right].$$

By definition of $T_{1-\varepsilon}$ we have, $\mathbb{E}\left[\ell_{\sigma_i}(\tilde{z}^{(i)})|\boldsymbol{z}\right] = (1-\varepsilon)\cdot\ell_{\sigma_i}(z^{(i)}) + \varepsilon\,\mathbb{E}[\ell_{\sigma_i}]$. Substituting the value we get

$$\mathrm{T}_{1-\varepsilon}\mathcal{F}(\boldsymbol{z}) = \sum_{\sigma} \hat{F}_\sigma \prod_{i=1}^{R} \Big((1-\varepsilon)\cdot\ell_{\sigma_i}(z^{(i)}) + \varepsilon\,\mathbb{E}[\ell_{\sigma_i}]\Big)\,.$$

By inspecting the above expression, the result follows. ∎

3.4.3 Influences

Definition 3.4.4. For a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, the *influence* of the ℓ^{th} coordinate is given by

$$\mathrm{Inf}_\ell(\mathcal{F}) = \mathop{\mathbb{E}}_{\boldsymbol{z}^{(-\ell)}\in\Omega^{R-1}}\left[\mathop{\mathbf{Var}}_{z^{(\ell)}\in\Omega}[\mathcal{F}(\boldsymbol{z})]\right].$$

Here $\boldsymbol{z}^{(-\ell)}$ consists of all but the ℓ^{th} coordinate of $\boldsymbol{z}$.

As in the case of boolean functions, influences of coordinates on a function $\mathcal{F} \in L^2(\Omega^R)$ can be expressed in terms of its multilinear expansion. We omit the fairly straightforward proof of the following proposition, which is along the lines of that of Proposition 3.0.5.

Proposition 3.0.12. *For a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, if $\mathcal{F}(\boldsymbol{z}) = \sum_\sigma \hat{\mathcal{F}}_\sigma \ell_\sigma(\boldsymbol{z})$ is its multilinear expansion with respect to an orthonormal ensemble $\mathcal{L}$ then,*

$$\mathrm{Inf}_\ell(\mathcal{F}) = \sum_{\sigma_\ell \neq 0} \hat{\mathcal{F}}_\sigma^2\,.$$

Lemma 3.0.2 (Sum of Influences Lemma)**.** *Given a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, if $\mathcal{H} = \mathrm{T}_{1-\varepsilon}\mathcal{F}$ then $\sum_{\ell\in[R]} \mathrm{Inf}_\ell(\mathcal{H}) \leqslant \frac{\mathbf{Var}[\mathcal{F}]}{2e\ln 1/(1-\varepsilon)} \leqslant \frac{\mathbf{Var}[\mathcal{F}]}{\varepsilon}$*

Proof. Let $\mathcal{F}(\boldsymbol{x}) = \sum_\sigma \hat{\mathcal{F}}(\sigma)\ell_\sigma(\boldsymbol{x})$ denote the multilinear expansion of $\mathcal{F}$ using an orthonormal basis $\mathcal{L}$. The function $\mathcal{H}$ is given by $\mathcal{H}(\boldsymbol{x}) = \sum_\sigma (1-\varepsilon)^{|\sigma|}\hat{\mathcal{F}}(\sigma)\ell_\sigma(x)$. Hence we get,

$$\begin{aligned}\sum_{\ell\in[R]} \mathrm{Inf}_\ell(\mathcal{H}) &= \sum_{i=1}^{R}\sum_{\sigma,\sigma_i\neq 0}(1-\varepsilon)^{2|\sigma|}\hat{\mathcal{F}}^2(\sigma) = \sum_\sigma (1-\varepsilon)^{2|\sigma|}|\sigma|\hat{\mathcal{F}}^2(\sigma)\\ &\leqslant \max_{\sigma\in\mathbb{N}^R}\left((1-\varepsilon)^{2|\sigma|}|\sigma|\right)\cdot\sum_{\sigma\neq\mathbf{0}}\hat{\mathcal{F}}(\sigma)^2 \leqslant \max_\sigma (1-\varepsilon)^{2|\sigma|}|\sigma|\cdot\mathbf{Var}[\mathcal{F}]\,.\end{aligned}$$

The function $h(x) = x(1-\varepsilon)^{2x}$ achieves a maximum at $x = -\frac{1}{2}\ln(1-\varepsilon)$. Substituting in the above equation, we get the desired conclusion. ■

It is easy to see that the convexity of influences holds in the general setting of functions over arbitrary product spaces.

Proposition 3.0.13 (Convexity of Influences)**.** *Let $\mathcal{F}$ be a random function from Ω^R to $\mathbb{R}$ and let $\mathcal{H} = \mathbb{E}[\mathcal{F}]$ denote the average function. Then,*

$$\mathbb{E}[\mathrm{Inf}_\ell(\mathcal{F})] \geqslant \mathrm{Inf}_\ell(\mathbb{E}[\mathcal{F}]) = \mathrm{Inf}_\ell(\mathcal{H})\,.$$

3.4.4 Formal Polynomials

Multilinear polynomials arise out of expressing a real-valued function $\mathcal{F}$ over a probability space Ω^R. Properties such as influence and noise stability of $\mathcal{F}$ are expressible in terms of the coefficients of the corresponding multilinear polynomial.

Now, we shall extend the notion of influences and noise stability to formal multilinear polynomials themselves. These notions are necessary in order to state the invariance principle.

Definition 3.4.5 (Influences for Polynomials)**.** For a multilinear polynomial $F(\boldsymbol{x}) = \sum_\sigma \hat{F}_\sigma x_\sigma$, define the variance of F and the influence of the ℓ^{th} coordinate as, i^{th} coordinate as follows:

$$\mathbf{Var}[F] = \sum_{\sigma\neq\mathbf{0}}\hat{F}_\sigma^2 \qquad\qquad \mathrm{Inf}_\ell(F) = \sum_{\sigma_\ell\neq\mathbf{0}}\hat{F}_\sigma^2\,.$$

It is easy to see that if $\mathcal{F}\in L^2(\Omega^R)$ is expressed as a multilinear polynomial F in an orthonormal ensemble $\mathcal{L}$ then

$$\mathbf{Var}[\mathcal{F}] = \mathbf{Var}[F] \qquad\qquad \mathrm{Inf}_\ell(\mathcal{F}) = \mathrm{Inf}_\ell(F)\,.$$

Thus, the above definition of influences and variance are the natural definitions to consider.

3.4.5 Vector Valued Functions and Polynomials

In many applications, it is natural to work with notions of influences, and noise operators for vector-valued functions, or tuples of multilinear polynomials. The notion of variance, influences and noise operators are generalized in the most natural way.

For $\boldsymbol{\mathcal{F}} = (\mathcal{F}_1, \ldots, \mathcal{F}_d)$ define

$$\begin{aligned} \mathrm{T}_{1-\varepsilon}\boldsymbol{\mathcal{F}} &= (\mathrm{T}_{1-\varepsilon}\mathcal{F}_1, \ldots, \mathrm{T}_{1-\varepsilon}\mathcal{F}_d)\,, \\ \mathbf{Var}[\boldsymbol{\mathcal{F}}] &= \sum_{i\in[d]} \mathbf{Var}[\mathcal{F}_i]\,, \\ \mathrm{Inf}_\ell(\boldsymbol{\mathcal{F}}) &= \sum_{i\in[d]} \mathrm{Inf}_\ell(\mathcal{F}_i)\,. \end{aligned}$$

Along similar lines, one defines these notions for vector-valued polynomials $\boldsymbol{F} = (F_1, \ldots, F_d)$.

More often than not, vector valued functions $\boldsymbol{\mathcal{F}}$ considered in this dissertation have a range equal to the q-dimensional simplex for a positive integer q (denoted by $\blacktriangle_q$).

Proposition 3.0.14. *Let $\boldsymbol{\mathcal{F}} : \Omega^R \to \blacktriangle_q$ be a vector-valued function over a probability space Ω and let $\boldsymbol{\mathcal{H}} = \mathrm{T}_{1-\varepsilon}\boldsymbol{\mathcal{F}}$. Then, $\mathbf{Var}[\boldsymbol{\mathcal{F}}] \leqslant 2$ and $\sum_{\ell\in R} \mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}) \leqslant \frac{2}{\varepsilon}$.*

Proof. Observe that for any two points $\boldsymbol{x}, \boldsymbol{y} \in \blacktriangle_q$, $\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 \leqslant 2$. This implies that $\mathbf{Var}[\boldsymbol{\mathcal{F}}] \leqslant 2$ and along with Lemma 3.0.2 implies that $\sum_{\ell\in R} \mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}) \leqslant \frac{2}{\varepsilon}$. ■

3.5 Gaussian Random Variables

A one-dimensional Gaussian random variable g has a distribution given by

$$\mu(g) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(g-\nu)^2}{2\sigma^2}}$$

where $\nu = \mathbb{E}[g]$ is the mean and $\sigma^2 = \mathbb{E}[(g-\mu)^2]$ is the variance of g.

A normal random variable is a Gaussian random variable with mean 0 and variance 1. Unless otherwise specified, a *Gaussian random variable* refers to a normal random variable (mean 0 and variance 1). Further, by *Gaussian space* we will refer to the probability space $\mathcal{G} = (\mathbb{R}, \mu)$ consisting of the following probability distribution over the set of real numbers $\mathbb{R}$:

$$\mu(x) = \frac{1}{\sqrt{2}} e^{-\frac{x^2}{2}}$$

Definition 3.5.1. For each integer $d \geqslant 0$, the d^{th} *Hermite polynomial* $H_d(x)$ is a uni-variate degree d polynomial such that

$$\mathop{\mathbb{E}}_{x\in\mathcal{G}}[H_d(x)H_{d'}(x)] = \begin{cases} 1 & \text{if } d = d' \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{G}$ is the Gaussian space. In other words, the set of Hermite polynomials form an orthonormal basis for the space of functions $L^2(\mathcal{G})$.

The distribution of an n-dimensional Gaussian random variable $\boldsymbol{g} = (g^{(1)}, \ldots, g^{(n)}) \in \mathbb{R}^n$ is specified by its means $\boldsymbol{\nu} = (\nu^{(1)}, \ldots, \nu^{(n)})$ and the covariance matrix $\Sigma = (\sigma_{ij})_{i,j\in[n]}$, i.e.,

$$\nu^{(i)} = \mathbb{E}[g^{(i)}] \qquad \sigma_{ij} = \mathbb{E}[g^{(i)}g^{(j)}] \qquad \forall i, j \in [n]\,.$$

The probability distribution of $\boldsymbol{g}$ is given by

$$\mu(\boldsymbol{g}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{(\boldsymbol{g}-\boldsymbol{\nu})^T\Sigma^{-1}(\boldsymbol{g}-\boldsymbol{\nu})}$$

where $|\Sigma|$ denotes the determinant of the matrix $|\Sigma|$.

Unless otherwise specified, a Gaussian vector refers to a vector all of whose coordinates are i.i.d normal random variables. The probability space $\mathcal{G}^n$ refers to the n-ary product of the Gaussian space $\mathcal{G}$. We have the following standard property of Gaussian random variables.

Property 3.1. For two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, and a n-dimensional Gaussian random vector $\boldsymbol{\zeta}$ whose coordinates are i.i.d normal random variables,

$$\mathop{\mathbb{E}}_{\boldsymbol{\zeta}}[\langle \boldsymbol{u}, \boldsymbol{\zeta}\rangle\langle \boldsymbol{v}, \boldsymbol{\zeta}\rangle] = \langle \boldsymbol{u}, \boldsymbol{v}\rangle$$

3.6 Invariance Principle

In its simplest form, the central limit theorem asserts the following:

> *"As n increases, the sum of n independent Bernoulli random variables (± 1 random variables) has approximately the same distribution as the sum of n independent normal (Gaussian with mean 0 and variance 1) random variables"*

Alternatively, as n increases, the value of the polynomial $F(\boldsymbol{x}) = \frac{1}{\sqrt{n}}(x^{(1)} + x^{(2)} + \ldots x^{(n)})$ has approximately the same distribution whether the random variables $x^{(i)}$ are i.i.d Bernoulli random variables or i.i.d normal random variables. More generally, the distribution of $F(\boldsymbol{x})$ is approximately the same as long as the random variables $x^{(i)}$ are independent with mean 0, variance 1 and satisfy certain mild regularity assumptions.

A phenomenon of this nature where the distribution of a function of random variables, depends solely on a small number of their moments is referred to an invariance. A natural approach to generalize of the above stated central limit theorem, is to replace the "sum" ($\frac{1}{\sqrt{n}}(x^{(1)} + x^{(2)} + \ldots x^{(n)})$) by other multivariate polynomials.

It is easy to see that invariance does not hold for arbitrary multivariate polynomials. For instance, consider the dictator function $F(\boldsymbol{x}) = x^{(1)}$ that outputs the first coordinate. It is clear that the polynomial F does not exhibit invariance. For instance, the distribution of F is different when substituting Bernoulli and Gaussian random variables for the coordinates of $\boldsymbol{x}$. It is easy to see that all polynomials that depend on a small number of coordinates (a *junta*) do not exhibit invariance.

The invariance principle for low degree polynomials was first shown by Rotar in 1979 [159]. More recently, invariance principles for low degree polynomials were shown in different settings in the work of Mossel, O'Donnell, and Olekschewicz [125] and Chatterjee [37]. The former of the two works also showed the Majority is Stablest conjecture, and has been influential in introducing the powerful tool of invariance to hardness of approximation.

Define functions $f_{[0,1]} : \mathbb{R} \to \mathbb{R}$ and $\xi : \mathbb{R}^q \to \mathbb{R}$ as follows:

$$f_{[0,1]}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leqslant x \leqslant 1 \\ 1 & \text{if } x > 1 \end{cases} \qquad \xi(\boldsymbol{x}) = \sum_{i \in [q]} (x_i - f_{[0,1]}(x_i))^2$$

The following invariance principle is an immediate consequence of the work of Mossel [124].

Theorem 3.2. *(Invariance Principle [124]) Fix $0 < \alpha, \varepsilon \leqslant 1/2$. Let Ω be a finite probability space such that every atom with non-zero probability has probability at least $\alpha \leqslant 1/2$. Let $\mathcal{L} = \{\ell_1, \ell_1, \ldots, \ell_m\}$ be an ensemble of random variables over Ω. Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be an ensemble of Gaussian random variables satisfying the following conditions:*

$$\mathbb{E}[\ell_i] = \mathbb{E}[g_i] \qquad \mathbb{E}[\ell_i^2] = \mathbb{E}[g_i^2] \qquad \mathbb{E}[\ell_i \ell_j] = \mathbb{E}[g_i g_j] \qquad \forall i, j \in [m]\,.$$

Let $\boldsymbol{F} = (F_1, \ldots, F_d)$ denote a vector valued multilinear polynomial. Let $H_i = T_{1-\varepsilon} F_i$, and $\boldsymbol{H} = (H_1, \ldots, H_d)$. If $\mathrm{Inf}_\ell(\boldsymbol{H}) \leqslant \tau$ and $\mathbf{Var}[H_\ell] \leqslant 1$ for all ℓ, then the following holds

1. *For every function $\Psi : \mathbb{R}^d \to \mathbb{R}$ that is thrice differentiable with all its partial derivatives up to order 3 bounded uniformly by C_0,*

$$\left| \mathbb{E}\left[\Psi(\boldsymbol{H}(\mathcal{L}^R))\right] - \mathbb{E}\left[\Psi(\boldsymbol{H}(\mathcal{G}^R))\right] \right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)}$$

   where $K = K(d, C_0) > 0$ is a constant depending on C_0, d.

2.
$$\left| \mathbb{E}[\xi(\boldsymbol{H}(\mathcal{L}^R))] - \mathbb{E}[\xi(\boldsymbol{H}(\mathcal{G}^R))] \right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)}$$

   where $K = K(d, C_0) > 0$ is a constant depending on C_0, d.

Proof. The theorem is implicit in the work of Mossel[124]. Here we describe some minor changes to deduce the result from Theorems 4.1 and 4.2 in [124].

Without loss of generality, the ensembles $\mathcal{L}$ and $\mathcal{G}$ can be assumed to be orthonormal. Otherwise, since the ensembles $\mathcal{L}$, $\mathcal{G}$ have matching inner products, there exists a linear transformation T that transforms both $\mathcal{L}$ and $\mathcal{G}$ into an orthonormal basis. Replace the polynomial $\boldsymbol{F}(\boldsymbol{x})$ by $\boldsymbol{F}(T^{-1}(\boldsymbol{x}))$ and the ensembles $\mathcal{L}, \mathcal{G}$ by orthonormal ensembles $T(\mathcal{L})$ and $T(\mathcal{G})$. Here we use the fact that the notions Inf and **Var** are independent of the basis, and would remain the same after the transformation T.

Truncate the polynomial $\mathbf{H}$ at degree $D = \log_{1-\varepsilon} \tau$ to obtain a vector valued polynomial $\mathbf{Q}$. Note that for each i, $\mathrm{Inf}_i(\mathbf{Q}) \leqslant \mathrm{Inf}_i(\mathbf{H}) \leqslant \tau$. Apply Theorem 4.1 in [124] for Q and the smooth functional Ψ,

$$\left| \mathbb{E}\left[\Psi(\mathbf{Q}(\mathcal{L}^R))\right] - \mathbb{E}\left[\Psi(\mathbf{Q}(\mathcal{G}^R))\right] \right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)}\,.$$

Since Ψ is a smooth functional we have,

$$\left| \mathbb{E}\left[\Psi(\mathbf{Q}(\mathcal{L}^R)) - \Psi(\mathbf{H}(\mathcal{L}^R))\right]\right| \leqslant C_0 \|\mathbf{Q}(\mathcal{L}^R) - \mathbf{H}(\mathcal{L}^R))\| \,.$$

However since $\mathbf{H} = \mathrm{T}_{1-\varepsilon}\boldsymbol{F}$,

$$\|\mathbf{Q}(\mathcal{L}^R) - \mathbf{H}(\mathcal{L}^R))\|^2 = \sum_{|\sigma|>\log_{1-\varepsilon}\tau} (1-\varepsilon)^{2\sigma}\hat{F}_\sigma^2 \leqslant \tau^2 \,\mathbf{Var}[F] \,.$$

Thus we get,

$$\left| \mathbb{E}\left[\Psi(\mathbf{Q}(\mathcal{L}^R)) - \Psi(\mathbf{H}(\mathcal{L}^R))\right]\right| \leqslant C_0\tau(\mathbf{Var}[F])^{\frac{1}{2}} \,,$$
$$\left| \mathbb{E}\left[\Psi(\mathbf{Q}(\mathcal{G}^R)) - \Psi(\mathbf{H}(\mathcal{G}^R))\right]\right| \leqslant C_0\tau(\mathbf{Var}[F])^{\frac{1}{2}} \,.$$

Along with Equation 3.6, the above two inequalities imply the first part of the result. To prove the second inequality, observe that if $\mathrm{Inf}_i^{\leqslant D}(\mathbf{H}) \leqslant \mathrm{Inf}_i(\mathbf{H}) \leqslant \tau$ for all i, D. Apply Theorem 4.2 in [124], on the one dimensional polynomial H_i to show that

$$\left| \mathbb{E}[\xi(H_i(\mathcal{L}^R))] - \mathbb{E}[\xi(H_i(\mathcal{G}^R))]\right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)} \,.$$

Summing up the expression over all i we get

$$\left| \mathbb{E}[\xi(\mathbf{H}(\mathcal{L}^R))] - \mathbb{E}[\xi(\mathbf{H}(\mathcal{G}^R))]\right| \leqslant d\tau^{K\varepsilon/\log(1/\alpha)} \,.$$

To finish the proof, observe that for a fixed d, we have $d\tau^{K\varepsilon/\log(1/\alpha)} = \tau^{K'\varepsilon/\log(1/\alpha)}$ ■

3.7 Noise Stability Bounds

An important application of invariance principle has been in showing upper bounds on the noise stability of functions over product spaces. In fact, all applications of invariance principle in hardness of approximation apart from those in this dissertation, rely only on the noise stability bounds, and do not directly need the invariance principle.

The work of Mossel et al. [125] utilized the invariance principle to show the Majority is Stablest theorem, which is essentially an upper bound on the noise stability of boolean functions, with no influential coordinates. We include the formal statement of the result below.

Theorem 3.3 (Majority is Stablest [125])**.** *Let $0 \leqslant \rho \leqslant 1$ and ε be given. Then there exists $\tau > 0$ such that if $\mathcal{F} : \{\pm 1\}^R \to [-1, 1]$ satisfies $\mathbb{E}[\mathcal{F}] = 0$ and $\mathrm{Inf}_\ell(\mathcal{F}) \leqslant \tau$ for all $\ell \in [R]$, then*

$$\mathbb{S}_\rho(\mathcal{F}) \leqslant \frac{2}{\pi}\arcsin\rho + \varepsilon$$

Here $\mathbb{S}_\rho(\mathcal{F}) = \sum_{S\in[R]} \rho^{|S|}\hat{\mathcal{F}}_S^2$ is the noise stability of $\mathcal{F}$.

To obtain noise stability bounds for functions over product spaces, the idea is to re-

late it to a certain Gaussian noise stability via the invariance principle. For the sake of completeness, we define the Gaussian noise stability below.

Definition 3.7.1 (Ornstein-Uhlenbeck Operator)**.** For $\rho \in [-1,1]$, the *Ornstein-Uhlenbeck* operator U_ρ operates on functions in $L^2(\mathcal{G}^n)$ as follows,

$$\mathrm{U}_\rho \mathcal{F}(\boldsymbol{z}) = \mathop{\mathbb{E}}_{\boldsymbol{\eta}\in\mathcal{G}^n}\left[\mathcal{F}(\rho\cdot\boldsymbol{z} + \sqrt{1-\rho^2}\cdot\boldsymbol{\eta})\right].$$

Definition 3.7.2 (Gaussian Noise Stability)**.** For $\rho \in [-1,1]$, the Gaussian noise stability at correlation ρ (noise $1-\rho$) of a function $\mathcal{F} \in L^2(\mathcal{G}^n)$ is $\langle \mathcal{F}, \mathrm{U}_\rho \mathcal{F}\rangle$.

For a multilinear polynomial $\mathcal{F}(\boldsymbol{z}) = \sum_{S\in[R]} \hat{\mathcal{F}}_S \prod_{\ell\in S} z^{(\ell)}$, the noise stability is given by $\langle \mathcal{F}, \mathrm{U}_\rho \mathcal{F}\rangle = \sum_{S\in[R]} \rho^{|S|}\hat{\mathcal{F}}_S^2$. Let $\mathcal{F}: \{\pm 1\}^R \to [-1,1]$ be a boolean function all of whose influences are low. Let F be the multilinear polynomial associated with $\mathcal{F}$. The polynomial F can be thought of as a function over the Gaussian space $\mathcal{G}^R$. Notice that the noise stability of $\mathcal{F}$ is identically equal to the Gaussian noise stability of F, i.e.,

$$\mathbb{S}_\rho(\mathcal{F}) = \sum_{S\subseteq[R]} \rho^{|S|}\hat{\mathcal{F}}_S^2 = \langle F, \mathrm{U}_\rho F\rangle.$$

Intuitively, the idea is to use the invariance principle to argue that like $\mathcal{F}$, the function F on $\mathcal{G}^R$ takes values in $[0,1]$. Then, one appeals to known bounds on Gaussian noise stability to finish the argument.

By Borell's isoperimetric result on the Gaussian space, among functions $\mathcal{F}: \mathcal{G}^R \to [0,1]$, with a given value of $\mathbb{E}[\mathcal{F}]$ the one that maximizes $\langle \mathcal{F}, \mathrm{U}_\rho \mathcal{F}\rangle$ are linear threshold functions of the form:

$$\mathcal{F}(\boldsymbol{z}) = \begin{cases} 1 & \text{if } z^{(1)} \geqslant t \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the noise stability bounds via the invariance principle are directly related to the noise stability of linear threshold functions on the Gaussian space. In this light, let us denote by Γ_ρ the Gaussian noise stability of linear thresholds as a function of their mean/expectation (μ). Formally, define Γ_ρ as follows.

Definition 3.7.3 (Γ_ρ)**.** Given $\mu \in [0,1]$, let $t = \Phi^{-1}(\mu)$ where Φ denotes the distribution function of the standard Gaussian. Then,

$$\Gamma_\rho(\mu) = \mathbb{P}\mathrm{r}[X \leqslant t, Y \leqslant t],$$

where (X,Y) is a two-dimensional Gaussian vector with covariance matrix $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

In this notation, the formal statement of Borell's isoperimetric theorem is as follows:

Theorem 3.4. *([27]) For $\mathcal{F}, \mathcal{F}' \in L_2(\mathbb{R}^R)$ with $\mathbb{E}[\mathcal{F}] = \mathbb{E}[\mathcal{F}'] = \mu$ and $0 \leqslant \varepsilon \leqslant 1$, $\langle \mathcal{F}, U_{1-\varepsilon}\mathcal{F}'\rangle \leqslant \Gamma_\rho(\mu)$*

The quantity Γ_ρ being a double integral can be computed easily to arbitrary accuracy. In many applications, the following asymptotic bound on the function Γ_ρ is sufficient.

Theorem 3.5 (Theorem B.2 [125]). *As $\mu \to 0$,*

$$\Gamma_\rho(\mu) \sim \mu^{2/(1+\rho)}(4\pi \ln(1/\mu))^{-\rho/(1+\rho)} \frac{(1+\rho)^{3/2}}{(1-\rho)^{1/2}} \leqslant \mu^{\frac{3-\rho}{2}} .$$

Finally, through the invariance principle and the Borell's isoperimetric theorem, Mossel et al. [125] show the following bound on the noise stability of functions over a product probability space.

Theorem 3.6 (Theorem 4.4, [125]). *Let Ω be a finite probability space with the least non-zero probability of an atom at least α. For every $\mu, \varepsilon, \gamma, \delta > 0$ there exists τ such that the following holds: For every function $\mathcal{F} : \Omega^R \to [0,1]$ with $\mu = \mathbb{E}[\mathcal{F}]$ and $\mathrm{Inf}_\ell(\mathrm{T}_{1-\gamma}\mathcal{F}) < \tau$ for all $\ell \in [R]$,*

$$\langle \mathcal{F}, \mathrm{T}_{1-\varepsilon}\mathcal{F} \rangle = \mathop{\mathbb{E}}_{\boldsymbol{z} \in \Omega^R} [\mathcal{F}(\boldsymbol{z}) \mathrm{T}_{1-\varepsilon}\mathcal{F}(\boldsymbol{z})] \leqslant \Gamma_{1-\varepsilon}(\mu) + \delta .$$

Part I

ALGORITHMIC TECHNIQUES

Chapter 4

LINEAR AND SEMIDEFINITE PROGRAMMING RELAXATIONS

The focus of this chapter is linear and semidefinite programming relaxations for generalized constraint satisfaction problems, while the next chapter addresses the question of rounding. Together, the two chapters yield approximation algorithms for the class of GCSPs.

4.1 Introduction

Given a combinatorial optimization problem Λ, there are numerous relaxations that could be considered to design approximation algorithms for Λ. In particular, there is no single canonical way to write a relaxation given the combinatorial optimization problem Λ. Different problems warrant different kinds of constraints in the relaxation. In some cases, multiple relaxations have been used to obtain approximation algorithms for the same combinatorial optimization problems.

Clearly, it would be extremely desirable to have a canonical relaxation for every combinatorial optimization problem. However, this is unrealistic to expect due to the sheer diversity of combinatorial optimization problems. This thesis demonstrates that for the class of Generalized Constraint Satisfaction Problems (GCSP), there is an SDP relaxation referred to as the LC relaxation, that is canonical in numerous ways. This is surprising since the class of GCSPs is a very large family of problems that includes maximization problems such as Max Cut and Max 3-Sat on one hand, and minimization problems such as Multiway Cut on the other.

In this chapter, we will motivate the LC relaxation and connect it to other linear and semidefinite programs that have appeared in literature for specific GCSPs. In the upcoming chapters, we will show that the LC-relaxation has the following properties that make it *canonical* for the class of GCSPs.

- The LC relaxation can be shown to be equivalent to several well known SDPs on restricting to specific GCSPs. For example, it is equivalent to the Goemans-Williamson relaxation on restricting to the case of Max Cut.

- Among GCSPs that admit a constant factor approximation, no stronger relaxation than LC has been utilized in an approximation algorithm. Thus, the LC relaxation yields the best known approximations for GCSPs in almost all regimes.

- Under the Unique Games Conjecture, the LC relaxation yields the optimal approximation for every GCSP, when the approximation factor is an absolute constant (See Chapter 6 and Chapter 7).

- Under the Unique Games Conjecture, for every metric labelling problem over a finite metric, the optimal approximation ratio is obtained by the earthmover linear program. The LC is a simple generalization of the earthmover linear program (EM-LP) (See Chapter 8).

- Under the Unique Games Conjecture, for every Ordering CSP, the optimal approximation is obtained by the LC relaxation (See Chapter 9).

– There is evidence to suggest that introducing certain additional constraints to the LC relaxation does not improve the approximation it yields for any GCSP. Formally, the integrality gap of the LC relaxation does not improve on introducing any valid constraints on at most $2^{O((\log \log n)^{\frac{1}{4}})}$ SDP vectors (See Chapter 12).

– In recent work, Steurer [151] showed that the LC relaxation can be solved in near-linear time up to an additive error ε for every GCSP . We refer the reader to Section 4.5 for more details.

Organization In the next section, we formally define notions of equivalence between relaxations. In Section 4.3, we motivate a linear program (SIMPLE LP relaxation), whose formal definition and its comparison with other well-known linear programs is presented in Section 4.4. The SIMPLE LP relaxation is strengthened to obtain the LC SDP in Section 4.5. In the same section, we also show a few simple properties of the LC relaxation. This is followed by a comparison of LC relaxation with other SDPs, particularly those for boolean GCSPs in Section 4.6. We present two different hierarchies of stronger relaxations that are natural strengthenings of the LC relaxation in Section 4.7. In the penultimate section Section 4.8, we show how approximately feasible solutions to LC relaxation can be converted into a completely feasible solutions with a small loss in the objective value. This property of a SDP relaxation that we refer to as *robustness* has numerous applications in subsequent chapters. Finally, in Section 4.9 we show that the *robustness* property holds for even the stronger SDP relaxations LH_r and SA_r, although with much weaker parameters.

4.2 Comparing Relaxations

With an infinitude of relaxations to choose from, it is necessary to compare relaxations, and study relations among them. The notions of *integrality gap*, and the *gap curve* (See Section 2.2) of a relaxation are measures useful to compare the quality of approximations. In particular, a relaxation with a larger value of integrality gap clearly yields a better approximation ratio.

In many scenarios, there is more that can be said about two relaxations than just the comparison of the quality of the approximations they yield. For instance, two relaxations could be exactly the same except for a renaming of the variables involved. In some cases, the variables in one relaxation are obtained by a linear transformation of variables in the other. To capture these notions of equivalence, we make the following definition:

Definition 4.2.1. Let conv_1 and conv_2 be two relaxations of a maximization problem Λ. For an instance $\mathfrak{I}$ of the problem Λ, let $\mathrm{conv}_1(\mathfrak{I})$ and $\mathrm{conv}_2(\mathfrak{I})$ denote the objective value of the optimum solutions to relaxations conv_1, conv_2 respectively.

– The relaxation conv_1 is said to be *stronger* than conv_2, if for every instance $\mathfrak{I}$ of the maximization problem Λ,

$$\mathrm{conv}_1(\mathfrak{I}) \geqslant \mathrm{conv}_2(\mathfrak{I})$$

The relaxation conv_2 is said to be *weaker* than conv_1 in this case. The inequality reverses for the case of a minimization problem.

– The relaxations conv_1 and conv_2 are said to be *equivalent* if for every instance $\mathfrak{I}$ of the problem Λ.

$$\text{conv}_1(\mathfrak{I}) = \text{conv}_2(\mathfrak{I})$$

In the examples we consider, a much stronger equivalence holds between relaxations. Specifically, there exists an efficiently computable bijection (possibly a linear transformation) from feasible solutions to conv_1 to those of conv_2. However, we refrain from formally defining these stronger notions of equivalence.

4.3 Local Distributions and Consistency

Let $\mathfrak{I} = (\mathcal{V}, \mathcal{P})$ be an instance of a GCSP Λ. Let q denote the size of the alphabet, and let k be the arity of the GCSP Λ. By definition, the goal is to find an assignment $y \in [q]^{\mathcal{V}}$ that maximizes $\mathbb{E}_{P \in \mathcal{P}}[P(y)]$.

$$\text{GCSP Instance } \mathfrak{I}: \qquad \max_{y \in [q]^{\mathcal{V}}} \mathop{\mathbb{E}}_{P \in \mathcal{P}}[P(y)]$$

This is an optimization problem over the discrete set of $q^{|\mathcal{V}|}$ assignments. To obtain a convex relaxation, we need to modify the domain of optimization so as to make it convex. The most natural technique to make a set convex would be to take its convex hull. Intuitively, this implies that the domain should not only include the assignments $[q]^{\mathcal{V}}$, but also convex combinations of assignments.

In this light, let us change the domain of optimization to the set of all probability distributions over assignments. Clearly, the set of probability distributions over assignments is a *convex set*, since the average of two distributions is again a probability distribution. Furthermore, any specific assignment $y \in [q]^{\mathcal{V}}$ is represented by the distribution that has all its mass on the assignment y.

Formally, let μ be a distribution over the set of all assignments $[q]^{\mathcal{V}}$. We can rewrite the optimization problem in terms of the distribution μ as follows:

$$\text{Integral Hull } \mathfrak{I}: \qquad \max_{\mu} \mathop{\mathbb{E}}_{y \in \mu} \mathop{\mathbb{E}}_{P \in \mathcal{P}}[P(y)]$$

Unfortunately, the above convex problem cannot be solved efficiently in general. To see this, observe that to represent a distribution over all assignments, one needs exponentially many ($q^{|\mathcal{V}|}$) different variables. In fact, it is easy to see that solving the above convex program would yield an optimal solution to $\mathfrak{I}$.

Towards obtaining a tractable relaxation, observe that the payoff functions P are "local"– in that they each payoff $P \in \mathcal{P}$ depends solely on at most k variables. Let $\mathcal{V}(P)$ denote the set of variables on which the payoff P depends. For a given payoff $P \in \mathcal{P}$, the corresponding value $\mathbb{E}_{y \in \mu}[P(y)]$ depends solely on the distribution of the coordinates in $\mathcal{V}(P)$.

Therefore, instead of requiring the entire probability distribution μ over all assignments $[q]^{\mathcal{V}}$, we can restrict our attention to the marginal distributions of μ. Before we proceed further, let us define some notation for marginal distributions. Given a distribution μ over $[q]^S$ for some set S, and a subset $T \subseteq S$, let $\mathsf{margin}_T\, \mu$ denote the *marginal distribution* on

the set T. Formally, let

$$\mathsf{margin}_T\,\mu(x) \stackrel{\text{def}}{=} \sum_{y\in[q]^{S\setminus T}} \mu(x,y)\,.$$

Here, (x,y) denotes the $[q]$-assignment to S that agrees with x on T and with y on $S\setminus T$.

For the sake of succinctness, let $\mu_P = \mathsf{margin}_{\mathcal{V}(P)}\,\mu$ denote the marginal distribution over the set $\mathcal{V}(P)$ for a payoff $P \in \mathrm{supp}(\mathcal{P})$. Let $\boldsymbol{\mu} = \{\mu_P | P \in \mathrm{supp}(\mathcal{P})\}$ denote the set of marginal distributions. In terms of the marginal distributions $\boldsymbol{\mu}$, one can write a convex program as follows:

$$\text{Convex Relaxation } \Im\text{:} \qquad \max_{\boldsymbol{\mu}} \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{x\in\mu_P} P(x)$$

Note that a marginal distribution μ_P can be represented by $q^{|\mathcal{V}(P)|}$ variables $\mu_{P,x}$ for all $x \in [q]^{\mathcal{V}(P)}$. Hence, the set of marginal distributions $\boldsymbol{\mu} = \{\mu_P | P \in \mathcal{P}\}$ can be represented using at most $|\mathrm{supp}(\mathcal{P})| \times q^k$ variables. The objective value can be represented as a linear function in these variables.

$$\mathop{\mathbb{E}}_{x\in\mu_P} P(x) = \sum_{x\in[q]^{\mathcal{V}(P)}} P(x)\mu_{P,x}$$

Notice that there is no guarantee that the marginal distributions $\boldsymbol{\mu}$ actually correspond to a global distribution μ over the set of all assignments $[q]^{\mathcal{V}}$. However, we can enforce some consistency constraints between the marginal distributions μ_P.

A natural consistency check between the distributions is the following: given two distributions μ_P and $\mu_{P'}$ such that $S = \mathcal{V}(P) \cap \mathcal{V}(P')$ is non-empty,

$$\mathsf{margin}_S\,\mu_P = \mathsf{margin}_S\,\mu_{P'}\,.$$

Specifically, the marginals of the distributions μ_P and $\mu_{P'}$ restricted on the same subset S must be equal.

The above consistency check between distributions can be enforced for subsets S of various sizes. Somewhat surprisingly, enforcing consistency only up to sets of size up to 2 is sufficient for many purposes.

4.4 A Simple LP Relaxation

In the previous section, we obtained the following natural linear programming relaxation for an arbitrary GCSP Λ.

LP Relaxation for GCSPs

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \qquad \text{(SIMPLE LP)}$$

$$\text{subject to} \quad \mathsf{margin}_S\,\mu_P = \mathsf{margin}_S\,\mu_{P'} \quad (P,P' \in \mathrm{supp}(\mathcal{P}),\ S \subseteq \mathcal{V}(P)\cap\mathcal{V}(P'),\ |S| = 2)\,, \tag{4.1}$$

$$\mu_P \in \blacktriangle([q]^{\mathcal{V}(P)}) \tag{4.2}$$

Recall that $\blacktriangle(S)$ denotes the set of probability distributions over the set S. In the above relaxation, the constraint $\mu_P \in \blacktriangle([q]^{\mathcal{V}(P)})$ implies that μ_P is a valid probability distribution over $[q]$-ary assignments to variables $\mathcal{V}(P)$.

Notice that we are enforcing the consistency constraints between marginals only for sets of size up to 2. The above linear program is a simple linear program for every GCSP, that already yields good approximations in some cases.

In this section, we compare the SIMPLE LP relaxation with some other linear programs that have appeared in the context of Multiway Cut and Metric Labeling problems.

4.4.1 Multiway Cut *relaxation*

Definition 4.4.1 (Multiway Cut). An instance $\mathfrak{I} = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of the Multiway Cut problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of terminals $\mathsf{L} \subset \mathcal{V}$. The objective is to delete a set of edges of minimum weight so as to separate every pair of terminals. The weights over edges are assumed to sum up to 1, in that $\mathcal{E}$ is a distribution over $\mathcal{V} \times \mathcal{V}$.

Formulating Multiway Cut as a GCSP, the variables are the vertices $\mathcal{V}$, and the domain is the set of terminals L. Each edge $e = (u, v)$ is associated with a payoff of the form $P(x, y) = -\mathbf{1}[x \neq y]$. Here we take the negative of the indicator, because we are posing Multiway Cut - a minimization problem, as a GCSP where the goal is to maximize the payoff.

The following linear programming relaxation for Multiway Cut was introduced in [46], and yields the best known approximation for the problem. Let us suppose the number of terminals $|\mathsf{L}| = q$. Intuitively, the LP asks for an embedding of the vertices $\mathcal{V}$ on the q-dimensional simplex $\blacktriangle_q$. For every vertex v, the corresponding point $X_v \in \Delta_k$ represents the probability distribution of each terminal being assigned to v. For example, each corner of the simplex represents a particular terminal. Every terminal t_ℓ is to be assigned to itself, and this is enforced by fixing X_{t_ℓ} to the ℓ^{th} corner of $\blacktriangle_q$. The objective to minimize is the weighted sum of the L_1 distances between adjacent vertices.

Formally, the relaxation is given by the following where $\boldsymbol{e}_\ell$ denotes the ℓ^{th} corner of the simplex $\blacktriangle_q$.

Simplex-based relaxation for Multiway Cut

$$\text{maximize} \quad \frac{1}{2} \mathop{\mathbb{E}}_{e=(u,v)\in\mathcal{E}} \|X_u - X_v\|_1 \qquad \text{(Simplex)}$$

$$\text{subject to} \quad X_u \in \blacktriangle_q \quad \forall u \in V \qquad (4.3)$$

$$X_{t_\ell} = \boldsymbol{e}_\ell \qquad (4.4)$$

Lemma 4.0.1. *For the case of* Multiway Cut *, the SIMPLE LP relaxation is equivalent to the simplex based linear program.*

Proof. To show an equivalence, we will see how to convert an optimal solution to the SIMPLE LP relaxation into one for the simplex based linear program with exactly the same objective value (up to the sign) and vice versa.

Let $\boldsymbol{\mu} = (\mu_e)_{e \in \mathcal{E}}$ be an optimal solution to the SIMPLE LP relaxation. Define X_v to be the marginal distribution over the assignment to v, of a distribution μ_e for an edge incident at v. Specifically, define

$$X_v = \mathsf{margin}_{\{v\}}\, \mu_e \qquad \text{for an edge } e \in \mathcal{E}, \text{ with } v \in e$$

More precisely, define $X_{v,a} = \mathsf{margin}_{\{v\}}\, \mu_e(a)$ for all $a \in [q]$. As $\mathsf{margin}_{\{v\}}\, \mu_e$ is a distribution over $[q]$, the point X_v lies on the simplex $\blacktriangle_q$. Furthermore, the consistency constraints among the local distributions in SIMPLE LP relaxation ensure that the X_v is well defined.

For a vertex v, it is useful to think of X_v as a distribution of mass 1 (of say *sand*) on the points $\mathsf{L} = \{1, 2, \ldots, q\}$. For an edge $e = (u, v)$, the local distribution μ_e can be thought of as a flow that transforms the distribution of *sand* from X_u to X_v. The payoff amounts to a cost of 1 for every unit of *sand* that is moved from one location to the other. Hence, the optimal flow that minimizes the total cost incurs a cost exactly equal to $\frac{1}{2}\|X_u - X_v\|_1$. In other words, for an optimal solution $\boldsymbol{\mu} = (\mu_e)_{e \in \mathcal{E}}$, the payoff on edge e is exactly,

$$\mathop{\mathbb{E}}_{(a,b) \in \mu_e} [P(a, b)] = -\frac{1}{2}\|X_u - X_v\|_1$$

Therefore, an optimal solution to SIMPLE LP yields a solution to the simplex based relaxation of the same value.

Conversely, let $\{X_v | v \in \mathcal{V}\}$ be an optimal solution to the simplex based relaxation. For each edge e, there exists a feasible set of flows for each (u, v) that convert distribution X_u to X_v, while incurring a cost of exactly $\frac{1}{2}\|X_u - X_v|_1$. These flows define the local distributions μ_e that achieve the same value as the optimal solution $\{X_v | v \in \mathcal{V}\}$ for the simplex-based relaxation. ■

4.4.2 Earth-Mover Linear Program

Here we will compare the SIMPLE LP relaxation with the earthmover linear program introduced in [39] for Metric Labeling problems. The best known approximation ratios for Metric Labeling problems are achieved using the earthmover linear program.

For the sake of completeness, we include the definition of Metric Labeling below.

Definition 4.4.2. A Metric Labeling problem is specified as $\Lambda = (\mathsf{L}, d)$ where d is a metric over the set of labels L.

We will use q to denote the number of labels $|\mathsf{L}|$

Definition 4.4.3 (Λ-Metric Labeling)**.** An instance $\mathfrak{I} = (\mathcal{V}, \mathcal{E}, \{\mathsf{L}(v)\}_{v \in \mathcal{V}})$ of the Λ-Metric Labeling problem consists of a set of vertices $\mathcal{V}$, a probability distribution $\mathcal{E}$ over pairs from $\mathcal{V} \times \mathcal{V}$ (equivalent to edges with weights) and a family of subsets $\{\mathsf{L}(v)\}_{v \in \mathcal{V}}$ of L. A valid labeling is a mapping $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each vertex, $v \in \mathcal{V}$, $\mathcal{L}(v)$ belongs to $\mathsf{L}(v)$. The cost of a labeling $\mathcal{L}$, $\mathrm{val}_{\mathfrak{I}}(\mathcal{L})$, is

$$\mathop{\mathbb{E}}_{(u,v)=e \in \mathcal{E}} d(\mathcal{L}(u), \mathcal{L}(v)).$$

The optimum value of the instance, $\mathrm{opt}(\Im)$, is the minimum cost labeling for the instance.

4.4.3 Earthmover Linear Program for Metric Labeling

The Earthmover linear programming (EM-LP) relaxation for Metric Labeling was introduced by [39]. Let $\Im = (\mathcal{V}, \mathcal{E}, \{\mathsf{L}(v)\}_{v\in\mathcal{V}})$ be an instance of metric labeling. Intuitively, the EM-LP program finds an embedding of the vertices $\mathcal{V}$ on the q-dimensional simplex $\blacktriangle_q$. For every vertex v, there is a variable $X_v = (X_{v,\ell})_{\ell\in[q]}$ which is a point on the q-ary simplex $\blacktriangle_q$. The point X_v represents the probability distribution of labels being assigned to v.

The labeling constraint $\mathcal{L}(v) \in \mathsf{L}(v)$ is enforced by a linear constraint on the probability distribution X_v. Specifically, one can include the following constraints,

$$X_{v,\ell} = 0 \qquad \text{for all } \ell \notin \mathsf{L}(v)\,.$$

These labeling constraints force the point X_v to lie in the face containing the allowed labels $\mathsf{L}(v)$, denoted by $\blacktriangle_{\mathsf{L}(v)}$. The objective is to minimize the weighted sum of the *earthmover* distance between adjacent vertices which is defined below.

Definition 4.4.4 (Earthmover Distance)**.** Given two points $X, Y \in \blacktriangle_q$, and a metric $d(i,j)$ on $[q]$, the earthmover distance, $d_{\bowtie}(X,Y)$ is given by the optimal value of the following LP:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i,j\in[q]} d(i,j)\mu_{ij} \\ \text{s.t.} \quad & \sum_i \mu_{ij} = Y_j \qquad \sum_j \mu_{ij} = X_i \qquad \forall i,j \in [q] \\ & \mu_{ij} \geqslant 0 \end{aligned}$$

In other words, the earthmover distance is the minimum cost of moving the probability mass from distribution X to Y, given the distance metric d on the labels. It is easy to see that this defines a metric on the simplex $\blacktriangle_q$. Thus, the earthmover distance generalizes a metric on q points to a metric on $\blacktriangle_q$ such that the distance between corner points is the same as the original metric. In this notation, the linear program of [39] is simply:

$$\begin{aligned} \text{Minimize} \quad & \mathop{\mathbb{E}}_{(u,v)\in\mathcal{E}} d_{\bowtie}(X_u, X_v) && && \text{(EM-LP)} \\ \text{s.t.} \quad & X_u \in \blacktriangle(\mathsf{L}(u)) && \forall u \in \mathcal{V} && (4.5) \end{aligned}$$

Here again, $\blacktriangle(\mathsf{L}(u))$ refers to the set of probability distributions over the subset of labels $\mathsf{L}(u)$.

Lemma 4.0.2. *For a metric labelling problem Λ, the earthmover linear program is equivalent to the SIMPLE LP relaxation.*

Proof. Consider an optimal solution to the earthmover linear program. It consists of $\{X_v\}_{v\in\mathcal{V}}$ and the optimal flows $\{\mu_{e,ij}\}_{i,j\in\mathsf{L}}$. Clearly, the flow variables $\mu_{e,ij}$ yield the local

distributions μ_e, which are not only a feasible solution but have the exact same objective value.

Conversely, given a solution to the SIMPLE LP relaxation, the X_v variables can be obtained by setting $X_v = \mathsf{margin}_{\{v\}}\, \mu_e$ for some $e \ni v$. Furthermore, the local distributions μ_e yield the flow variables for the earthmover relaxation. Trivially, the resulting solution has the same objective value.

■

4.5 A Simple SDP Relaxation

In this section, we strengthen the SIMPLE LP relaxation to obtain a semidefinite programming relaxation. The basic idea is to enforce the distributions μ_e to be realized as inner products of vectors.

As earlier, let $\Im = (\mathcal{V}, \mathcal{P})$ be an instance of a GCSP Λ over a domain $[q]$ and arity k. The variables of the program consist of

– A collection of vectors $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in[q]}$. In the integral solution $\boldsymbol{b}_{i,a}$ is 1 if variable i is assigned a, and 0 otherwise.

  The intent for the vector solution $\{\boldsymbol{b}_{i,a}\}$ is that the vectors correspond to distributions over integral assignments. Specifically, in the intended solution, all vectors have only $\{0,1\}$-coordinates and for every i and every coordinate r, exactly one of the vectors $\boldsymbol{b}_{i,1}, \ldots, \boldsymbol{b}_{i,q}$ has a 1 in the r^{th} coordinate.

– A collection $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ of distributions over local assignments. For each payoff $P \in \mathcal{P}$, μ_P is a distribution over $[q]^{\mathcal{V}(P)}$ corresponding to assignments for the variables $\mathcal{V}(P)$.

The details of the relaxation are as follows:

LC Relaxation

$$\begin{aligned} \text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) && \text{(LC)} \\ \text{subject to} \quad & \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_P}\Big\{x_i = a, x_j = b\Big\}\,; \quad P \in \mathrm{supp}(\mathcal{P}),\ \ i,j \in \mathcal{V}(P),\ \ a,b \in [q]\,. \\ & \mu_P \in \blacktriangle([q]^{\mathcal{V}(P)}) \end{aligned}$$

In the above definition of LC relaxation, the set $\blacktriangle\left([q]^{\mathcal{V}(P)}\right)$ refers to the set of probability distributions over the $[q]$-ary assignments to $\mathcal{V}(P)$.

Notice that we have omitted the consistency condition between the marginal distributions that were part of the SIMPLE LP relaxation. Specifically, these were constraints of the form $\mathsf{margin}_S\, \mu_P = \mathsf{margin}_S\, \mu_{P'}$ for two local distributions μ_P and μ'_P. These consistency constraints are enforced here via the constraints on the inner products of vectors. For a set

$S = \{i, j\} \subseteq \mathcal{V}(P) \cap \mathcal{V}(P')$ that lies in the intersection of two payoffs P, P' and $a, b \in [q]$,

$$\begin{aligned}\mathsf{margin}_S\, \mu_P(a,b) &= \Pr_{x \in \mu_P}\Big\{x_i = a, x_j = b\Big\} \\ &= \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x \in \mu_{P'}}\Big\{x_i = a, x_j = b\Big\} = \mathsf{margin}_S\, \mu_{P'}(a,b)\end{aligned}$$

The total number of variables in the above semidefinite program is $poly(q^k, |\mathrm{supp}(\mathcal{P})|)$. Therefore, the above semidefinite program can be solved up to an additive error of ε, in time $poly(q^k, |\mathrm{supp}(\mathcal{P})|, \log 1/\varepsilon)$ using interior point methods. More recently, building on the work of Arora-Kale [13], Steurer [151] exhibited an algorithm running in near linear time that computes the optimal SDP solution up to an additive error ε. For the sake of completeness, we restate the result of Steurer [151] below:

Theorem 4.1 (Theorem 1.3, [151])**.** *There is an algorithm $\mathcal{A}$ that on input a GCSP instance $\Im$ on n variables and $m \geqslant n$ payoffs, alphabet size q and arity k, finds a SDP solution to the LC relaxation whose objective value is at least $\mathsf{LC}(\Im) - \varepsilon$. The running time of the algorithm $\mathcal{A}$ is bounded by $poly(q^k/\varepsilon) \times (m + n \log^2 n)$.*

Remark 4.5.1. Note that the LC relaxation can be solved in polynomial time even for $q = poly(n)$. Thus, the LC relaxation can be written for GCSPs whose domain size grows with the input. Ordering constraint satisfaction problems (see Chapter 9 for definitions) such as Maximum Acyclic Subgraph can be posed as a GCSP with a domain size n. In Chapter 9, we will show that the resulting relaxation yields the optimal approximation ratio for every OCSP.

4.5.1 Additional Properties

While the LC relaxation appears fairly minimal, feasible solutions to it satisfy several additional useful properties. We state some of these properties here.

Observation 4.5.1. *For every variable $i \in \mathcal{V}$, the set of vectors $\{\boldsymbol{b}_{i,a} | a \in [q]\}$ satisfy:*

$$\begin{aligned}\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{i,a'}\rangle &= 0 && \forall\ a, a' \in [q], a \neq a' \\ \sum_{a \in [q]} \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{i,a}\rangle &= 1\end{aligned}$$

Proof. Let P be a payoff such that $i \in \mathcal{V}(P)$, and let μ_P denote the associated distribution. Both of the above equations follow trivially by observing that $\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{i,a'}\rangle = \Pr_{x \in \mu_P}\{x_i = a, x_i = a'\}$. ∎

Observation 4.5.2. *Given an arbitrary feasible solution $(\boldsymbol{V}, \boldsymbol{\mu})$, there exists a feasible*

solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$ *with the same objective value and a vector* $\boldsymbol{b}_0$ *such that*

$$\sum_{a\in[q]} \boldsymbol{b}^*_{i,a} = \boldsymbol{b}_0 \qquad \forall i \in \mathcal{V}\,, \tag{4.6}$$

$$\langle \boldsymbol{b}^*_{i,a}, \boldsymbol{b}_0\rangle = \|\boldsymbol{b}^*_{i,a}\|_2^2 \qquad \forall i \in \mathcal{V}, a \in [q]\,, \tag{4.7}$$

$$\|\boldsymbol{b}_0\|_2^2 = 1 \tag{4.8}$$

Proof. From Observation 4.5.1, for every vertex $\sum_{a\in[q]} \boldsymbol{b}_{i,a}$ is a unit vector. Furthermore, for two variables i, j that participate together in a payoff P,

$$\langle \sum_{a\in[q]} \boldsymbol{b}_{i,a}, \sum_{a'\in[q]} \boldsymbol{b}_{j,a'}\rangle = \sum_{a\in[q],a'\in[q]} \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,a'}\rangle = \sum_{a\in[q],a'\in[q]} \Pr_{x\in\mu_P}\left\{x_i = a, x_j = a'\right\} = 1\,.$$

Note that the inner product between two unit vectors is 1 if and only if the two are equal. Consequently, for two variables $i, j \in \mathcal{V}(P)$ that belong to the same payoff P, we have $\sum_{a\in[q]} \boldsymbol{b}_{i,a} = \sum_{a'\in[q]} \boldsymbol{b}_{j,a'}$.

Consider the hypergraph H whose vertices are the variables $\mathcal{V}$, and hyperedges are the sets $\{\mathcal{V}(P)|P \in \mathrm{supp}(\mathcal{P})\}$. Let $C_1 \cup C_2 \cup \ldots C_t = H$ be the decomposition of H into connected components. For all variables i within a connected component C_ℓ, the vector $\sum_{a\in[q]} \boldsymbol{b}_{i,a}$ are equal. Let $\boldsymbol{b}_0^{(\ell)}$ denote the vector $\sum_{a\in[q]} \boldsymbol{b}_{i,a}$ corresponding to $i \in C_\ell$.

For each component C_ℓ for $\ell > 1$, rotate all the corresponding vectors, so that $\boldsymbol{b}_0^{(\ell)}$ coincides with $\boldsymbol{b}_0^{(1)}$. The transformed SDP vectors $\{\boldsymbol{b}^*_{i,a}\}$ form the new SDP vectors. The set of distributions $\boldsymbol{\mu}^*$ is the same as $\boldsymbol{\mu}$. Set $\boldsymbol{b}_0 = \boldsymbol{b}_0^{(1)}$.

The transformations preserve all inner products within connected components C_ℓ. In particular, for any given payoff P, the inner products between SDP vectors $\{\boldsymbol{b}_{i,a}|i \in \mathcal{V}(P), a \in [q]\}$ remain unchanged. Therefore, the inner products of transformed vectors still match the local distributions $\{\mu_P|P \in \mathcal{P}\}$.

By definition, the vectors satisfy $\sum_{a\in[q]} \boldsymbol{b}^*_{i,a} = \boldsymbol{b}_0$ for all $i \in \mathcal{V}$. Equations 4.6 and 4.7 follow easily from the definition of $\boldsymbol{b}_0$ and orthogonality of vectors $\{\boldsymbol{b}^*_{i,a}|a \in [q]\}$ for every $i \in \mathcal{V}$. The formal proofs are included below for convenience.

$$\langle \boldsymbol{b}^*_{i,a}, \boldsymbol{b}_0\rangle = \langle \boldsymbol{b}^*_{i,a}, \sum_{a'\in[q]} \boldsymbol{b}^*_{i,a'}\rangle = \|\boldsymbol{b}^*_{i,a}\|_2^2 + \sum_{\substack{a\neq a'\\ a,a'\in[q]}} \langle \boldsymbol{b}^*_{i,a}, \boldsymbol{b}^*_{i,a'}\rangle = \|\boldsymbol{b}^*_{i,a}\|_2^2$$

$$\langle \boldsymbol{b}_0, \boldsymbol{b}_0\rangle = \langle \sum_{a\in[q]} \boldsymbol{b}^*_{i,a}, \sum_{a'\in[q]} \boldsymbol{b}^*_{i,a'}\rangle = \sum_{a,a'\in[q]} \langle \boldsymbol{b}^*_{i,a}, \boldsymbol{b}^*_{i,a'}\rangle = \sum_{a\in[q]} \langle \boldsymbol{b}^*_{i,a}, \boldsymbol{b}^*_{i,a}\rangle = 1$$

■

4.6 Comparison with Relaxations in Literature

To the best of our knowledge, the only instances where a semidefinite program *stronger* than the LC relaxation has been used to approximate a GCSP is [41]. This work exhibits an algorithm for Unique Games that uses techniques from the breakthrough work of Arora-

Rao-Vazirani [16], and thereby relies on a semidefinite program with triangle inequalities. Furthermore, even in this case, the stronger SDP only improves the approximation in a sub-constant regime where the input is a Unique Games instance that is $1-\varepsilon$ satisfiable for $\varepsilon \leqslant \frac{1}{\log n}$.

Therefore, for every GCSP when we are interested in approximation factors that are constant, no SDP stronger than LC has proved useful.

We remark here that in some works like [35], the SDP relaxations used are stronger than LC as they are stated. However, by Observation 4.5.2 the LC is equivalent to certain stronger SDP relaxations with additional constraints. Moreover, a close examination of the rounding schemes reveal that some of the constraints of the semidefinite program could be omitted.

In the rest of the section, we will compare the LC relaxation with some well-known SDPs, for GCSPs over the boolean alphabet.

$\{\pm 1\}$-relaxations In the LC relaxation, assignments from the alphabet $[q]$ to a variable x_i were encoded using q different variables $\{\boldsymbol{b}_{i,1},\dots,\boldsymbol{b}_{i,q}\}$, exactly one of which can be 1. For combinatorial optimization problems over the boolean domain such as Max Cut or Sparsest Cut, it is natural to use relaxations that arise out of a $\{\pm 1\}$-encoding of solutions.

Let us consider a GCSP Λ over the domain $\{0,1\}$. By convention, we will encode $0 \to 1$ and $1 \to -1$, i.e., $a \to (-1)^a$. Let $\mathfrak{I}$ be an instance of Λ over a set of variables $\mathcal{V}$. For each variable $i \in \mathcal{V}$, we introduce a vector $\boldsymbol{v}_i$ in a $\{\pm 1\}$-relaxation. In the integral solution, the vector $\boldsymbol{v}_i$ is intended to be either -1 or $+1$ when i is assigned 0 or 1 respectively. More generally, the vector $\boldsymbol{v}_i$ is intended to be a vector with $\{\pm 1\}$ coordinates. The LC relaxation can be equivalently formulated as a $\{\pm 1\}$-relaxations in the following manner.

An *SDP solution* $(\boldsymbol{V}, \boldsymbol{\mu})$ for the LC-BIN relaxation of an instance $\mathfrak{I}$ consists of the following:

- A collection of vectors $\{\boldsymbol{v}_i\}_{i\in\mathcal{V}}$. In the integral solution $\boldsymbol{v}_i$ is 1 if variable i is assigned 0, and -1 otherwise.
- A collection $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ of distributions over local assignments $\{\pm 1\}^{\mathcal{V}(P)}$.
- A unit vector $\boldsymbol{b}_0$ which is intended to be equal to 1.

LC-BIN Relaxation

$$\begin{aligned} \text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) && && \text{(LC-BIN)}\\ \text{subject to} \quad & \langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle = \mathop{\mathbb{E}}_{x\sim\mu_P} x_i x_j && (P\in\mathrm{supp}(\mathcal{P}),\ \ i,j\in\mathcal{V}(P)\ )\,. \\ & \langle \boldsymbol{v}_i, \boldsymbol{b}_0\rangle = \mathop{\mathbb{E}}_{x\sim\mu_P} x_i && (P\in\mathrm{supp}(\mathcal{P}),\ \ i\in\mathcal{V}(P)\ )\,. \\ & \mu_P \in \blacktriangle(\{\pm 1\}^{\mathcal{V}(P)}) \end{aligned}$$

Lemma 4.1.1. *The LC-BIN relaxation is equivalent to the LC relaxation for a GCSP Λ over the boolean domain.*

Proof. Let $(\{\boldsymbol{v}_i\}_{i\in\mathcal{V}} \cup \{\boldsymbol{b}_0\}, \boldsymbol{\mu})$ form a feasible solution to the LC-BIN relaxation. Define vectors $\boldsymbol{b}_{i,a}$ as follows,

$$\boldsymbol{b}_{i,0} = \frac{(\boldsymbol{b}_0 + \boldsymbol{v}_i)}{2}\,, \quad \boldsymbol{b}_{i,1} = \frac{(\boldsymbol{b}_0 - \boldsymbol{v}_i)}{2} \qquad \forall i \in \mathcal{V}\,.$$

For a vector $y \in \{0,1\}^d$, we will write $(-1)^y$ to denote the $\{\pm 1\}$-vector $\left((-1)^{x_i}\right)_{i=1}^d$. For each payoff, define a probability distribution μ'_P over $\{\pm 1\}^{\mathcal{V}(P)}$ as $\mu'_P(y) = \mu_P\big((-1)^y\big)$. We claim that $(\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}\}_{i\in\mathcal{V}}, \{\mu'_P\}_{P\in\mathcal{P}})$ is a feasible solution to the LC relaxation. To see this, observe that for every $i, j \in \mathcal{V}$, $P \in \mathcal{P}$ and $a, b \in \{0,1\}$ we have,

$$\begin{aligned}
\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle &= \frac{1}{4}\langle (\boldsymbol{b}_0 + (-1)^a \boldsymbol{v}_i)\,, (\boldsymbol{b}_0 + (-1)^a \boldsymbol{v}_i)\rangle\,,\\
&= \frac{1}{4}\left(1 + (-1)^a \mathop{\mathbb{E}}_{x\in\mu_P} x_i + (-1)^b \mathop{\mathbb{E}}_{x\in\mu_P} x_j + (-1)^{a+b} \mathop{\mathbb{E}}_{x\in\mu_P} x_i x_j\right)\,,\\
&= \frac{1}{4}\mathop{\mathbb{E}}_{x\in\mu_P}\left[(1 + (-1)^a x_i)\cdot\left(1 + (-1)^b x_j\right)\right]\,,\\
&= \mathop{\mathbb{P}\mathrm{r}}_{x\in\mu_P}\left\{x_i = (-1)^a, x_j = (-1)^b\right\} = \mathop{\mathbb{P}\mathrm{r}}_{y\in\mu'_P}\left\{y_i = a, y_j = b\right\}\,.
\end{aligned}$$

Conversely, let $(\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}\}_{i\in\mathcal{V}}, \boldsymbol{\mu})$ be a feasible SDP solution for the LC relaxation. Without loss of generality, we may assume that there exists a unit vector $\boldsymbol{b}_0$ satisfying the properties outlined in Observation 4.5.2. Define the new SDP solution $(\{\boldsymbol{v}_i\}_{i\in\mathcal{V}} \cup \{\boldsymbol{b}_0\}, \boldsymbol{\mu}')$ as follows,

$$\begin{aligned}
\boldsymbol{v}_i &= \boldsymbol{b}_{i,0} - \boldsymbol{b}_{i,1} && \forall i \in \mathcal{V}\,,\\
\mu'_P(x) &= \mu_P\left(\frac{\mathbf{1} - x}{2}\right) && \forall P \in \mathcal{P}, x \in \{\pm 1\}^{\mathcal{V}}(P)\,.
\end{aligned}$$

Here $\mathbf{1}$ denotes the vector all of whose components are equal to 1. ■

Boolean 2-CSP

Lemma 4.1.2. *For a GCSP of arity 2 over the boolean domain $\{0,1\}$, the LC is equivalent to the BS relaxation described below.*

$$\textit{Maximize} \quad \mathop{\mathbb{E}}_{P\in\mathcal{P}}\Big[\sum_{a,b\in\{0,1\}} P(a,b)\cdot\Big(\frac{1}{4}\langle \boldsymbol{b}_0+(-1)^a\boldsymbol{v}_i, \boldsymbol{b}_0+(-1)^b\boldsymbol{v}_j\rangle\Big)\Big] \tag{BS}$$

$$\begin{aligned}\textit{Subject to } \langle(\boldsymbol{b}_0\pm\boldsymbol{v}_i),(\boldsymbol{b}_0\pm\boldsymbol{v}_j)\rangle &\geqslant 0 \qquad \forall i,j\in\mathcal{V}\\ |v_i|^2 &= 1 \qquad \forall i\in\mathcal{V}\\ |v_i|^2 &= 1 \qquad \forall i\in\mathcal{V}\end{aligned}$$

Proof. By Lemma 4.1.1, it is sufficient to show equivalence to the LC-BIN relaxation.

Firstly, the above relaxation is *weaker* than the LC-BIN. Specifically, the constraints of LC-BIN yield,

$$\langle(\boldsymbol{b}_0\pm\boldsymbol{v}_i),(\boldsymbol{b}_0\pm\boldsymbol{v}_j)\rangle = \mathop{\mathbb{E}}_{x\in\mu_P}(1\pm x_i)(1\pm x_j)\geqslant 0$$

.

Conversely, to show that LC-BIN is *weaker* than the BS, fix a payoff P, and let $\mathcal{V}(P)=\{i,j\}$. Define a local distribution μ_P as follows:

$$\mu_P(a,b)=\frac{1}{4}\langle(\boldsymbol{b}_0\pm\boldsymbol{v}_i),(\boldsymbol{b}_0\pm\boldsymbol{v}_j)\rangle$$

The constraints of the SDP ensure that $\mu_P(a,b)\geqslant 0$ for all $a,b\in\{\pm1\}$. Furthermore, it follows immediately from the definition that $\mu_P(0,0)+\mu_P(0,1)+\mu_P(1,0)+\mu_P(1,1)=1$. Finally, it is a straight forward calculation to show that $\mathbb{E}_{x\in\mu_P}x_ix_j=\langle\boldsymbol{v}_i,\boldsymbol{v}_j\rangle$ and $\mathbb{E}_{x\in\mu_P}x_i=\langle\boldsymbol{b}_0,\boldsymbol{v}_j\rangle$. Thus, $(\{\boldsymbol{v}_i\}_{i\in\mathcal{V}},\{\mu_P\}_{P\in\mathcal{P}})$ is a feasible solution to the LC-BIN relaxation. ■

Remark 4.6.1. The constraints of the form $\langle(\boldsymbol{b}_0\pm\boldsymbol{v}_i),(\boldsymbol{b}_0\pm\boldsymbol{v}_j)\rangle\geqslant 0$ are just the *triangle inequalities* between the vectors $\boldsymbol{b}_0,\boldsymbol{v}_i$ and $\boldsymbol{v}_j$.

In [114], a variant of the BS SDP relaxation is used to obtain the best known approximation for the Max 2-Sat problem. The SDP relaxation in [114] imposes the *triangle inequalities* on every $\boldsymbol{b}_0,\boldsymbol{v}_i,\boldsymbol{v}_j$ for all $i,j\in\mathcal{V}$. Instead, the BS relaxation imposes the triangle inequalities only for pairs $i,j\in\mathcal{V}$ that participate together in a payoff/constraint. However, the rounding scheme presented in [114] still yields the same approximation ratio when used with the seemingly weaker relaxation LC-BIN. Specifically, the analysis of the rounding scheme in [114] is *local* in that it depends solely on the geometry of vectors within individual constraints. The BS SDP relaxation

Max Cut Wrapping up the section, we finally compare the relaxation to the Goemans-Williamson SDP relaxation for the classic problem of Max Cut. We restate the GW SDP relaxation for the sake of convenience.

	$\mathsf{GW}(G)$ Relaxation		(GW)
Maximize	$\frac{1}{4}\sum_{(v_i,v_j)\in E}\|\boldsymbol{v}_i-\boldsymbol{v}_j\|_2^2$		(Total Squared Length of Edges)
Subject to	$\|\boldsymbol{v}_i\|_2^2=1$	$\forall i\in\mathcal{V}$	(all vectors $\boldsymbol{v}_i$ are unit vectors)

Lemma 4.1.3. *For* Max Cut *the* LC *relaxation is equivalent to the* GW *relaxation.*

Proof. By Lemma 4.1.1 and Lemma 4.1.2 , it is sufficient to show equivalence between GW and the BS relaxation.

Clearly, BS is a stronger relaxation than GW. Given a solution $\{\boldsymbol{v}_i\}_{i\in\mathcal{V}}$ to the GW relaxation, set $\boldsymbol{b}_0$ to be a unit vector orthogonal to all $\boldsymbol{v}_i$. Then, the SDP solution $\{\boldsymbol{v}_i\}_{i\in\mathcal{V}}\cup\{\boldsymbol{b}_0\}$ forms a feasible solution to the BS, thereby finishing the proof. ■

4.7 Stronger Relaxations

Towards obtaining better approximations, a natural avenue is to utilize stronger LP/SDP relaxations that include greater number of constraints. There are numerous choices of additional constraints that can be included to strengthen a given convex relaxation. In fact, there are several hierarchies of increasingly stronger relaxations such as the Lovász-Schriver [118], Lasserre [110] and Sherali-Adams hierarchies [150] that have been proposed in literature.

Here, we describe two hierarchies of relaxations that are natural strengthenings of the LC relaxation, and are closely related to the Sherali-Adams hierarchy [150].

LH_r relaxation In the LC relaxation, the inner products of SDP vectors $\{\boldsymbol{b}_{i,a}|i\in\mathcal{V},a\in[q]\}$ are constrained to agree with the local distributions $\{\mu_P|P\in\mathrm{supp}(\mathcal{P})\}$. However, the relaxation contains local distributions μ_S only for sets S that are $S=\mathcal{V}(P)$ for some payoff $P\in\mathrm{supp}(\mathcal{P})$. A natural way to strengthen the relaxation is to include local distributions for every set S of size say r. We refer to the resulting SDP as the LH_r relaxation.

For a constant r, it is easy to see that the LH_r SDP has $poly(n^r,q^k)$ constraints for a GCSP instance $\mathfrak{S}$ with n variables, over alphabet q and arity k. Hence, the LH_r relaxation can be solved in polynomial time for constant r. For the sake of convenience, we include the detailed definition of the LH_r relaxation.

Let $\mathfrak{S}$ be a GCSP instance over a set of variables $\mathcal{V}$, alphabet size q and arity k. A feasible solution to the LH_r relaxation consists of the following:

1. A *collection of (local) distributions* $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$, where $\mu_S\colon[q]^S\to\mathbb{R}_+$ is a distribution over $[q]$-assignments to S, that is, $\mu_S\in\blacktriangle([q]^S)$.

2. A *(global) vector solution* $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in[q]}$, where $\boldsymbol{b}_{i,a}\in\mathbb{R}^d$ for every $i\in\mathcal{V}$ and $a\in[q]$.

The intention for the local distributions $\{\mu_S\}$ is again that they arise as the marginal distribution of a global distribution $\mu\colon [q]^{\mathcal{V}} \to \mathbb{R}_+$ over $[q]$-assignments to the variables $\mathcal{V}$. The intention for the vector solution $\{\boldsymbol{b}_{i,a}\}$ is that all vectors have only $\{0,1\}$-coordinates and that for every i and every coordinate ℓ, exactly one of the vectors $\boldsymbol{b}_{i,1}, \ldots, \boldsymbol{b}_{i,q}$ has a 1 in the ℓ^{th} coordinate.

LH_r-Relaxation.

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \tag{LH_r}$$

$$\text{subject to} \quad \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_S}\Big\{x_i = a, x_j = b\Big\} \qquad S \subseteq \mathcal{V},\ |S| \leqslant r,\ \ i,j \in S,\ \ a,b \in [q]\,, \tag{4.9}$$

$$\mu_S \in \blacktriangle\left([q]^S\right) \tag{4.10}$$

Here, $\blacktriangle\left([q]^S\right)$ denotes probability distributions over $[q]^S$. As usual, we denote by $\mathsf{LH}_r(\Im)$ the value of an optimal solution to this relaxation.

The above relaxation succinctly encodes all possible inequalities on up to r vectors. The next remark makes this observation precise.

Remark 4.7.1. A linear inequality on the inner products of a subset of vectors $\{\boldsymbol{b}_{i,a}\}_{i\in S, a\in[q]}$ for $S \subseteq \mathcal{V}$ is *valid* if it inequality if it holds for all distributions over $[q]$-assignments to the variables S. A feasible solution to the LH_r-relaxation satisfies all valid inequalities on sets of up to r vectors.

4.7.1 SA_r*-Relaxation*

Notice that the local distributions in the LH_r-relaxation have redundancies. Specifically, consider two sets $A, B \subseteq \mathcal{V}$ such that $A \subset B$ and $|A|, |B| \leqslant r$. The local distribution μ_B induces a distribution $\mathsf{margin}_A\, \mu_B$ over assignments to the set A, since $A \subset B$. (Here, $\mathsf{margin}_A\, \mu_B$ denotes the marginal of μ_B on the set A) It is but natural to enforce that $\mathsf{margin}_A\, \mu_B$ and μ_A be the same distribution.

SA_r-Relaxation:

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \tag{SA_r}$$

$$\text{subject to} \quad \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_S}\Big\{x_i = a, x_j = b\Big\} \qquad S \subseteq \mathcal{V},\ \ |S| \leqslant r,\ \ i,j \in S,\ \ a,b \in [q]\,, \tag{4.11}$$

$$\|\mathsf{margin}_{A\cap B}\, \mu_A - \mathsf{margin}_{A\cap B}\, \mu_B\|_1 = 0 \qquad A, B \subseteq \mathcal{V},\ \ |A|, |B| \leqslant r\,. \tag{4.12}$$

$$\mu_S \in \blacktriangle\left([q]^S\right) \tag{4.13}$$

Again, $\blacktriangle([q]^S)$ denotes the set of probability distributions over $[q]$-ary assignments for the variables in S. As usual, we denote by $\mathsf{SA}_r(\Im)$ the value of an optimal solution to this relaxation.

Remark 4.7.2. The SA_r relaxation is closely related to the r^{th} level of the Sherali–Adams hierarchy. In fact, SA_r is obtained from the basic SDP relaxation by r-rounds Sherali–Adams lift-and-project. In other words, we are optimizing over the intersection of the basic SDP relaxation and the Sherali–Adams relaxation.

$\{\pm 1\}$-relaxations The LH_r and SA_r SDPs can also be written as $\{\pm 1\}$-relaxations for GCSPs over boolean alphabet. For the sake of completeness, we include the formulations below.

LH_r-Relaxation ($\{\pm 1\}$-version)

$$\begin{aligned}
\text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \\
\text{subject to} \quad & \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = \mathop{\mathbb{E}}_{x\sim\mu_S} x_i x_j && S \subseteq \mathcal{V},\ |S| \leqslant r,\ \ i,j \in S\,, \\
& \langle \boldsymbol{v}_i, \boldsymbol{b}_0 \rangle = \mathop{\mathbb{E}}_{x\sim\mu_P} x_i && P \in \operatorname{supp}(\mathcal{P}),\ \ i \in \mathcal{V}(P)\ .\\
& \mu_S \in \blacktriangle\left(\{\pm 1\}^S\right)
\end{aligned}$$

SA_r-Relaxation ($\{\pm 1\}$-version)

$$\begin{aligned}
\text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) && && (\mathsf{SA}_r)\\
\text{subject to} \quad & \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = \mathop{\mathbb{E}}_{x\sim\mu_S} x_i x_j && S \subseteq \mathcal{V},\ |S| \leqslant r,\ \ i,j \in S\,, \\
& \langle \boldsymbol{v}_i, \boldsymbol{b}_0 \rangle = \mathop{\mathbb{E}}_{x\sim\mu_P} x_i\,, \\
& \mathsf{margin}_{A\cap B}\,\mu_A = \mathsf{margin}_{A\cap B}\,\mu_B && A,B \subseteq \mathcal{V},\ |A|,|B| \leqslant r\,.\\
& \mu_S \in \blacktriangle\left(\{\pm 1\}^S\right)
\end{aligned}$$

Here, $\boldsymbol{b}_0 \in \mathbb{R}^d$ is an arbitrary fixed unit vector.

4.8 Robustness and Smoothing of the LC relaxation

In this section, we will be interested in the robustness of the SDP relaxations LC,LH_r and SA_r to "noise". More precisely, suppose $(\boldsymbol{V}, \boldsymbol{\mu})$ is a an approximately feasible solution in that it satisfies all the constraints of one of these SDP relaxations within a tiny error of $\pm\varepsilon$. Then, the robustness of the SDP relaxation refers to how close the solution must be to a completely feasible solution. Robustness of a SDP relaxation will prove useful in rounding the SDP solutions (Chapter 5), constructing integrality gaps (Chapter 12), reductions from integrality gaps to dictatorship tests (Chapter 7) and even in solving the SDPs efficiently [151].

Formally, an ε-infeasible solution to a SDP relaxation is defined as follows.

Definition 4.8.1. An SDP solution $\{\boldsymbol{v}_{i,a}\}_{i\in\mathcal{V},a\in\mathbb{F}_q}$, $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ is said to be *ε-infeasible* for a SDP relaxation if it satisfies all the constraints of the program up to an additive error of ε.

As defined earlier, the LC relaxation is reasonably robust. However, including certain additional constraints into the relaxation make the robustness argument simpler. Specifically, we rewrite LC in the following equivalent fashion.

LC Relaxation (Equivalent Version)

$$\begin{aligned}
\text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \\
\text{subject to} \quad & \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_P}\Big\{x_i = a, x_j = b\Big\} \quad P \in \operatorname{supp}(\mathcal{P}),\ \ i,j \in \mathcal{V}(P),\ \ a,b \in [q]\,. \qquad (4.14)\\
& \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_0\rangle = \|\boldsymbol{b}_{i,a}\|_2^2 \qquad \forall i \in \mathcal{V}, a \in [q]\,, \qquad (4.15)\\
& \|\boldsymbol{b}_0\|_2^2 = 1 \qquad (4.16)\\
& \mu_P \in \blacktriangle([q]^{\mathcal{V}(P)})
\end{aligned}$$

To the original definition of LC relaxation, we have included two additional constraints (4.15) and (4.16). As was shown in Observation 4.5.1 and Observation 4.5.2, any solution to the LC can be transformed to satisfy these additional constraints, without any loss in objective value. Therefore, the above relaxation is equivalent to the LC relaxation.

Theorem 4.2 (Robustness of LC)**.** *Let $\mathcal{P}$ be a Λ-CSP instance on variable set $\mathcal{V}$. Suppose that $\{\boldsymbol{b}_{i,a}\}_{i\in V, a\in[q]}$, $\{\mu_P\}_{P\in\operatorname{supp}(\mathcal{P})}$ is an ε-infeasible SDP solution for $\mathcal{P}$ of value α. Here, ε-infeasible means that all consistency constraints (4.14)–(4.16) of the relaxation LC are satisfied up to an additive error of at most ε. Then,*

$$\operatorname{sdp}(\mathcal{P}) \geqslant \alpha - \sqrt{\varepsilon}\cdot \operatorname{poly}(kq)\,.$$

4.8.1 Surgery & Smoothing

Let $\{\boldsymbol{b}_{i,a}\}$, $\{\mu_P\}$ be an ε-infeasible SDP solution for a Λ-CSP instance $\mathcal{P}$ on the variable set $V = [n]$. Recall that an ε-infeasible SDP solution satisfies ,

$$\left|\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle - \Pr_{x\sim\mu_P}\Big\{x_i = a,\ x_j = b\Big\}\right| \leqslant \varepsilon \qquad (4.17)$$

for all $P \in \operatorname{supp}(\mathcal{P})$, $i,j \in \mathcal{V}(P)$, and $a,b \in [q]$ and for all $i \in \mathcal{V}(P)$ and $a \in [q]$,

$$\left|\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_0\rangle - \|\boldsymbol{b}_{i,a}\|_2^2\right| \leqslant \varepsilon \qquad (4.18)$$

We construct a feasible solution that is close to the given SDP solution in two steps.

In the first step, called "surgery", we construct vectors $\{\boldsymbol{u}_{i,a}\}$ and $\boldsymbol{b}_0$ that satisfy the equality constraints on SDP vectors, i.e., $\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,b}\rangle = 0$ for all $a \neq b \in [q]$ and all $i \in \mathcal{V}$ and $\sum_{a\in[q]} \boldsymbol{u}_{i,a} = \boldsymbol{b}_0$ for all $i \in \mathcal{V}$.

In the second step, called "smoothing", we construct a feasible SDP solution $\{\boldsymbol{w}_{i,a}\}$, $\{\mu'_P\}$. In this step, the vectors and the local distributions are "smoothed" which allows us to modify the local distributions so that they match the vectors perfectly.

Lemma 4.2.1. *The vectors $\{\boldsymbol{b}_{i,a}\} \cup \{\boldsymbol{b}_0\}$ can be transformed to vectors $\{\boldsymbol{u}_{i,a}\} \cup \{\boldsymbol{u}_0\}$ such that for all $a \neq b \in [q]$ and all $i \in V$,*

$$\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,b} \rangle = 0\,, \tag{4.19}$$

and for all $i \in V$,

$$\sum_{a \in [q]} \boldsymbol{u}_{i,a} = \boldsymbol{u}_0\,. \tag{4.20}$$

Furthermore, for $i \in V$ and $a \in [q]$,

$$\|\boldsymbol{u}_{i,a} - \boldsymbol{b}_{i,a}\| \leqslant \sqrt{\varepsilon} \cdot \mathrm{poly}(q)\,. \tag{4.21}$$

In particular, the SDP solution $\{\boldsymbol{u}_{i,a}\}$, $\{\mu_P\}$ is η-infeasible for $\eta = \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$.

Proof. First, the length of $\boldsymbol{b}_0$ is in the range, $[1-\varepsilon, 1+\varepsilon]$. Normalize the vector $\boldsymbol{b}_0$ to a unit vector to obtain $\boldsymbol{u}_0$. From (4.17) and (4.18) it follows that $\|\boldsymbol{b}_{i,a}\|^2 \leqslant 1+\varepsilon$ and $|\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{i,b} \rangle| \leqslant \varepsilon$. for all $a \neq b \in [q]$. Therefore, if we apply the Gram–Schmidt orthogonalization process on the vectors $\boldsymbol{b}_{i,1}, \ldots, \boldsymbol{b}_{i,q}$, the resulting vectors $\boldsymbol{v}'_{i,1}, \ldots, \boldsymbol{v}'_{i,q}$ satisfy $\|\boldsymbol{b}_{i,a} - \boldsymbol{b}'_{i,a}\| \leqslant O(\varepsilon \cdot q)$. For every variable $i \in V$, we compute a rescaling factor α_i such that $\boldsymbol{b}_{i,0} := \sum_{a \in [q]} \alpha_i \boldsymbol{v}'_{i,a}$ is a unit vector. Note that $\alpha_i = 1 \pm \varepsilon \cdot \mathrm{poly}(q)$. Furthermore, $\langle \boldsymbol{b}_{i,0}, \boldsymbol{b}_0 \rangle \geqslant 1 - \varepsilon \cdot \mathrm{poly}(q)$. Therefore, the angle $\angle(\boldsymbol{b}_{i,0}, \boldsymbol{b}_0) = \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$. For every variable $i \in V$, we define a rotation $\boldsymbol{U_i}$ which maps the vector $\boldsymbol{b}_{i,0}$ to $\boldsymbol{b}_0$ and acts as the identity on the space orthogonal to the plane $\mathrm{span}\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_0\}$. We claim that the vector $\boldsymbol{u}_{i,a} := \alpha_i \boldsymbol{U_i} \boldsymbol{v}'_{i,a}$ satisfy the conditions of the lemma. By construction, the vectors satisfy the constraints (4.19) and (4.20). Since $\boldsymbol{U_i}$ is a rotation by an angle of at most $\sqrt{\varepsilon} \cdot \mathrm{poly}(q)$, we have $\|\boldsymbol{U_i} - \boldsymbol{I}\| \leqslant \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$ and therefore $\|\boldsymbol{u}_{i,a} - \alpha_i \boldsymbol{v}'_{i,a}\| \leqslant \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$. Previous observations imply that $\|\alpha_i \boldsymbol{v}'_{i,a} - \boldsymbol{b}_{i,a}\| \leqslant \varepsilon \cdot \mathrm{poly}(q)$. Thus, the vectors $\{\boldsymbol{u}_{i,a}\}$ satisfy also the third condition (4.21). ■

The existence of a local distribution μ_P imposes constraints on the vectors corresponding to $\mathcal{V}(P)$. Specifically, the inner products of vectors corresponding to $\mathcal{V}(P)$ must lie in a certain polytope Q_P of constant dimension, to ensure the existence of a matching local distribution μ_P. The SDP solution $\{\boldsymbol{u}_{i,a}\}$ has local distributions that match up to an error of η. In other words, for every payoff P, the vectors corresponding to $\mathcal{V}(P)$ are within η distance from the corresponding polytope Q_P.

The idea of smoothing is to take a convex combination of the SDP solution $\{\boldsymbol{u}_{i,a}\}$, with the SDP solution corresponding to the uniform distribution over all assignments. By a suitable basis change, the local polytopes Q_P can be made full-dimensional, in that they are defined by a set of inequalities (no equations involved). The SDP solution corresponding to the uniform distribution over all assignments, lies at the center of each of these local polytopes Q_P. As $\{\boldsymbol{u}_{i,a}\}$ is only η away from each of these polytopes, it moves into the polytope on taking convex combination with the center. The above intuition is formalized in the following lemma.

4.8.2 Smoothing

Lemma 4.2.2 (Smoothing)**.** *The local distributions $\{\mu_P\}$ can be transformed to distributions $\{\mu'_P\}$ such that for all $P \in \mathrm{supp}(\mathcal{P})$, $i \neq j \in \mathcal{V}(P)$, and $a, b \in [q]$,*

$$\Pr_{x\sim\mu'_P}\Big\{x_i = a,\ x_j = b\Big\} = (1-\delta)\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b}\rangle + \delta \cdot \tfrac{1}{q^2}\,, \tag{4.22}$$

where $\delta = q^4k^2\eta$. Furthermore, for every $P \in \mathrm{supp}(\mathcal{P})$,

$$\|\mu_P - \mu'_P\|_1 \leqslant 3\delta$$

Proof. Let us fix a payoff function $P \in \mathrm{supp}(\mathcal{P})$. Let $S = \mathcal{V}(P)$. We may assume that $S = \{1, \ldots, k\}$. We can think of μ_P as a function $\mathcal{F}\colon [q]^k \to \mathbb{R}$ such that $\mathcal{F}(x)$ is the probability of the assignment x under the distribution μ_P. For the case $q = 2$, the constraint (4.22) translates to a condition on the degree-2 Fourier coefficients of $\mathcal{F}$. For larger q, we will use a suitable generalization of the Fourier bases. We refer the reader to Section 3.4 for an introduction to multilinear expansion of functions.

Let $\chi_1, \ldots, \chi_q$ be an orthonormal basis of the vector space $\{\mathcal{F}\colon [q] \to \mathbb{R}\}$ such that $\chi_1 \equiv 1$. (Here, orthonormal means $\mathbb{E}_{a\in[q]}\chi_i(a)\chi_j(a) = \delta_{ij}$ for all $i, j \in [q]$). By tensoring this basis, we obtain the orthonormal basis $\{\chi_\sigma \mid \sigma \in [q]^k\}$ of the vector space $\{\mathcal{F}\colon [q]^k \to \mathbb{R}\}$. For $\sigma \in [q]^k$, we have $\chi_\sigma(x) = \chi_{\sigma_1}(x_1)\cdots\chi_{\sigma_k}(x_k)$. For a function $\mathcal{F}\colon [q]^k \to \mathbb{R}$, we denote by $\hat{\mathcal{F}}(\sigma)$ the χ_σ-coefficient of $\mathcal{F}$, i.e., $\hat{f}(\sigma) := \sum_{x\in[q]^k}\mathcal{F}(x)\chi_\sigma(x)$. Note that $\mathcal{F} = \mathbb{E}_{\sigma\in[q]^k}\hat{\mathcal{F}}(\sigma)\chi_\sigma$. Therefore, if we let $\mathcal{F}$ again be the function corresponding to μ_P, then for all $i \neq j \in S$ and $a, b \in [q]$ we have

$$\Pr_{x\sim\mu_P}\Big\{x_i = a,\ x_j = b\Big\} = \sum_{\substack{x\in[q]^k\\ x_i=a, x_j=b}} \mathop{\mathbb{E}}_{\sigma\in[q]^k}\hat{\mathcal{F}}(\sigma)\chi_\sigma(x) \tag{4.23}$$

$$= \mathop{\mathbb{E}}_{\sigma\in[q]^2}\hat{\mathcal{F}_{ij}}(\sigma)\chi_\sigma(a,b) \tag{4.24}$$

where $\hat{\mathcal{F}}_{ij}(s,t)$ is defined as the coefficient $\hat{\mathcal{F}}(\sigma)$ for $\sigma_i = s$, $\sigma_j = t$ and $\sigma_r = 1$ for all $r \in [q] \setminus \{i, j\}$. In the second equality we used that for every σ with $\sigma_r \neq 1$ for some $r \in [q] \setminus \{i, j\}$, the sum over the values of χ_σ in (4.23) vanishes.

For every variable pair $i \neq j \in S$, let $\mathcal{G}_{ij}\colon [q]^2 \to \mathbb{R}$ be the function $\mathcal{G}_{ij}(a,b) = \langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b}\rangle$. Similarly, we let $\mathcal{G}_i\colon [q] \to \mathbb{R}$ be the function $\mathcal{G}_i(a) = \langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,a}\rangle = \langle \boldsymbol{u}_{i,a}, \boldsymbol{b}_0\rangle$. We define a function $\mathcal{F}'\colon [q]^k \to \mathbb{R}$ as follows

$$\hat{\mathcal{F}}'(\sigma) = \begin{cases} \hat{\mathcal{G}}_i(s) & \text{if } \sigma_i = s \text{ and } \sigma_r = 1 \text{ for all } r \in [q]\setminus\{i\},\\ \hat{\mathcal{G}_{ij}}(s,t) & \text{if } \sigma_i = s, \sigma_j = t \text{ and } \sigma_r = 1 \text{ for } r \in [q]\setminus\{i,j\},\\ \hat{\mathcal{F}}(\sigma) & \text{otherwise.}\end{cases}$$

The conditions (4.19) and (4.20) imply that $\hat{\mathcal{G}}_i(s) = \hat{\mathcal{G}_{ij}}(s,1)$ for all $i \neq j \in S$. We also have $\hat{\mathcal{F}}(\mathbb{1}) = \hat{\mathcal{G}}_i(\mathbb{1}) = \hat{\mathcal{G}_{ij}}(\mathbb{1}) = 1$. Therefore, the identity in (4.23)–(4.24) applied to $\mathcal{F}'$ shows

that for all $i, j \in S$ and $a, b \in [q]$,

$$\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b} \rangle = \sum_{\substack{x \in [q]^k \\ x_i = a, x_j = b}} \mathop{\mathbb{E}}_{\sigma \in [q]^k} \hat{\mathcal{F}}'(\sigma) \chi_\sigma(x) = \sum_{\substack{x \in [q]^k \\ x_i = a, x_j = b}} \mathcal{F}'(x)\,. \tag{4.25}$$

We could finish the proof at this point if the function $\mathcal{F}'$ corresponded to a distribution μ'_P over assignments $[q]^k$. The function $\mathcal{F}'$ satisfies $\sum_{x \in [q]^k} \mathcal{F}'(x) = \hat{\mathcal{F}}(\mathbb{1}) = 1$ However, in general, the function $\mathcal{F}'$ might take negative values. We will show that these values cannot be too negative and that the function can be made into a proper distribution by smoothing.

Let K be an upper bound on the values of the functions $\chi_1, \ldots, \chi_q$. From the orthonormality of the functions, it follows that $K \leqslant \sqrt{q}$. Let $\mathcal{F}_{ij}(a, b) = \Pr_{x \sim \mu_P}\{x_i = a,\ x_j = b\}$. Recall that we computed in (4.24) the coefficients of $\mathcal{F}_{ij}$ in the basis $\{\chi_{s,t} \mid s, t \in [q]\}$. Since the SDP solution $\{\boldsymbol{u}_{i,a}\}$, $\{\mu_P\}$ is η-infeasible, we have

$$\hat{\mathcal{G}_{ij}}(s,t) = \sum_{a,b \in [q]} \mathcal{G}_{ij}(a,b)\chi_{st}(a,b) = \sum_{a,b \in [q]} \mathcal{F}_{ij}(a,b)\chi_{st}(a,b) \pm K^2 q^2 \eta = \hat{\mathcal{F}_{ij}}(s,t) \pm K^2 q^2 \eta.$$

Therefore, $|\hat{\mathcal{F}}(\sigma) - \hat{\mathcal{F}}'(\sigma)| \leqslant K^2 q^2 \eta$ for all $\sigma \in [q]^k$. Thus,

$$\mathcal{F}'(x) \;=\; \mathop{\mathbb{E}}_{\sigma \in [q]^k} \hat{\mathcal{F}}'(\sigma)\chi_\sigma(x) \;=\; \mathop{\mathbb{E}}_{\sigma \in [q]^k} \hat{\mathcal{F}}(\sigma)\chi_\sigma(x) \;\pm\; \delta/q^k \;=\; \mathcal{F}(x) \;\pm\; \delta/q^k\,, \tag{4.26}$$

where $\delta := K^4 k^2 q^2 \eta$. Hence, if we let $\mathcal{H} = (1 - \delta) \cdot \mathcal{F}' + \delta \cdot U$, where $U \colon [q]^k \to \mathbb{R}$ is the uniform distribution $U \equiv 1/q^k$, then

$$\mathcal{H} = (1 - \delta)\mathcal{F}' + \delta/q^k \geqslant (1 - \delta)\mathcal{F} \geqslant 0.$$

It follows that $\mathcal{H}$ corresponds to a distribution μ'_P over assignments $[q]^k$. Furthermore, from (4.25) it follows that for all $i \neq j \in S$ and $a, b \in [q]$,

$$\Pr_{x \sim \mu'_P}\Big\{x_i = a,\ x_j = b\Big\} = (1 - \delta)\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b} \rangle + \delta \cdot \tfrac{1}{q^2}\,.$$

Finally, let us estimate the statistical distance between the distributions μ_P and μ'_P,

$$\begin{aligned} \|\mathcal{F} - \mathcal{H}\|_1 &= \|\delta(\mathcal{F} - U) + (1 - \delta)(\mathcal{F} - \mathcal{F}')\|_1 \\ &\leqslant 2\delta + \|\mathcal{F} - \mathcal{F}'\|_1 \quad \text{(using triangle inequality)} \\ &\leqslant 3\delta \quad \text{(using (4.26))}\,. \end{aligned}$$

In this way, we can construct a suitable distribution μ'_P for every $P \in \operatorname{supp}(\mathcal{P})$, which proves the lemma. ■

4.8.3 Proof of Theorem 4.2 (Robustness of LC)

Let us consider an ε-infeasible SDP solution $\{\boldsymbol{b}_{i,a}\}$, $\{\mu_P\}$ for a Λ-CSP instance $\mathcal{P}$. Suppose that this SDP solution has value α. First, we construct vector $\{\boldsymbol{u}_{i,a}\}$ as in Lemma 4.2.1. These vectors together with the original local distributions $\{\mu_P\}$ form an η-infeasible SDP solution for $\mathcal{P}$, where $\eta = \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$. Next, we construct local distributions $\{\mu'_P\}$ as in Lemma 4.2.2. Define new vectors

$$\boldsymbol{w}_{i,a} \stackrel{\mathrm{def}}{=} \sqrt{1-\delta} \cdot \boldsymbol{u}_{i,a} \oplus \sqrt{\delta} \cdot \boldsymbol{u}'_{i,a} \,,$$

where $\oplus$ denotes the direct sum of vectors and $\{\boldsymbol{u}'_{i,a}\}$ are vectors corresponding to the uniform average over all feasible SDP solutions (which satisfy $\langle \boldsymbol{u}'_{i,a}, \boldsymbol{u}'_{jb} \rangle = 1/q^2$ for all $i \neq j \in V$ and all $a, b \in [q]$). From Lemma 4.2.1 and Lemma 4.2.2 it follows that $\{\boldsymbol{w}_{i,a}\}$, $\{\mu'_P\}$ is a feasible SDP solution for $\mathcal{P}$. It remains to estimate the value of this feasible SDP solution:

$$\begin{aligned}
\mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu'_P} P(x) &= \alpha - \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \sum_{x\in[q]^{\mathcal{V}(P)}} P(x) \left(\mu(x) - \mu'(x)\right) \\
&\geqslant \alpha - \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \|\mu - \mu'\|_1 \\
&\geqslant \alpha - \eta \cdot \mathrm{poly}(kq) \,.
\end{aligned}$$

For the first inequality, we used that $|P(x)| \leqslant 1$. The second inequality follows from Lemma 4.2.2. (In the last calculation, we just verified that the value of SDP solutions is Lipschitz in the statistical distance of the local distributions.) ■

4.9 Robustness of LH_r and SA_r relaxations

In this section, we will show that the LH_r and SA_r have the following robustness property. Notice that the closeness of the completely feasible solution is expressed in terms of the L_1 distance between the local distributions. Recall that the objective value in these relaxations is expressed as an expectation over the local distributions. Therefore, a bound on the L_1 distance between local distributions also corresponds to a bound on the change in the SDP value.

Theorem 4.3. *Given an ε-infeasible solution $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in\mathbb{F}_q}$, $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ to the LH_r relaxation, there exists a feasible solution $\{\boldsymbol{b}'_{i,a}\}, \{\mu'_S\}_{S\subset\mathcal{V},|S|\leqslant r}$ for LH_r such that for all subsets $S \subseteq \mathcal{V}, |S| \leqslant r$, $\|\mu_S - \mu'_S\|_1 \leqslant \mathrm{poly}(q) \cdot r^2\varepsilon$.*

Theorem 4.4. *Given an ε-infeasible solution $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in\mathbb{F}_q}$, $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ to the SA_r relaxation, there exists a feasible solution $\{\boldsymbol{b}'_{i,a}\}, \{\mu'_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ for SA_r such that for all subsets $S \subseteq \mathcal{V}, |S| \leqslant r$, $\|\mu_S - \mu'_S\|_1 \leqslant \mathrm{poly}(q) \cdot \varepsilon \cdot q^r$.*

The proof follows along the lines of the corresponding proof for LC relaxation presented in the previous section.

Let $\{\chi_1, \ldots, \chi_q\}$ be an orthonormal basis for the vector space $\{\mathcal{F}\colon [q] \to \mathbb{R}\}$ such that $\chi_1(a) = 1$ for all $a \in [q]$. (Here, orthonormal means $\mathbb{E}_{a\in[q]} \chi_i(a)\chi_j(a) = \delta_{ij}$ for all $i, j \in [q]$.)

For $r \in \mathbb{N}$, let $\{\chi_\sigma \mid \sigma \in [q]^r\}$ be the orthonormal basis of the vector space $\{\mathcal{F}\colon [q]^r \to \mathbb{R}\}$ defined by

$$\chi_\sigma(x) \stackrel{\text{def}}{=} \chi_{\sigma_1}(x_1) \cdot \cdots \cdot \chi_{\sigma_r}(x_r)\,, \tag{4.27}$$

where $\sigma = (\sigma_1, \ldots, \sigma_r) \in [q]^r$ and $x = (x_1, \ldots, x_r) \in [q]^q$. Again, a function $\mathcal{F}\colon [q]^r \to \mathbb{R}$ can be written as the multilinear polynomial,

$$\hat{\mathcal{F}}(\sigma) \stackrel{\text{def}}{=} \sum_{x \in [q]^r} \mathcal{F}(x)\chi_\sigma(x)\,. \tag{4.28}$$

Using the fact $\mathbb{E}_{\sigma \in [q]^r} \chi_\sigma(x)\chi_\sigma(y) = \delta_{xy}$ for all $x, y \in [q]^r$, we see that

$$\mathcal{F} = \mathop{\mathbb{E}}_{\sigma \in [q]^r} \hat{\mathcal{F}}(\sigma)\chi_\sigma\,.$$

We define the following norm for functions $\hat{\mathcal{F}}\colon [q]^r \to \mathbb{R}$,

$$\|\hat{\mathcal{F}}\|_1 \stackrel{\text{def}}{=} \sum_{\sigma \in [q]^r} |\hat{\mathcal{F}}(\sigma)|\,.$$

We say $\mathcal{F}\colon [q]^r \to \mathbb{R}$ is a *distribution* if $\mathcal{F}(x) \geqslant 0$ for all $x \in [q]^r$ and $\sum_{x \in [q]^r} \mathcal{F}(x) = 1$. We define

$$K \stackrel{\text{def}}{=} \max_{\sigma \in [q]^r, x \in [q]^r} |\chi_\sigma(x)|\,.$$

In the next lemma, we give a proof of the following intuitive fact: If a function $\mathcal{G}\colon [q]^q \to \mathbb{R}$ satisfies the normalization constraint $\sum_{x \in [q]^q} \mathcal{G}(x) = 1$ and it is close to a distribution in the sense that there exists a distribution $\mathcal{F}$ such that $\|\hat{\mathcal{F}} - \hat{\mathcal{G}}\|$ is small, then $\mathcal{G}$ can be made to a distribution by "smoothing" it. Here, smoothing means to move slightly towards the uniform distribution (where every assignment has probability q^{-r}).

Lemma 4.4.1. *Let $\mathcal{F}, \mathcal{G}\colon [q]^r \to \mathbb{R}$ be two functions with $\hat{\mathcal{F}}(\mathbb{1}) = \hat{\mathcal{G}}(\mathbb{1}) = 1$. Suppose $\mathcal{F}$ is a distribution. Then, the following function is also a distribution*

$$(1 - \varepsilon)\mathcal{G} + \varepsilon q^{-r} \qquad \textit{where } \varepsilon = \|\hat{\mathcal{F}} - \hat{\mathcal{G}}\|_1 \cdot K\,.$$

Proof. It is clear that the function $\mathcal{H} = (1 - \varepsilon)\mathcal{G} + \varepsilon q^{-r}$ satisfies the constraint $\hat{\mathcal{H}}(\mathbb{1}) = 1$.

For every $x \in [q]^r$, we have

$$\begin{aligned}
\mathcal{H}(x) &= (1-\varepsilon)\mathcal{G}(x) + \varepsilon q^{-r} \\
&\geqslant (1-\varepsilon)\left(\mathcal{G}(x) - \mathcal{F}(x)\right) + \varepsilon q^{-r} \qquad \text{(using } \mathcal{F}(x) \geqslant 0\text{)} \\
&= \varepsilon q^{-r} + (1-\varepsilon) \mathop{\mathbb{E}}_{\sigma \in [q]^r} \left(\hat{\mathcal{G}}(\sigma) - \hat{\mathcal{F}}(\sigma)\right) \chi_\sigma(x) \\
&\geqslant \varepsilon q^{-r} - (1-\varepsilon) \mathop{\mathbb{E}}_{\sigma \in [q]^r} \left|\hat{\mathcal{G}}(\sigma) - \hat{\mathcal{F}}(\sigma)\right| \cdot K \\
&= \varepsilon q^{-r} - (1-\varepsilon) K \|\hat{\mathcal{F}} - \hat{\mathcal{G}}\|_1 \cdot q^{-r} \\
&\geqslant 0\,. \qquad \text{(by our choice of } \varepsilon\text{)}
\end{aligned}$$

■

Let V be a set. For a function $\mathcal{F}\colon [q]^V \to \mathbb{R}$ and a subset $S \subseteq V$, we define the function $\mathsf{margin}_S \mathcal{F}\colon [q]^S \to \mathbb{R}$ as

$$\mathsf{margin}_S \mathcal{F}(x) \stackrel{\text{def}}{=} \sum_{y \in [q]^{V \setminus S}} \mathcal{F}(x, y)\,.$$

Note that if $\mathcal{F}$ is a distribution over $[q]$-assignments to V then $\mathsf{margin}_S \mathcal{F}$ is its marginal distribution over $[q]$-assignments to T.

Lemma 4.4.2. *For every $\mathcal{F}\colon [q]^V \to \mathbb{R}$ and $S \subseteq V$,*

$$\mathsf{margin}_S \mathcal{F} = \mathop{\mathbb{E}}_{\sigma \in [q]^S} \hat{\mathcal{F}}(\sigma, \mathbb{1}) \chi_\sigma\,.$$

Here, $\sigma, \mathbb{1}$ denotes the $[q]$-assignment to V that agrees with σ on S and assigns 1 to all variables in $V \setminus S$.

Proof.

$$\begin{aligned}
\mathsf{margin}_S \mathcal{F}(x) &= \sum_{y \in [q]^{V \setminus S}} \mathcal{F}(x, y) \\
&= \sum_{y \in [q]^{V \setminus S}} \mathop{\mathbb{E}}_{\sigma \in [q]^V} \hat{\mathcal{F}}(\sigma) \chi_\sigma(x, y) \\
&= \sum_{y \in [q]^{V \setminus S}} \mathop{\mathbb{E}}_{\sigma \in [q]^S} \mathop{\mathbb{E}}_{\sigma' \in [q]^{V \setminus S}} \hat{\mathcal{F}}(\sigma) \chi_\sigma(x) \chi_{\sigma'}(y) \\
&= \mathop{\mathbb{E}}_{\sigma \in [q]^S} \mathop{\mathbb{E}}_{\sigma' \in [q]^{V \setminus S}} \hat{\mathcal{F}}(\sigma, \sigma') \chi_\sigma(x) \cdot \sum_{y \in [q]^{V \setminus S}} \chi_{\sigma'}(y) \\
&= \mathop{\mathbb{E}}_{\sigma \in [q]^S} \hat{\mathcal{F}}(\sigma, \mathbb{1}) \chi_\sigma(x)\,. \quad \text{(using } \textstyle\sum_{y \in [q]^{V \setminus S}} \chi_{\sigma'}(y) = 0 \text{ for } \sigma' \neq \mathbb{1}\text{.)}
\end{aligned}$$

■

The margin operator has the following useful property (which is clear from its definition).

Lemma 4.4.3. *For every function $\mathcal{F}\colon [q]^V \to \mathbb{R}$ and any sets $T \subseteq S \subseteq V$,*

$$\mathsf{margin}_T \, \mathsf{margin}_S \, \mathcal{F} = \mathsf{margin}_T \, \mathcal{F} \, .$$

Lemma 4.4.4. *Let V be a set and let $\{\mu_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ be a collection of distributions. Suppose that for all sets $A, B \subseteq V$ with $|A|, |B| \leqslant r$,*

$$\|\mathsf{margin}_{A\cap B} \, \mu_A - \mathsf{margin}_{A\cap B} \, \mu_B\|_1 \leqslant \eta \, .$$

Then, there exists a collection of distributions $\{\mu'_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ such that

- *for all $A, B \subseteq V$ with $|A|, |B| \leqslant r$,*

$$\mathsf{margin}_{A\cap B} \, \mu'_A = \mathsf{margin}_{A\cap B} \, \mu'_B \, .$$

- *for all $S \subseteq V$ with $|S| \leqslant r$,*

$$\|\mu'_S - \mu_S\|_1 \leqslant O(\eta q^r K^2) \, ,$$

The previous lemma is not enough to establish the robustness of our SDP relaxations. The issue is that we not only require that the distributions are consistent among themselves but they should also also be consistent with the SDP vectors.

The following lemma allows us to deal with this issue.

Lemma 4.4.5 (SA_r Smoothing)**.** *Let V be a set and let $\{\mu_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ be a collection of distributions. Suppose that*

- *for all sets $A, B \subseteq V$ with $|A|, |B| \leqslant r$,*

$$\|\mathsf{margin}_{A\cap B} \, \mu_A - \mathsf{margin}_{A\cap B} \, \mu_B\|_1 \leqslant \eta \, .$$

- *for all sets $A, B \subseteq V$ with $|A|, |B| \leqslant 2$,*

$$\mathsf{margin}_{A\cap B} \, \mu_A = \mathsf{margin}_{A\cap B} \, \mu_B \, .$$

Then, for $\varepsilon \geqslant q^r K^2 \eta$, there exists a collection of distributions $\{\mu'_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ such that

- *for all $A, B \subseteq V$ with $|A|, |B| \leqslant r$,*

$$\mathsf{margin}_{A\cap B} \, \mu'_A = \mathsf{margin}_{A\cap B} \, \mu'_B \, . \tag{4.29}$$

- *for all $S \subseteq V$ with $|S| \leqslant r$,*

$$\|\mu'_S - \mu_S\|_1 \leqslant O(K^2 \eta q^r) \, , \tag{4.30}$$

– *for all* $S \subseteq V$ *with* $|S| \leqslant 2$,

$$\mu'_S = (1-\varepsilon)\mu_S + \varepsilon \cdot q^{-|S|}\,. \tag{4.31}$$

Proof. For $\sigma \in [q]^V$, let $\mathrm{supp}(\sigma)$ denote the set of coordinates of σ not equal to 1, and let $|\sigma|$ denote the number of such coordinates,

$$\mathrm{supp}(\sigma) \stackrel{\text{def}}{=} \{i \in V \mid \sigma_i \neq 1\} \quad \text{and} \quad |\sigma| \stackrel{\text{def}}{=} |\mathrm{supp}(\sigma)|\,.$$

For every $\sigma \in [q]^V$ with $|\sigma| \leqslant r$, we define

$$\hat{\mathcal{F}}(\sigma) := \mathop{\mathbb{E}}_{x\sim\mu_S} \chi_\sigma(x) \qquad \text{where } S = \mathrm{supp}(\sigma)\,.$$

For every σ with $|\sigma| > r$, we set $\hat{\mathcal{F}}(\sigma) := 0$. We define μ'_S in terms of $\mathcal{F} = \mathbb{E}_\sigma \hat{\mathcal{F}}(\sigma)\chi_\sigma$,

$$\mu'_S := \mathsf{margin}_S\,(1-\varepsilon)\mathcal{F} + \varepsilon q^{-|V|}\,.$$

By Lemma 4.4.3, this choice of μ'_S satisfies condition (4.29).

First, let us argue that the functions μ'_S are distributions. Let $S \subseteq V$ with $|S| \leqslant r$. For $\sigma \in [q]^S$ with $T := \mathrm{supp}(\sigma) \subseteq S$, we have

$$\begin{aligned}
|\hat{\mathcal{F}}(\sigma,\mathbb{1}) - \mathop{\mathbb{E}}_{x\sim\mu_S}\chi_\sigma(x)| &= |\mathop{\mathbb{E}}_{x\sim\mu_T}\chi_\sigma(x) - \mathop{\mathbb{E}}_{x\sim\mu_S}\chi_\sigma(x)| \\
&\leqslant \|\mu_T - \mathsf{margin}_T\,\mu_S\|_1 \cdot \max|\chi_\sigma| \\
&\leqslant \eta \cdot K\,.
\end{aligned} \tag{4.32}$$

Let $\mathcal{F}_S$ denote the function $\mathsf{margin}_S\,\mathcal{F}$. By Lemma 4.4.2, $\hat{\mathcal{F}}_S(\sigma) = \hat{\mathcal{F}}(\sigma,\mathbb{1})$ for all $\sigma \in [q]^S$. Hence, $\|\hat{\mathcal{F}} - \hat{\mu}_S\|_1 \leqslant q^r \cdot K\eta$. It follows that for $\varepsilon \geqslant q^r K^2 \eta$, the function $\mu'_S = (1-\varepsilon)\mathcal{F}_S + \varepsilon q^{-|S|}$ is a distribution (using Lemma 4.4.1).

Next, let us verify that (4.30) holds. We have

$$\begin{aligned}
\|\mu'_S - \mu_S\|_1 &\leqslant O(\varepsilon) + \|\mathsf{margin}_S\,\mathcal{F} - \mu_S\|_1 \\
&\stackrel{\text{La. 4.4.2}}{=} O(\varepsilon) + \left\| \mathop{\mathbb{E}}_{\sigma\in[q]^S} \left(\hat{\mathcal{F}}(\sigma,\mathbb{1}) - \mathop{\mathbb{E}}_{x\sim\mu_S}\chi_\sigma(x) \right) \chi_\sigma \right\|_1 \\
&\stackrel{(4.32)}{\leqslant} O\left(\eta K^2 \cdot q^r\right) \quad (\text{using } |\hat{\mathcal{F}}(\sigma,\mathbb{1}) - \hat{\mu}_S(\sigma)| \leqslant \eta K \text{ and } |\chi_\sigma(x)| \leqslant K)\,.
\end{aligned}$$

Finally, we show that the new distributions satisfy (4.31). Let $S \subseteq V$ be a set of size at most 2. It follows from the consistency assumption that for all $\sigma \in [q]^S$, we have $\hat{\mathcal{F}}(\sigma,\mathbb{1}) = \hat{\mu}_S(\sigma)$. Hence, $\mathcal{F}_S = \mu_S$, which implies (4.31).

■

Lemma 4.4.6 (LH_r Smoothing)**.** *Let* V *be a set and let* $\{\mu_S \colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ *be a collection of distributions. Suppose that*

– *for all sets* $A, B \subseteq V$ *with* $|A|, |B| \leqslant r$,

$$\|\mathsf{margin}_{A\cap B}\,\mu_A - \mathsf{margin}_{A\cap B}\,\mu_B\|_1 \leqslant \eta\,.$$

– *for all sets* $A, B \subseteq V$ *with* $|A|, |B| \leqslant 2$,

$$\mathsf{margin}_{A\cap B}\,\mu_A = \mathsf{margin}_{A\cap B}\,\mu_B\,.$$

Then, for $\varepsilon \geqslant qR^2K^2\eta$, *there exists a collection of distributions* $\{\mu'_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ *such that*

– *for all* $A, B \subseteq V$ *with* $|A|, |B| \leqslant r$ *with* $|A \cap B| \leqslant 2$,

$$\mathsf{margin}_{A\cap B}\,\mu'_A = \mathsf{margin}_{A\cap B}\,\mu'_B\,. \tag{4.33}$$

– *for all* $S \subseteq V$ *with* $|S| \leqslant r$,

$$\|\mu'_S - \mu_S\|_1 \leqslant O(K^2\eta q r^2)\,, \tag{4.34}$$

– *for all* $S \subseteq V$ *with* $|S| \leqslant 2$,

$$\mu'_S = (1-\varepsilon)\mu_S + \varepsilon \cdot q^{-|S|}\,. \tag{4.35}$$

Proof. The proof is along the lines of the proof of the previous lemma.

Define $\hat{\mathcal{F}}\colon [q]^r \to \mathbb{R}$ as before. We define new functions $\{\mu^*_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ such that

$$\hat{\mu}^*_S(\sigma) = \begin{cases} \hat{\mu}_S(\sigma) & \text{if } \mathrm{supp}(\sigma) > 2, \\ \hat{\mathcal{F}}(\sigma, \mathbb{1}) & \text{if } 1 \leqslant \mathrm{supp}(\sigma) \leqslant 2, \\ 1 & \text{otherwise.} \end{cases}$$

Since $|\hat{\mathcal{F}}(\sigma, \mathbb{1}) - \hat{\mu}_S(\sigma)| \leqslant K\eta$ (see proof of previous lemma), we can upper bound $\|\hat{\mu}^*_S - \hat{\mu}_S\|_1 \leqslant qr^2 \cdot K\eta$ (there are not more than qR^2 different $\sigma \in [q]^S$ with $\hat{\mathcal{F}}(\sigma, \mathbb{1}) \neq \hat{\mu}_S(\sigma)$. By Lemma 4.4.1, for $\varepsilon \geqslant qR^2K^2\eta$, the functions $\{\mu'_S\colon [q]^S \to \mathbb{R} \mid S \subseteq V, |S| \leqslant r\}$ defined by $\mu'_S := (1-\varepsilon)\mu^*_S + \varepsilon q^{-|S|}$ are the desired distributions. We can check that the assertions of the lemma are satisfied in the same way as for the proof of the previous lemma. ■

Proofs of Theorem 12.7 and Theorem 12.8 (Sketch). Without loss of generality, we may assume that the original vector assignment $\{\boldsymbol{b}_{i,a}\}$ is perfectly consistent with the marginals of up to order 2. This is because, given an arbitrary vector assignment we can apply Lemma 4.2.1 to obtain SDP vectors such that for every pair of vectors there exists a local distribution agreeing with inner products. Now, change the marginals on sets of size 2 to those determined by these inner products. We apply Lemma 4.4.6 or Lemma 4.4.5 to the local distributions $\{\mu_S\}$ of the ε-infeasible LH_r or SA_r solution, respectively. We get a new set of local distributions $\{\mu'_S\}$ that have the desired consistency properties. It remains to

change the vectors so that their inner product match the corresponding probabilities in the local distributions. Let $\{\boldsymbol{u}_{i,a}\}$ be the vector assignment that corresponds to the uniform distribution over all possible assignments to the variables (this vector assignment is the geometric center of the set of all vector assignments). Then, we define the new vector assignment $\{\boldsymbol{v}_{i,a}\}$ as

$$\boldsymbol{v}_{i,a} = \sqrt{1-\delta} \cdot \boldsymbol{b}_{i,a} \oplus \sqrt{\delta}\boldsymbol{u}_{i,a}\,,$$

where δ is the smoothing parameter in Lemma 4.4.6 or Lemma 4.4.5. It is easy to verify that $\{\boldsymbol{v}_{i,a}\}$ together with $\{\mu'_S\}$ form a feasible LH_r or SA_r solution. ■

Chapter 5

A GENERIC ROUNDING SCHEME

5.1 Introduction

Despite all the successes in designing approximation algorithms using SDP relaxations, rounding the solution to a semidefinite program remains a difficult task. Contrast this to linear programming which has seen the development of primal-dual [158] and iterative rounding techniques [88, 111], leading to simple combinatorial algorithms. Part of the problem is that the approximation ratios involved in SDP based algorithms are irrational numbers stemming from the geometry of vectors. Even for problems like Max 3-Sat where the optimal approximation ratio is a simple fraction like $\frac{7}{8}$ [93], the analysis of the rounding procedure is fairly involved.

In the previous chapter, we introduced the LC relaxation for the class of generalized constraint satisfaction problems (GCSP). The class of GCSPs is the natural generalization of CSPs obtained by replacing predicates with bounded real valued payoff functions (Definition 2.4.1). Here, we present a generic rounding scheme that *achieves the integrality gap* of the LC relaxation for every GCSP.

In Chapter 7, we will see that the LC relaxation yields the optimal approximation ratio for every CSP, under the Unique Games Conjecture (UGC). Thus the rounding scheme along with the LC relaxation yields a generic algorithm for CSPs that achieves the optimal approximation ratio under UGC.

Furthermore, as seen in Chapter 4, the LC relaxation is stronger than or equivalent to nearly every relaxation used in the literature to approximate CSPs. Hence, irrespective of the truth of the UGC, the generic rounding scheme presented here, for every CSP, yields an approximation at least as good as the best known algorithm [65, 32, 114, 93, 79, 35, 36, 62, 78, 122, 162, 66, 54, 156, 161, 164] in literature.

This thesis also demonstrates that the integrality gap of the LC relaxation cannot be reduced by adding large classes of valid inequalities in the fashion of Sherali–Adams LP hierarchies (See Chapter 12). As a consequence, the rounding scheme presented here achieves the integrality gap of these stronger relaxations too, for every GCSP.

Even for seemingly simple CSPs like Max-2SAT or Max-3SAT, the analyses of the best known algorithms are quite involved and require computer-aided calculations. Our results do not obviate these calculations (indeed the calculations required to determine the approximation ratio of our algorithm for concrete CSPs appear to be practically infeasible). However, in contrast to all previous works on approximating CSPs, our analysis provides a simple explanation why the approximation guarantee of the algorithm approaches the integrality gap of the relaxation: If the rounding algorithm has an approximation ratio α on an instance $\mathfrak{I}$, then the SDP has a gap of $\approx \alpha$ on a related instance $\mathfrak{I}'$ which is obtained from $\mathfrak{I}$ by identifying variables. In other words, an instance is hard to round only if it is a "blow-up" of an integrality gap instance.

The rounding scheme presented in this chapter can be succinctly summarized as follows: Reduce the dimension of the SDP solution by randomly projecting it into a constant-dimensional space, identify all variables whose projected vectors are close to each other, and solve the resulting instance by brute force! The analysis is elementary in that it avoids the use of typical machinery from Unique Games reductions such as dictatorship tests, Fourier analysis or the Invariance principle.

A common theme of this chapter and Chapter 12 is a *robustness lemma* for SDP relaxations which asserts that approximately feasible solutions can be made feasible by "smoothing" without changing the objective value significantly.

5.2 Result

To state the result of this chapter precisely, we need to define the SDP integrality gap curve $\mathsf{Gap}_\Lambda(c)$ for a CSP Λ. Let $\mathrm{sdp}(\Im)$ denote the objective value of an optimal solution for the LC relaxation of an instance $\Im$. Throughout this chapter, when we refer to "the" SDP or "the SDP value" we will refer to the LC relaxation.

Let $\mathrm{opt}(\Im)$ denote the value of the optimal solution to $\Im$. The integrality gap curve $\mathsf{Gap}_\Lambda(c)$ is the minimum value of $\mathrm{opt}(\Im)$, given that $\mathrm{sdp}(\Im) = c$ where the minimum is over all instances $\Im$ of the problem Λ. Formally,

$$\mathsf{Gap}_\Lambda(c) = \inf_{\Im \in \Lambda, \mathrm{sdp}(\Im)=c} \mathrm{opt}(\Im)$$

Theorem 5.1. *For every GCSP Λ and for every $\eta > 0$, there exists a polynomial time approximation algorithm for Λ that returns an assignment of value at least $\mathsf{Gap}_\Lambda(c-\eta) - \eta$ on an instance $\Im$ with SDP value c. The algorithm runs in time $\exp(\exp(\mathrm{poly}(^{kq}/_{\eta})))$.*

The above result also holds in the more general setting where predicates are replaced by bounded real-valued payoff functions. For a traditional CSP Λ consisting of predicates, the above theorem implies the following corollary.

Corollary 5.1.1. *Given a CSP Λ with positive valued payoff functions, for every $\eta > 0$, there exists a polynomial time approximation algorithm for Λ with approximation ratio at most the integrality gap ratio $\mathsf{GapRatio}_\Lambda$ defined as,*

$$\mathsf{GapRatio}_\Lambda \stackrel{\mathrm{def}}{=} \sup_{\Im \in \Lambda} \frac{\mathrm{sdp}(\Im)}{\mathrm{opt}(\Im)}.$$

The algorithm runs in time $\exp(\exp(\mathrm{poly}(^{kq}/_{\eta})))$.

On the downside, the proof of optimality of the rounding scheme is non-explicit. To show the optimality of the rounding scheme, we proceed as follows: given an instance $\Im$ on which the rounding scheme only achieves an α approximation, we exhibit an instance on which the integrality gap of the SDP is at least α. In particular, this yields no information on the approximation ratio α achieved by the rounding scheme. To address this issue, we also present an algorithm to compute the integrality gap of LC for any given CSP Λ.

Theorem 5.2. *For every constant $\eta > 0$ and every CSP Λ, the integrality gap curve $\mathsf{Gap}_\Lambda(c)$ can be computed to an additive approximation of η in time $\exp(\exp(\mathrm{poly}(^{kq}/_{\eta})))$.*

5.3 Proof Overview

In this section, we elucidate how these are employed to obtain rounding schemes for CSPs. We begin by describing the generic SDP relaxation LC for a well known CSP - Max3SAT.

Fix a Max3SAT instance $\Im$ consisting of variables $\mathcal{V} = \{y_1, \ldots, y_n\}$ and clauses $\mathcal{P} = \{P_1, \ldots, P_m\}$. The variables in LC are as follows:

- For each variable y_i, introduce two vector variables $\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}\}$. In the intended solution, the assignment $y_i = 1$ is represented by $\boldsymbol{b}_{i,0} = 0$ and $\boldsymbol{b}_{i,1} = 1$, while $y_i = 0$ implies $\boldsymbol{b}_{i,0} = 1, \boldsymbol{b}_{i,1} = 0$.
- For each clause we will introduce 8 variables to denote the 8 different states possible. For instance, with the clause $P = (y_1 \vee y_2 \vee y_3)$ we shall associate 8 variables $\mu_P = \{\mu_{(P,000)}, \mu_{(P,001)}, \ldots, \mu_{(P,111)}\}$. In general, the variables μ_P form a probability distribution locally over integral solutions.

The relaxation LC has the minimal set of constraints necessary to ensure that for every clause $P \in \mathcal{P}$, the following hold: Firstly, μ_P is a valid probability distribution over local assignments ($\{0,1\}^3$). Further, the inner products of the vectors corresponding to variables in P match the distribution μ_P. The objective value to be maximized can be written in terms of the local integral distributions μ_P as follows:

$$\sum_{P \in \mathcal{P}} \sum_{x \in \{0,1\}^3} P(x) \mu_{P,x}$$

We wish to point out that the relaxation LC is an extremely minimal SDP relaxation. For instance, if two variables y_i, y_j do not occur in a clause together, then LC does not impose any constraints on the inner products of the corresponding vectors $\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}, \boldsymbol{b}_{j,0}, \boldsymbol{b}_{j,1}\}$. Specifically, the inner product of $\boldsymbol{b}_{i,0}$ and $\boldsymbol{b}_{j,0}$ could take negative values in a feasible solution.

Given the SDP solution to the instance $\Im$, we construct a constant sized Max3SAT instance $\Im'$ which serves as a model for $\Im$. More specifically, we construct a partition $S_1 \cup S_2 \cup \ldots S_m = \mathcal{V}$ of the set of variables $\mathcal{V}$ into m subsets for some constant m. The instance $\Im'$ is over m variables $\{s_1, s_2, \ldots, s_m\}$ corresponding to subsets $S_1, \ldots, S_m$. Essentially, the instance $\Im'$ is obtained by merging all the variables in each of the sets S_i to a corresponding variable s_i. We will refer to $\Im'$ as a *folding* of the instance $\Im$.

Observe that any assignment $\mathcal{A}'$ to $\Im'$ yields a corresponding assignment $\mathcal{A}$ to $\Im$ by simple unfolding, i.e., assign $\mathcal{A}(y_j) = \mathcal{A}'(s_i)$ for every variable y_j in the set S_i. Clearly, the fraction of clauses satisfied by assignment $\mathcal{A}$ on $\Im$ is exactly the same as that satisfied by $\mathcal{A}'$ on $\Im'$. Observe that any folding operation immediately yields a rounding scheme - "Find the optimal assignment to $\Im'$ by brute force, and unfold it to an assignment for $\Im$."

To show the optimality of this scheme, the crucial property we require of the *folding* operation is that it preserves the SDP value. Clearly, any *folding* operation can only decrease the value of the optimum for the SDP relaxation, i.e, $\text{sdp}(\Im') \leqslant \text{sdp}(\Im)$. We will exhibit a folded instance $\Im'$ such that $\text{sdp}(\Im') \approx \text{sdp}(\Im)$. More precisely, we will exhibit a folded instance $\Im'$ with approximately the same clauses as $\Im$, and roughly the same SDP value. Such a folded instance $\Im'$ will serve as a certificate for the optimality of the rounding scheme. Recall that the folded instance is an integrality gap instance with a SDP value $\text{sdp}(\Im') \approx \text{sdp}(\Im)$ and optimum value $\text{opt}(\Im')$. By definition, the scheme returns an assignment of value $\text{opt}(\Im')$ on the instance $\Im$ with SDP value $\text{sdp}(\Im)$. Thus the rounding scheme achieves an approximation no worse than the integrality gap of the SDP.

At this juncture, we would like to draw a parallel between this approach and the work of Frieze-Kannan [63] on approximating dense instances of NP-hard problems. Given a dense instance of Max Cut, they construct a finite model that *approximates* the instance using the Szemeredi Regularity lemma. This finite model is nothing but a *folding* of the instance that preserves the optimum value for Max Cut. In contrast, we construct a finite model for arbitrary instances that need not be dense, while preserving an arguably simpler property - the SDP optimum.

Summarizing the discussion, the problem of rounding has been reduced to finding an algorithm to merge variables in the instance into a few clusters, while preserving the SDP value. Intuitively, the most natural way to preserve the SDP value would be to merge variables whose SDP vectors are close to each other. In other words, we would like to cluster the SDP vectors $\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}\}$ into constant number of clusters. A first attempt at such a clustering would be as follows: partition the ambient space in to bins of diameter at most η, and merge all the SDP vectors that fall into the same bin. The number of clusters created is at most the number of bins in the partition.

In general, the optimum SDP vectors $\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}\}$ lie in a space of dimension equal to the number of variables in the SDP (say n). A partition of the n-dimensional sphere into bins of diameter at most η, would require roughly $(1/\eta)^n$ bins, while our goal is to use a constant number of bins. Simply put, there is little chance that n vectors in a n-dimensional space are clustered into a few clusters. To address this issue, we pursue the most natural approach: first perform a dimension reduction on the SDP vectors by using random projections, and then cluster them together.

Heuristically, for large enough constant d, when projected in to a random d-dimensional space, at least a $1 - \eta$ fraction of the inner products will change by at most η. Further, merging variables within the same bin of diameter η, could affect the inner products by at most η. Thus the SDP value of the folded instance should be within $O(\eta)$ of the original SDP value. The number of variables in the folded instance would be $(1/\eta)^d$ - a constant.

Making the above heuristic argument precise forms the technical core of this chapter. While this is easy for some 2-CSPs like Max Cut, extending it to CSPs of larger arity and alphabet size is non-trivial. The central issue to be addressed is how to respect all the constraints of the SDP during dimension reduction. In fact, for stronger SDP relaxations such as the one in [16], it is unclear whether a dimension reduction can be carried out at all. For a subset of CSP variables involved in a constraint P, the LC relaxation requires the inner products of the corresponding SDP vectors to be consistent with a local integral distribution μ_P. This translates into the SDP vectors satisfying special constraints amongst themselves. For instance, even for a CSP of arity 3 such as Max3SAT, this implies the triangle inequalities on every 3-tuple of variables involved in a clause.

To make the argument precise, we use the smoothing operation defined in [136] which in some sense introduces noise to the SDP vectors. Interestingly, the smoothing operation was applied for an entirely different purpose in [136]. For every CSP instance, there is a canonical SDP solution $\{\boldsymbol{u}_{i,0}, \boldsymbol{u}_{i,1}\}$ corresponding to the uniform distribution over all possible integral solutions. Given an arbitrary SDP solution $\{\boldsymbol{b}_{i,0}, \boldsymbol{b}_{i,1}\}$, the smoothed solution is defined by $\boldsymbol{v}^*_{i,a} = (1-\eta)\boldsymbol{b}_{i,a} \circ \sqrt{2\eta - \eta^2}\boldsymbol{u}_{i,a}$, where $\circ$ denotes the concatenation operation. Clearly, the SDP objective value changes by at most $O(\eta)$ due to smoothing. We observe that

if the vectors $\{\boldsymbol{b}_{i,a}\}$ are close to satisfying a valid inequality (say the triangle inequality) approximately, then by smoothing, the new solution $\{\boldsymbol{v}^*_{i,a}\}$ satisfies the inequality exactly. We present a separate argument to handle the equality constraints in the SDP.

In the original instance $\mathfrak{I}$, for every clause P, the inner products of the vectors involved match a local integral distribution μ_P. After random projection and discretization, for at least $1-\eta$ fraction of the clauses in $\mathfrak{I}$, the corresponding inner products match a local integral distribution up to an error η. Let us refer to these $1-\eta$ fraction of the clauses as *good*. Apply the smoothing operation on the discretized SDP solution. For each *good* clause, the smoothed SDP solution is consistent with a local integral distribution. To finish the argument, we discard the η-fraction of the *bad* clauses from the folded instance $\mathfrak{I}'$. By the definition of LC, once a *bad* clause P is dropped from the instance, it is no longer necessary to satisfy the SDP constraints corresponding to P. Hence, we conclude $\mathrm{sdp}(\mathfrak{I}') \approx \mathrm{sdp}(\mathfrak{I})$.

Mathematical Tools: We use the method of expanding functions over product spaces as multilinear polynomials from Section 3.4

5.4 Preliminaries

We refer the reader to Chapter 2 for the formal definition of GCSPs. The rounding scheme presented in this chapter uses ε-nets for the unit ball. In this light, we present formal definition of ε-net here.

Definition 5.4.1. Let $B(\mathbf{0},1)$ denote the unit ball in the d-dimensional space $\mathbb{R}^d$. An ε-net for the unit ball $B(\mathbf{0},1)$ is a finite set N of points in $B(\mathbf{0},1)$ such that for every $\boldsymbol{x} \in B(\mathbf{0},1)$ there exists a point $y \in N$ such that $\|\boldsymbol{x}-\boldsymbol{y}\|_2 \leqslant \varepsilon$.

The following is a fairly trivial bound on the size of ε-nets.

Fact 5.4.1. *There exists an absolute constant c such that for every positive integer d and $\varepsilon > 0$, there exists an ε-net N for the unit ball $B(\mathbf{0},1)$ in $\mathbb{R}^d$ such that $|N| \leqslant (c/\varepsilon)^d$.*

5.4.1 LC Relaxation

We restate the LC relaxation for the convenience of the reader. Given an instance $\mathfrak{I} = (\mathcal{V},\mathcal{P})$, the LC relaxation consists of vectors $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in[q]}$ and a collection $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ of distributions over local assignments and a unit vector $\boldsymbol{b}_0$. Each distribution μ_P is over $[q]^{\mathcal{V}(P)}$ (the set of assignments to the variable set $\mathcal{V}(P)$).

LC Relaxation

$$\begin{aligned}
\text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \\
\text{subject to} \quad & \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_P}\left\{x_i = a, x_j = b\right\} \quad P \in \operatorname{supp}(\mathcal{P}),\ \ i,j \in \mathcal{V}(P),\ \ a,b \in [q]\,. && (5.1)\\
& \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_0\rangle = \|\boldsymbol{b}_{i,a}\|_2^2 \qquad \forall i \in \mathcal{V}, a \in [q]\,, && (5.2)\\
& \|\boldsymbol{b}_0\|_2^2 = 1 && (5.3)\\
& \mu_P \in \blacktriangle([q]^{\mathcal{V}(P)})
\end{aligned}$$

To the original definition of LC relaxation, we have included two additional constraints (5.2) and (5.3). As was shown in Observation 4.5.1 and Observation 4.5.2, any solution to the LC can be transformed to satisfy these additional constraints, without any loss in objective value. As seen in Section 4.8, these additional constraints make the LC relaxation *robust* in that, a solution that approximately satisfies all these constraints can be modified into an exact solution.

5.5 Rounding General CSPs

In the following, let $\mathcal{P}$ be a Λ-CSP instance on the variable set $\mathcal{V} = [n]$. We will also assume that we can associate in a unique way an optimal SDP solution with every Λ-CSP instance.

Variable folding For a mapping $\varphi\colon \mathcal{V} \to W$, we define a new Λ-CSP instance $\mathcal{P}/\varphi$ on the variable set W by identifying variables of $\mathcal{P}$ that get mapped to the same variable in W. Formally, the payoff functions in $\mathcal{P}/\varphi$ are of the form $P(x_{\phi(1)}, \ldots, x_{\phi(n)})$ for $x \in [q]^W$. Since any assignment for $\mathcal{P}/\phi$ corresponds to an assignment for $\mathcal{P}$, we can note the following fact.

Fact 5.5.1. $\operatorname{opt}(\mathcal{P}) \geqslant \operatorname{opt}(\mathcal{P}/\phi)$.

In general, the optimal value of the folded instance might be significantly lower than the optimal value of the original instance. However, we will show that we can always find a variable folding that approximately preserves the SDP value of an instance that is close to original instance.

Theorem 5.3. *Given $\varepsilon > 0$ and a Λ-CSP instance $\mathcal{P}$, we can efficiently compute another Λ-CSP instance $\mathcal{P}'$ and a variable folding ϕ such that*

1. *$\mathcal{P}'$ is obtained by discarding an ε fraction of payoffs from the instance $\mathcal{P}$. Formally, $V(\mathcal{P}') = V(\mathcal{P})$ and the total variational distance between distributions $\mathcal{P}, \mathcal{P}'$ is bounded by ε, i.e., $\|\mathcal{P} - \mathcal{P}'\|_{\mathrm{TV}} \leqslant \varepsilon$,*

2. $\operatorname{sdp}(\mathcal{P}'/\phi) \geqslant \operatorname{sdp}(P) - \varepsilon$,

3. *the variable set of $\mathcal{P}'/\phi$ has cardinality* $\exp(\operatorname{poly}(kq/\varepsilon))$.

Given the above theorem, we can immediately show the main results of the chapter.

Proof of Theorem 5.1. Given a Λ-CSP instance $\mathcal{P}$ with variable set $\mathcal{V} = [n]$, we first compute another instance $\mathcal{P}'$ and a variable folding ϕ according to Lemma 5.3. Since $\mathcal{P}'/\phi$ has only $\exp(\mathrm{poly}(kq/\varepsilon))$ variables, we can compute an optimal assignment for $\mathcal{P}'/\phi$ in time $\exp(\exp(\mathrm{poly}(kq/\varepsilon)))$. This assignment can be unfolded to an assignment $x \in [q]^n$ with the same value for $\mathcal{P}'$. Since $\|\mathcal{P}-\mathcal{P}'\|_{\mathrm{TV}} \leqslant \varepsilon$, the assignment x has value at least $\mathrm{opt}(\mathcal{P}'/\phi)-\varepsilon$ for the instance $\mathcal{P}$. By definition of Gap_Λ, we have $\mathrm{opt}(\mathcal{P}'/\phi) \geqslant \mathsf{Gap}_\Lambda(\mathrm{sdp}(\mathcal{P}'/\phi)) \geqslant \mathsf{Gap}_\Lambda(\mathrm{sdp}(\mathcal{P}) - \varepsilon)$. Hence the assignment $x \in [q]^n$ has value at least $\mathsf{Gap}_\Lambda(\mathrm{sdp}(\mathcal{P}) - \varepsilon) - \varepsilon$ as claimed. ■

Proof of Theorem 5.2. By Theorem 5.3, to compute the SDP integrality gap within ε, it is sufficient to go over all instances of size $\exp(\mathrm{poly}(kq/\varepsilon))$. Thus the algorithm would just discretize the space of instances with $\exp(\mathrm{poly}(kq/\varepsilon))$ many variables, and compute the SDP and optimum value for each instance. ■

The rest of this section is devoted to the proof of Theorem 5.3. The construction of $\mathcal{P}'/\phi$ is described below:

Construction of $\mathcal{P}'/\phi$

Dimension reduction Let $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},\ a\in[q]}$, $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}, \boldsymbol{b}_0$ be an SDP solution for a Λ-CSP instance $\mathcal{P}$ on the variable set $\mathcal{V} = [n]$. Suppose that $\boldsymbol{b}_{i,a} \in \mathbb{R}^D$. We apply the following procedure to reduce the dimension from D to d.

1. Sample a $d \times D$ Gaussian matrix $\boldsymbol{\Phi}$, where each entry is independently distributed according to the Gaussian distribution $N(0, 1/d)$.

2. For every vector $\boldsymbol{b}_{i,a}$, compute its image $\boldsymbol{u}_{i,a}$ under the map $\boldsymbol{\Phi}$,

$$\boldsymbol{u}_{i,a} \stackrel{\text{def}}{=} \boldsymbol{\Phi}\boldsymbol{b}_{i,a}\,.$$

   Furthermore, define $\boldsymbol{u}_0 := \boldsymbol{\Phi}\boldsymbol{b}_0$.

Discarding bad constraints Let $B_\varepsilon \subseteq \mathrm{supp}(\mathcal{P})$ be the set of payoff functions P such that the vectors $\boldsymbol{u}_{i,a}$ and the distributions μ_P violate one of the SDP constraints corresponding to P by more than ε. Define the instance $\mathcal{P}'$ on the set of variables $\mathcal{V}$ by removing all payoff functions in B_ε from $\mathcal{P}$. Formally, $\mathcal{P}'$ is obtained by conditioning the distribution $\mathcal{P}$ on the event $P \notin B_\varepsilon$.

Folding by Discretization Let N be an ε-net for the unit ball in $\mathbb{R}^d$. We have $|N| \leqslant (c/\varepsilon)^d$ for some absolute constant c by Fact 5.4.1. For every vector $\boldsymbol{u}_{i,a}$, let $\boldsymbol{w}_{i,a}$ denote its closest vector in N. We identify variables of $\mathcal{P}'$ that have the same vectors $\boldsymbol{w}_{i,a}$. Formally, we output the Λ-CSP instance $\mathcal{P}'/\phi$ where $\phi\colon \mathcal{V} \to N^q$ is defined as

$$\phi(i) \stackrel{\text{def}}{=} (\boldsymbol{w}_{i,1}, \ldots, \boldsymbol{w}_{i,q})\,.$$

5.5.1 Property of Dimension Reduction

The key property of the dimension reduction is that it preserves inner products between vectors approximately.

Lemma 5.3.1 (Inner products are preserved approximately)**.** *For any two vectors $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^D$ in the unit ball,*

$$\Pr_{\boldsymbol{\Phi}}\left\{\left|\langle \boldsymbol{\Phi}\,\boldsymbol{v}_1, \boldsymbol{\Phi}\,\boldsymbol{v}_2\rangle - \langle \boldsymbol{v}_1, \boldsymbol{v}_2\rangle\right| \geqslant \tfrac{t}{\sqrt{d}}\right\} \leqslant O\left(1/t^2\right)\,.$$

Proof. Note that we may assume that both vectors are unit vectors (otherwise, we can normalize them). Suppose that $\langle \boldsymbol{v}_1, \boldsymbol{v}_2\rangle = \alpha$. By rotational invariance, we can assume that $\boldsymbol{v}_1 = (1, 0)$ and $\boldsymbol{v}_2 = (\alpha, \beta)$, where $\beta = \sqrt{1-\alpha^2}$. Hence, $\langle \boldsymbol{\Phi}\,\boldsymbol{v}_1, \boldsymbol{\Phi}\,\boldsymbol{v}_2\rangle$ has the same

distribution as

$$\frac{1}{d}\left(\sum_{i=1}^{d}\alpha\xi_i^2+\beta\xi_i\xi_i'\right),$$

where $\xi_1,\xi_1',\ldots,\xi_d,\xi_d'$ are independent standard Gaussian variables (mean 0 and standard deviation 1).

For each i, the expectation of $\alpha\xi_i^2+\beta\xi_i\xi_i'$ is equal to α and the variance is bounded (at most 2). Hence, the expectation of $\langle \mathbf{\Phi}\,\boldsymbol{v}_1, \mathbf{\Phi}\,\boldsymbol{v}_2\rangle$ is equal to α and the standard deviation is $O(1/\sqrt{d})$. The lemma follows from Chebychev's inequality. ■

It is clear that the dimension-reduced vectors $\boldsymbol{u}_{i,a}$ together with the distributions μ_P need not form a feasible SDP solution. However, we can deduce from Lemma 5.3.1 that with good probability most of the constraints will be nearly satisfied. It follows that not too many payoffs are discarded from $\mathcal{P}$ to construct $\mathcal{P}'$.

Lemma 5.3.2. *For every payoff $P\in\operatorname{supp}(\mathcal{P})$,*

$$\Pr_{\mathbf{\Phi}}\{P\in B_\varepsilon\}\leqslant O\left(\tfrac{k^2q^2}{\varepsilon^2 d}\right).$$

Proof. Fix a payoff $P\in\operatorname{supp}(\mathcal{P})$. There are k^2q^2 SDP constraints associated with the payoff P, all of the form (5.1) in the LC program. By Lemma 5.3.1, each inner product is preserved up to an error ε with probability at most $1-d\varepsilon^2$ (substitute $t=\varepsilon\sqrt{d}$ in Lemma 5.3.1). By a union bound over all the k^2q^2 constraints, we get that $\Pr_{\mathbf{\Phi}}\{P\in B_\varepsilon\}$ is at most $k^2q^2/\varepsilon^2 d$. ■

5.5.2 Discretization

Consider two vectors $\boldsymbol{u}_{i,a}$ and $\boldsymbol{u}_{j,b}$ in the unit ball. It is clear that if we move the vectors to their closest point in N, their inner product changes by at most 2ε. (Since N is an ε-net of the unit ball, each vector is moved by at most ε.)

A minor technical issue is that some of the points $\boldsymbol{u}_{i,a}$ might be outside of the unit ball. However, vectors of norm more than $\sqrt{1+\varepsilon}$ can be ignored, because they violate the constraint $\langle \boldsymbol{u}_{i,a},\boldsymbol{u}_{i,a}\rangle\leqslant 1$ by more than ε.

In particular, the following lemma holds.

Lemma 5.3.3. *For small enough $\varepsilon>0$, suppose the vectors $\boldsymbol{u}_{i,a}$ satisfy all constraints corresponding to some payoff function $P\in\operatorname{supp}(\mathcal{P}')$ up to an error of ε. Then, the vectors $\boldsymbol{w}_{i,a}$ satisfy all constraints corresponding to P up to an error of 4ε.*

The discretization into the ε-net changes each inner product by at most 2ε. The vectors $\boldsymbol{u}_{i,a}$ could be ε away from the unit ball, which in turn introduces another 2ε error in the inner products.

Here we are using the fact that for each payoff $P\in\operatorname{supp}(\mathcal{P}')$, the corresponding constraints in the relaxation $\operatorname{sdp}(\mathcal{P}')$ involve just a single inner product. We also use the fact the vectors $\boldsymbol{u}_{i,a}$ for a variable $i\in\mathcal{V}(P)$ with $P\in\operatorname{supp}(\mathcal{P}')$ have norms at most $\sqrt{1+\varepsilon}$.

5.5.3 Robustness of SDP Relaxation LC

To finish the proof of Theorem 5.3, we need to construct a completely feasible SDP solution to $\mathcal{P}'/\phi$ from the vectors $\boldsymbol{w}_{i,a}$ which nearly satisfy all the constraints.

We will use the following theorem concerning the robustness of LC relaxation from Section 4.8.

Theorem 5.4 (Robustness of LC)**.** *Let $\mathcal{P}$ be a Λ-CSP instance on variable set $\mathcal{V}$. Suppose that $\{\boldsymbol{b}_{i,a}\}_{i\in V,a\in[q]}$, $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ is an ε-infeasible SDP solution for $\mathcal{P}$ of value α. Here, ε-infeasible means that all consistency constraints (5.1)–(5.3) of the relaxation LC are satisfied up to an additive error of at most ε. Then,*

$$\mathrm{sdp}(\mathcal{P}) \geqslant \alpha - \sqrt{\varepsilon}\cdot \mathrm{poly}(kq)\,.$$

5.5.4 Proof of Theorem 5.3

Assuming Theorem 5.4 (Robustness of LC) we can now complete the proof of Theorem 5.3.

For simplicity, we assume that the SDP solution $\{\boldsymbol{b}_{i,a}\}$, $\{\mu_P\}$ that was used in the construction of $\mathcal{P}'/\phi$ has value $\mathrm{sdp}(\mathcal{P})$. (The proof also works if the value of this SDP solution is close to the optimal value.)

Recall that $B_\varepsilon \subseteq \mathrm{supp}(\mathcal{P})$ is the set of payoff functions P whose constraints are violated by more than ε by the dimension-reduced vectors $\boldsymbol{u}_{i,a}$. For $d \gg k^2q^2/\varepsilon^3$, Lemma 5.3.2 implies that with high probability, $\|\mathcal{P}-\mathcal{P}'\|_{\mathrm{TV}} \leqslant \varepsilon$. Note that the vectors $\{\boldsymbol{u}_{i,a}\}$ together with the original local distributions $\{\mu_P\}$ form an ε-infeasible SDP solution for $\mathcal{P}'$. Hence, by Lemma 5.3.3, the SDP solution $\{\boldsymbol{w}_{i,a}\}$, $\{\mu_P\}$ is 4ε-infeasible. The value of this SDP solution for the instance $\mathcal{P}'$ is at least $\mathrm{sdp}(\mathcal{P}) - \|\mathcal{P}-\mathcal{P}'\|_1 \geqslant \mathrm{sdp}(\mathcal{P}) - \varepsilon$. The key observation is now that the SDP solution $\{\boldsymbol{w}_{i,a}\}$, $\{\mu_P\}$ is also a solution for the folded instance $\mathcal{P}'/\phi$. To see this, observe that all the vectors that are merged to a single variable under the folding ϕ, have the same SDP vector (due to discretization using the ε-net). Therefore, we see that $\mathcal{P}'/\phi$ has a 4ε-infeasible SDP solution of value at least $\mathrm{sdp}(\mathcal{P}) - \varepsilon$. Theorem 5.4 (Robustness of LC) asserts that in this situation we can conclude $\mathrm{sdp}(\mathcal{P}'/\phi) \geqslant \mathrm{sdp}(\mathcal{P}) - \sqrt{\varepsilon}\cdot\mathrm{poly}(kq)$. Finally, we observe that the cardinality of the variable set of $\mathcal{P}'/\phi$ is at most $|N|^q \leqslant (c/\varepsilon)^{dq} = 2^{\mathrm{poly}(kq/\varepsilon)}$. ■

Part II

THE UNIQUE GAMES BARRIER

Chapter 6

DICTATORSHIP TESTS, ROUNDING SCHEMES AND UNIQUE GAMES CONJECTURE

One of the main contributions of this thesis is the direct connection it establishes between SDP integrality gaps and UG hardness results. In this chapter, we present an exposition of the connections that have emerged in this thesis and other works, between SDP integrality gaps, UG-hardness results, and objects commonly referred to as "Dictatorship tests". In the subsequent chapters, we will demonstrate how these connections have implications on the approximability of several large classes of problems.

Organization We begin the chapter by introducing the notion of dictatorship tests using Max Cut as an example. In Section 6.2, we survey the surprising connections that have emerged between the three objects of interest in this chapter, dictatorship tests, UG-hardness results and SDP integrality gaps. In the next section, we present the reduction from dictatorship test to UG-hardness result for Max Cut from [99]. A reduction from integrality gaps to dictatorship tests is presented in sections 6.4 and 6.5. While we present the reduction and an intuitive idea of the proof in Section 6.4, a formal proof is presented in Section 6.5. This reduction and its various applications is one of the major contributions of this dissertation. In the subsequent section, we will show how the analysis of this reduction can be used to obtain a rounding scheme for the semidefinite program. In Section 6.7, we present the reduction of Khot-Vishnoi [104] from UG hardness results to SDP integrality gaps. Finally, we will study the surprisingly strong implications of these reductions in Section 6.8

Mathematical Tools: This chapter uses harmonic analysis of boolean functions, and associated notions of influences and noise stability (Section 3.3). Invariance principle (Section 3.6) lies at the heart of the reduction from integrality gaps to dictatorship tests presented in Section 6.7. However, we present a simple version of the invariance principle suited for the application at hand in Section 6.5.

6.1 Dictatorship Tests

The motivation for the problem of dictatorship testing arises from hardness of approximation and PCP constructions. To show that an optimization problem Λ is **NP**-hard to approximate, one constructs a reduction from a well-known problem that is **NP**-hard to approximate such as Label Cover to Λ. Given an instance Φ of the Label Cover problem, a hardness reduction produces an instance Φ' of the problem Λ. The instance Φ' has a large optimum value if and only if Φ has a high optimum. Dictatorship tests serve as "gadgets" that encode solutions to Label Cover, as solutions to the problem Λ. In fact, constructing an appropriate dictatorship test almost always translates into a corresponding hardness result based on the Unique Games Conjecture.

Dictatorship tests, or long-code tests as they are also referred to, were originally conceived purely from the insight of error correcting codes. Let us suppose we are to encode a message that could take one of R different values $\{m_1, \ldots, m_R\}$. The long code encoding of the message m_ℓ is a bit string of length 2^R consisting of the truth table of the function $\mathcal{F}(x_1, \ldots, x_R) = x_\ell$. This encoding is *maximally redundant* in that any binary encoding with more than 2^R bits would contain 2 bits that are identical for all R messages. Intuitively, the greater the redundancy in the encoding, the easier it is to perform the reduction.

While long code tests/dictatorship tests were originally conceived from a coding-theoretic perspective, somewhat surprisingly these objects are intimately connected to semidefinite programs.

6.1.1 The case of Max Cut

The nature of dictatorship test needed for a hardness reduction varies with the specific problem one is trying to show is hard. To keep things concrete and simple, we will restrict our attention to the Max Cut problem.

A dictatorship test DICT for the Max Cut problem consists of a graph on the set of vertices $\{\pm 1\}^R$. By convention, the graph DICT is a weighted graph where the edge weights form a probability distribution (sum up to 1). We will write $(\boldsymbol{z}, \boldsymbol{z}') \in \mathsf{DICT}$ to denote an edge sampled from the graph DICT (here $\boldsymbol{z}, \boldsymbol{z}' \in \{\pm 1\}^R$).

A cut of the DICT graph can be thought of as a boolean function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$. For a boolean function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$, let $\mathsf{DICT}(\mathcal{F})$ denote the value of the cut. The value of a cut $\mathcal{F}$ is given by

$$\mathsf{DICT}(\mathcal{F}) = \frac{1}{2} \mathop{\mathbb{E}}_{(\boldsymbol{z},\boldsymbol{z}') \in \mathsf{DICT}} \left[1 - \mathcal{F}(\boldsymbol{z})\mathcal{F}(\boldsymbol{z}')\right]$$

and is the probability that $\boldsymbol{z}$,$\boldsymbol{z}'$ are on different sides of the cut. It is also useful to define $\mathsf{DICT}(\mathcal{F})$ for non-boolean functions $\mathcal{F} : \{\pm 1\}^R \to [-1, 1]$ that take values in the interval $[-1, 1]$. To this end, we will interpret a value $\mathcal{F}(\boldsymbol{z}) \in [-1, 1]$ as a random variable that takes $\{\pm 1\}$ values. Specifically, we think of a number $a \in [-1, 1]$ as the following random variable

$$a = \begin{cases} -1 & \text{with probability } \frac{1-a}{2} \\ 1 & \text{with probability } \frac{1+a}{2} \end{cases} \tag{6.1}$$

With this interpretation, the natural definition of $\mathsf{DICT}(\mathcal{F})$ for such a function is as follows:

$$\mathsf{DICT}(\mathcal{F}) = \frac{1}{2} \mathop{\mathbb{E}}_{(\boldsymbol{z},\boldsymbol{z}') \in \mathsf{DICT}} \left[1 - \mathcal{F}(\boldsymbol{z})\mathcal{F}(\boldsymbol{z}')\right].$$

Indeed, the above expression is equal to the expected value of the cut obtained by randomly rounding the values of the function $\mathcal{F} : \{\pm 1\}^R \to [-1, 1]$ to $\{\pm 1\}$ as described in Equation 6.1.

The *dictator cuts* are given by the functions $\mathcal{F}(\boldsymbol{z}) = z_\ell$ for some $\ell \in [R]$. The Completeness

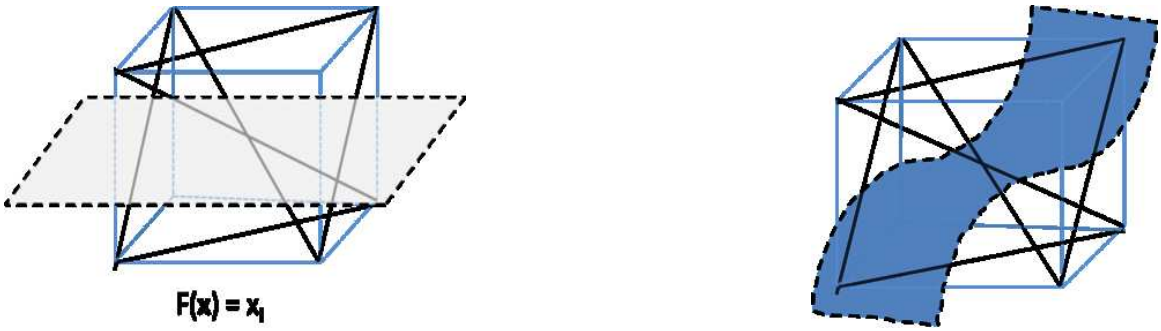

Figure 6.1: Dictator and Non-Dictator cuts

of the test DICT is the minimum value of a dictator cut, i.e.,

$$\mathsf{Completeness}(\mathsf{DICT}) = \min_{\ell \in [R]} \mathsf{DICT}(z_\ell)$$

The soundness of the dictatorship test is the value of cuts that are *far from every dictator*. We will formalize the notion of being *far from every dictator* is formalized using influences as follows:

Definition 6.1.1 ((τ,ε)-quasirandom)**.** A function $\mathcal{F} : \{\pm 1\}^R \to [-1,1]$ is said to be (τ,ε)-quasirandom if for all $\ell \in [R]$, $\mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}) \leqslant \tau$.

Definition 6.1.2 ($\mathsf{Soundness}_{\tau,\varepsilon}$)**.** For a dictatorship test DICT over $\{\pm 1\}^R$ and $\varepsilon, \tau > 0$, define the soundness of DICT as

$$\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}) = \max_{\substack{\mathcal{F}:\{\pm 1\}^R \to [-1,1] \\ \mathcal{F} \text{ is } (\tau,\varepsilon)-\text{quasirandom}}} \mathsf{DICT}(\mathcal{F})\,.$$

6.1.2 Property Testing Perspective

The goal of a property testing algorithm is to determine if an object satisfies a certain property by making very few queries to it. For instance, given a function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$, one would wish to determine if $\mathcal{F}$ is a constant function by making very few queries (say constantly many) to the value of $\mathcal{F}$. However this is an impossible to achieve and one relaxes the goal to determining with good probability if the object satisfies the property or is *far* from satisfying the property. A function $\mathcal{F}$ can be said to be ε-far from being a constant, if it differs from every constant function on ε-fraction of the inputs. Hence, in this case one would wish to distinguish whether the $\mathcal{F}$ is a constant function or is ε-far from being constant.

Dictatorship tests are specific examples of property testing algorithms, where the concerned objects are boolean functions and the property being tested is whether the function is a dictator. Given a boolean function $\mathcal{F}(x_1, x_2, \ldots, x_R)$ on R bits, the goal is to distinguish

whether $\mathcal{F}$ is a dictator ($\mathcal{F}(x) = x_i$ for some i) or is *far* from being a dictator by making constantly many queries to $\mathcal{F}$.

Formally, a dictatorship test is a randomized procedure DICT that, given the truth table of $\mathcal{F}$, queries a few locations (say 2) in the truth table, tests a predicate P on the values it queried and outputs ACCEPT or REJECT. The randomized procedure will often be referred to as the *Verifier*.

The main parameters of interest in a dictatorship test are :

- **Completeness**(c) Every dictator function $\mathcal{F}(x) = x_i$ is accepted with probability at least c.
- **Soundness**(s) Any function $\mathcal{F}$ which is *far* from every dictator is accepted with probability at most s.
- Number of queries made, and the predicate P used by the test.

This notion of dictatorship tests as a verifiers is equivalent to the earlier definition of dictatorship tests as gadgets.

Let us restrict our attention to dictatorship tests for Max Cut. Let DICT be a verifier corresponding to a dictatorship test for functions on $\{\pm 1\}^R$. Let us suppose that DICT makes two queries, and always tests a predicate $P(x, y) = \mathbf{1}_{[x \neq y]}$. The distribution of queries of the DICT yield a weighted graph on $\{\pm 1\}^R$, sum of whose weights sum to 1. Every boolean function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$ corresponds to a *cut* of this graph. The probability of the verifier accepting $\mathcal{F}$ is exactly the fraction of edges cut by the corresponding cut.

Conversely, let DICT be a graph on $\{\pm 1\}^R$ with edge weights summing to 1. Consider a verifier that picks a random edge $(\boldsymbol{x}, \boldsymbol{y})$ in DICT, and tests if $\mathcal{F}(\boldsymbol{x}) \neq \mathcal{F}(\boldsymbol{y})$. It is easy to see that the probability of acceptance of a function $\mathcal{F}$ is exactly the fraction of edges cut by the corresponding cut.

Both formulations of dictatorship tests are convenient for use in different contexts, and it is thus important to be mindful of the two.

6.2 Emerging Connections (history)

Dictatorship tests/long-code tests were introduced into hardness of approximation by the work of Bellare,Goldreich, and Sudan [22]. Long-code tests have since been the main proof strategy in numerous influential works such as the hardness of approximation results for Max 3SAT [86] and Maximum Clique [82]. We refer the reader to [98] for a survey on the various applications of dictatorship testing in hardness of approximation.

With the advent of the Unique Games Conjecture, dictatorship tests have started playing an even greater role in hardness of approximation. In fact, the work of Khot [97] that introduced the Unique Games Conjecture made use of long-code tests to obtain a hardness for Max 2-Sat. Like most previous works, the notion of being *far from a dictator* was defined there in terms of non-existence of sparse Fourier coefficients of large weight.

The notion of dictatorship tests as defined earlier in this chapter made its appearance in the work of Khot et al. [99]. Although influences of functions figured earlier in the work

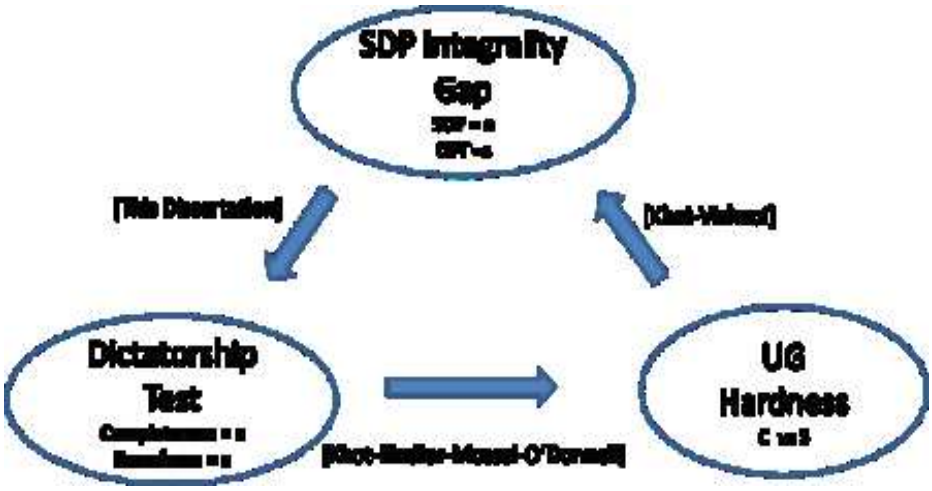

Figure 6.2: Connections Between SDP Integrality Gaps, UG-hardness Results and Dictatorship Tests

of Dinur and Safra [50], it is [99] that used them to define the notion of being far from a dictator. The authors used this notion of dictatorship testing to obtain a UG-hardness result for Max Cut that matches the approximation obtained by Goemans-Williamson algorithm. This work has since served as a general template to obtain UG-hardness results starting with dictatorship tests [17, 18, 136]

In fact, it is true as a rule of thumb that an appropriate dictatorship test often leads to a UG-hardness result. Specifically, given a dictatorship test with completeness c and soundness s such that the verifier only tests a predicate P, it often implies a c-vs-s UG-hardness result for the constraint satisfaction problem where the constraints are of the form - predicate P applied to subset of variables. However, we stress that in many cases (such as Ordering CSPs Chapter 9), executing this conversion from dictatorship tests into UG-hardness results poses considerable challenges.

The next connection emerged with the influential work of Khot-Vishnoi [104] who exhibited a reduction from a UG-hardness result to a SDP integrality gap. First, the paper constructed a UG-hardness reduction for Sparsest Cut. Going a step further, the work constructed a SDP integrality gap instance Φ for Unique games and executed the hardness reduction for Sparsest Cut on Φ. Surprisingly, the instance of Sparsest Cut so produced was a SDP integrality gap! This yielded a super-constant integrality gap for Sparsest Cut thus disproving an earlier conjecture of Goemans and Linial. Among the many reasons that make this work remarkable, the reduction from a UG-hardness result to an SDP integrality gap is clearly an important one. Since this work, the reduction from UG-hardness results to SDP integrality gaps has been executed for various problems like Maximum Acyclic Subgraph [73] and Metric Labeling [121]. As a rule of thumb, it could be said that a c-vs-s UG-hardness result for a problem Λ yields an instance of Λ with SDP value c while the integral optimum is only s.

This dissertation exhibits a reduction from SDP integrality gaps to dictatorship tests. Roughly speaking, we show that starting with a c-vs-s SDP integrality gap for a problem Λ, one can construct a dictatorship test DICT with completeness c and soundness s. Moreover, DICT is a test that could be used for showing a UG-hardness for Λ, in that it is a gadget for the problem Λ.

This completes the cycle of reductions between dictatorship tests, SDP integrality gaps and UG-hardness results. Not only does this prove a certain equivalence of the three notions, it has led to optimal UG-hardness results and matching approximation algorithms for various classes of problems such as constraint satisfaction problems. Furthermore, direct reductions from integrality gaps to dictatorship tests have paved the way to obtaining optimal hardness results for combinatorial optimization problems for which even integrality gaps are unknown. Specifically, for problems such as k-Way Cut for which the correct value of the integrality gap is unknown, the techniques of this dissertation show that a certain linear program yields the optimal approximation under UGC whatever the approximation factor may be. Finally, the reduction establishes a formal connection between integrality gaps and hardness results that was long suspected, but rarely formalized.

6.3 From Dictatorship Tests to UG-hardness Results

In this section, we will see how dictatorship tests can be translated into obtain Unique Games based hardness result.Formally, we show

Theorem 6.1. *For every $\varepsilon, \tau, \eta > 0$, there exists positive integer k_0 such that, if $\mathsf{DICT}^\varepsilon$ is a dictatorship test for* Max Cut *over $\{\pm 1\}^k$ for some $k > k_0$, then given a* Max Cut *instance Ψ, it is UG-hard to distinguish between the following two cases:*

- *There exists a cut of Ψ of value at least $\mathsf{Completeness}(\mathsf{DICT}^\varepsilon) - \eta$,*
- *No cut of the graph Ψ has value greater than $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^\varepsilon) + \eta$.*

Roughly speaking, if we construct a dictatorship test $\mathsf{DICT}^\varepsilon$ for every large dimension, with completeness c and soundness s, then we get a c vs s-UG-hardness for Max Cut.

To prove the above theorem, we will exhibit a reduction from unique games problem to Max Cut, via the dictatorship test. Specifically, we will show the following reduction.

Reduction Let $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi, [R])$ be a bipartite Unique Games instance. Let $\mathsf{DICT}^\varepsilon$ denote a dictatorship test for Max Cut over the R-dimensional hypercube ($\{\pm 1\}^R$).

Starting from the Unique Games instance Φ, we shall construct an instance $\Psi = (\mathcal{V}_\Psi, E_\Psi)$ of the Max Cut problem. The graph Ψ contains 2^R vertices indexed by $\{\pm 1\}^R$ for every vertex of the Unique Games instance Φ. Formally, the set of vertices Ψ is given by $\mathcal{V}_\Psi = \mathcal{V}_\Phi \times \{\pm 1\}^R$.

For a Unique Games vertex $v \in \mathcal{V}_\Phi$, the set of 2^R vertices $\{(v, \boldsymbol{x}) | \boldsymbol{x} \in \{\pm 1\}^R\}$ is the *long code* corresponding to the Unique Games vertex v. The choice of a label for the Unique Games vertex v, is encoded as a cut of the corresponding set of 2^R vertices.

Specifically, if the vertex v is assigned a label $\ell \in [R]$, then the ℓ-th dictator cut is chosen for the corresponding set of vertices.

$$v \text{ is assigned } \ell \in [R] \implies \text{For all } \boldsymbol{x} \in \{\pm 1\}^R, (v, \boldsymbol{x}) \text{ is assigned } x_\ell \,.$$

In terms of coding theory, we are encoding the label assigned to vertex v, using its *long code* which is a code of length 2^R.

To describe the edges E_Ψ, we first set up some notation. For a vector $\boldsymbol{x} \in \{\pm 1\}^R$ and a permutation $\pi : [R] \to [R]$ define the vector $\pi \circ \boldsymbol{x} \in \{\pm 1\}^R$ as follows:

$$\pi \circ \boldsymbol{x} = (x_{\pi(1)}, x_{\pi(2)}, \cdots, x_{\pi(R)})$$

As usual, the graphs $\mathsf{DICT}^\varepsilon$ and Ψ are weighted graphs, where the edge weights form a probability distribution (sum of edge weights = 1). The edges E_Ψ can be sampled using the following procedure:

Edges of the graph Ψ

- Sample a random vertex $w \in \mathcal{W}_\Phi$. Pick two random neighbors v, v' of the vertex w independently at random.
- Sample an edge $(\boldsymbol{z}, \boldsymbol{z}')$ from the graph $\mathsf{DICT}^\varepsilon$.
- Output the edge between $(v, \pi_{w \leftarrow v} \circ \boldsymbol{z})$ and $(v', \pi_{w \leftarrow v'} \circ \boldsymbol{z}')$

By definition, a *cut* of the graph Ψ consists of a map $\mathcal{F} : \mathcal{V}_\Psi \to \{\pm 1\}$. For each vertex $v \in \mathcal{V}_\Phi$, let the function $\mathcal{F}^v : \{\pm 1\}^R \to \{\pm 1\}$ denote the cut restricted to the long code corresponding to vertex v.

Now, let us calculate the value of an arbitrary cut given by functions $\{\mathcal{F}^v : \{\pm 1\}^R \to \{\pm 1\} | \forall v \in \mathcal{V}_\Phi\}$. We can write the value of the cut as follows:

$$\mathrm{val}(\{\mathcal{F}^v\}) = \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{v, v' \in N(w)} \mathop{\mathbb{E}}_{\boldsymbol{z}, \boldsymbol{z}'} \left[\mathbf{1}[\mathcal{F}^v(\pi_{w \leftarrow v} \circ \boldsymbol{z}) \neq \mathcal{F}^{v'}(\pi_{w \leftarrow v'} \circ \boldsymbol{z}')] \right].$$

For two numbers $a, b \in \{\pm 1\}$, we can write $\mathbf{1}[a \neq b] = 1/2(1 - ab)$. Substituting this expression, we obtain

$$\mathrm{val}(\{\mathcal{F}^v\}) = \frac{1}{2} \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{v, v' \in N(w)} \mathop{\mathbb{E}}_{\boldsymbol{z}, \boldsymbol{z}'} \left[1 - \mathcal{F}^v(\pi_{w \leftarrow v} \circ \boldsymbol{z}) \cdot \mathcal{F}^{v'}(\pi_{w \leftarrow v'} \circ \boldsymbol{z}') \right].$$

Notice that the choices of neighbors v, v' are independent of each other and of the choices of $\boldsymbol{z}$ and $\boldsymbol{z}'$. Consequently, we can rewrite the above expression as,

$$\mathrm{val}(\{\mathcal{F}^v\}) = \frac{1}{2} \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{\boldsymbol{z}, \boldsymbol{z}'} \left[1 - \mathop{\mathbb{E}}_{v \in N(w)} \left[\mathcal{F}^v(\pi_{w \leftarrow v} \circ \boldsymbol{z}) \right] \cdot \mathop{\mathbb{E}}_{v' \in N(w)} \left[\mathcal{F}^{v'}(\pi_{w \leftarrow v'} \circ \boldsymbol{z}') \right] \right].$$

For a $w \in \mathcal{W}_\Phi$, define the function $\mathcal{F}^w : \{\pm 1\}^R \to [-1, 1]$ as follows:

$$\mathcal{F}^w(\boldsymbol{z}) = \mathop{\mathbb{E}}_{v \in N(w)} [\mathcal{F}^v(\pi_{w \leftarrow v} \circ \boldsymbol{z})] \qquad \text{for all } \boldsymbol{z} \in \{\pm 1\}^R .$$

The function $\mathcal{F}^w$ is an average of the functions $\mathcal{F}^v$ corresponding to its neighbors $v \in N(w)$, composed with the appropriate permutations. In terms of the functions $\mathcal{F}^w$, we get

$$\mathrm{val}(\{\mathcal{F}^v\}) = \frac{1}{2} \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{\boldsymbol{z}, \boldsymbol{z}'} \left[1 - \mathcal{F}^w(\boldsymbol{z}) \cdot \mathcal{F}^w(\boldsymbol{z}')\right] = \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \left[\mathsf{DICT}^\varepsilon(\mathcal{F}^w)\right] . \tag{6.2}$$

To complete the proof of Theorem 6.1, we analyze the value of the Max Cut instance Ψ in the completeness and soundness cases. Specifically, we will show

Theorem 6.2. *For all $\delta, \gamma > 0$, given an instance Φ of* Unique Games, *the* Max Cut *instance Ψ satisfies the following properties:*

- Completeness: *If Φ is a $(1-\gamma)$-strongly satisfiable instance of Unique Games, then* $\mathrm{opt}(\Psi) \geqslant (1 - \gamma)\mathsf{Completeness}(\mathit{DICT}^\varepsilon)$

- Soundness: $\mathrm{opt}(\Phi) \leqslant \delta \Rightarrow \mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathit{DICT}^\varepsilon) + \frac{\delta}{\varepsilon^2 \tau^3}$.

Completeness Let $\mathcal{A} : \mathcal{V}_\Phi \cup \mathcal{W}_\Phi \to [R]$ be an assignment to the Unique Games instance that strongly satisfies $1 - \gamma$-fraction of the vertices in $\mathcal{W}_\Phi$. Recall that, an assignment *strongly satisfies* a vertex $w \in \mathcal{W}_\Phi$ if $\mathcal{A}(w) = \pi_{w \leftarrow v}(\mathcal{A}(v))$ for all neighbors v of the vertex w.

Define a *cut* for the graph Ψ as follows:

$$\mathcal{F}^v(\boldsymbol{z}) = z_{\mathcal{A}(v)} \qquad \text{for all vertices } v \in \mathcal{V}_\Psi .$$

In other words, for each Unique Games vertex $v \in \mathcal{V}_\Phi$, the cut of $\{v\} \times \{\pm 1\}^R$ corresponds to the long code of the label $\mathcal{A}(v)$.

Consider a vertex $w \in \mathcal{W}_\Phi$ that is *strongly satisfied* by the assignment $\mathcal{A}$. By definition, for each of its neighbors $v \in N(w)$, we have $\pi_{w \leftarrow v}(\mathcal{A}(v)) = \mathcal{A}(w)$. For such a vertex w, the function $\mathcal{F}^w$ is given by

$$\begin{aligned}
\mathcal{F}^w(\boldsymbol{z}) &= \mathop{\mathbb{E}}_{v \in N(w)} \left[\mathcal{F}^v(\pi_{w \leftarrow v} \circ \boldsymbol{z})\right] , \\
&= \mathop{\mathbb{E}}_{v \in N(w)} \left[(\pi_{w \leftarrow v} \circ \boldsymbol{z})_{\mathcal{A}(v)}\right] , \\
&= \mathop{\mathbb{E}}_{v \in N(w)} \left[z_{\pi_{w \leftarrow v}(\mathcal{A}(v))}\right] , \\
&= \mathop{\mathbb{E}}_{v \in N(w)} \left[z_{\mathcal{A}(w)}\right] = z_{\mathcal{A}(w)} . \qquad (\because \forall v \in N(w), \pi_{w \leftarrow v}(\mathcal{A}(v)) = \mathcal{A}(w)) .
\end{aligned}$$

For a strongly satisfied vertex w, the fraction of edges cut is given by

$$\mathsf{DICT}^{\varepsilon}(\mathcal{F}^w) = \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'}\left[1-\mathcal{F}^w(\boldsymbol{z})\mathcal{F}^w(\boldsymbol{z}')\right] = \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'}\left[1-z_{\mathcal{A}(w)}z'_{\mathcal{A}(w)}\right] \tag{6.3}$$

$$= \mathsf{DICT}(z_{\mathcal{A}(w)}) \geqslant \mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}) \tag{6.4}$$

For a vertex w that is not strongly satisfied, trivially we have $\mathsf{DICT}^{\varepsilon}(\mathcal{F}^w) \geqslant 0$. Using Equation 6.3, we can estimate the value of the cut

$$\begin{aligned}\mathrm{val}(\{\mathcal{F}^v\}) &= \mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\left[\mathsf{DICT}^{\varepsilon}(\mathcal{F}^w)\right]\\ &\geqslant \mathop{\mathbb{P}\mathrm{r}}_{w\in\mathcal{W}_\Phi}[w \text{ is strongly satisfied}]\cdot\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon})\\ &\qquad + \mathop{\mathbb{P}\mathrm{r}}_{w\in\mathcal{W}_\Phi}[w \text{ is NOT strongly satisfied}]\cdot 0\\ &\geqslant (1-\gamma)\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon})\,.\end{aligned}$$

Soundness For the sake of brevity, let us denote $\eta = \delta/\varepsilon^2\tau^3$. Let us suppose there is a *cut* of the graph Ψ, with value greater than $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}) + \eta$. Then we have

$$\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}[\mathsf{DICT}^{\varepsilon}(\mathcal{F}^w)] > \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}) + \eta\,.$$

Since $\mathsf{DICT}^{\varepsilon}(\mathcal{F}^w)$ is always bounded above by 1. Hence, for at least η fraction of the vertices $w \in \mathcal{W}_\Phi$, $\mathsf{DICT}^{\varepsilon}(\mathcal{F}^w) > \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon})$ by Markov's inequality. Henceforth we refer to these vertices as *good* vertices.

By definition of $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon})$, for every *good* vertex $w \in \mathcal{W}_\Phi$, the function $\boldsymbol{\mathcal{F}}^w$ is not (τ,ε)-quasirandom. In particular, for a *good* vertex $w \in \mathcal{W}_\Phi$, there exists coordinates $\ell \in [R]$ that are *influential* for the function $\mathcal{F}^w$, i.e., $\mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}^w) \geqslant \tau$.

For each vertex $w \in \mathcal{W}_\Phi$, define the set of labels $\mathsf{L}(w)$ as

$$\mathsf{L}(w) = \{\ell | \mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}^w) > \tau\}\,.$$

Similarly, for each $v \in \mathcal{V}_\Phi$ define,

$$\mathsf{L}(v) = \{\ell | \mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}^w) > \tau/2\}\,.$$

By the Sum of Influences Lemma (Lemma 3.0.2), we have $|\mathsf{L}(w)| \leqslant 1/\varepsilon\tau$ for a vertex $w \in \mathcal{W}_\Phi$ and $|\mathsf{L}(v)| \leqslant 3/\varepsilon\tau$ for $v \in \mathcal{V}_\Phi$.

For each of the *good* vertices $w \in \mathcal{W}_\Phi$ have a non-empty label set $\mathsf{L}(w)$. Fix a *good* vertex w with a nonempty label set $\mathsf{L}(w)$. Fix a label $\ell \in \mathsf{L}(w)$. By definition of $\mathsf{L}(w)$, we have $\mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}^w) \geqslant \tau$. The function $T_{1-\varepsilon}\mathcal{F}^w$ is given by $T_{1-\varepsilon}\mathcal{F}^w(\boldsymbol{z}) = \mathbb{E}_{v\in N(w)}[T_{1-\varepsilon}\mathcal{F}^v(\pi_{w\leftarrow v}\circ \boldsymbol{z})]$. By convexity of influences (see Proposition 3.0.13), if $\mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}^w) > \tau$ then

$$\mathop{\mathbb{E}}_{v\in N(w)}[\mathrm{Inf}_{\pi^{-1}_{w\leftarrow v}(\ell)}(T_{1-\varepsilon}\mathcal{F}^v)] \geqslant \tau \tag{6.5}$$

Since the range of the function $\mathcal{F}^v$ is $\{\pm 1\}$, we have $\mathrm{Inf}_\ell(T_{1-\varepsilon}\mathcal{F}^v) \leqslant 1$ for all v, ℓ. Hence for

at least a $\tau/2$ fraction of neighbors $v \in N(w)$, $\mathrm{Inf}_{\pi_{v \leftarrow w}(\ell)}(T_{1-\varepsilon}\mathcal{F}^v) \geqslant \frac{\tau}{2}$. Summarizing the above argument, for a *good* vertex w and a label $\ell \in \mathsf{L}(w)$, the coordinate $\pi_{v \leftarrow w}(\ell) \in \mathsf{L}(v)$ for at least a $\tau/2$-fraction of the neighbors $v \in N(w)$.

Define a labeling $\mathcal{A}$ for the Unique Games instance Φ as follows: For each vertex $u \in \mathcal{W}_\Phi \cup \mathcal{V}_\Phi$, assign a random label from $\mathsf{L}(u)$ if it is nonempty, else assign a uniformly random label. Specifically,

$$\mathcal{A}(u) = \begin{cases} \text{a random label in } \mathsf{L}(u) & \text{if } \mathsf{L}(u) \text{ is nonempty} \\ \text{an arbitrary label} & \text{if } \mathsf{L}(u) \text{ is empty}\,. \end{cases}$$

For a *good* vertex w, at least $\tau/2$ fraction of the edges (w, v) are satisfied with probability $\frac{1}{|\mathsf{L}(v)||\mathsf{L}(w)|} \geqslant 2\varepsilon^2\tau^2$ by the assignment $\mathcal{A}$. At least η fraction of the vertices $w \in \mathcal{W}_\Phi$ are *good* vertices. Therefore, in expectation, the assignment $\mathcal{A}$ satisfies at least $\eta \cdot \tau/3 \cdot 3\varepsilon^2\tau^2$ fraction of the edges, which is greater than δ by design.

6.4 From Integrality Gaps to Dictatorship Tests

A black box reduction from integrality gaps to dictatorship tests is one of the primary contributions of the thesis. In this section, we present an exposition of the technique for the Max Cut problem.

Let $G = (V, E)$ be an an arbitrary instance of the Max Cut problem. Specifically, G is a weighted graph over a set of vertices $V = \{v_1, \ldots, v_n\}$, whose edge weights sum up to 1 (by convention). Thus, the set of edges E will also be thought of as a probability distribution over edges. We begin by recalling the Goemans-Williamson semidefinite programming relaxation for Max Cut. The variables of the GW SDP consist of a set of vectors $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$, one vector for each vertex in the graph G.

Goemans Williamson SDP

$\mathsf{GW}(G)$ Relaxation

Maximize	$\mathrm{val}(\boldsymbol{V}) = \frac{1}{2} \mathop{\mathbb{E}}_{(v_i,v_j)\in E} \left[1 - \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle\right]$		(Average Squared Length of Edges)
Subject to	$\|\boldsymbol{v}_i\|_2^2 = 1$	$\forall i, 1 \leqslant i \leqslant n$	(all vectors $\boldsymbol{v}_i$ are unit vectors)

The above relaxation is identical to the SDP relaxation GW presented in Chapter 2, with the objective function $\frac{1}{4}\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$ rewritten as,

$$\frac{1}{4}\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 = \frac{1}{4}\Big(\|\boldsymbol{v}_i\|_2^2 + \|\boldsymbol{v}_j\|_2^2 - 2\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle\Big) = \frac{1}{2}(1 - \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle)\,.$$

6.4.1 Intuition

We begin by presenting the intuition behind the black box reduction.

Dimension Reduction Without loss of generality, the SDP solution $\boldsymbol{V}$ can be assumed to lie in a large constant dimensional space. Specifically, given an arbitrary SDP solution $\boldsymbol{V}$ in n-dimensional space, project it into a random subspace of dimension R – a large constant. Random projections approximately preserve the lengths of vectors and distances between them. Hence, roughly speaking, the vectors produced after random projection yield a low-dimensional SDP solution to the same graph G.

Formally, sample R random Gaussian vectors $\{\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_R\}$ of the same dimension as the SDP vectors $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$. Here R is to be thought of as a large constant independent of the size of the graph G. Define a solution to the GW SDP relaxation as follows:

$$\boldsymbol{w}_i = \frac{1}{\sqrt{\sum_{j \in [R]} \langle \boldsymbol{v}_i, \boldsymbol{\zeta}_j \rangle^2}} \Big(\langle \boldsymbol{v}_i, \boldsymbol{\zeta}_1 \rangle, \ldots, \langle \boldsymbol{v}_i, \boldsymbol{\zeta}_R \rangle \Big) \text{ for all vertices } v_i \text{ in graph } G$$

The vector $\boldsymbol{w}_i$ is just the projection of the vector $\boldsymbol{v}_i$ along directions $\{\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_R\}$, normalized to unit length. Since they are of unit length, the vectors $\boldsymbol{w}_i$ form a feasible SDP solution to GW SDP.

For every $\eta > 0$, by choosing R to be a sufficiently large constant, if is fairly well known that the following can be ensured: the distance between any two vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ is preserved up to $(1 \pm \varepsilon)$-multiplicative factor with probability at least $1 - \varepsilon$. A formal proof of this statement can be seen in Lemma 5.3.1. Consequently, there exists some choice of $\{\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_R\}$ such that the vectors $\boldsymbol{w}_i$ form a low-dimensional SDP solution with roughly the same value as $\{\boldsymbol{v}_i\}$, i.e., $\mathrm{val}(\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}) \geqslant \mathrm{val}(\boldsymbol{V}) - \eta$.

Henceforth, without loss of generality, we will assume that the SDP solution $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ consist of R-dimensional vectors for a large enough constant R.

Sphere Graph A *graph on the unit sphere*, will consist of a set of unit vectors, and weighted edges between them. As usual, the weights of the graph form a probability distribution, in that they sum up to 1.

The SDP solution $\boldsymbol{V}$ for a graph G, yields a *graph on the R-dimensional unit sphere*, that is isomorphic to G. Recall that the objective value of the GW SDP is the average squared length of the edges. Hence, the SDP value remains unchanged under the following transformations:

- **Rotation** Any rotation of the SDP vectors $\boldsymbol{V}$ about the origin preserves the lengths of edges and the distances between them. Thus, rotating the SDP solution $\boldsymbol{V}$ yields another feasible solution with the same objective value.

- **Union of Rotations** Let $\{T_1\boldsymbol{v}_1, \ldots, T_1\boldsymbol{v}_n\}$ and $\{T_2\boldsymbol{v}_1, \ldots, T_2\boldsymbol{v}_n\}$ be two solutions obtained by applying rotations T_1, T_2 to the SDP vectors $\boldsymbol{V}$. Let G_1, G_2 be the associated graphs on the unit sphere. Let G' denote the union of the two graphs, i.e., $G' = G_1 \cup G_2$. The set of all distinct vectors in $T_1\boldsymbol{V} \cup T_2\boldsymbol{V}$ are the vertices of G'. The edge distribution of G' is the average of the edge distributions of G_1 and G_2.

  The average squared lengths of edges in both $T_1\boldsymbol{V}$ and $T_2\boldsymbol{V}$ are equal to $\mathrm{val}(\boldsymbol{V})$. Hence, the average squared edge length in G' is also equal to $\mathrm{val}(\boldsymbol{V})$. Thus, taking

the union of two rotations of a graph preserves the SDP value.

Define the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ as follows:

Sphere Graph $\mathcal{S}_{\boldsymbol{V}}$: Union of all possible rotations of the graph G (on the set of vectors $\{\boldsymbol{w}_i\}$) on the R-dimensional unit sphere.

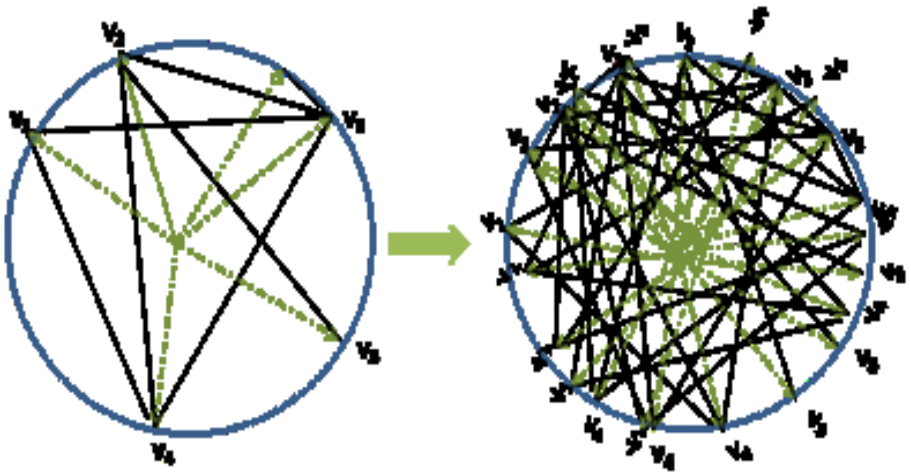

Figure 6.3: Construction of Sphere Graph

Clearly the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ is an infinite graph. The sphere graph $\mathcal{S}_{\boldsymbol{V}}$ is solely a conceptual tool, and an explicit representation is never needed in the reduction. Nevertheless, due to its symmetry, indeed the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ can be represented succinctly.

By construction, the SDP value of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ is the same as that of the original graph G. However, we will argue that $\mathcal{S}_{\boldsymbol{V}}$ is as hard an instance of Max Cut as the original graph G. In fact, given a cut for the sphere graph $\mathcal{S}_{\boldsymbol{V}}$, it is possible to retrieve a cut for the original graph G with the same objective value.

Let us suppose that $\mathcal{F} : \mathcal{S}_{\boldsymbol{V}} \to \{\pm 1\}$ is a cut of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ that cuts a c-fraction of the edges. Notice that $\mathcal{S}_{\boldsymbol{V}}$ consists of a union of infinitely many copies (or rotations) of the graph G. Therefore, on at least one of the copies of G, the cut $\mathcal{F}$ must cut a c-fraction of the edges. Indeed, if we have oracle access to the cut function $\mathcal{F}$, we can efficiently construct a cut of the graph G with the same value as $\mathcal{F}$ using the following rounding procedure:

$\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$

- Sample a rotation T of the unit sphere, uniformly at random.
- Output the cut induced by $\mathcal{F} : \mathcal{S}_{\boldsymbol{V}} \to \{\pm 1\}$ on the copy $T\boldsymbol{V}$ of the graph G.

The expected value of the cut output by the $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$ procedure is equal to the value of the cut $\mathcal{F}$ on the sphere graph $\mathcal{S}_{\boldsymbol{V}}$. An immediate consequence is that,

$$\mathrm{opt}(\mathcal{S}_{\boldsymbol{V}}) \leqslant \mathrm{opt}(G)\,. \tag{6.6}$$

The sphere graph $\mathcal{S}_{\boldsymbol{V}}$ inherits the GW SDP value as G, while the optimum value $\mathrm{opt}(\mathcal{S}_{\boldsymbol{V}})$ is at most that of the graph G. In this light, the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ is a *harder* instance of Max Cut than the original graph G.

It is easy to see that the following is an equivalent definition for the sphere graph $\mathcal{S}_{\boldsymbol{V}}$.

Definition 6.4.1 (Sphere Graph $\mathcal{S}_{\boldsymbol{V}}$)**.** Given a feasible solution $\boldsymbol{V}$ to the GW SDP, the set of vertices of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ is the set of all points on the R-dimensional unit sphere. To sample an edge of $\mathcal{S}_{\boldsymbol{V}}$ use the following procedure:

- Sample an edge (v_i, v_j) in the graph G,
- Sample two points $(\boldsymbol{g}, \boldsymbol{g}')$ on the sphere at a squared distance $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$ uniformly at random.
- Output the edge between $(\boldsymbol{g}, \boldsymbol{g}')$.

Hypercube Graph Finally, we describe the construction of the graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ on the R-dimensional hypercube. Here we refer to the hypercube suitably normalized to make all its points lie on the unit sphere.

$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$

The set of vertices of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ are points in $\left\{-\frac{1}{\sqrt{R}}, \frac{1}{\sqrt{R}}\right\}^R$. An edge of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ can be sampled as follows:

- Sample an edge (v_i, v_j) in the graph G.
- Sample two points $(\boldsymbol{z}, \boldsymbol{z}')$ in $\{-\frac{1}{\sqrt{R}}, \frac{1}{\sqrt{R}}\}^R$, at squared distance $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$ uniformly at random.
- Output the edge between $(\boldsymbol{z}, \boldsymbol{z}')$.

It is likely that there are no pair of points on the hypercube $\left\{-\frac{1}{\sqrt{R}}, \frac{1}{\sqrt{R}}\right\}^R$ at a distance exactly equal to $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$. For the sake of exposition, we will ignore this issue for now. To remedy this issue in the final construction, for each edge (v_i, v_j), we introduce a probability distribution over edges such that the expected length of an edge is indeed $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$.

Completeness Consider the ℓ^{th} dictator cut $\mathcal{F} : \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}} \to \{\pm 1\}$ given $\mathcal{F}(\boldsymbol{z}) = \sqrt{R} z_\ell$. This corresponds to the axis-parallel cut of the $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ graph along the ℓ^{th} axis of the hypercube. Let us estimate the value of the cut $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F})$. An edge $(\boldsymbol{z}, \boldsymbol{z}')$ is *cut* by the

ℓ^{th} dictator cut if and only if $z_\ell \neq z'_\ell$. Therefore, the value of the ℓ^{th} dictator cut $\mathcal{F}$ is given by:

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) = \mathop{\mathbb{E}}_{(v_i,v_j)\in G} \mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \Big[\mathbf{1}[z_\ell \neq z'_\ell]\Big]$$

Notice that two points $\boldsymbol{z}, \boldsymbol{z}'$ in $\{\pm 1\}^R$ at a squared distance $d = \|\boldsymbol{z} - \boldsymbol{z}'\|_2^2$ differ on exactly $d/4$ coordinates. Hence, two random points at distance $\boldsymbol{z}, \boldsymbol{z}'$ at a squared distance d differ on a given coordinate with probability $d/4$. Therefore, we can rewrite the expression for $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F})$ as follows:

$$\begin{aligned}\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) &= \frac{1}{4} \mathop{\mathbb{E}}_{(v_i,v_j)\in G} \Big[\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2\Big] \\ &= \frac{1}{2} \mathop{\mathbb{E}}_{(v_i,v_j)\in G} \Big[1 - \langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle\Big] = \mathrm{val}(\boldsymbol{V})\,.\end{aligned}$$

Hence, the completeness $\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon})$ test is at least $\mathrm{val}(\boldsymbol{V})$.

Soundness Consider a cut $\mathcal{F} : \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}} \to \{\pm 1\}$ that is *far from every dictator*. Intuitively, the cut is not parallel to any of the axis of the hypercube. Note the strong similarity in the construction of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ and the hypercube graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$. In both cases, we sampled two random points at a distance equal to the edge length. In fact, the hypercube graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ is a subgraph of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$. The existence of special directions (the axes of the hypercube) is what distinguishes the hypercube graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ from the sphere graph $\mathcal{S}_{\boldsymbol{V}}$. Thus, roughly speaking, a cut $\mathcal{F}$ that is not parallel to any axis must be unable to distinguish between the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ and the hypercube graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$. If we visualize the cut $\mathcal{F}$ of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ as a geometric surface not parallel to any axis (see Figure 6.4), then the same geometric surface viewed as a cut of the sphere graph must separate roughly the same fraction of edges.

Indeed, the above intuition can be made precise if the cut $\mathcal{F}$ is sufficiently smooth (low degree). The cut $\mathcal{F} : \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}} \to \{\pm 1\}$ can be expressed as a multilinear polynomial F (by Fourier expansion), thus extending the cut function $\mathcal{F}$ from $\{-1/\sqrt{R}, 1/\sqrt{R}\}^R$ to $\mathbb{R}^R$. The function $\mathcal{F}$ is *smooth* if the corresponding polynomial polynomial F is *low degree*. If $\mathcal{F}$ is *smooth* and *far from every dictator*, then one can show that,

$$\text{Value of } \mathcal{F} \text{ on } \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}} \approx \text{ Value of } F \text{ on } \mathcal{S}_{\boldsymbol{V}}$$

By inequality 6.6, the maximum value of a cut of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ is at most $\mathrm{opt}(G)$. Therefore, for any cut $\mathcal{F} : \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}} \to \{\pm 1\}$ that is *smooth* and *far from every dictator*, we get $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) \precsim \mathrm{opt}(G)$.

Ignoring the *smoothness* condition for now, the above argument shows that the soundness of the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ is at most $\mathrm{opt}(G)$. Summarizing the above discussion, starting from a SDP solution $\{\boldsymbol{v}_i\}$ for a graph G, we constructed hypercube graph (dictatorship test) $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ such that $\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \geqslant \mathrm{val}(\{\boldsymbol{v}_i\})$, and $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \precsim \mathrm{opt}(G)$.

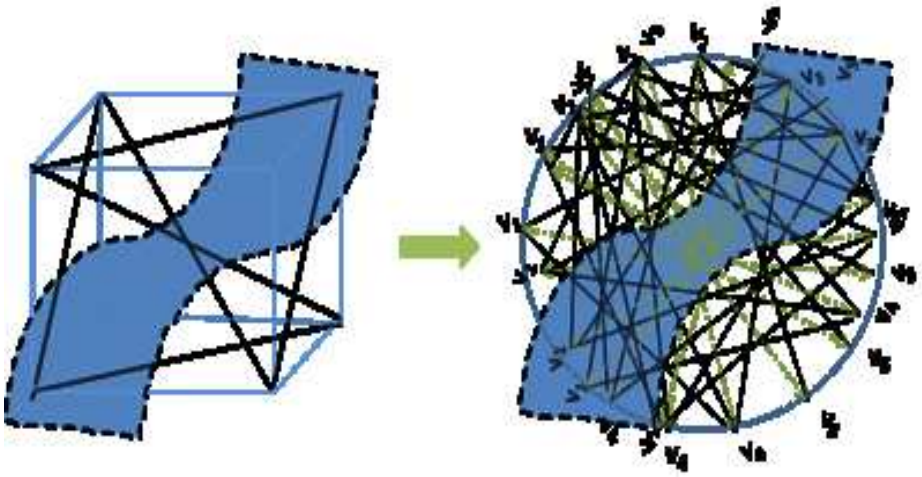

Figure 6.4: Extending the Cut from $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ to $\mathcal{S}_{\boldsymbol{V}}$

By suitably modifying the construction of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$, the *smoothness* requirement for the cut can be dropped. The basic idea is fairly simple yet powerful. In the definition of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$, while introducing an edge between $(\boldsymbol{z}, \boldsymbol{z}')$, perturb each coordinate of $\boldsymbol{z}$ and $\boldsymbol{z}'$ with a tiny probability ε to obtain $\tilde{\boldsymbol{z}}$ and $\tilde{\boldsymbol{z}}'$ respectively, then introduce the edge $(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{z}}')$ instead of $(\boldsymbol{z}, \boldsymbol{z}')$. The introduction of noise to the vertices $\boldsymbol{z}$ and $\boldsymbol{z}'$ has an averaging effect on the cut function, thus making it smooth.

6.5 Formal Proof of Reduction

Let $G = (V, E)$ be an arbitrary instance of Max Cut. Let $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ be a feasible solution to the GW SDP relaxation.

Locally, for every edge $e = (v_i, v_j)$ in G, there exists a distribution over $\{\pm 1\}$ assignments that match the SDP inner products. In other words, there exists $\{\pm 1\}$ valued random variables z_i, z_j such that

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = \mathbb{E}[z_i \cdot z_j] \,.$$

For each edge $e = (v_i, v_j)$, let μ_e denote the local integral distribution over $\{\pm 1\}$ assignments.

The details of the construction of dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ are as follows:

$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ (MaxCut Example)
The set of vertices of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ consists of the R-dimensional hypercube $\{\pm 1\}^R$. The distribution of edges in $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ is the one induced by the following sampling procedure:

- Sample an edge $e = (v_i, v_j) \in E$ in the graph G.
- Sample R times independently from the distribution μ_e to obtain $\boldsymbol{z}_i^R = (z_i^{(1)}, \ldots, z_i^{(R)})$ and $\boldsymbol{z}_j^R = (z_j^{(1)}, \ldots, z_j^{(R)})$, both in $\{\pm 1\}^R$.
- Perturb each coordinate of $\boldsymbol{z}_i^R$ and $\boldsymbol{z}_j^R$ independently with probability ε to obtain $\tilde{\boldsymbol{z}}_i^R, \tilde{\boldsymbol{z}}_j^R$ respectively. Formally, for each $\ell \in [R]$,

$$\tilde{z}_i^{(\ell)} = \begin{cases} z_i^{(\ell)} & \text{with probability } 1-\varepsilon \\ \text{uniformly random value in } \{\pm 1\} & \text{with probability } \varepsilon \end{cases}$$

- Output the edge $(\tilde{\boldsymbol{z}}_i^R, \tilde{\boldsymbol{z}}_j^R)$.

Theorem 6.3. *There exist absolute constants C, K such that for all $\varepsilon, \tau \in [0,1]$, for any graph G and an SDP solution $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ for the GW-SDP relaxation of G,*

- $\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \geqslant \mathrm{val}(\boldsymbol{V}) - 2\varepsilon$
- $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \leqslant \mathrm{opt}(G) + C\tau^{K\varepsilon}$.

Let $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$ be a cut of the $\mathsf{DICT}^{\varepsilon}$ graph. The fraction of edges cut by $\mathcal{F}$ is given by

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) = \frac{1}{2} \mathop{\mathbb{E}}_{(v_i,v_j)\in E} \mathop{\mathbb{E}}_{\boldsymbol{z}_i^R, \boldsymbol{z}_j^R} \mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_i^R, \tilde{\boldsymbol{z}}_j^R} \left[1 - \mathcal{F}(\tilde{\boldsymbol{z}}_i^R) \cdot \mathcal{F}(\tilde{\boldsymbol{z}}_j^R)\right] \tag{6.7}$$

In the above expression, the expectation over $\tilde{\boldsymbol{z}}_i$ $\tilde{\boldsymbol{z}}_j$ refers to an expectation over the noise. Thus, formally the inner expectation should be written as over $\tilde{\boldsymbol{z}}_i | \boldsymbol{z}_i$ and $\tilde{\boldsymbol{z}}_j | \boldsymbol{z}_j$.
Completeness : Consider the ℓ^{th} dictator cut given by $\mathcal{F}(\boldsymbol{z}^R) = z^{(\ell)}$. With probability $(1-\varepsilon)^2$, the perturbation does not affect the ℓ^{th} coordinate of $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$. In other words, with probability $(1-\varepsilon)^2$, we have $\tilde{z}_j^{(\ell)} = z_j^{(\ell)}$ and $\tilde{z}_i^{(\ell)} = z_i^{(\ell)}$. Hence,

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) \geqslant (1-\varepsilon)^2 \cdot \frac{1}{2} \mathop{\mathbb{E}}_{e} \mathop{\mathbb{E}}_{\boldsymbol{z}_i^R, \boldsymbol{z}_j^R} \left[1 - z_i^{(\ell)} \cdot z_j^{(\ell)}\right]$$

Observe that if the edge $e = (v_i, v_j)$ in G is sampled, then the distribution μ_e is used to generate each coordinates of $\boldsymbol{z}_i^R$ and $\boldsymbol{z}_j^R$. Specifically, this means that the coordinates $z_i^{(\ell)}$ and $z_j^{(\ell)}$ satisfy,

$$\mathop{\mathbb{E}}_{\boldsymbol{z}_i^R, \boldsymbol{z}_j^R} \left[1 - z_i^{(\ell)} \cdot z_j^{(\ell)}\right] = \mathop{\mathbb{E}}_{\mu_e} [1 - z_i^{(\ell)} \cdot z_j^{(\ell)}] = 1 - \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle .$$

Therefore, $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) \geqslant (1-\varepsilon)^2 \cdot \frac{1}{2}\,\mathbb{E}_e\,[1-\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle] \geqslant (1-\varepsilon)^2 \cdot \mathrm{val}(\boldsymbol{V})$.

Soundness : For the sake of analysis, we will construct a graph $\mathcal{G}_{\boldsymbol{V}}$, roughly similar to the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ described earlier.

Gaussian Graph $\mathcal{G}_{\boldsymbol{V}}$

The vertices of $\mathcal{G}_{\boldsymbol{V}}$ are points in $\mathbb{R}^R$. The graph $\mathcal{G}_{\boldsymbol{V}}$ is the union of all random projections of the SDP solution $\boldsymbol{V}$ in to R dimensions. Formally, an edge of $\mathcal{G}_{\boldsymbol{V}}$ can be sampled as follows:

- Sample R random Gaussian vectors $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}$ of the same dimension as the SDP solution $\boldsymbol{V}$.
- Project the SDP vectors $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ along directions $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}$ to obtain a copy of G in a $\mathbb{R}^R$. Formally define
$$\boldsymbol{g}_i^R = (\langle \boldsymbol{v}_i, \boldsymbol{\zeta}^{(1)}\rangle, \ldots, \langle \boldsymbol{v}_i, \boldsymbol{\zeta}^{(R)}\rangle)\,.$$
- Sample an edge $e = (v_i, v_j)$ in G, and output the corresponding edge $(\boldsymbol{g}_i^R, \boldsymbol{g}_j^R)$ in $\mathbb{R}^R$

As lengths of vectors are approximately preserved under random projections, most of the vectors are $\{\boldsymbol{g}_i^R\}$ are roughly unit vectors. Hence, the Gaussian graph $\mathcal{G}_{\boldsymbol{V}}$ is a slightly fudged version of the sphere graph $\mathcal{S}_{\boldsymbol{V}}$ described earlier.

As the graph $\mathcal{G}_{\boldsymbol{V}}$ consists of a union of several isomorphic copies of G, the following claim is an immediate consequence.

Claim 6.3.1. $\mathrm{opt}(\mathcal{G}_{\boldsymbol{V}}) \leqslant \mathrm{opt}(G)$.

Let us suppose that $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$ is a (τ, ε)-quasirandom function. For the sake of succinctness, let us denote $\mathcal{H} = T_{1-\varepsilon}\mathcal{F}$. Essentially, $\mathcal{H}(\boldsymbol{z}^R)$ is the expected value returned by $\mathcal{F}$ on querying a perturbation of the input $\boldsymbol{z}^R$. Thus the function $\mathcal{H}$ is a *smooth* version of $\mathcal{F}$, obtained by averaging the values of $\mathcal{F}$.

Now we will extend the cut $\mathcal{F}$ from the hypercube graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ to the Gaussian graph $\mathcal{G}_{\boldsymbol{V}}$. To this end, we write the functions $\mathcal{H}, \mathcal{F}$ as a multilinear polynomials in the coordinates of $\boldsymbol{z}^R = (z^{(1)}, \ldots, z^{(R)})$. In particular, the Fourier expansion of $\mathcal{F}$ and $\mathcal{H}$ yields the intended multilinear polynomials.

$$F(\boldsymbol{x}) = \sum_{\sigma} \hat{\mathcal{F}}_\sigma \prod_{i\in\sigma} x^{(i)} \qquad \text{and} \quad H(\boldsymbol{x}) = \sum_{\sigma} (1-\varepsilon)^{|\sigma|} \hat{\mathcal{F}}_\sigma \prod_{i\in\sigma} x^{(i)}\,.$$

The polynomials F and H yield natural extensions of the cut functions $\mathcal{F}$ and $\mathcal{H}$ from $\{\pm 1\}^R$ to $\mathbb{R}^R$. However, unlike the original cut function $\mathcal{F}$, the range of its extension need not be restricted to $\{\pm 1\}$. To ensure that the extension defines a cut of the graph $\mathcal{G}_{\boldsymbol{V}}$, we will *round* the extension in the most natural fashion. Formally, define the rounding function

$f_{[-1,1]}$ as follows:

$$f_{[-1,1]}(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leqslant x \leqslant 1 \\ 1 & \text{if } x > 1\,. \end{cases}$$

The extension H^* of the cut $\mathcal{F}$ to $\mathcal{G}_{\boldsymbol{V}}$ is given by

$$H^*(\boldsymbol{g}^R) = f_{[-1,1]}(H(\boldsymbol{g}^R)) \qquad \text{where } H(\boldsymbol{g}^R) = \sum_{\sigma} (1-\varepsilon)^{|\sigma|} \hat{\mathcal{F}}_\sigma \prod_{j \in \sigma} g^{(j)}$$

Let $\mathrm{val}(H^*)$ denote the value of the cut H^* of the graph $\mathcal{G}_{\boldsymbol{V}}$. Now we will show the following claim.

Claim 6.3.2. *There exists an absolute constant $K > 0$ such that for a (τ, ε)-quasirandom function $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$,*

$$\mathrm{val}(H^*) = \mathit{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) \pm \tau^{K\varepsilon}\,.$$

By definition of $\mathrm{opt}(\mathcal{G}_{\boldsymbol{V}})$, we have $\mathrm{val}(H^*) \leqslant \mathrm{opt}(\mathcal{G}_{\boldsymbol{V}})$. Along with Claim 6.3.1, this implies that $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \leqslant \mathrm{opt}(G) + \tau^{K\varepsilon}$, completing the proof of Theorem 6.3.

Proof of Claim 6.3.2. Returning to the definition of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$, notice that the random variable $\tilde{\boldsymbol{z}}^R_i$ depends only on $\boldsymbol{z}^R_i$. Thus, the value of a cut $\mathcal{F} : \{\pm 1\}^R \to \{\pm 1\}$ can be rewritten as,

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) = \frac{1}{2} \mathop{\mathbb{E}}_{e} \mathop{\mathbb{E}}_{\boldsymbol{z}^R_i, \boldsymbol{z}^R_j} \Big[1 - \mathbb{E}[\mathcal{F}(\tilde{\boldsymbol{z}}^R_i) | \boldsymbol{z}^R_i] \cdot \mathbb{E}[\mathcal{F}(\tilde{\boldsymbol{z}}^R_j) | \boldsymbol{z}^R_j]\Big]$$

By the definition of the noise operator $T_{1-\varepsilon}$,$T_{1-\varepsilon}\mathcal{F}(\boldsymbol{z}^R) = \mathbb{E}_{\tilde{z}^R}[\mathcal{F}(\tilde{\boldsymbol{z}}^R)|\boldsymbol{z}^R]$. Hence $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ can be rewritten as

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) = \frac{1}{2} \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{z}^R_i, \boldsymbol{z}^R_j} \Big[1 - \mathcal{H}(\boldsymbol{z}^R_i) \cdot \mathcal{H}(\boldsymbol{z}^R_j)\Big] = \frac{1}{2} \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{z}^R_i, \boldsymbol{z}^R_j} \Big[1 - H(\boldsymbol{z}^R_i) \cdot H(\boldsymbol{z}^R_j)\Big]$$

By definition of the Gaussian graph $\mathcal{G}_{\boldsymbol{V}}$, we have

$$\mathrm{val}(H^*) = \frac{1}{2} \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{g}^R_i, \boldsymbol{g}^R_j} \Big[1 - H^*(\boldsymbol{g}^R_i) \cdot H^*(\boldsymbol{g}^R_j)\Big]$$

Firstly, let us denote by $P : [-1,1]^2 \to [-1,1]$ the function given by $P(x,y) = 1 - xy$. Let us restrict our attention to a particular edge $e = (v_1, v_2)$. For this edge, we will show that

$$\mathop{\mathbb{E}}_{\boldsymbol{z}^R_1, \boldsymbol{z}^R_2} \big[P(H(\boldsymbol{z}^R_1), H(\boldsymbol{z}^R_2))\big] = \mathop{\mathbb{E}}_{\boldsymbol{g}^R_1, \boldsymbol{g}^R_2} \big[P\left(H^*(\boldsymbol{g}^R_1), H^*(\boldsymbol{g}^R_2)\right)\big] \pm \tau^{K\varepsilon} \tag{6.8}$$

By averaging the above equality over all edges e in the graph G, the claim follows. We will use the invariance principle to show the above claim.

Here is a statement of the invariance principle (see Section 3.6) tailored to the application at hand.

Theorem 6.4 (Invariance Principle [125]). *Let $\boldsymbol{z} = \{z_1, z_2\}$ and $\mathbf{G} = \{g_1, g_2\}$ be two sets of random variables such that:*

$$\mathbb{E}[z_i] = \mathbb{E}[g_i] = 0 \qquad \mathbb{E}[z_i^2] = \mathbb{E}[g_i^2] = 1 \qquad \text{for all } i \in [2]$$

and $\mathbb{E}[z_1 z_2] = \mathbb{E}[g_1 g_2]$. Let $\boldsymbol{z}^R, \mathbf{G}^R$ denote R independent copies of the random variables $\boldsymbol{z}$ and $\mathbf{G}$.

There is an absolute constant $K > 0$, such that for all $\tau, \varepsilon > 0$ the following holds:

If F be a multilinear polynomial given by $F(\boldsymbol{x}) = \sum_\sigma \hat{F}_\sigma \prod_{i\in\sigma} x^{(i)}$, and if $H(\boldsymbol{x}) = T_{1-\varepsilon}F(\boldsymbol{x}) = \sum_\sigma (1-\varepsilon)^{|\sigma|}\hat{F}_\sigma \prod_{i\in\sigma} x^{(i)}$ be such that $\mathrm{Inf}_\ell(H) \leqslant \tau$ for all $\ell \in [R]$ then,

1. *For every function $\Psi : \mathbb{R}^2 \to \mathbb{R}$ that is thrice differentiable with all its partial derivatives up to order 3 bounded uniformly by C_0,*

$$\left| \mathbb{E}\left[\Psi(H(\boldsymbol{z}_1^R), H(\boldsymbol{z}_2^R))\right] - \mathbb{E}\left[\Psi(H(\boldsymbol{g}_1^R), H(\boldsymbol{g}_2^R))\right] \right| \leqslant \tau^{K\varepsilon}$$

2. *Define the function $\xi : \mathbb{R}^2 \to \mathbb{R}$ as $\xi(\boldsymbol{x}) = \sum_{i\in[2]} (x_i - f_{[-1,1]}(x_i))^2$ Then, we have $\left| \mathbb{E}[\xi(H(\boldsymbol{z}_1^n), H(\boldsymbol{z}_2^n))] - \mathbb{E}[\xi(H(\boldsymbol{g}_1^n), H(\boldsymbol{g}_2^n))] \right| \leqslant \tau^{K\varepsilon}$*

By design, for each edge $e = (v_i, v_j)$ the pairs of random variables $\{z_i, z_j\}$ and $\{g_i, g_j\}$ satisfy,

$$\begin{aligned} \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}[g_i] = \mathop{\mathbb{E}}_{\mu_e}[z_i] = 0 \qquad & \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}[g_i^2] = \mathop{\mathbb{E}}_{\mu_e}[z_i^2] = 1 \\ \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}[g_j] = \mathop{\mathbb{E}}_{\mu_e}[z_j] = 0 \qquad & \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}[g_j^2] = \mathop{\mathbb{E}}_{\mu_e}[z_j^2] = 1 \\ \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}[g_i g_j] = \mathop{\mathbb{E}}_{\mu_e}[z_i z_j] &= \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle . \end{aligned}$$

The predicate/payoff is currently defined as $P(x, y) = 1 - xy$ in the domain $[-1, 1]^2$. Notice that the function $P(x, y) = 1 - xy$ by itself does not have uniformly bounded derivatives in $\mathbb{R}^2$. Extend the payoff P to a smooth function over the entire space $\mathbb{R}^2$, with all its partial derivatives up to order 3 bounded uniformly throughout $\mathbb{R}^2$. Further, it is easy to ensure that the extension satisfies the following Lipschitz condition for some large enough constant $C > 0$,

$$|P(x, y) - P(x', y')| \leqslant C(|x - x'| + |y - y'|) \qquad \forall (x, y), (x', y') \in \mathbb{R}^2 . \tag{6.9}$$

We will prove Equation 6.8 in two steps.

Step I : Apply the Invariance Principle with the ensembles $\boldsymbol{z} = \{z_1, z_2\}$ and $\mathbf{G} = \{g_1, g_2\}$, for the vector of multilinear polynomials $\mathbf{H}$ and the smooth function $\Psi = P$. This yields,

$$\mathop{\mathbb{E}}_{\boldsymbol{z}_1^R, \boldsymbol{z}_2^R}\left[P(H(\boldsymbol{z}_1^R), H(\boldsymbol{z}_2^R))\right] = \mathop{\mathbb{E}}_{\boldsymbol{g}_1^R, \boldsymbol{g}_2^R}\left[P(H(\boldsymbol{g}_1^R), H(\boldsymbol{g}_2^R))\right] \pm \tau^{K\varepsilon} \tag{6.10}$$

Step II : In this step, we bound the effect of the rounding operation used in extending the cut $\mathcal{F}$ from $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ to $\mathcal{G}_{\boldsymbol{V}}$.

As $\mathcal{F}$ is a cut of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$, its range is $\{\pm 1\}$. Hence, the corresponding polynomial F takes $\{\pm 1\}$ values on inputs from $\{\pm 1\}^R$. As $H = T_{1-\varepsilon}F$ is an average of the values of F, the values $H(\boldsymbol{z}_1^R)$ and $H(\boldsymbol{z}_2^R)$ are always in the range $[-1, 1]$.

By the invariance principle, the random variable $(H(\boldsymbol{z}_1^R), H(\boldsymbol{z}_2^R))$ has approximately the same behaviour as $(H(\boldsymbol{g}_1^R), H(\boldsymbol{g}_2^R))$. Roughly speaking, this implies that the values $H(\boldsymbol{g}_1^R), H(\boldsymbol{g}_2^R)$ are also nearly always in the range $[-1, 1]$. Hence intuitively, the rounding operation must have little effect on the value of the cut.

This intuition is formalized by the second claim in the invariance principle. The function ξ measures the squared deviation from the range $[-1, 1]$. For random variables $(\boldsymbol{z}_1^R, \boldsymbol{z}_2^R)$, clearly we have $\mathbb{E}[\xi(H(\boldsymbol{z}_1^R), H(\boldsymbol{z}_2^R))] = 0$. By the invariance principle applied to polynomial H we get,

$$\mathbb{E}[\xi(H(\boldsymbol{g}_1^n), H(\boldsymbol{g}_2^n))] \leqslant \mathbb{E}[\xi(H(\boldsymbol{z}_1^n), H(\boldsymbol{z}_2^n))] + \tau^{K\varepsilon} = 0 + \tau^{K\varepsilon} = \tau^{K\varepsilon} \tag{6.11}$$

Using the Lipschitz condition satisfied by the payoff, we can write:

$$\begin{aligned}
&\Big| \mathop{\mathbb{E}}_{\boldsymbol{g}_1^R, \boldsymbol{g}_2^R} \big[P(H^*(\boldsymbol{g}_1^R), H^*(\boldsymbol{g}_2^R))\big] - \mathop{\mathbb{E}}_{\boldsymbol{g}_1^R, \boldsymbol{g}_2^R} \big[P(H(\boldsymbol{g}_1^R), H(\boldsymbol{g}_2^R))\big] \Big| \\
&\quad \leqslant C \mathop{\mathbb{E}}_{\boldsymbol{g}_1^R, \boldsymbol{g}_2^R} \Big[\big|H^*(\boldsymbol{g}_1^R) - H(\boldsymbol{g}_1^R)\big| + \big|H^*(\boldsymbol{g}_2^R) - H(\boldsymbol{g}_2^R)\big| \Big] \\
&\quad \leqslant C \Big(2 \mathop{\mathbb{E}}_{\boldsymbol{g}_1^R, \boldsymbol{g}_2^R} \Big[\big|H^*(\boldsymbol{g}_1^R) - H(\boldsymbol{g}_1^R)\big|^2 + \big|H^*(\boldsymbol{g}_2^R) - H(\boldsymbol{g}_2^R)\big|^2 \Big] \Big)^{1/2} && \text{by Cauchy-Schwartz ineq} \\
&\quad \leqslant C \Big(2 \mathop{\mathbb{E}}_{\boldsymbol{g}_1^R, \boldsymbol{g}_2^R} \Big[\xi(H(\boldsymbol{g}_1^R), H(\boldsymbol{g}_2^R)) \Big] \Big)^{1/2} && \text{(by Definition of } \xi) \\
&\quad \leqslant 2C\tau^{K\varepsilon} \leqslant \tau^{K'\varepsilon} && \text{(by Equation 6.11)}
\end{aligned}$$

Along with Equation 6.10, the above inequality implies Equation 6.8. This finishes the proof of Claim 6.3.2.

■

6.6 Dictatorship Tests and Rounding Schemes

The proof of soundness in Section 6.4 can be translated into an efficient rounding scheme. Specifically, given a cut $\mathcal{F}$ of the graph $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$, let H^* denote its extension to the Gaussian graph $\mathcal{G}_{\boldsymbol{V}}$. The idea of the rounding scheme $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$ is to sample a random copy of the graph G inside the Gaussian graph $\mathcal{G}_{\boldsymbol{V}}$ and output the cut induced by H^* on the copy. The details of the rounding scheme are described in Figure 6.6.

Let $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}(\boldsymbol{V})$ denote the expected value of the cut returned by the above rounding scheme on an SDP solution $\boldsymbol{V}$. Then we can write,

$$\mathsf{Round}^{\varepsilon}_{\mathcal{F}}(\boldsymbol{V}) = \frac{1}{2} \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{g}_i^R, \boldsymbol{g}_j^R} \Big[1 - H^*(\boldsymbol{g}_i^R) \cdot H^*(\boldsymbol{g}_j^R)\Big] = \frac{1}{2} \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{g}_i^R, \boldsymbol{g}_j^R} \Big[P(H^*(\boldsymbol{g}_i^R), H^*(\boldsymbol{g}_j^R))\Big].$$

$\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$
Input: SDP solution $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ for the GW SDP relaxation of the graph G.

- Sample R random Gaussian vectors $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}$ of the same dimension as the SDP solution $\boldsymbol{V}$.
- Project the SDP vectors $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ along directions $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}$. Let
$$\boldsymbol{g}_i^R = (\langle \boldsymbol{v}_i, \boldsymbol{\zeta}^{(1)} \rangle, \ldots, \langle \boldsymbol{v}_i, \boldsymbol{\zeta}^{(R)} \rangle)\,.$$
- Compute $H^*(\boldsymbol{g}_i^R)$ for all $i \in [n]$ as follows:
$$H^*(\boldsymbol{g}_i^R) = f_{[-1,1]}(H(\boldsymbol{g}_i^R)) \quad \text{where } H(\boldsymbol{g}_i^R) = T_{1-\varepsilon}F(\boldsymbol{g}_i^R) = \sum_{\sigma}(1-\varepsilon)^{|\sigma|}\hat{\mathcal{F}}_\sigma \prod_{j\in\sigma} g_i^{(j)}\,.$$
- Assign vertex v_i the value 1 with probability $(1 + H^*(\boldsymbol{g}_i^R))/2$ and -1 with the remaining probability.

Figure 6.5: Rounding Scheme $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$

Here P denotes the smooth extension of the payoff function from $[-1,1]^2$ to $\mathbb{R}^2$. The following is an immediate consequence of Claim 6.3.2,

Theorem 6.5. *There is a constant $K > 0$ such that for a (τ, ε)-quasirandom function $\mathcal{F} : \{\pm 1\}^R \to [-1, 1]$,*
$$\mathsf{Round}^{\varepsilon}_{\mathcal{F}}(\boldsymbol{V}) = \mathit{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}) \pm \tau^{K\varepsilon}$$

On one hand, the above theorem exposes an interesting duality between rounding schemes and dictatorship tests.

It is clearly desirable to execute the scheme $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$ for a (τ, ε)-quasirandom function $\mathcal{F}$ that maximizes the acceptance probability against $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$. However, it could be difficult to explcitly find a (τ, ε)-quasirandom function $\mathcal{F}$ that maximizes this acceptance probability for an SDP solution $\boldsymbol{V}$.

The crucial insight is that the size of the domain 2^R is an absolute constant independent of the SDP solution $\boldsymbol{V}$. Furthermore, the performance of the rounding scheme $\mathsf{Round}_{\mathcal{F}}(\boldsymbol{V})$ is a continuous function of $\mathcal{F}$. Formally, we have the following lemma:

Lemma 6.5.1. *There exists an absolute constant C_0 such that if $\mathcal{F}, \mathcal{F}' : \{\pm 1\} \to [-1, 1]$ are two functions and let $\boldsymbol{V}$ be a GW SDP relaxation for a graph G. Then,*
$$|\mathsf{Round}^{\varepsilon}_{\mathcal{F}}(\boldsymbol{V}) - \mathsf{Round}^{\varepsilon}_{\mathcal{F}'}(\boldsymbol{V})| \leqslant 2C_0\|\mathcal{F} - \mathcal{F}'\|_2$$

Proof. Starting with the expression for $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$, we can rewrite

$$
\begin{aligned}
|\mathsf{Round}^{\varepsilon}_{\mathcal{F}} - \mathsf{Round}^{\varepsilon}_{\mathcal{F}'}| &= \Big| \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{g}_i^R,\boldsymbol{g}_j^R} \Big[P(H^*(\boldsymbol{g}_i^R), H^*(\boldsymbol{g}_j^R)) - P(H'^*(\boldsymbol{g}_i^R), H'^*(\boldsymbol{g}_j^R)) \Big] \Big| \\
&\leqslant C_0 \mathop{\mathbb{E}}_{e=(v_i,v_j)} \mathop{\mathbb{E}}_{\boldsymbol{g}_i^R,\boldsymbol{g}_j^R} \Big[|H^*(\boldsymbol{g}_i^R) - H'^*(\boldsymbol{g}_i^R)| + |H^*(\boldsymbol{g}_j^R)) - H'^*(\boldsymbol{g}_j^R)| \Big] \\
&\qquad\qquad \text{(by the Lipschitz condition for payoff} P \\
&\leqslant C_0 \mathop{\mathbb{E}}_{e=(v_i,v_j)} \Big[\big(\mathop{\mathbb{E}}_{\boldsymbol{g}_i^R} |H^*(\boldsymbol{g}_i^R) - H'^*(\boldsymbol{g}_i^R)|^2 \big)^{1/2} + \big(\mathop{\mathbb{E}}_{\boldsymbol{g}_j^R} |H^*(\boldsymbol{g}_j^R) - H'^*(\boldsymbol{g}_j^R)|^2 \big)^{1/2} \Big] \\
&\qquad\qquad \text{(by Cauchy Schwartz inequality)} \\
&\leqslant 2C_0 \|H^* - H'^*\|_2
\end{aligned}
$$

To finish the proof of the claim, observe that

$$\|H^* - H'^*\|_2 \leqslant \|H - H'\|_2 \leqslant \|\mathcal{F} - \mathcal{F}'\|_2 \,,$$

where the two inequalities use the fact that the operators $f_{[-1,1]}$ and $T_{1-\varepsilon}$ are contractive. ■

Using the continuity of $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$, the space of functions can be discretized, and searched by bruteforce. Every function $\mathcal{F} : \{\pm 1\}^R \to [-1,1]$ can be thought of as a point in $\mathbb{R}^{2^d}$. The natural metric between two functions given by $\|\mathcal{F} - \mathcal{F}'\|_2 = \mathbb{E}_{\boldsymbol{x}}[(\mathcal{F}(\boldsymbol{x}) - \mathcal{F}'(\boldsymbol{x}))^2]^{1/2}$ is a scaled version of the natural ℓ_2 metric on $\mathbb{R}^{2^d}$. Thus to discretize the space of functions, it is sufficient to pick a κ-net for the unit ball in $\mathbb{R}^{2^d}$. The formal definition of a κ-net is as follows:

Definition 6.6.1. Let $B(\mathbf{0},1)$ denote the unit ball in the d-dimensional space $\mathbb{R}^d$. A κ-net for the unit ball $B(\mathbf{0},1)$ is a finite set N of points in $B(\mathbf{0},1)$ such that for every $\boldsymbol{x} \in B(\mathbf{0},1)$ there exists a point $y \in N$ such that $\|\boldsymbol{x} - \boldsymbol{y}\|_2 \leqslant \varepsilon$.

$\mathsf{Round}^{\varepsilon}_{\kappa}$ scheme
Input : A feasible solution to the GW SDP relaxation.
Let $S_\kappa = \{\mathcal{F}_1, \cdots, \mathcal{F}_M\}$ be a set of functions such that for every $\mathcal{F} : \{\pm 1\}^R \to [-1,1]$ there exists $\mathcal{F}_i \in S_\kappa$ satisfying $\|\mathcal{F}_i - \mathcal{F}\|_2 \leqslant \kappa$.

- For each function $\mathcal{F}_i \in S_\kappa$, run the subroutine $\mathsf{Round}^{\varepsilon}_{\mathcal{F}_i}$ on the SDP solution
- Output the assignment obtained with the largest objective value.

As an immediate consequence of Theorem 6.5 and Lemma 6.5.1 we get the following theorem.

Theorem 6.6. *For every $\eta, \varepsilon > 0$, there exists choices of κ, τ such that*

$$\mathsf{Round}^{\varepsilon}_{\kappa}(\boldsymbol{V}) \geqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathit{DICT}^{\varepsilon}_{\boldsymbol{V}}) - \eta$$

Proof. Recall that $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}})$ was defined as

$$\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) = \max_{\substack{\mathcal{F}^*:\{\pm1\}^R\to[-1,1]\\ \mathcal{F}^* \text{ is } (\tau,\varepsilon)-\text{quasirandom}}} \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F})\,.$$

Let $\mathcal{F}^* : \{\pm1\}^R \to [-1,1]$ be the function for which the maximum is achieved in the above definition. Let $\mathcal{F}_j \in S_\kappa$ denote the closest point to $\mathcal{F}^*$ in the κ-net S_κ, i.e., $\|\mathcal{F}^* - \mathcal{F}_j\|_2 \leqslant \kappa$.

Since $\mathcal{F}^*$ is (τ,ε)-quasirandom, by Theorem 6.5 we have

$$\mathsf{Round}^{\varepsilon}_{\mathcal{F}^*}(\boldsymbol{V}) = \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}(\mathcal{F}^*) \pm \tau^{K\varepsilon} = \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \pm \tau^{K\varepsilon}$$

By the continuity of $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$ (Lemma 6.5.1), we have $|\mathsf{Round}^{\varepsilon}_{\mathcal{F}^*}(\boldsymbol{V}) - \mathsf{Round}^{\varepsilon}_{\mathcal{F}_j}(\boldsymbol{V})| \leqslant 2C_0\|\mathcal{F}^* - \mathcal{F}_j\|_2 = 2C_0\kappa$. Consequently we obtain the conclusion of the theorem.

$$\mathsf{Round}^{\varepsilon}_{\kappa}(\boldsymbol{V}) \geqslant \mathsf{Round}^{\varepsilon}_{\mathcal{F}_j}(\boldsymbol{V}) \geqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) - \tau^{K\varepsilon} - 2C_0\kappa \geqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) - \eta$$

■

Comparison with Half space Rounding The Goemans-Williamson algorithm [65] for Max Cut uses a halfspace rounding wherein a single random projection $\{g_i | g_i = \langle \boldsymbol{v}_i, \boldsymbol{\zeta}\rangle, i \in [n]\}$ of the SDP solution $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ is sampled, and for each $i \in [n]$, the vertex v_i is assigned $\operatorname{sgn}(g_i)$.

Equivalently, since the sum of several Gaussian random variables is also Gaussian, the halfspace rounding can be rephrased in the following manner: follows:

- Project the SDP solution $\boldsymbol{V} = \{\boldsymbol{v}_1, \ldots, v_n\}$ along R random Gaussian directions $\{\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}\}$ to obtain

$$\boldsymbol{g}^R_i = (\langle \boldsymbol{v}_i, \boldsymbol{\zeta}^{(1)}\rangle, \ldots, \langle \boldsymbol{v}_i, \boldsymbol{\zeta}^{(R)}\rangle) \qquad \text{for each } i \in [n]$$

- For each $i \in [n]$, assign $\operatorname{sgn}(g^{(1)}_i + g^{(2)}_i + \ldots, g^{(R)}_i)$ to vertex v_i

For every $\tau, \varepsilon > 0$, with a large enough choice of R, the function $\mathcal{F}^*(\boldsymbol{x}) = \operatorname{sgn}(x_1 + \ldots + x_R)$ is (τ,ε)-quasirandom. During the bruteforce search, the $\mathsf{Round}^{\varepsilon}_{\kappa}$ scheme will iterate over a function $\mathcal{F}$ that is close to $\mathcal{F}^*$. Therefore, the rounding scheme $\mathsf{Round}^{\varepsilon}_{\kappa}$ achieves an approximation that is at least as good as that of the Goemans-Williamson algorithm.

The design of a rounding scheme such as the Goemans-Williamson halfspace rounding often requires ingenuity, and knowledge about the nature of the CSP involved. By yielding a constant sized search space of rounding schemes Theorem 6.5 removes the need for ingenuity, thus making it more amenable to generalization for arbitrary CSPs. On the flip-side, specific rounding schemes such as the halfspace rounding are vastly more efficient than the generic rounding scheme outlined here, although both are polynomial time algorithms.

Comparison with the generic rounding scheme in Chapter 5 The $\mathsf{Round}^{\varepsilon}_{\kappa}$ scheme is arguably the same as the generic rounding scheme for CSPs presented in Chapter 5.

To see this, let us pick an η-net for the R-dimensional sphere for sufficiently small η, and subdivide the sphere into Voronoi cells for the η-net. Being a smooth extension of the cut $\mathcal{F}$, the value of H^* is roughly constant on each of the cells. A brute force search over the space of all functions $\mathcal{F}$ amounts to a search over all possible assignments to the cells. The $\mathsf{Round}^{\varepsilon}_{\kappa}$ algorithm randomly projects the SDP vectors along R directions, and each vertex v_i is assigned the value $H^*(\boldsymbol{g}_i^R)$. In particular, all vertices whose projections fall into the same Voronoi cell, will get assigned the same value.

Therefore, the execution of the $\mathsf{Round}^{\varepsilon}_{\kappa}$ scheme can be rephrased as follows: Project the SDP vectors along R random directions and merge vertices whose projections fall in to the same Voronoi cell. Finally, perform a brute force search over the constant sized graph that is obtained after merging the vertices. Indeed, this is exactly how the rounding scheme presented in Chapter 5 proceeds towards rounding the SDP solution.

6.7 From UG-hardness to Integrality Gaps

A UG-hardness result for a problem Λ almost always yields an SDP integrality gap instance for Λ. Clearly, a UG-hardness result for Λ, involves a polynomial time reduction Red from Unique Games to the problem Λ. Specifically, given a UG instance Φ, the reduction Red produces an instance $\mathsf{Red}(\Phi)$ such that,

$$\mathrm{opt}(\Phi) \geqslant 1-\gamma \Longrightarrow \mathrm{opt}(\mathsf{Red}(\Phi)) \geqslant c \qquad \text{and} \qquad \mathrm{opt}(\Phi) \leqslant \delta \Longrightarrow \mathrm{opt}(\mathsf{Red}(\Phi)) \leqslant s\,.$$

To obtain an SDP integrality gap using the reduction, one starts with a SDP integrality gap instance Φ for Unique Games. Formally, let Φ be an instance of Unique Games with $\mathrm{sdp}(\Phi) \geqslant 1-\gamma$ while $\mathrm{opt}(\Phi) \leqslant \delta$. Consider the instance $\Im$ of problem Λ produced by executing the UG-hardness reduction Red on the instance Φ. As $\mathrm{opt}(\Phi) \leqslant \delta$ clearly we have $\mathrm{opt}(\Im) \leqslant s$. Surprisingly, the SDP solution for Φ can be composed with the reduction in order to demonstrate that $\mathrm{sdp}(\Im) \geqslant c$. Thus a c vs s-UG hardness result yields an integrality gap instance where the SDP value is c, while the integral optimum is at most s.

In this section, we will demonstrate the above proof technique for our running example of the Max Cut problem. We will show the following theorem.

Theorem 6.7. *For all $\gamma, \delta, \tau, \varepsilon > 0$, given a dictatorship test DICT over $\{\pm 1\}^R$ for $R \geqslant (\frac{1}{\delta})^{\frac{100}{\gamma}}$, there exists a* Max Cut *instance Ψ such that* $\mathrm{sdp}(\Psi) \geqslant (1-\gamma)\mathsf{Completeness}(\mathsf{DICT})$ *and* $\mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}) + \frac{\delta}{\varepsilon^2\tau^3}$.

We begin by recalling the properties of a SDP integrality gap instance for Unique Games.

Definition 6.7.1. A *weak gap* instance Φ for Unique Games consists of $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi, [R])$, and a set of SDP vectors $\mathcal{B} = \{B_v\}_{v\in\mathcal{V}_\Phi}$ where $B_v = \{\boldsymbol{b}_{v,i} | i \in [R]\}$. The SDP vectors $\mathcal{B}$ form a feasible solution for the following SDP relaxation.

$$
\begin{aligned}
&\text{Maximize} && \mathrm{val}(\mathcal{B}) = \mathop{\mathbb{E}}_{(v,w)\in E} \sum_{\ell\in[R]} \langle \boldsymbol{b}_{v,\pi_{v\leftarrow w}(\ell)}, \boldsymbol{b}_{w,\ell}\rangle && \text{(UG)}\\
&\text{Subject to} && \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'}\rangle \geqslant 0,\ \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v,\ell'}\rangle = 0 && \forall v, v' \in \mathcal{V}_\Phi, \ell, \ell' \in [R]\\
& && \sum_{\ell\in[R]} |\boldsymbol{b}_{v,\ell}|^2 = 1 && \forall v \in \mathcal{V}_\Phi\\
& && \Big|\sum_{\ell\in[R]} \boldsymbol{b}_{v,\ell} - \sum_{\ell\in[R]} \boldsymbol{b}_{v',\ell}\Big|^2 = 0 && \forall v, v' \in \mathcal{V}_\Phi
\end{aligned}
$$

The SDP vectors $\mathcal{B}$ can be assumed to satisfy the following additional properties:

– There exists a unit vector $\boldsymbol{b}_0$ such that for each $v \in \mathcal{V}_\Phi$, $\sum_{\ell\in[R]} \boldsymbol{b}_{v,\ell} = \boldsymbol{b}_0$ and $\langle \boldsymbol{b}_0, \boldsymbol{b}_{v,\ell}\rangle = |\boldsymbol{b}_{v,\ell}|^2$ for all $\ell \in [R]$. Observe that $\sum_{\ell,\ell'} \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'}\rangle = \langle \boldsymbol{b}_0, \boldsymbol{b}_0\rangle = 1$.

– The collection $\mathcal{B}$ of orthonormal sets is a good SDP solution for Φ, in that $\mathrm{val}(\mathcal{B}) \geqslant 1-\gamma$. Furthermore, it can be assumed that

$$
\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi} \mathop{\mathbb{E}}_{v,v'\in N(w)} \Big(\sum_{\ell\in[R]} \langle \boldsymbol{b}_{v,\pi_{v\leftarrow w}(\ell)}, \boldsymbol{b}_{v',\pi_{v'\leftarrow w}(\ell)}\rangle \Big) \geqslant 1-\gamma. \tag{6.12}
$$

– (**Strong Matching Property**) For every pair of vertices $u, u' \in \mathcal{V}_\Phi \cup \mathcal{W}_\Phi$, the sets B_u and $B_{u'}$ satisfy the following *strong matching property*: There exist R disjoint matchings between $B_u, B_{u'}$ given by bijections $\pi^{(1)}, \ldots, \pi^{(R)} : B_u \to B_{u'}$ such that for all $i \in [R], \boldsymbol{b}, \boldsymbol{b}' \in B_u$, we have $\langle \boldsymbol{b}, \pi^{(i)}(\boldsymbol{b})\rangle = \langle \boldsymbol{b}', \pi^{(i)}(\boldsymbol{b}')\rangle$.

– For every edge $e = (w, v) \in E$, the vector sets B_w and B_v have significant correlation under the permutation $\pi = \pi_{v\leftarrow w}$. Specifically,

$$
\forall \ell \in [R]. \quad \langle \boldsymbol{b}_{w,\ell}, \boldsymbol{b}_{v,\pi(\ell)}\rangle^2 \geqslant 0.99.
$$

Let Φ denote a weak gap instance for Unique Games with $\mathrm{sdp}(\Phi) \geqslant 1-\gamma$ and $\mathrm{opt}(\Phi) \leqslant \delta$. By the work of Khot and Vishnoi [104], there exists a weak gap instance for Unique Games over an alphabet size $R \geqslant (\frac{1}{\delta})^{\frac{100}{\gamma}}$. Execute the reduction to Max Cut outlined in Section 6.3 starting with the instance Φ. We claim that the resulting Max Cut instance Ψ is an SDP integrality gap for Max Cut.

Since $\mathrm{opt}(\Phi) \leqslant \delta$, it immediately follows from Theorem 6.2 that $\mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^\varepsilon) + \delta/\varepsilon^2\tau^3$. All that is left to finish the proof of Theorem 6.7, is to show that $\mathrm{sdp}(\Psi) \geqslant (1-\gamma)\mathsf{Completeness}(\mathsf{DICT}^\varepsilon)$. Towards this goal, we will use the SDP vectors for Φ to exhibit an SDP solution for the GW relaxation of Max Cut for the graph Ψ.

The idea behind the construction of SDP vectors is pretty simple. Let us pretend for the moment that the SDP solution $\mathcal{B}$ for the unique games instance Φ is *integral*. Specifically,

let us suppose that the SDP solution $\mathcal{B}$ consists of one-dimensional vectors that are either 0 or 1. Thus for a vertex $u \in \mathcal{W}_\Phi \cup \mathcal{V}_\Phi$, exactly one of the vectors $\{\boldsymbol{b}_{v,\ell}\}_{\ell \in [R]}$ is equal to 1 while the remaining are identically zero vectors. Furthermore if $\boldsymbol{b}_{v,\ell} = 1$ then it is interpreted as assigning the label ℓ to the vertex v.

The vertices of the MAX CUT instance Ψ consists of a long code for every vertex $v \in \mathcal{V}_\Phi$, i.e., $\mathcal{V}_\Psi = \mathcal{V}_\Phi \times \{\pm 1\}^R$. Consider a vertex $(v, \boldsymbol{x})$ in the graph Ψ. The reduction encodes the choice of a label ℓ to a UNIQUE GAMES vertex v, as the ℓ^{th} dictator cut for the set of vertices $\{v\} \times \{\pm 1\}^R$. Therefore, if the vertex v is assigned label ℓ, then $(v, \boldsymbol{x})$ is to be assigned $x^{(\ell)}$. Consequently, the value assigned to vertex $(v, \boldsymbol{x})$ is given by

$$\boldsymbol{V}_{(v,\boldsymbol{x})} = \sum_{\ell \in [R]} x^{(\ell)} \boldsymbol{b}_{v,\ell} \,.$$

Clearly, if the SDP vectors $\boldsymbol{b}_{v,\ell}$ are integral, then the solution $\boldsymbol{V}_{(v,\boldsymbol{x})}$ is the intended solution for the MAX CUT instance Ψ. More generally, the SDP vectors $\{\boldsymbol{b}_{v,\ell}\}$ are to be thought of as random variables $\{b_{v,\ell}\}$ that take *integral* values. In other words, each $b_{v,\ell}$ is a random variable that takes values 0 or 1, and satisfies

$$\mathbb{E}[b_{v,\ell} b_{v',\ell'}] = \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'} \rangle$$

Equivalently, the SDP vectors $\mathcal{B}$ correspond to a probability distribution over labellings to the UNIQUE GAMES instance Φ.

Consider a vertex w and two of its neighbors v, v'. Since $\mathcal{B}$ is an SDP solution with high objective value (6.12), the probability distribution over labellings almost always assigns labels that satisfy the edges (w, v) and (w, v'). With high probability, the distribution over labellings assigns $w \to \ell$, $v \to \pi_{v \leftarrow w}(\ell)$ and $v' \to \pi_{v' \leftarrow w}(\ell)$ for some label $\ell \in [R]$. Equivalently, with high probability we have $b_{w,\ell} = b_{v,\pi_{v \leftarrow w}(\ell)} = b_{v',\pi_{v' \leftarrow w}(\ell)} = 1$ for some label ℓ. In turn, the long codes corresponding to v and v' are assigned $\pi_{v \to w}(\ell)^{\text{th}}$ and $\pi_{v' \to w}(\ell')^{\text{th}}$ dictator cuts respectively. On assigning such matching dictator cuts, at least $\mathsf{Completeness}(\mathsf{DICT}^\varepsilon)$ fraction of edges between the long codes of v and v' are cut. As this happens with high probability, the fraction of edges cut is roughly $\mathsf{Completeness}(\mathsf{DICT}^\varepsilon)$.

While the above argument outlines the intuition behind the SDP value, the formal proof is a fairly easy calculation. To check the feasibility of the SDP solution $\{\boldsymbol{V}_{(v,\boldsymbol{x})} | v \in \mathcal{V}_\Phi, \boldsymbol{x} \in \{\pm 1\}^R\}$, all that is required is that the vectors $\boldsymbol{V}_{v,\boldsymbol{x}}$ are unit vectors. Indeed the vectors $\boldsymbol{V}_{v,\boldsymbol{x}}$ are unit vectors as shown below:

$$\begin{aligned}
\langle \boldsymbol{V}_{(v,\boldsymbol{x})}, \boldsymbol{V}_{(v,\boldsymbol{x})} \rangle &= \sum_{\ell,\ell' \in [R]} x^{(\ell)} x^{(\ell')} \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v,\ell'} \rangle \\
&= \sum_{\ell \in [R]} (x^{(\ell)})^2 \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v,\ell} \rangle \qquad (textsince \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v,\ell'} \rangle = 0 \qquad \text{for all } \ell \neq \ell') \\
&= \sum_{\ell \in [R]} |\boldsymbol{b}_{v,\ell}|^2 = 1
\end{aligned}$$

Now we shall turn to the analysis of the value of SDP solution.

$$\mathrm{val}(\{\boldsymbol{V}_{v,\boldsymbol{x}}\}) = \frac{1}{2} \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{v,v' \in N(w)} \mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - \langle \boldsymbol{V}_{v,\pi_{w \leftarrow v} \circ \boldsymbol{z}}, \boldsymbol{V}_{v',\pi_{w \leftarrow v'} \circ \boldsymbol{z}'} \rangle\right].$$

Let us denote $\boldsymbol{x} = \pi_{w \leftarrow v} \circ \boldsymbol{z}$ and $\boldsymbol{y} = \pi_{w \leftarrow v'} \circ \boldsymbol{z}'$. Rewrite the inner expectation as,

$$\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - \langle \boldsymbol{V}_{v,\pi_{w \leftarrow v} \circ \boldsymbol{z}}, \boldsymbol{V}_{v',\pi_{w \leftarrow v'} \circ \boldsymbol{z}'} \rangle\right] \quad = \quad \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[1 - \langle \boldsymbol{V}_{v,\boldsymbol{x}}, \boldsymbol{V}_{v',\boldsymbol{x}'} \rangle\right]$$

Using $\boldsymbol{V}_{v,\boldsymbol{x}} = \sum_\ell x^{(\ell)} \boldsymbol{b}_{v,\ell}$ and $\sum_{\ell,\ell' \in [R]} \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'} \rangle = \langle \boldsymbol{b}_0, \boldsymbol{b}_0 \rangle = 1$, we can rewrite

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - \langle \boldsymbol{V}_{v,\pi_{w \leftarrow v} \circ \boldsymbol{z}}, \boldsymbol{V}_{v',\pi_{w \leftarrow v'} \circ \boldsymbol{z}'} \rangle\right] &= \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \Big[\sum_{\ell,\ell' \in [R]} \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'} \rangle - \sum_{\ell,\ell' \in [R]} x^{(\ell)} y^{(\ell')} \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'} \rangle \Big] \\
&= \sum_{\ell,\ell' \in [R]} \langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'} \rangle \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[1 - x^{(\ell)} y^{(\ell')}\right].
\end{aligned}$$

Observe that $\langle \boldsymbol{b}_{v,\ell}, \boldsymbol{b}_{v',\ell'} \rangle \geqslant 0$ and $\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[1 - x^{(\ell)} y^{(\ell')}] \geqslant 0$ for all ℓ, ℓ'. Dropping all terms other than terms of the form $\pi_{v \leftarrow w}(\ell), \pi_{v' \leftarrow w}(\ell)$ for $\ell \in [R]$, we get:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - \langle \boldsymbol{V}_{v,\pi_{w \leftarrow v} \circ \boldsymbol{z}}, \boldsymbol{V}_{v',\pi_{w \leftarrow v'} \circ \boldsymbol{z}'} \rangle\right] \geqslant \\
&\qquad \sum_{\ell \in [R]} \langle \boldsymbol{b}_{v,\pi_{v \leftarrow w}(\ell)}, \boldsymbol{b}_{v',\pi_{v' \leftarrow w}(\ell)} \rangle \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left[1 - x^{(\pi_{v \leftarrow w}(\ell))} y^{(\pi_{v' \leftarrow w}(\ell'))}\right].
\end{aligned}$$

By definition of $\boldsymbol{x}$ and $\boldsymbol{y}$, we have $x^{(\pi_{v \leftarrow w}(\ell))} = z^{(\ell)}$ and $y^{(\pi_{v' \leftarrow w}(\ell))} = z'^{(\ell)}$. Consequently,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - \langle \boldsymbol{V}_{v,\pi_{w \leftarrow v} \circ \boldsymbol{z}}, \boldsymbol{V}_{v',\pi_{w \leftarrow v'} \circ \boldsymbol{z}'} \rangle\right] &\geqslant \sum_{\ell \in [R]} \langle \boldsymbol{b}_{v,\pi_{v \leftarrow w}(\ell)}, \boldsymbol{b}_{v',\pi_{v' \leftarrow w}(\ell)} \rangle \mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - z^{(\ell)} z'^{(\ell)}\right] \\
&\geqslant \Big(\sum_{\ell \in [R]} \langle \boldsymbol{b}_{v,\pi_{v \leftarrow w}(\ell)}, \boldsymbol{b}_{v',\pi_{v' \leftarrow w}(\ell)} \rangle \Big) \cdot \Big(2 \cdot \mathsf{Completeness}(\mathsf{DICT}^\varepsilon) \Big).
\end{aligned}$$

Substituting this back in the expression for $\mathrm{val}(\boldsymbol{V}_{v,x})$, we obtain the desired conclusion.

$$\begin{aligned}
\mathrm{val}(\{\boldsymbol{V}_{v,x}\}) &= \frac{1}{2} \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{v,v' \in N(w)} \mathop{\mathbb{E}}_{\boldsymbol{z},\boldsymbol{z}'} \left[1 - \langle \boldsymbol{V}_{v,\pi_{w \leftarrow v} \circ \boldsymbol{z}}, \boldsymbol{V}_{v',\pi_{w \leftarrow v'} \circ \boldsymbol{z}'} \rangle\right] \\
&\geqslant \mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{v,v' \in N(w)} \Big(\sum_{\ell \in [R]} \langle \boldsymbol{b}_{v,\pi_{v \leftarrow w}(\ell)}, \boldsymbol{b}_{v',\pi_{v' \leftarrow w}(\ell)} \rangle \Big) \cdot \mathsf{Completeness}(\mathsf{DICT}^\varepsilon) \\
&\geqslant (1 - \gamma) \mathsf{Completeness}(\mathsf{DICT}^\varepsilon) \qquad \text{by 6.12}
\end{aligned}$$

6.8 Implications

In this section, we study some of the implications of the connections exposed in this chapter. Let $\mathsf{Gap}_{\textsc{Max Cut}}$, $\mathsf{UGhard}_{\textsc{Max Cut}}$ denote the SDP integrality gap and UG-hardness curves for the Max Cut problem.

Parameters We will choose parameters $R, \tau, \varepsilon, \kappa, \gamma, \delta$ such that all the reductions outlined in this chapter incur an additive error of at most η for some $\eta > 0$. With this in mind, first set $\varepsilon = \frac{\eta}{10}$ and fix τ such that the error $C\tau^{K\varepsilon}$ in Theorem 6.3 is at most $\frac{\eta}{10}$. A value of $\tau = 2^{-O(\frac{\log \eta}{\eta})}$ suffices for this purpose. Choose a value of $\kappa = O(\eta)$ such that the error in Theorem 6.6 is at most $\frac{\eta}{10}$.

Set $\gamma = \frac{\eta}{10}$ and $\delta = \frac{\eta\varepsilon^2\tau^3}{10} = 2^{-O(\frac{\log \eta}{\eta})}$ to ensure that the reduction in Theorem 6.2 incurs an error of at most $\frac{\eta}{10}$ in both the completeness and soundness cases. Finally, fix the value of the alphabet size of Unique Games R as $2^{O(1/\eta^3)}$. This value is large enough to ensure that there exists a weak gap instance of Unique Games with SDP value $1-\gamma$ and optimum value δ.

Optimal UG-hardness Let us compose the conversion from SDP integrality gaps to dictatorship tests (Theorem 6.3) with the reduction from dictatorship tests to UG-hardness result (Theorem 6.1). As an immediate consequence, we obtain a UG-hardness result for Max Cut that matches the SDP integrality gap. Formally,

Theorem 6.8. *For all $\eta > 0$ and $\frac{1}{2} \leqslant c \leqslant 1$, it is UG-hard to distinguish between* Max Cut *instances with value at least $c - \eta$ from those with value $\mathsf{Gap}_{\text{Max Cut}}(c)$, i.e.,*

$$\mathsf{UGhard}_{\text{Max Cut}}(c - \eta) \leqslant \mathsf{Gap}_{\text{Max Cut}}(c) + \eta$$

Proof. Let G be a graph such that $\mathrm{sdp}(G) = c$ while $\mathrm{opt}(G) \leqslant \mathsf{Gap}_{\text{Max Cut}}(c) + \frac{\eta}{4}$. The existence of such a graph is guaranteed by the definition of the SDP integrality gap curve $\mathsf{Gap}_{\text{Max Cut}}$. Apply Theorem 6.3 for the graph G with its optimal SDP solution and the above defined values of τ, ε. The claim follows by using the resulting dictatorship test in UG-hardness reduction (Theorem 6.2). ■

Optimal Rounding Scheme The following theorem shows that the rounding scheme $\mathsf{Round}^{\varepsilon}_{\kappa}$ achieves the integrality gap of the semidefinite program. Let $\mathsf{RoundingCurve}(c)$ denote the rounding curve associated with the $\mathsf{Round}^{\varepsilon}_{\kappa}$ scheme.

Theorem 6.9. *For all $\eta > 0$, there exists choices of ε, κ such that*

$$\mathsf{RoundingCurve}(c) \geqslant \mathsf{Gap}_{\text{Max Cut}}(c - \eta) - \eta$$

Proof. The choices of κ, ε as a function of η are as outlined earlier in this section.

By definition of $\mathsf{RoundingCurve}$, there exists an instance G and an SDP solution $\boldsymbol{V}$ for G such that $\mathrm{val}(\boldsymbol{V}) \geqslant c$ while $\mathsf{Round}^{\varepsilon}_{\kappa}(\boldsymbol{V}) \leqslant \mathsf{RoundingCurve}(c) + \frac{\eta}{2}$. Consider the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ associated with SDP solution $\boldsymbol{V}$. By Theorem 6.3, we have $\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \geqslant \mathrm{val}(\boldsymbol{V}) - 2\varepsilon \geqslant c - \frac{\eta}{2}$. Further by Theorem 6.6 $\mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \leqslant \mathsf{Round}^{\varepsilon}_{\kappa}(\boldsymbol{V}) + \frac{\eta}{4} \leqslant \mathsf{RoundingCurve}(c) + \frac{3\eta}{4}$.

Now we can use the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}$ in Theorem 6.7 to obtain a SDP integrality gap. Therefore by Theorem 6.7 we obtain a SDP integrality gap Ψ for Max Cut such that

$$\mathrm{sdp}(\Psi) \geqslant (1 - \gamma)\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) \geqslant c - \eta\,,$$

while,

$$\mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}}) + \delta/\varepsilon^2\tau^3 \leqslant \mathsf{RoundingCurve}(c) + \eta\,.$$

Thus Ψ is an instance of Max Cut with SDP value $c-\eta$ and optimum value at most $\mathsf{RoundingCurve}(c)+\eta$. By definition of the $\mathsf{Gap}_{\text{Max Cut}}$ curve, the optimum value of Ψ is at least $\mathsf{Gap}_{\text{Max Cut}}(c-\eta)$. As an immediate consequence one gets $\mathsf{RoundingCurve}(c) \geqslant \mathsf{Gap}_{\text{Max Cut}}(c-\eta)-\eta$. ■

The following corollary follows immediately from the above two Theorems.

Corollary 6.9.1. *The GW relaxation along with the rounding scheme* $\mathsf{Round}^{\varepsilon}_{\kappa}$ *form an approximation algorithm for* Max Cut *whose approximation curve* $\alpha(c)$ *satisfies for all* $\eta > 0$*:*

$$\alpha(c) \geqslant \mathsf{Gap}_{\text{Max Cut}}(c-\eta) - \eta \geqslant \mathsf{UGhard}_{\text{Max Cut}}(c-2\eta) - 2\eta$$

Roughly speaking, the GW relaxation along with the $\mathsf{Round}^{\varepsilon}_{\kappa}$ yield an algorithm that achieves the optimal approximation under UGC. The important subtlety involved is the continuity of the curves $\mathsf{Gap}_{\text{Max Cut}}$ and $\mathsf{UGhard}_{\text{Max Cut}}$.

The continuity of these curves is not an issue if one is solely interested in the worst case approximation ratio over all c. In particular, if we define $\mathsf{GapRatio}_{\text{Max Cut}}$ and $\mathsf{UGhardThreshold}_{\text{Max Cut}}$ as

$$\mathsf{GapRatio}_{\text{Max Cut}} = \inf_{c} \frac{\mathsf{Gap}_{\text{Max Cut}}(c)}{c} \qquad \mathsf{UGhardThreshold}_{\text{Max Cut}} = \inf_{c} \frac{\mathsf{UGhard}_{\text{Max Cut}}(c)}{c}\,,$$

then, we have the following corollary.

Corollary 6.9.2. $\mathsf{GapRatio}_{\text{Max Cut}} = \mathsf{UGhardThreshold}_{\text{Max Cut}}$ *and further the algorithm consisting of GW relaxation along with the* $\mathsf{Round}^{\varepsilon}_{\kappa}$ *rounding scheme achieves an approximation* $\mathsf{GapRatio}_{\text{Max Cut}} - \eta$ *for all* η*.*

The $\mathsf{Gap}_{\text{Max Cut}}$ and $\mathsf{UGhard}_{\text{Max Cut}}$ were shown to be continuous by O'Donnell and Wu (Corollary 5.4, [132]). Therefore, the above theorems yield matching UG-hardness results and approximation algorithms for Max Cut. While all the above stated theorems and corollaries generalize to arbitrary CSPs, the continuity of the curves involved does not hold for arbitrary CSPs.

Computing Integrality Gaps In this section, we have obtained a UG-hardness result and a matching approximation algorithm for Max Cut. However, the results are implicit in that they do not shed light on the value of the approximation ratio, or the SDP integrality gap. Now we will see how the connection between dictatorship tests and SDP integrality gaps can be harnessed towards computing the curves $\mathsf{Gap}_{\text{Max Cut}}$ and $\mathsf{UGhard}_{\text{Max Cut}}$.

In Theorem 6.3 and Theorem 6.7, we have effectively established an equivalence between SDP integrality gaps and dictatorship tests over a large constant dimensional hypercube. The integrality gap curve $\mathsf{Gap}_{\text{Max Cut}}(c)$ is the worst case value of the optimum over all instances with SDP value c. As there are infinitely many instances of all sizes with SDP value c, it is unclear how $\mathsf{Gap}_{\text{Max Cut}}(c)$ can be computed in finite time.

The crucial observation is that the set of all dictatorship tests over a constant dimensional hypercube is a compact set, that can be easily discretized. Recall that a dictatorship test over $\{\pm 1\}^R$ is nothing but a weighted graph over $\{\pm 1\}^R$, whose edge weights sum up to 1. Hence, the space of all dictatorship tests can be identified with probability distributions over $\{\pm 1\}^R \times \{\pm 1\}^R$ - a compact set.

Define $\mathsf{Soundness}_{\tau,\varepsilon}(c)$ as follows:

$$\mathsf{Soundness}_{\tau,\varepsilon}(c) = \inf_{\substack{\mathsf{DICT}-\text{ a dictatorship test over } \{\pm 1\}^R \\ \mathsf{Completeness}(\mathsf{DICT})=c}} \mathsf{Soundness}_{\tau,\varepsilon}(\mathsf{DICT})$$

By Theorem 6.7, there exists a Max Cut instance Ψ with $\mathrm{sdp}(\Psi) \geqslant (1-\gamma)c \geqslant c - \eta$ and $\mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\tau,\varepsilon}(c) + \eta$. Therefore we have

$$\mathsf{Soundness}_{\tau,\varepsilon}(c) \geqslant \mathsf{Gap}_{\text{Max Cut}}(c-\eta) - \eta\,.$$

Furthermore by Theorem 6.3, we have

$$\mathsf{Soundness}_{\tau,\varepsilon}(c) \leqslant \mathsf{Gap}_{\text{Max Cut}}(c+\eta) + \eta$$

Observe that for any $\eta > 0$, by iterating over a sufficiently fine κ-net over the space of dictatorship tests, the value $\mathsf{Soundness}_{\tau,\varepsilon}(c)$ can be computed within an accuracy of η.

Corollary 6.9.3. *There exists an algorithm that for any $\eta > 0$ and $\frac{1}{2} < c < 1$, runs in time $\exp(\exp(\exp(O(1/\eta^3))))$ and computes a real number θ such that*

$$\mathsf{Gap}_{\text{Max Cut}}(c-\eta) - 2\eta \leqslant \theta \leqslant \mathsf{Gap}_{\text{Max Cut}}(c+\eta) + 2\eta\,.$$

In particular, the algorithm can be used to estimate $\mathsf{GapRatio}_{\text{Max Cut}}$ to any desired accuracy η. Moreover, the continuity of the curve $\mathsf{Gap}_{\text{Max Cut}}$ ensures that θ is indeed a good approximation for $\mathsf{Gap}_{\text{Max Cut}}(c)$ too.

Chapter 7

GENERAL CONSTRAINT SATISFACTION PROBLEMS

The connections between SDP integrality gaps, Dictatorship tests and Unique Games hardness results were explored in Chapter 6. In this chapter, we will generalize these connections and their implications to the class of generalized constraint satisfaction problems.

7.1 Results

In the next few sections, we generalize the reductions between SDP integrality gaps, dictatorship tests and UG-hardness results outlined in Chapter 6 to arbitrary GCSPs. As in the case of Max Cut (Chapter 6), several interesting results emerge as implications of these connections.

To state these implications let us recall some definitions. Recall that a generalized constraint satisfaction problem (GCSP) is a the natural generalization of CSPs where we allow both positive and negative payoff functions (See Definition 2.4.1. For a GCSP Λ, Gap_Λ and UGhard_Λ denote the associated SDP integrality gap curve and the UG-hardness curve. Since, the value of any assignment to a GCSP instance lies in the range $[-1, 1]$, the curves Gap_Λ and UGhard_Λ are defined in the range $[-1, 1]$. $\mathsf{GapRatio}_\Lambda$ and $\mathsf{UGhardThreshold}_\Lambda$ are much coarser measures of approximation that are defined as,

$$\mathsf{GapRatio}_\Lambda = \inf_c \frac{\mathsf{Gap}_\Lambda(c)}{c} \qquad \mathsf{UGhardThreshold}_\Lambda = \inf_c \frac{\mathsf{UGhard}_\Lambda(c)}{c}\,.$$

First, using the conversion from SDP integrality gaps and UG hardness results via dictatorship tests, we will show that the best approximation to every GCSP problem Λ is given by LC relaxation. The formal statement of the result is as follows:

Theorem 7.1. *For a* GCSP Λ*, for all* $\eta > 0$ *and* $-1 < c \leqslant 1$*, it is UG-hard to distinguish between instances of* Λ *with value at least* $c - \eta$ *from those with value* $\mathsf{Gap}_\Lambda(c)$*, i.e.,*

$$\mathsf{UGhard}_\Lambda(c - \eta) \leqslant \mathsf{Gap}_\Lambda(c) + \eta$$

The above theorem immediately implies a relation between the coarser measures $\mathsf{GapRatio}_\Lambda$ and $\mathsf{UGhardThreshold}_\Lambda$.

Corollary 7.1.1. *Let* Λ *be a* GCSP *such that* $\mathsf{GapRatio}_\Lambda$ *and* $\mathsf{UGhardThreshold}_\Lambda$ *are both finite. Then,*

$$\mathsf{GapRatio}_\Lambda = \mathsf{UGhardThreshold}_\Lambda$$

Qualitatively, the result shows that if UGC is true, then LC is the strongest SDP for every GCSP. Thus if UGC is true, then stronger SDPs obtained through the Lovasz-Schriver, Lasserre and Sherali-Adams hierarchies do not yield better approximation ratios for any GCSP. The proof of the reduction from integrality gaps to dictatorship tests yields a rounding scheme for the GCSP Λ.

Theorem 7.2. *For all* $\eta > 0$*, there exists a rounding scheme* Round_η *running in time* $\exp(\exp(\exp(O(1/\eta^3)))) \cdot \mathrm{poly}(n)$ *such that, if* $\mathsf{RoundingCurve}(c)$ *denotes the rounding curve associated with the* Round_η *scheme then,*

$$\mathsf{RoundingCurve}(c) \geqslant \mathsf{Gap}_\Lambda(c - \eta) - \eta \qquad \forall c \in (-1, 1]$$

The following corollary is a restatement of the above result as an approximation algorithm for Λ.

Corollary 7.2.1. *For all $\eta > 0$, the LC relaxation along with the rounding scheme Round_η form an approximation algorithm for Λ whose approximation curve $\alpha(c)$ satisfies:*

$$\alpha(c) \geqslant \mathsf{Gap}_\Lambda(c-\eta) - \eta \geqslant \mathsf{UGhard}_\Lambda(c-2\eta) - 2\eta$$

Roughly speaking, for every GCSP, the Round_η scheme achieves the integrality gap of the LC relaxation, and is optimal under **UGC**, at every value of c. The caveat is that the curves Gap_Λ and UGhard_Λ need not be continuous, and therefore $\mathsf{Gap}_\Lambda(c-\eta)$ could be vastly different from $\mathsf{Gap}_\Lambda(c)$.

In many special cases of interest, the continuity of the curves Gap_Λ and UGhard_Λ is not an issue. For instance, let us suppose Λ is a constraint satisfaction problem (CSP). Recall that CSPs are a special case of GCSPs where the payoff functions take $\{0,1\}$ values. For GCSPs with positive payoffs, the optimum of every instance is a number strictly bounded away from 0. In other words, the curves Gap_Λ and UGhard_Λ are defined in a range $[\theta, 1]$ for some $\theta > 0$. Let us further restrict our attention to approximation ratios, instead of approximation curves that are more refined measures.

An immediate consequence of Theorem 7.2, the Round_η algorithm achieves an approximation ratio that is within η of the integrality gap and the optimal possible under **UGC** for every constraint satisfaction problem.

Corollary 7.2.2. *Let Λ be a constraint satisfaction problem or more generally a* GCSP *with positive payoffs. For every $\eta > 0$, the algorithm consisting of LC relaxation along with the Round_η rounding scheme runs in $\exp(\exp(\exp(O(1/\eta^3)))) \cdot \mathrm{poly}(n)$ time and achieves an approximation $\mathsf{GapRatio}_\Lambda - \eta = \mathsf{UGhardThreshold}_\Lambda - \eta$.*

The above results generalize a large number of algorithmic and UG-hardness results in literature. However, the results are implicit in that they do not shed light on the value of the approximation ratio or the UG-hardness threshold. The actual values of approximation ration or the SDP integrality gap are a function of the predicates involved in the GCSP Λ. It would be rather surprising if general black box reductions such as those presented here can explicitly determine these quantities. On a positive note, the black box reductions do yield an algorithm to compute these quantities for a GCSP.

Theorem 7.3. *There exists an algorithm that for any $\eta > 0$ and $-1 < c < 1$, runs in time $\exp(\exp(\exp(O(1/\eta^3))))$ and computes a real number θ such that*

$$\mathsf{Gap}_\Lambda(c-\eta) - 2\eta \leqslant \theta \leqslant \mathsf{Gap}_\Lambda(c+\eta) + 2\eta\,.$$

Again, the continuity of the curves involved can be ignored in the following special case.

Corollary 7.3.1. *Let Λ be a constraint satisfaction problem or more generally a* GCSP *with positive payoffs. For every $\eta > 0$, there exists an algorithm that runs in $\exp(\exp(\exp(O(1/\eta^3))))$ time and computes $\mathsf{GapRatio}_\Lambda$ within an additive error of η.*

Organization: The chapter begins by recalling the formal definition of GCSPs, the LC relaxation and the invariance principle in Section 7.2. The reduction from integrality gap to dictatorship tests is presented in Section 7.3. This reduction is among the major contributions of this thesis, and its proof is presented in Section 7.4. In the subsequent section, we present the reduction from dictatorship tests to UG-hardness results which essentially follows from the work of Khot et al. [99]. We demonstrate that the soundness analysis of the reduction from integrality gaps to dictatorship tests (Section 7.3) can be converted into an efficient rounding scheme in Section 7.6. The last leg of the reduction from UG hardness results to SDP integrality gaps is sketched in Section 7.7, a fairly straightforward generalization from [104]. Finally, in Section 7.8 we use the reductions to derive Theorem 7.1, Theorem 7.2 and Theorem 7.3 as immediate implications.
Mathematical Tools: The chapter uses multilinear expansion of functions over product spaces, and the associated notions of influences and noise operators (Section 3.4), Gaussian random variables (Section 3.5) and the invariance principle (Section 3.6).

7.2 Preliminaries

For the sake of convenience, we recall the definition of GCSP s here.

Definition 7.2.1. A Generalized Constraint Satisfaction Problem (GCSP) Λ is specified by a family of *payoff functions* $\Lambda = \{P | P : [q]^k \to [-1, 1]\}$. The integer k is referred to as the arity of the GCSP Λ, while q denotes the domain size.

A payoff function is said to be of *type* Λ if it belongs to the family Λ.

Definition 7.2.2 (Λ-GenerlizedConstraintSatisfactionProblem)**.** (GCSP) of Generalized Constraint Satisfaction Problem Λ is given by $\Im = (\mathcal{V}, \mathcal{P})$ where

- $\mathcal{V}$ is the set of variables that are to be assigned values in $[q]$. For notational convenience, we will associate $\mathcal{V}$ with the set $[N] = \{1, \ldots, N\}$ for $N = |\mathcal{V}|$.
- A function $P' : [q]^{\mathcal{V}} \to [-1, 1]$ is said to be of type Λ, if $P'(y) = P(y_{i_1}, \ldots, y_{i_k})$ for some $P \in \Lambda$ and some $i_1, i_2, \ldots, i_k \in \mathcal{V}$. $\mathcal{P}$ is a probability distribution over a payoffs of type Λ.

The objective is to find an assignment $y \in [q]^{\mathcal{V}}$ to the variables that maximizes the expected payoff denoted by $\mathrm{val}(y)$, i.e.,

$$\mathrm{val}(y) = \mathop{\mathbb{E}}_{P \sim \mathcal{P}} \left[P(y) \right].$$

We define the value $\mathrm{opt}(\Im)$ as

$$\mathrm{opt}(\Im) \stackrel{\text{def}}{=} \max_{y \in [q]^{\mathcal{V}}} \mathrm{val}(y).$$

For a payoff P' of type Λ, let $\mathcal{V}(P') \subseteq \mathcal{V}$ denote the set of variables on which P' depends on. Further, the arity of the GCSP Λ will be denoted by k.

7.2.1 SDP Relaxation

We will now recall the LC relaxation for a Λ-GCSP here for the convenience of the reader. Given an instance $\mathfrak{I} = (\mathcal{V}, \mathcal{P})$, the LC relaxation consists of vectors $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in[q]}$ and a collection $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ of distributions over local assignments and a unit vector $\boldsymbol{b}_0$. Each distribution μ_P is over $[q]^{\mathcal{V}(P)}$ (the set of assignments to the variable set $\mathcal{V}(P)$).

LC Relaxation (Equivalent Version)

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x)$$

$$\text{subject to} \quad \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \mathop{\mathbb{P}\mathrm{r}}_{x\sim\mu_P}\Big\{x_i = a, x_j = b\Big\} \quad P\in\mathrm{supp}(\mathcal{P}),\ \ i,j\in\mathcal{V}(P),\ \ a,b\in[q]\,. \tag{7.1}$$

$$\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_0\rangle = \|\boldsymbol{b}_{i,a}\|_2^2 \qquad \forall i\in\mathcal{V}, a\in[q]\,, \tag{7.2}$$

$$\|\boldsymbol{b}_0\|_2^2 = 1 \tag{7.3}$$

$$\mu_P \in \blacktriangle([q]^{\mathcal{V}(P)})$$

7.2.2 Averaged Functions

A function $\mathsf{F} : [q]^R \to [q]$ is said to be the ith dictator function if $\mathsf{F}(\boldsymbol{x}) = x_i$. In its simplest form, the goal of a dictatorship test is to distinguish dictator functions from functions which do not correlate with any dictators. Formally, given oracle access to a function $\mathsf{F} : [q]^R \to [q]$, a dictatorship test queries the value of F on few random locations and concludes whether F is a dictator or is far from being a dictator.

For the purposes of showing UG-hardness results, it is often necessary to construct a dictatorship test for an average of several functions of the form $\mathsf{F} : [q]^R \to [q]$. In this light, we will construct a dictatorship test for the class of functions from the domain $[q]^R$ to the range $\blacktriangle_q$.

Definition 7.2.3. For every function $\mathsf{F} : [q]^R \to [q]$, the corresponding Δ_q-representation is a function $\boldsymbol{\mathcal{F}} : [q]^R \to \Delta_q$ given by

$$\boldsymbol{\mathcal{F}}(\boldsymbol{x}) = \boldsymbol{e}_{\mathsf{F}(x)}$$

where $\boldsymbol{e}_j$ is the jth basis vector in $\mathbb{R}^q$.

Given a function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ consider the distribution over functions $\boldsymbol{\mathcal{F}}' : [q]^R \to \Delta_q$ given by the following sampling procedure: For each $x \in [q]^R$, set the value of $\boldsymbol{\mathcal{F}}'(x)$ independently as

$$\boldsymbol{\mathcal{F}}'(x) = \boldsymbol{e}_a \qquad \text{with probability } \boldsymbol{\mathcal{F}}(x)_a \qquad \text{for all } a\in[q]\,.$$

It is easy to check that for each $\boldsymbol{x}\in[q]^R$, we have $\boldsymbol{\mathcal{F}}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{\mathcal{F}}'(\boldsymbol{x})]$.

With the above definitions, a function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ is a dictator if $\boldsymbol{\mathcal{F}}(\boldsymbol{x}) = \boldsymbol{e}_{x_i}$ for some $i\in[R]$. On the other hand, a function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ is *far from every dictator* if the

influences of each of the variables is small. The formal definition of this notion is presented in Definition 7.3.3.

We will use the typeface F, G to denote functions from $[q]^R$ to $[q]$, while denoting their corresponding extensions to $\blacktriangle_q$ by $\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}}$.

The domain of the payoff functions P in a GCSP Λ can also be extended from $[q]^k$ to $\blacktriangle_q^k$. First, for a payoff $P : [q]^k \to [-1, 1]$ define the corresponding Δ_q-representation $P' : \Delta_q^k \to [-1, 1]$ as,

$$P'(\boldsymbol{e}_{a_1}, \ldots, \boldsymbol{e}_{a_k}) = P(a_1, \ldots, a_k) \qquad \text{for all } (a_1, \ldots, a_k) \in [q]^k .$$

The function P' can be extended to the domain $\blacktriangle_q^k$ using the multilinear extension. Recall that each of the vectors $\boldsymbol{x}_i \in \blacktriangle_q$ is given by $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,q})$. Define the extension of P' as,

$$P'(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \sum_{\sigma \in [q]^k} P'(\sigma) \prod_{i=1}^{k} x_{i,\sigma_i} \qquad \text{for all } \boldsymbol{x}_1, \ldots \boldsymbol{x}_k \in \blacktriangle_q .$$

For the sake of intuition, let us suppose X_i is a independent random variable taking value $\boldsymbol{e}_a$ with probability $x_{i,a}$ for all $a \in [q]$. Then, by the multilinearity of the extension above,

$$\mathbb{E}[P'(X_1, \ldots, X_k)] = P'(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) ,$$

as expected for an average. Abusing notation, we will use P to denote the original payoff of the GCSP and the corresponding payoff on $\blacktriangle_q^k$.

7.2.3 Invariance Principle

Define functions $f_{[0,1]} : \mathbb{R} \to \mathbb{R}$ and $\xi : \mathbb{R}^q \to \mathbb{R}$ as follows:

$$f_{[0,1]}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leqslant x \leqslant 1 \\ 1 & \text{if } x > 1 \end{cases} \qquad \xi(\boldsymbol{x}) = \sum_{i \in [q]} (x_i - f_{[0,1]}(x_i))^2$$

We recall the invariance principle (Theorem 3.2) here for the reader's convenience.

Theorem 7.4. *(Invariance Principle [124]) Fix $0 < \alpha, \varepsilon \leqslant 1/2$. Let Ω be a finite probability space such that every atom with non-zero probability has probability at least $\alpha \leqslant 1/2$. Let $\mathcal{L} = \{\ell_1, \ell_1, \ldots, \ell_m\}$ be an ensemble of random variables over Ω. Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be an ensemble of Gaussian random variables satisfying the following conditions:*

$$\mathbb{E}[\ell_i] = \mathbb{E}[g_i] \qquad \mathbb{E}[\ell_i^2] = \mathbb{E}[g_i^2] \qquad \mathbb{E}[\ell_i \ell_j] = \mathbb{E}[g_i g_j] \qquad \forall i, j \in [m] .$$

Let $\boldsymbol{F} = (F_1, \ldots, F_d)$ denote a vector valued multilinear polynomial. Let $H_i = T_{1-\varepsilon} F_i$, and $\boldsymbol{H} = (H_1, \ldots, H_d)$. If $\operatorname{Inf}_\ell(\boldsymbol{H}) \leqslant \tau$ and $\mathbf{Var}[H_\ell] \leqslant 1$ for all ℓ, then the following holds

1. *For every function $\Psi : \mathbb{R}^d \to \mathbb{R}$ that is thrice differentiable with all its partial deriva-*

tives up to order 3 *bounded uniformly by* C_0,

$$\left| \mathbb{E}\left[\Psi(\boldsymbol{H}(\mathcal{L}^R))\right] - \mathbb{E}\left[\Psi(\boldsymbol{H}(\mathcal{G}^R))\right] \right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)}$$

where $K = K(d, C_0) > 0$ *is a constant depending on* C_0, d.

2.
$$\left| \mathbb{E}[\xi(\boldsymbol{H}(\mathcal{L}^R))] - \mathbb{E}[\xi(\boldsymbol{H}(\mathcal{G}^R))] \right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)}$$

where $K = K(d, C_0) > 0$ *is a constant depending on* C_0, d.

7.2.4 Smoothing

The reduction from SDP integrality gaps to dictatorship tests, requires the SDP solution to satisfy a certain *smoothness* property defined below. Intuitively, a *smooth* SDP solution satisfies all the inequalities of the relaxation with an extra slack. Roughly speaking, the feasible solution lies in the interior of the polytope. Formally, the *smoothness* property is defined as follows.

Definition 7.2.4. For $\alpha > 0$, a feasible SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ to the LC relaxation of a Λ-GCSP instance $\Im = (\mathcal{V}, \mathcal{P})$ is said to be α-smooth if for every $P \in \mathcal{P}$ and $x \in [q]^{\mathcal{V}(P)}$, we have $\mu_P(x) \geqslant \alpha$.

An arbitrary feasible solution to the LC relaxation can be transformed into a *smooth* solution with a slight loss in the objective value.

Lemma 7.4.1 (Smoothing)**.** *For all* $\frac{1}{q^k} > \alpha > 0$*, given a feasible solution* $(\boldsymbol{V}, \boldsymbol{\mu})$ *for the* LC *relaxation, there exists an* α*-smooth SDP solution* $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$ *such that*

$$\mathrm{val}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) = \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\alpha q^k$$

where k, q *are the arity and domain size of* GCSP Λ.

Proof. Let $(\boldsymbol{V}', \boldsymbol{\mu}')$ denote the SDP solution that corresponds to the uniform distribution over all assignments from $[q]^{\mathcal{V}}$. Recall here that the intended solutions for the LC relaxation correspond to distributions over all assignments $[q]^{\mathcal{V}}$.

By definition of $(\boldsymbol{V}', \boldsymbol{\mu}')$, for each payoff P the local distribution μ'_P is the uniform distribution over $[q]^{\mathcal{V}(P)}$. As the arity of GCSP Λ is k, for each distribution μ'_P and $x \in [q]^{\mathcal{V}(P)}$ we have $\mu'_P(x) \geqslant \frac{1}{q^k}$. Consequently, the SDP solution $(\boldsymbol{V}', \boldsymbol{\mu}')$ is a $\frac{1}{q^k}$-smooth solution.

To make an SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ smooth, the idea is to take a convex combination with the solution $(\boldsymbol{V}', \boldsymbol{\mu}')$ corresponding to the uniform distribution over all assignments. Formally, define the new SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$ as follows:

1. For each $i \in \mathcal{V}, a \in [q]$, define $\boldsymbol{b}^*_{i,a} = \sqrt{(1 - \alpha q^k)}\boldsymbol{b}_{i,a} \oplus \sqrt{\alpha q^k}\boldsymbol{b}'_{i,a}$, where $\oplus$ denotes the direct-sum between the two vectors.

2. For each payoff $P \in \mathcal{P}$, let $\mu^*_P = (1 - \alpha q^k)\mu_P + \alpha q^k \mu'_P$.

It is straightforward to check that inner products of the SDP vectors $\boldsymbol{b}^*_{i,a}$ agree with the local distributions μ^*_P. Furthermore, for every $x \in \mathcal{V}(P)$ of a payoff P, we have $\mu^*_P(x) \geqslant \alpha q^k \cdot \mu'_P(x) \geqslant \alpha$. Finally, the objective value of the SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$ can be bounded as follows:

$$\mathrm{val}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) = (1 - \alpha q^k)\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) + \alpha q^k \mathrm{val}(\boldsymbol{V}', \boldsymbol{\mu}') \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\alpha q^k$$

The final inequality above follows from the fact that for every GCSP all payoffs are within $[-1, 1]$, and hence $\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}), \mathrm{val}(\boldsymbol{V}', \boldsymbol{\mu}') \in [-1, 1]$.

■

7.3 From Integrality Gaps to Dictatorship Tests

In this section, we will exhibit a generic conversion from an arbitrary SDP integrality gap instance $\Im = (\mathcal{V}, \mathcal{P})$ for the Λ-GCSP, to a dictatorship test for functions on $[q]^R$.

7.3.1 Construction of Dictatorship Test

Let $\Im$ be an instance of a GCSP Λ of arity k and domain $[q]$. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote an α-smooth solution for LC relaxation of $\Im$. By Lemma 7.4.1, there exists such an α-smooth SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$, with objective value $\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) \geqslant \mathsf{LC}(\Im) - 2\alpha q^k$

Let $\mathcal{V} = \{1, \ldots, N\}$ denote the variables in the GCSP instance $\Im$. For each $i \in [N]$, $\Omega_i = ([q], \mu_i)$ will refer to a probability space with atoms $[q] = \{1, 2, \ldots, q\}$. In Ω_i, the probability of occurrence of an atom $c \in [q]$ is given by $\mu_i(c) = \|\boldsymbol{b}_{i,c}\|^2$.

Define the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ for functions $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ as follows:

$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ Test

- Sample a payoff $P \sim \mathcal{P}$. Let $\mathcal{V}(P) = \{s_1, \ldots, s_k\}$.
- Sample $\boldsymbol{z}_P = \{\boldsymbol{z}_{s_1}, \ldots, \boldsymbol{z}_{s_k}\}$ from the product distribution μ^R_P, i.e., for each $1 \leqslant j \leqslant n$, $z^{(j)}_P = \{z^{(j)}_{s_1}, \ldots, z^{(j)}_{s_k}\}$ is sampled using the distribution μ_P.
- Perturb $\boldsymbol{z}_P$ by random noise to obtain $\tilde{\boldsymbol{z}}_P$. More precisely, for each $s_i \in \mathcal{V}(P)$ and each $j \in [R]$, sample $\tilde{z}^{(j)}_{s_i}$ as follows,

$$\tilde{z}^{(j)}_{s_i} = \begin{cases} z^{(j)}_{s_i} & \text{with probability } 1 - \varepsilon\,, \\ \text{new sample from } \Omega_{s_i} & \text{with probability } \varepsilon\,. \end{cases}$$

- Query the function values $\boldsymbol{\mathcal{F}}(\tilde{\boldsymbol{z}}_{s_1}), \ldots, \boldsymbol{\mathcal{F}}(\tilde{\boldsymbol{z}}_{s_k})$.
- Return a payoff given by $P\Big(\boldsymbol{\mathcal{F}}\big(\tilde{\boldsymbol{z}}_{s_1}\big), \ldots, \boldsymbol{\mathcal{F}}\big(\tilde{\boldsymbol{z}}_{s_k}\big)\Big)$.

Definition 7.3.1. $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}})$ is the expected payoff returned by the verifier of dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ on the function $\boldsymbol{\mathcal{F}}$ as input.

7.3.2 Completeness of $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$

Lemma 7.4.2.

$$\mathsf{Completeness}(\mathit{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) \geqslant \mathrm{val}(\boldsymbol{V},\boldsymbol{\mu}) - 2\varepsilon k$$

Proof. Consider a dictator function $\boldsymbol{\mathcal{F}}(\boldsymbol{z}) = \boldsymbol{e}_{z^{(j)}}$. The expected payoff returned by the verifier $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}})$ is given by

$$\mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_P}\Big[P\Big(\boldsymbol{\mathcal{F}}\big(\tilde{\boldsymbol{z}}_{s_1}\big),\ldots,\boldsymbol{\mathcal{F}}\big(\tilde{\boldsymbol{z}}_{s_k}\big)\Big)\Big] = \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_P}\Big[P\Big(\tilde{z}^{(j)}_{s_1},\ldots,\tilde{z}^{(j)}_{s_k}\Big)\Big]$$

With probability $(1-\varepsilon)^k$, $\tilde{z}^{(j)}_{s_i} = z^{(j)}_{s_i}$ for each $s_i \in \mathcal{V}(P)$. Further the payoff P takes values in $[-1,1]$, when all their inputs belong to $\blacktriangle_q$. Hence a lower bound for the expected payoff is given by

$$\begin{aligned}\mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_P}\Big[P\Big(\boldsymbol{\mathcal{F}}\big(\tilde{z}_{s_1}\big),\ldots,\boldsymbol{\mathcal{F}}\big(\tilde{z}_{s_k}\big)\Big)\Big] &\geqslant (1-\varepsilon)^k \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\Big[P\Big(z^{(j)}_{s_1},\ldots,z^{(j)}_{s_k}\Big)\Big] + (1-(1-\varepsilon)^k)\cdot(-1)\\ &\geqslant \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\Big[P\Big(z^{(j)}_{s_1},\ldots,z^{(j)}_{s_k}\Big)\Big] - 2\varepsilon k\end{aligned}$$

The j^{th} coordinates $\boldsymbol{z}^{(j)}_S = \{z^{(j)}_{s_1},\ldots,z^{(j)}_{s_k}\}$ are generated from the local probability distribution μ_P. Therefore we get,

$$\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) \geqslant \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\Big[P\Big(z^{(j)}_{s_1},\ldots,z^{(j)}_{s_k}\Big)\Big] - 2\varepsilon k = \mathrm{val}(\boldsymbol{V},\boldsymbol{\mu}) - 2\varepsilon k\,,$$

thus finishing the proof. ∎

7.3.3 Quasi-random functions

To complete the construction of the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$, we need to bound the expected payoff for functions that are quasi-random, i.e., *far from being a dictator*. In this work, we will use a special notion of quasi-randomness that depends on the SDP solution $(\boldsymbol{V},\boldsymbol{\mu})$ to the instance $\mathcal{I}$.

The natural definition of quasi-randomness used in numerous works is to bound the influence of each coordinate. Formally, a function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ is quasi-random if the influence $\mathrm{Inf}_\ell(\boldsymbol{\mathcal{F}})$ is small for all coordinates $\ell \in [R]$. However, an underlying probability measure on the domain $[q]$ is necessary in order to define influences.

For the construction $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$, there is no single natural choice of probability measure. In fact, for each variable i in the original GCSP instance $\mathcal{I}$ there is a corresponding probability distribution over $[q]$ given by $\mathbb{P}\mathrm{r}[i=a] = \|\boldsymbol{b}_{i,a}\|^2$. Therefore, we will define the relative notion of "quasi-random with respect to $(\boldsymbol{V},\boldsymbol{\mu})$". Roughly speaking, we shall call a function "quasi-random" if all its influences are low under nearly all of the probability distributions corresponding to variables $i \in \mathcal{V}$.

Formally, for each $i \in [N]$, let $\Omega_i = ([q], \mu_i)$ denote the probability space with atoms $[q]$ and the probability of the atom $a \in [q]$ given by $\mu_i(a) = \|\boldsymbol{b}_{i,a}\|^2$. Recall that the instance $\mathcal{G}$ has N variables $\mathcal{V} = \{1, \ldots, N\}$.

Let us fix a function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$. For each i, let $\boldsymbol{\mathcal{F}}_i$ denote the function $\boldsymbol{\mathcal{F}}$ interpreted as a function over the product probability space Ω_i^R.

First, we define the notion of being quasi-random with respect to a payoff.

Definition 7.3.2. A function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ is said to (τ, ε)-quasirandom with respect to a payoff $P \in \mathcal{P}$ if for each $s \in \mathcal{V}(P)$ the following holds:

$$\max_{\ell \in [R]} \mathrm{Inf}_\ell(T_{1-\varepsilon} \boldsymbol{\mathcal{F}}_s) \leqslant \tau$$

Extending the notion further, we make the following definition.

Definition 7.3.3. A function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ is said to be $(\beta, \tau, \varepsilon)$-quasirandom with respect to a SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ for a GCSP instance $\mathcal{G} = (\mathcal{V}, \mathcal{P})$ if the following holds: For a choice of payoff $P \sim \mathcal{P}$, with probability at least $1 - \beta$, $\boldsymbol{\mathcal{F}}$ is (τ, ε)-quasirandom with respect to P.

Definition 7.3.4. For $\beta, \tau > 0$, define

$$\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) = \sup_{\substack{\boldsymbol{\mathcal{F}}:[q]^R \to \blacktriangle_q \\ \boldsymbol{\mathcal{F}} \text{ is } (\beta,\tau,\varepsilon)-\text{quasirandom w.r.t.}(\boldsymbol{V},\boldsymbol{\mu})}} \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}})$$

7.4 Soundness of Dictatorship Test $\mathsf{DICT}^{\varepsilon}_{\mathbf{V},\mu}$

7.4.1 Rounding Scheme

In this section, we will present the soundness proof of the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$. To this end, given a function $\mathcal{F} : [q]^R \to \blacktriangle_q$, we will exhibit a randomized rounding scheme $\mathsf{Round}^{\varepsilon}_{\mathcal{F}}$ for the SDP relaxation LC. The details of the rounding scheme are presented in Figure 7.1.

Definition 7.4.1. $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu})$ is the expected value of the assignment output by the rounding scheme $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}$ on the SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$. Here the expectation is over the random choices of the algorithm $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}$.

The following lemma forms the core of the soundness analysis of the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$.

Lemma 7.4.3. *There exists constants $C = C(q,k), K = K(q,k)$ such that for all $\varepsilon, \tau, \alpha, \beta$, if a function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ is $(\beta, \tau, \varepsilon)$ quasirandom with respect to a α-smooth SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ to a* GCSP *instance $\mathcal{G}$ then*

$$\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathit{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) \pm C(\beta + \tau^{K\varepsilon/\log(1/\alpha)})$$

In the rest of this section, we will present the proof of the above lemma. To this end, we first develop some machinery concerning payoff functions, and certain random variables associated with the dictatorship test.

$\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}$ Scheme

Input: An SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ for a GCSP instance $\Im = (\mathcal{V}, \mathcal{P})$.
Setup: For each $s \in \mathcal{V}$, the probability space $\Omega_s = ([q], \mu_s)$ consists of atoms $[q]$ with the distribution $\mu_s(c) = |\boldsymbol{b}_{s,c}|^2$. Let $\boldsymbol{\mathcal{F}}_s$ denote the function obtained by interpreting the function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ as a function over Ω_s^R. Further, let us denote $\boldsymbol{\mathcal{H}}_s = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_s$. Let $\boldsymbol{F}_s(\boldsymbol{x}), \boldsymbol{H}_s(\boldsymbol{x})$ denote the multilinear polynomials corresponding to functions $\boldsymbol{\mathcal{F}}_s, \boldsymbol{\mathcal{H}}_s$ respectively.
Scheme: Sample R vectors $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}$ with same dimension as vectors in $\boldsymbol{V}$, such that each coordinate of $\boldsymbol{\zeta}^{(i)}$ for all i is an i.i.d normal random variable.
For each $s \in \mathcal{V}$ do

- For all $j \in [R]$ and $c \in [q]$, compute the projection $h^{(j)}_{s,c}$ of the vector $\boldsymbol{b}_{s,c}$ as follows:

$$g^{(j)}_{s,c} = \|\boldsymbol{b}_{s,c}\|^2 + \left[\langle(\boldsymbol{b}_{s,c} - \|\boldsymbol{b}_{s,c}\|^2\boldsymbol{b}_0), \boldsymbol{\zeta}^{(j)}\rangle\right]$$

  Let $\boldsymbol{g}^{(j)}_s = (g^{(j)}_{s,1}, \ldots, g^{(j)}_{s,q})$ and $\boldsymbol{g}^R_s = (\boldsymbol{g}^{(1)}_s, \ldots, \boldsymbol{g}^{(R)}_s)$.

- Evaluate the multilinear polynomial $\boldsymbol{H}_s$ with $\boldsymbol{g}^R_s$ as inputs to obtain $\mathbf{p}_s$. Formally, let $\mathbf{p}_s = \boldsymbol{H}_s(\boldsymbol{g}^R_s)$. By Fact 3.0.11, with access to the function $\boldsymbol{\mathcal{F}}$, the computation of $\mathbf{p}_s = \boldsymbol{H}_s(\boldsymbol{g}^R_s)$ can be implemented as follows:

  1. For each $j \in [R]$ and $c \in [q]$, let

$$h^{(j)}_{s,c} = \|\boldsymbol{b}_{s,c}\|^2 + (1-\varepsilon)\left[\langle(\boldsymbol{b}_{s,c} - \|\boldsymbol{b}_{s,c}\|^2\boldsymbol{b}_0), \boldsymbol{\zeta}^{(j)}\rangle\right]$$

  2. Compute $\mathbf{p}_s = (p_{s,1}, \ldots, p_{s,k})$ as follows: $\mathbf{p}_s = \sum_{\sigma \in [q]^R} \boldsymbol{\mathcal{F}}(\sigma) \prod_{j=1}^R h^{(j)}_{s,\sigma_j}$

- Round $\mathbf{p}_s$ to $\mathbf{p}^*_s \in \blacktriangle_q$ using the following procedure.

$$f_{[0,1]}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leqslant x \leqslant 1 \\ 1 & \text{if } x > 1 \end{cases} \qquad \text{Scale}(x_0, x_1, \ldots, x_{q-1}) = \begin{cases} \frac{1}{\sum_i x_i}(x_0, \ldots, x_{q-1}) \text{if } \sum_i x_i \neq 0 \\ (1, 0, 0, \ldots, 0) \text{ if } \sum_i x_i = 0 \end{cases}$$

$$\mathbf{p}^*_s = \text{Scale}(f_{[0,1]}(p_{s,1}), \ldots, f_{[0,1]}(p_{s,q}))$$

- Assign the GCSP variable $s \in \mathcal{V}$ the value $j \in [q]$ with probability $p^*_{s,j}$.

Figure 7.1: Rounding Scheme $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}$ for a function $\boldsymbol{\mathcal{F}}$

Payoff Functions For the sake of the proof, we will extend the payoff functions P in the family Λ to *smooth* real valued functions on $\mathbb{R}^{qk}$.Given a payoff $P : \blacktriangle_q^k \to [-1, 1]$, we extend the function P *smoothly* from $\blacktriangle_q^k$ to $\mathbb{R}^{qk}$. Specifically, the *smooth* extension of P satisfies the following properties for a constant $C_0(q, k)$ depending on q and k.

- (**Property I**) All the partial derivatives of P up to order 3 are uniformly bounded by $C_0(q, k)$.

- (**Property II:**) For all $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}, \{\mathbf{y}_1, \ldots, \mathbf{y}_k\} \in \mathbb{R}^{qk}$,

$$|P(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) - P(\mathbf{y}_1, \ldots, \mathbf{y}_k)| \leqslant C_0(q, k) \sum_{i=1}^{k} \|\boldsymbol{x}_i - \mathbf{y}_i\|_2$$

Local and Global Ensembles Now, we shall describe two ensembles of random variables, namely the local integral ensembles $\mathcal{L}_P$ for each payoff P, and a global Gaussian ensemble $\mathcal{G}$.

Recall that, for each payoff P, μ_P is a distribution over assignments to the GCSP variables in set $\mathcal{V}(P)$.

Definition 7.4.2. Let P be a payoff with $\mathcal{V}(P) = \{s_1, \ldots, s_k\}$. The corresponding *Local Integral Ensemble* $\mathcal{L}_P$ is a set of random variables $\mathcal{L}_P = \{\boldsymbol{\ell}_{s_1}, \ldots, \boldsymbol{\ell}_{s_k}\}$ each taking values in Δ_q generated as follows,

- Sample an assignment to $\mathcal{V}(P) = \{s_1, \ldots, s_k\}$ from the distribution μ_P.
- Set $\boldsymbol{\ell}_{s_i} = \boldsymbol{e}_{s_i}$ for all $i \in [k]$. Here $\boldsymbol{e}_j$ is the jth basis vector in $\mathbb{R}^q$ for all $j \in [q]$.

Definition 7.4.3. The *Global* Ensemble $\mathcal{G} = \{\boldsymbol{g}_s | s \in \mathcal{V}, j \in [q]\}$ are generated by setting $\boldsymbol{g}_s = \{g_{s,1}, \ldots, g_{s,q}\}$ where

$$g_{s,j} = \|\boldsymbol{b}_{s,j}\|^2 + \langle (\boldsymbol{b}_{s,j} - |\boldsymbol{b}_{s,j}|^2 \boldsymbol{b}_0), \boldsymbol{\zeta} \rangle \qquad \forall j \in [q]\,,$$

and $\boldsymbol{\zeta}$ is a normal Gaussian random vector of appropriate dimension.

It is easy to see that the local and global integral ensembles have matching moments up to degree two.

Observation 7.4.1. *For any set $P \in \mathcal{P}$, the global ensemble $\mathcal{G}$ matches the following moments of the local integral ensemble $\mathcal{L}_P$*

$$\mathbb{E}[g_{s,j}] = \mathbb{E}[\ell_{s,j}] \qquad \mathbb{E}[g_{s,j} \cdot g_{s',j'}] = \mathbb{E}[\ell_{s,j} \cdot \ell_{s',j'}] \qquad \forall j, j' \in [q], s, s' \in \mathcal{V}(P)$$

Proof. By definition of the local ensemble $\mathcal{L}_P$, property of the LC relaxation

$$\mathbb{E}[\ell_{s,j}] = \Pr_{\mu_P}[s = j] = \|\boldsymbol{b}_{s,j}\|^2 \qquad \mathbb{E}[\ell_{s,j} \cdot \ell_{s',j'}] = \Pr_{\mu_P}[s = j \wedge s' = j'] = \langle \boldsymbol{b}_{s,j}, \boldsymbol{b}_{s',j'} \rangle \qquad (7.4)$$

For the global ensemble, we can write,

$$\mathbb{E}[g_{s,j}] = \|\boldsymbol{b}_{s,j}\|^2 + \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}\left[\langle\left(\boldsymbol{b}_{s,j} - \|\boldsymbol{b}_{s,j}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle\right] = \|\boldsymbol{b}_{s,j}\|^2. \tag{7.5}$$

Further, we can write

$$\mathbb{E}[g_{s,j} \cdot g_{s',j'}] = \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}\Big[\Big(\|\boldsymbol{b}_{s,j}\|^2 + \langle\left(\boldsymbol{b}_{s,j} - \|\boldsymbol{b}_{s,j}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle\Big)\Big(\|\boldsymbol{b}_{s',j'}\|^2 + \langle\left(\boldsymbol{b}_{s',j'} - \|\boldsymbol{b}_{s',j'}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle\Big)\Big]$$

Using $\mathbb{E}[\langle\left(\boldsymbol{b}_{s,j} - \|\boldsymbol{b}_{s,j}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle] = \mathbb{E}[\langle\left(\boldsymbol{b}_{s',j'} - \|\boldsymbol{b}_{s',j'}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle] = 0$, we can rewrite the above equation as

$$\mathbb{E}[g_{s,j} \cdot g_{s',j'}] = \|\boldsymbol{b}_{s,j}\|^2\|\boldsymbol{b}_{s',j'}\|^2 + \mathop{\mathbb{E}}_{\boldsymbol{\zeta}}\Big[\langle\left(\boldsymbol{b}_{s,j} - \|\boldsymbol{b}_{s,j}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle\langle\left(\boldsymbol{b}_{s',j'} - \|\boldsymbol{b}_{s',j'}\|^2\boldsymbol{b}_0\right), \boldsymbol{\zeta}\rangle\Big].$$

Recall that $\mathbb{E}_{\boldsymbol{\zeta}}[\langle \boldsymbol{u}, \boldsymbol{\zeta}\rangle\langle \boldsymbol{v}, \boldsymbol{\zeta}\rangle] = \langle \boldsymbol{u}, \boldsymbol{v}\rangle$ (Property 3.1) for all vectors $\boldsymbol{u}, \boldsymbol{v}$. Therefore we get,

$$\begin{aligned}
\mathbb{E}[g_{s,j} \cdot g_{s',j'}] &= \|\boldsymbol{b}_{s,j}\|^2\|\boldsymbol{b}_{s',j'}\|^2 + \langle\left(\boldsymbol{b}_{s,j} - \|\boldsymbol{b}_{s,j}\|^2\boldsymbol{b}_0\right), \left(\boldsymbol{b}_{s',j'} - \|\boldsymbol{b}_{s',j'}\|^2\boldsymbol{b}_0\right)\rangle \\
&= \|\boldsymbol{b}_{s,j}\|^2\|\boldsymbol{b}_{s',j'}\|^2 + \langle \boldsymbol{b}_{s,j}, \boldsymbol{b}_{s',j'}\rangle - \|\boldsymbol{b}_{s,j}\|^2\|\boldsymbol{b}_{s',j'}\|^2 - \|\boldsymbol{b}_{s,j}\|^2\|\boldsymbol{b}_{s',j'}\|^2 + \|\boldsymbol{b}_{s,j}\|^2\|\boldsymbol{b}_{s',j'}\|^2 \\
&\qquad\qquad (\text{using } \boldsymbol{b}_0 \cdot \boldsymbol{b}_{i,a} = \|\boldsymbol{b}_{i,a}\|^2 \text{for all } i, a) \\
&= \langle \boldsymbol{b}_{s,j}, \boldsymbol{b}_{s',j'}\rangle
\end{aligned} \tag{7.6}$$

The claim follows from Equations 7.4, 7.5 and 7.6. ■

Finally, we have developed the appropriate definitions and machinery to prove Lemma 7.4.3.

Proof of Lemma 7.4.3. For each i, the probability space Ω_i has the set of atoms $[q]$. Thus the function $\boldsymbol{\mathcal{F}}$ can be interpreted as a function over the domain Ω_i^R, instead of $[q]^R$. We shall use $\boldsymbol{\mathcal{F}}_i$ to denote the function obtained by interpreting the function $\boldsymbol{\mathcal{F}}$ as a function over Ω_i^R. Further, let us denote $\boldsymbol{\mathcal{H}}_i = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_i$. Let $\boldsymbol{F}_i(\boldsymbol{x}), \boldsymbol{H}_i(\boldsymbol{x})$ denote the multilinear polynomials corresponding to functions $\boldsymbol{\mathcal{F}}_i, \boldsymbol{\mathcal{H}}_i$ respectively. Here again, we shall use the standard basis over Ω_i to obtain the multilinear expansion.

The expected payoff returned by the verifier in the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is given by:

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) = \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_P}\Big[P\Big(\boldsymbol{\mathcal{F}}_{s_1}(\tilde{\boldsymbol{z}}_{s_1}), \ldots, \boldsymbol{\mathcal{F}}_{s_k}(\tilde{\boldsymbol{z}}_{s_k})\Big)\Big]$$

Each vector $\boldsymbol{z}_{s_i}$ is independently perturbed to obtain $\tilde{\boldsymbol{z}}_{s_i}$. The payoff functions P are multilinear when restricted to the domain $\blacktriangle_q$. Consequently, we can write

$$\begin{aligned}
\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) &= \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\Big[P\Big(\mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_{s_1}}[\boldsymbol{\mathcal{F}}_{s_1}(\tilde{\boldsymbol{z}}_{s_1})|\boldsymbol{z}_{s_1}], \ldots, \mathop{\mathbb{E}}_{\tilde{\boldsymbol{z}}_{s_1}}[\boldsymbol{\mathcal{F}}_{s_k}(\tilde{\boldsymbol{z}}_{s_k}|\boldsymbol{z}_{s_k}])\Big)\Big] \\
&= \mathop{\mathbb{E}}_{P\in\mathcal{P}}\mathop{\mathbb{E}}_{\boldsymbol{z}_P}\Big[P\Big(\boldsymbol{\mathcal{H}}_{s_1}(\boldsymbol{z}_{s_1}), \ldots, \boldsymbol{\mathcal{H}}_{s_k}(\boldsymbol{z}_{s_k})\Big)\Big]
\end{aligned}$$

The last equality is due to the fact $\mathbb{E}_{\tilde{\boldsymbol{z}}_{s_i}}[\boldsymbol{\mathcal{F}}_{s_i}(\tilde{\boldsymbol{z}}_{s_i})|\boldsymbol{z}_{s_i}] = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_{s_i}(\boldsymbol{z}_{s_i}) = \boldsymbol{\mathcal{H}}_{s_i}(\boldsymbol{z}_{s_i})$. For each $s \in \mathcal{V}(P)$, the coordinates of $\boldsymbol{z}_s$ are generated by the distribution μ_P. Therefore the above

expectation can be written in terms of the polynomials $\boldsymbol{H}_s$ applied integral ensemble $\mathcal{L}_P$. Specifically, we can write

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\boldsymbol{z}_P} \Big[P\Big(\boldsymbol{\mathcal{H}}_{s_1}(\boldsymbol{z}_{s_1}),\ldots,\boldsymbol{\mathcal{H}}_{s_k}(\boldsymbol{z}_{s_k})\Big)\Big] = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathcal{L}_P^R} \Big[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{\ell}^R_{s_1}),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{\ell}^R_{s_k})\Big)\Big] \tag{7.7}$$

Now we shall arithmetize the total payoff of the assignment returned by the rounding scheme $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}$. Note that the random variables $\boldsymbol{g}_i^R$ are nothing but samples from the Global Ensemble $\mathcal{G}$. Let us denote by $\boldsymbol{H}^*_s(\boldsymbol{g}^R_s)$ the rounding of $\boldsymbol{H}_s(\boldsymbol{g}^R_s)$. In other words, $\boldsymbol{H}^*(\boldsymbol{g}^R_s) = \mathbf{p}^*_s$. Clearly, we have

$$\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V},\boldsymbol{\mu}) = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathcal{G}^R} \Big[P\Big(\boldsymbol{H}^*_{s_1}(\boldsymbol{g}^R_{s_1}),\ldots,\boldsymbol{H}^*_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\Big]$$

Call a payoff $P\sim\mathcal{P}$ *good*, if $\boldsymbol{\mathcal{F}}$ is (τ,ε)-quasirandom with respect to P. Fix a *good* payoff P and let $\mathcal{V}(P) = \{s_1,\ldots,s_k\}$. From Lemma 7.4.5, for a good payoff P,

$$\Big|\mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}^*_{s_1}(\boldsymbol{g}^R_{s_1}),\ldots,\boldsymbol{H}^*_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\Big] - \mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{g}^R_{s_1}),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\Big]\Big| \leqslant C\tau^{K\varepsilon/\log(1/\alpha)}$$

Using Lemma 7.4.4,

$$\Big|\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\Big[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{\ell}^R_{s_1}),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{\ell}^R_{s_k})\Big)\Big] - \mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{g}^R_{s_1}),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\Big]\Big| \leqslant \tau^{K\varepsilon/\log(1/\alpha)}$$

Adding the two inequalities, for every *good* payoff P the following holds:

$$\Big|\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\Big[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{\ell}^R_{s_1}),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{\ell}^R_{s_k})\Big)\Big] - \mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}^*_{s_1}(\boldsymbol{g}^R_{s_1}),\ldots,\boldsymbol{H}^*_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\Big]\Big| \leqslant (C+1)\tau^{K\varepsilon/\log(1/\alpha)}$$

Recall that all the payoff functions P are bounded in the range $[-1,1]$. Thus for every payoff $P\in\mathcal{P}$,

$$\Big|\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\Big[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{\ell}^R_{s_1}),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{\ell}^R_{s_k})\Big)\Big] - \mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}^*_{s_1}(\boldsymbol{g}^R_{s_1}),\ldots,\boldsymbol{H}^*_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\Big]\Big| \leqslant 2$$

In particular, the above inequality holds for all payoffs P which are *not good*. Using the previous two inequalities, the lemma follows.

$$\begin{aligned}
&|\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V},\boldsymbol{\mu}) - \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\mathcal{F})| \\
&\qquad\leqslant \Pr_{P\in\mathcal{P}}[P \text{ is good}]\cdot(C+1)\tau^{K\varepsilon/\log(1/\alpha)} + (1-\Pr_{P\in\mathcal{P}}[P \text{ is good}])\cdot 2\,, \\
&\qquad\leqslant (1-\beta)\cdot(C+1)\tau^{K\varepsilon/\log(1/\alpha)} + \beta\cdot 2 \\
&\qquad< (C+1)\tau^{K\varepsilon/\log(1/\alpha)} + 2\beta\,.
\end{aligned}$$

■

Lemma 7.4.4. *If the function $\mathcal{F}$ is (τ,ε)-quasirandom with respect to a payoff P with $\mathcal{V}(P)=\{s_1,\ldots,s_k\}$ then*

$$\left|\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\left[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{\ell}_{s_1}^R),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{\ell}_{s_k}^R)\Big)\right]-\mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{g}_{s_1}^R),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{g}_{s_k}^R)\Big)\right]\right|\leqslant\tau^{K\varepsilon/\log(1/\alpha)}$$

where $K=K(q,k)$ is a constant depending on q,k.

Proof. Without loss of generality we may assume $\mathcal{V}(P)=\{1,2,\ldots,k\}$. Let us denote $\mathcal{G}_P=\{\boldsymbol{g}_1,\ldots,\boldsymbol{g}_k\}$. The idea of the proof is to apply the invariance principle on the vector of multilinear polynomials $(\boldsymbol{H}_1,\ldots,\boldsymbol{H}_k)$ and ensembles $\mathcal{L}_P$ and $\mathcal{G}_P$, with the payoff P as a smooth functional.

To begin with, recall that $\boldsymbol{H}_i$ is a multilinear polynomial representing the function $\boldsymbol{\mathcal{H}}_i=T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_i$. Let $\boldsymbol{F}_i$ be the multilinear polynomial representing function the $\boldsymbol{\mathcal{F}}_i$.

Consider the joint probability space $\Omega_P=(\Omega_1,\Omega_2,\ldots,\Omega_k)$. Note that the different probability spaces Ω_i are not independent of each other. In the terminology of [124], the probability spaces Ω_i are coarsenings of Ω_P.

The probability space Ω_P consists of atoms $[q]^{\mathcal{V}(P)}=[q]^k$ and the distribution μ_P. As $(\boldsymbol{V},\boldsymbol{\mu})$ is an α-smooth SDP solution, the smallest probability of an atom in Ω_P is at least α.

By definition, the ensemble $\mathcal{L}_P=\{\boldsymbol{\ell}_1,\ldots,\boldsymbol{\ell}_k\}$ is an ensemble of random variables over the probability space Ω_P. By Observation 7.4.1, the ensemble $\mathcal{G}_P$ is a Gaussian ensemble whose moments up to degree two match with those of $\mathcal{L}_P$.

Apply the invariance principle with the finite probability space Ω_P, the ensembles $\mathcal{L}_P$, $\mathcal{G}_P$, smooth function P and vector of multilinear polynomials: $(\boldsymbol{F}_1,\boldsymbol{F}_2,\ldots,\boldsymbol{F}_k)$. As a consequence, we get

$$\left|\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[P(\boldsymbol{H}_1(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k(\boldsymbol{g}_k^R))\right]-\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\left[P(\boldsymbol{H}_1(\boldsymbol{\ell}_1^R),\ldots,\boldsymbol{H}_k(\boldsymbol{\ell}_k^R))\right]\right|\leqslant\tau^{K\varepsilon/\log(1/\alpha)},$$

where the constant K depends on q and k. The result follows by observing that expectation over $\mathcal{G}_P$ is equivalent to an expectation over $\mathcal{G}$. ∎

7.4.2 Bounding the Rounding Error

Here we will prove the following lemma that bounds the loss of payoff incurred while rounding $\mathbf{p}_s$ to $\mathbf{p}_s^*$ in the rounding scheme $\mathsf{Round}_{\boldsymbol{\mathcal{F}}}^{\varepsilon}$. With this lemma, we finish the proof of soundness of $\mathsf{DICT}_{\boldsymbol{V},\boldsymbol{\mu}}^{\varepsilon}$ (Lemma 7.4.3). Formally, we will show:

Lemma 7.4.5. *If a function $\mathcal{F}$ is (τ,ε)-quasirandom with respect to a payoff P with $\mathcal{V}(P)=\{s_1,\ldots,s_k\}$, then*

$$\left|\mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}^*_{s_1}(\boldsymbol{g}_{s_1}^R),\ldots,\boldsymbol{H}^*_{s_k}(\boldsymbol{g}_{s_k}^R)\Big)\right]-\mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{g}_{s_1}^R),\ldots,\boldsymbol{H}_{s_k}(\boldsymbol{g}_{s_k}^R)\Big)\right]\right|\leqslant C\tau^{K\varepsilon/\log(1/\alpha)},$$

where C,K are constants depending on q,k,α and ε.

Before we present the proof of the above lemma, we show two claims that will be useful for the proof.

Claim 7.4.1. *For all* $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$, $s \in \mathcal{V}$ *and all choices of* $\boldsymbol{\zeta}^{(1)}, \ldots, \boldsymbol{\zeta}^{(R)}$, $\sum_{a\in[q]} p_{s,a} = 1$.

Proof. Let us suppose that the function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ consists of q functions denoted by $\boldsymbol{\mathcal{F}} = (f_1, \ldots, f_q)$. By definition,

$$\begin{aligned}
\sum_{a\in[q]} p_{s,a} &= \sum_{a\in[q]} \sum_{\sigma\in[q]^R} f_a(\sigma) \prod_{j=1}^{R} h^{(j)}_{s,\sigma_j} \\
&= \sum_{\sigma\in[q]^R} \Big(\sum_{a\in[q]} f_a(\sigma) \Big) \prod_{j=1}^{R} h^{(j)}_{s,\sigma_j} \\
&= \sum_{\sigma\in[q]^R} \prod_{j=1}^{R} h^{(j)}_{s,\sigma_j} \qquad \Big(\because \sum_{a\in[q]} f_a(\sigma) = 1, \text{ since } \boldsymbol{\mathcal{F}}(\sigma) \in \blacktriangle_q \Big) \\
&= \prod_{j=1}^{R} \Big(\sum_{\sigma_j\in[q]} h^{(j)}_{s,\sigma_j} \Big) \qquad (7.8)
\end{aligned}$$

Observe that,

$$\sum_{j\in[q]} h_{s,j} = \sum_{j\in[q]} \|\boldsymbol{b}_{s,j}\|^2 + (1-\varepsilon)\langle \Big(\sum_{j\in[q]} \boldsymbol{b}_{s,j} - (\sum_{j\in[q]} \|\boldsymbol{b}_{s,j}\|^2)\boldsymbol{b}_0 \Big), \boldsymbol{\zeta} \rangle$$

Since $\sum_{j\in[q]} \boldsymbol{b}_{s,j} = \boldsymbol{b}_0$ and $\sum_{j\in[q]} \|\boldsymbol{b}_{s,j}\|^2 = 1$, we get $\sum_{j\in[q]} h_{s,j} = 1+(1-\varepsilon)\langle (\boldsymbol{b}_0 - \boldsymbol{b}_0), \boldsymbol{\zeta} \rangle = 1$. Substituting in Equation 7.8, the claim follows. ■

Claim 7.4.2. *Let* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^q$ *be such that* $\sum_i a_i = \sum_i b_i = 1$. *Define* $\mathbf{a}^*$ *and* $\mathbf{b}^*$ *as follows:*

$$f_{[0,1]}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leqslant x \leqslant 1 \\ 1 & \text{if } x > 1 \end{cases} \quad \text{Scale}(x_0, x_1, \ldots, x_{q-1}) = \begin{cases} \frac{1}{\sum_i x_i} \big(x_0, \ldots, x_{q-1}\big) \text{ if } \sum_i x_i \neq 0 \\ (1, 0, 0, \ldots, 0) \text{ if } \sum_i x_i = 0 \end{cases}$$

$$\mathbf{a}^* = \text{Scale}(f_{[0,1]}(a_1), \ldots, f_{[0,1]}(a_q)) \qquad \mathbf{b}^* = \text{Scale}(f_{[0,1]}(b_1), \ldots, f_{[0,1]}(b_q))$$

Then,

$$\sum_{i\in[q]} (a_i - a^*_i)^2 \leqslant (2q+2)\boldsymbol{\zeta}(\mathbf{a}) \qquad \sum_{i\in[q]} (b^*_i - a^*_i)^2 \leqslant (2q+2) \sum_i (b_i - a_i)^2$$

Proof. Let $\tilde{a} = (f_{[0,1]}(a_1), \ldots, f_{[0,1]}(a_q))$ and $\tilde{b} = (f_{[0,1]}(b_1), \ldots, f_{[0,1]}(b_q))$. First, observe that $\sum_i \tilde{a}_i > 0$ and $\sum_i \tilde{b}_i > 0$. Further, by definition of the function $\boldsymbol{\zeta}$, we have $\boldsymbol{\zeta}(\mathbf{a}) =$

$\sum_i(\tilde{a}_i - a_i)^2$.

$$\begin{aligned}
\sum_i(\tilde{a}_i - a_i^*)^2 &= \sum_i \Big(\tilde{a}_i - \frac{\tilde{a}_i}{\sum_i \tilde{a}_i}\Big)^2 && \text{(by definition)} \\
&= \Big(\sum_i \tilde{a}_i - 1\Big)^2 \sum_i \frac{\tilde{a}_i^2}{(\sum_i \tilde{a}_i)^2} \\
&\leqslant \Big(\sum_i \tilde{a}_i - 1\Big)^2 \\
&= \Big(\sum_i (\tilde{a}_i - a_i)\Big)^2 && (\because \sum_i a_i = 1) \\
&\leqslant q \sum_i (\tilde{a}_i - a_i)^2 && \text{(Cauchy-Schwartz inequality)} \qquad (7.9)
\end{aligned}$$

Now to finish the proof of the first assertion.

$$\begin{aligned}
\sum_i (a_i - a_i^*)^2 &\leqslant 2\big(\sum_i (a_i - \tilde{a}_i)^2 + \sum_i (\tilde{a}_i - a_i^*)^2\big) \\
&\leqslant (2q+2)\sum_i (a_i - \tilde{a}_i)^2 && \text{(using Equation7.9)} \\
&= (2q+2)\boldsymbol{\zeta}(\mathbf{a})
\end{aligned}$$

$$\begin{aligned}
\sum_i (a_i^* - b_i^*) &= \sum_i \Big(\frac{\tilde{a}_i}{\sum_i \tilde{a}_i} - \frac{\tilde{b_i}}{\sum_i \tilde{b_i}}\Big)^2 \\
&= \frac{1}{(\sum_i \tilde{b}_i)^2} \sum_i \Big(\frac{\tilde{a}_i(\sum_i \tilde{b}_i - \sum_i \tilde{a}_i)}{\sum_i \tilde{a}_i} - (\tilde{b}_i - \tilde{a}_i)\Big)^2 \\
&\leqslant \frac{2}{(\sum_i \tilde{b}_i)^2} \sum_i \Big[\Big(\frac{\tilde{a}_i(\sum_i \tilde{b}_i - \sum_i \tilde{a}_i)}{\sum_i \tilde{a}_i}\Big)^2 + (\tilde{b}_i - \tilde{a}_i)^2\Big] \\
&\leqslant \frac{2}{(\sum_i \tilde{b}_i)^2} \Big[\frac{\sum_i \tilde{a}_i^2}{(\sum_i \tilde{a}_i)^2} (\sum_i \tilde{b}_i - \sum_i \tilde{a}_i)^2 + \sum_i (\tilde{b}_i - \tilde{a}_i)^2\Big]
\end{aligned}$$

Since the numbers $\tilde{a}_i$ are non-negative, we have $\sum_i \tilde{a}_i^2 \leqslant (\sum_i \tilde{a}_i)^2$. Using this, we get

$$\sum_i (a_i^* - b_i^*) \leqslant \frac{2}{(\sum_i \tilde{b}_i)^2} \Big[(\sum_i \tilde{b}_i - \sum_i \tilde{a}_i)^2 + \sum_i (\tilde{b}_i - \tilde{a}_i)^2\Big]$$

By Cauchy Schwartz inequality, we know $(\sum_i \tilde{b}_i - \sum_i \tilde{a}_i)^2 \leqslant q \sum_i (b_i - a_i)^2$. Hence,

$$\sum_i (a_i^* - b_i^*) \leqslant \frac{2q+2}{(\sum_i \tilde{b}_i)^2} \sum_i (\tilde{b}_i - \tilde{a}_i)^2$$

Observe that by definition of the function $f_{[0,1]}$, if $\sum_i b_i = 1$ then $\sum_i \tilde{b}_i \geqslant 1$. Using this result in the above inequality, the result follows. ■

Proof of Lemma 7.4.5. Without loss of generality, we may assume that $\mathcal{V}(P) = \{1, \ldots, k\}$. The function $\boldsymbol{\mathcal{H}}_i = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_i$ is over the domain $[q]^R$ and has the range $\blacktriangle_q$. The polynomials $\boldsymbol{H}_i = (H_{(i,0)}, \ldots, H_{(i,q-1)})$ are a representation of $\boldsymbol{\mathcal{H}}_i$ in a standard basis.

Intuitively, the invariance principle asserts that the distribution of the random variable $\boldsymbol{H}_s(\boldsymbol{g}_s^R)$ is roughly the same as that of $\boldsymbol{H}_s(\boldsymbol{\ell}_s^R)$. Observe that on inputs from the local distribution $\boldsymbol{\ell}_s^R$, the value $\boldsymbol{H}_s(\boldsymbol{\ell}_s^R)$ is always contained in $\blacktriangle_q$. This suggests that the random variables $\boldsymbol{H}_s(\boldsymbol{g}_s^R)$ is *nearly always* close to $\blacktriangle_q$. For a point $\mathbf{p} \in \blacktriangle_q$, its rounded value $\mathbf{p}^* = \mathbf{p}$. Thus, the rounding of $\boldsymbol{H}_s(\boldsymbol{g}_s^R)$ only slightly changes its value, i.e., $\boldsymbol{H}_s(\boldsymbol{g}_s^R) \approx \boldsymbol{H}_s^*(\boldsymbol{g}_s^R)$. Recall that the payoff functions P is smooth in that they satisfy Property II (7.4.1). Therefore if $\boldsymbol{H}_s(\boldsymbol{g}_s^R) \approx \boldsymbol{H}_s^*(\boldsymbol{g}_s^R)$ for all s, the two quantities in the above claim are approximately equal.

From Property II (7.4.1) that the payoff functions satisfy,

$$\left| \mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}_{s_1}^*(\boldsymbol{g}_{s_1}^R), \ldots, \boldsymbol{H}_{s_k}^*(\boldsymbol{g}_{s_k}^R)\Big)\right] - \mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{g}_{s_1}^R), \ldots, \boldsymbol{H}_{s_k}(\boldsymbol{g}_{s_k}^R)\Big)\right]\right| \leqslant C_0 \sum_{i=1}^{|S|} \left(\mathbb{E}\left[\|\boldsymbol{H}_i^*(\boldsymbol{g}_i^R) - \boldsymbol{H}_i(\boldsymbol{g}_i^R)\|_2^2\right]\right)^{\frac{1}{2}} \quad (7.10)$$

By Claim 7.4.1, we have $\sum_{a\in[q]} p_{s,a} = 1$ for all variables $s \in \mathcal{V}(P)$. Therefore, by Claim 7.4.2, for all $s \in \mathcal{V}(P)$,

$$\mathbb{E}\left[\|\boldsymbol{H}_s^*(\boldsymbol{g}_s^R) - \boldsymbol{H}_s(\boldsymbol{g}_s^R)\|_2^2\right] \leqslant (2q+2)\,\mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\boldsymbol{g}_s^R))\right]$$

Since $\boldsymbol{H}_i(\boldsymbol{\ell}_s^R) \in \blacktriangle_q$ we have $\mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\boldsymbol{\ell}_s^R))\right] = 0$. Rewrite the above inequality as

$$\mathbb{E}\left[\|\boldsymbol{H}_s^*(\boldsymbol{g}_s^R) - \boldsymbol{H}_s(\boldsymbol{g}_s^R)\|_2^2\right] \leqslant (2q+2)\left|\mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\boldsymbol{g}_s^R))\right] - \mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\ell_s^R))\right]\right|.$$

Now we shall apply the invariance principle to bound the quantity on the right hand side. Applying the invariance principle on the vector valued function $\boldsymbol{F}_s$, and ensembles $\boldsymbol{\ell}_s^R$, $\boldsymbol{g}_s^R$,we get:

$$\left|\mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\boldsymbol{\ell}_s^R))\right] - \mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\boldsymbol{g}_s^R))\right]\right| \leqslant \tau^{K\varepsilon/\log(1/\alpha)},$$

for a constant K depending on α, ε and q. Consequently,

$$\mathbb{E}\left[\|\boldsymbol{H}_s^*(\boldsymbol{g}_s^R) - \boldsymbol{H}_s(\boldsymbol{g}_s^R)\|_2^2\right] \leqslant (2q+2)\left|\mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\boldsymbol{g}_s^R))\right] - \mathbb{E}\left[\boldsymbol{\zeta}(\boldsymbol{H}_s(\ell_s^R))\right]\right| \leqslant (2q+2)\tau^{K\varepsilon/\log(1/\alpha)}$$

Substituting in 7.10 we get the required result.

$$\left| \mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}_{s_1}^*(\boldsymbol{g}_{s_1}^R), \ldots, \boldsymbol{H}_{s_k}^*(\boldsymbol{g}_{s_k}^R)\Big)\right] - \mathop{\mathbb{E}}_{\mathcal{G}^R}\left[P\Big(\boldsymbol{H}_{s_1}(\boldsymbol{g}_{s_1}^R), \ldots, \boldsymbol{H}_{s_k}(\boldsymbol{g}_{s_k}^R)\Big)\right]\right| \leqslant C\tau^{K\varepsilon/\log(1/\alpha)}$$

■

7.4.3 Putting It Together

Now we are ready to show the following conversion from integrality gaps to dictatorship tests.

Theorem 7.5. *There exists constants $C = C(q,k)$ and $K = K(q,k)$ such that for all $\alpha, \varepsilon, \beta, \tau$ the following holds: Given a Λ-GCSP instance $\Im$, along with a α-smooth SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$, the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ satisfies the following properties:*

- $\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\varepsilon k.$
- $\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) \leqslant \mathrm{opt}(\Im) + C(\beta + \tau^{K\varepsilon/\log(1/\alpha)})$
  i.e., For every function $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ that is (β,τ,ε)-quasi-random with respect to $(\boldsymbol{V}, \boldsymbol{\mu})$,
  $$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) \leqslant \mathrm{opt}(\Im) + C(\beta + \tau^{K\varepsilon/\log(1/\alpha)})$$

Proof. Let $C = C(q,k)$ and $K = K(q,k)$ be the constants obtained in Lemma 7.4.3.

The claim about the completeness of the dictatorship test follows directly from Lemma 7.4.2 and the choice of ε. For a function $\boldsymbol{\mathcal{F}}$ which is (β,τ,ε)-quasi-random with respect to $\Im$, by Lemma 7.4.3, the expected value of the assignment returned by the rounding scheme $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}$ satisfies,

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) \leqslant \mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) + C(\beta + \tau^{K\varepsilon/\log(1/\alpha)}) .$$

By definition of $\mathrm{opt}(\Im)$, we have $\mathsf{Round}^{\varepsilon}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) \leqslant \mathrm{opt}(\Im)$. Therefore the claim follows. ■

7.5 From Dictatorship Tests to UG-hardness Results

In this section, we will make use of the dictatorship test shown in Section 7.3 to obtain Unique Games based hardness result.

Let $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi, [R])$ be a bipartite Unique Games instance. Further let $\Im = (\mathcal{V}, \mathcal{P})$ be an instance of a GCSP problem Λ. Let DICT be a dictatorship test obtained starting from a SDP solution for $\Im$.

Starting from the Unique Games instance Φ, we shall construct an instance $\Im(\Phi)$ of the GCSP problem Λ. For each vertex $v \in \mathcal{V}_\Phi$, we shall introduce a long code over $[q]^R$. More precisely, the instance $\Im(\Phi)$ is given by $\Im(\Phi) = (\mathcal{V}_\Phi \times [q]^R, \mathcal{P}')$. All the payoff functions in $\mathcal{P}'$ are of type Λ ensuring that $\Im(\Phi)$ is also an instance of the GCSP problem Λ. Since the set of variables of $\Im(\Phi)$ is given by $\mathcal{V}_\Phi \times [q]^R$, an assignment to $\Im(\Phi)$ consists of a set of functions,

$$\mathsf{F}^v : [q]^R \to [q] \quad \text{for each} \quad v \in \mathcal{V}_\Phi$$

For each $v \in \mathcal{V}_\Phi$, define $\boldsymbol{\mathcal{F}}^v : [q]^R \to \blacktriangle_q$ as follows:

$$\boldsymbol{\mathcal{F}}^v(\boldsymbol{z}) = \boldsymbol{e}_{\mathsf{F}^v(\boldsymbol{z})}$$

For a permutation $\pi : [R] \to [R]$ and $\boldsymbol{z} \in [q]^R$, define $\pi(\boldsymbol{z}) \in [q]^R$ as $\big(\pi(\boldsymbol{z})\big)^{(j)} = z^{(\pi^{-1}(j))}$ for all $j \in [R]$. For each $w \in \mathcal{W}_\Phi$, define a function $\boldsymbol{\mathcal{F}}^w : [q]^R \to \blacktriangle_q$,

$$\boldsymbol{\mathcal{F}}^w(\boldsymbol{z}) = \mathop{\mathbb{E}}_{v \in N(w)} \left[\boldsymbol{\mathcal{F}}^v(\pi_{v \leftarrow w}(\boldsymbol{z}))\right]$$

The basic idea behind converting a dictatorship test to Unique Games hardness is similar to Khot et.al.[99]. Roughly speaking, the verifier performs the dictatorship test DICT on the functions $\boldsymbol{\mathcal{F}}^w$ for $w \in \mathcal{W}_\Phi$. Note that the functions $\boldsymbol{\mathcal{F}}^w$ are not explicitly available to the verifier. However, this access can be simulated by accessing $\boldsymbol{\mathcal{F}}^v$ for a random neighbor $v \in N(w)$.

$\mathsf{Oracle}(\boldsymbol{\mathcal{F}}^w)$

- On a query $\boldsymbol{\mathcal{F}}^w(\boldsymbol{z})$, Pick a random neighbor $v \in N(w)$, and return $\boldsymbol{\mathcal{F}}^v(\pi_{v \leftarrow w}(\boldsymbol{z}))$.

$\mathsf{Verifier}(\Im(\Phi))$

- Pick $w \in \mathcal{W}_\Phi$ at random.
- Perform the test DICT on $\boldsymbol{\mathcal{F}}^w$, by transferring each of the queries to the $\mathsf{Oracle}(\boldsymbol{\mathcal{F}}^w)$.

The queries of the $\mathsf{Verifier}(\Im(\Phi))$ through the oracle, translate into tests/payoffs over the functions $\boldsymbol{\mathcal{F}}^v$. In turn, this is equivalent to tests/payoffs on the values of functions $\{\mathsf{F}^v | v \in \mathcal{V}_\Phi\}$. Summarizing, the set of all tests of the above verifier yield a GCSP instance over the variables $\mathcal{V}_\Phi \times [q]^R$.

Theorem 7.6 (Dictatorship Tests $\Rightarrow$ UG-hardness)**.** *Let DICT be a dictatorship test obtained from a SDP solution to a instance $\Im$ of a* GCSP Λ*. For all values of $\tau, \beta, \varepsilon, \delta, \gamma > 0$, given a* Unique Games *instance Φ, $\Im(\Phi)$ is an instance of* GCSP Λ *such that,*

- Completeness: *Φ is a $(1-\gamma)$-strong satisfiable instance then*

$$\mathrm{opt}(\Im(\Phi)) \geqslant \mathsf{Completeness}(\mathit{DICT}) - 2\gamma\,.$$

- Soundness: $\mathrm{opt}(\Phi) \leqslant \delta \Rightarrow \mathrm{opt}(\Im(\Phi)) \leqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathit{DICT}) + \frac{100\delta}{k^2\beta\varepsilon^2\tau^3}$.

Let $\boldsymbol{z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k | \boldsymbol{z}_i \in [q]^R\}$ be random variables denoting the query locations of $\mathsf{Verifier}(\Im(\Phi))$. Further, let P denote the payoff/test that the $\mathsf{Verifier}(\Im(\Phi))$ decides to perform on these locations. Arithmetizing the expected payoff returned by the verifier we get,

$$\mathop{\mathbb{E}}_{w \in \mathcal{W}_\Phi} \mathop{\mathbb{E}}_{\boldsymbol{z}} \mathop{\mathbb{E}}_{v_1,\ldots,v_k \in N(w)} \left[P\Big(\boldsymbol{\mathcal{F}}^{v_1}(\pi_{v_1 \leftarrow w}(\boldsymbol{z}_1)), \ldots, \boldsymbol{\mathcal{F}}^{v_k}(\pi_{v_k \leftarrow w}(\boldsymbol{z}_k))\Big)\right]$$

The payoff functions P are multilinear in the region $\blacktriangle_q$. The choices of the oracle $v_1, \ldots, v_k \in N(w)$ are independent of each other, and the verifier's query locations. In this light, we can

write the expected payoff as,

$$\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{\boldsymbol{z}}\Big[P\Big(\mathop{\mathbb{E}}_{v_1\in N(w)}[\boldsymbol{\mathcal{F}}^{v_1}(\pi_{v_1\leftarrow w}(\boldsymbol{z}_1))],\ldots,\mathop{\mathbb{E}}_{v_k\in N(w)}[\boldsymbol{\mathcal{F}}^{v_k}(\pi_{v_k\leftarrow w}(\boldsymbol{z}_k))]\Big)\Big]$$

Hence the expected payoff is just equal to $\mathbb{E}_{w\in\mathcal{W}_\Phi}[\mathsf{DICT}(\boldsymbol{\mathcal{F}}^w)]$

Completeness: If Φ is a $(1-\gamma)$-perfect satisfiable instance (see Definition 2.5.3), then there exists an assignment $\mathcal{A}$ such that for $(1-\gamma)$-fraction of the vertices $w\in\mathcal{W}_\Phi$, all the edges (v,w) incident at w are satisfied by $\mathcal{A}$.

Let us refer to such vertices $w\in\mathcal{W}_\Phi$ as *good* vertices. The assignment to the GCSP instance $\Im(\Phi)$ is given by the following set of functions: $\mathsf{F}^v(\boldsymbol{z})=z^{(\mathcal{A}(v))}$, or equivalently $\boldsymbol{\mathcal{F}}^v(\boldsymbol{z})=\boldsymbol{e}_{z^{(\mathcal{A}(v))}}$ for all $v\in\mathcal{V}_\Phi$. For every *good* vertex $w\in\mathcal{W}_\Phi$, we have:

$$\boldsymbol{\mathcal{F}}^w(\boldsymbol{z})=\mathop{\mathbb{E}}_{v\in N(w)}\left[\boldsymbol{\mathcal{F}}^v(\pi_{v\leftarrow w}(\boldsymbol{z}))\right]=\mathop{\mathbb{E}}_{v\in N(w)}\left[\boldsymbol{e}_{z^{(\pi^{-1}_{v\leftarrow w}(\mathcal{A}(v)))}}\right]=\boldsymbol{e}_{z^{(\mathcal{A}(w))}},$$

where the final step in the above calculation used the fact that $\pi^{-1}_{v\leftarrow w}(\mathcal{A}(v))=\mathcal{A}(w)$ for all $v\in N(w)$. Therefore the functions $\boldsymbol{\mathcal{F}}^w$ are dictator functions for every *good* vertex $w\in\mathcal{W}_\Phi$. With at least $(1-\gamma)$ fraction of the vertices in $\mathcal{W}_\Phi$ being *good*, the expected payoff is at least

$$(1-\gamma)\cdot(\mathsf{Completeness}(\mathsf{DICT}))+\gamma\cdot(-1)\geqslant\mathsf{Completeness}(\mathsf{DICT})-2\gamma$$

Soundness: Suppose there is an assignment to the variables $\mathcal{V}_\Phi\times[q]^R$ whose payoff is greater than $\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT})+\eta$. Then we have,

$$\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}[\mathsf{DICT}(\boldsymbol{\mathcal{F}}^w)]>\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT})+\eta$$

As all the payoff functions are bounded by 1, for at least η fraction of the vertices $w\in\mathcal{W}_\Phi$, $\mathsf{DICT}(\boldsymbol{\mathcal{F}}^w)>\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT})$. Henceforth we refer to these vertices as *good* vertices. By definition of $\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT})$, for every *good* vertex $w\in\mathcal{W}_\Phi$, the function $\boldsymbol{\mathcal{F}}^w$ is not (β,τ,ε)-quasirandom with respect to $\Im$.

Consider a *good* vertex $w\in\mathcal{W}_\Phi$. For a random choice of payoff P from the distribution $\mathcal{P}$, with probability at least β, the function $\boldsymbol{\mathcal{F}}^w$ is not (τ,ε)-quasirandom with respect to S. By an averaging argument, there exists a payoff P such that for at least β-fraction of the *good* vertices $w\in\mathcal{W}_\Phi$, the function $\boldsymbol{\mathcal{F}}^w$ is not (τ,ε)-quasirandom with respect to P. Fix such a payoff and let $\mathcal{V}(P)=\{s_1,\ldots,s_k\}$. For convenience, let us denote $\boldsymbol{\mathcal{H}}^u_s=T_{1-\varepsilon}\boldsymbol{\mathcal{F}}^u_s$ for each $u\in\mathcal{W}_\Phi\cup\mathcal{V}_\Phi$ and $s\in\mathcal{V}$.

For each vertex $w\in\mathcal{W}_\Phi$, define the set of labels $\mathsf{L}(w)$ as

$$\mathsf{L}(w)=\{\ell|\exists s\in\mathcal{V}(P),\mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}^w_s)>\tau\}.$$

Similarly, for each $v\in\mathcal{V}_\Phi$ define,

$$\mathsf{L}(v)=\{\ell|\exists s\in\mathcal{V}(P),\mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}^v_s)>\tau/3\}.$$

Consider the following Labeling for the Unique Games instance Φ: For each vertex

$u \in \mathcal{W}_\Phi \cup \mathcal{V}_\Phi$, assign a random label from $\mathsf{L}(u)$ if it is nonempty, else assign a uniformly random label.

At least η fraction of the vertices $w \in \mathcal{W}_\Phi$ are *good* vertices. By the choice of payoff P, at least β fraction of the *good* vertices $w \in \mathcal{W}_\Phi$ have a non-empty label set $\mathsf{L}(w)$. Fix a *good* vertex w with a nonempty label set $\mathsf{L}(w)$. Consider a label $\ell \in \mathsf{L}(w)$. By definition of $\mathsf{L}(w)$, we have $\mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}_s^w) \geqslant \tau$ for some $s \in \mathcal{V}(P)$. The function $\boldsymbol{\mathcal{H}}_s^w$ is given by $\boldsymbol{\mathcal{H}}_s^w(\boldsymbol{z}) = \mathbb{E}_{v \in N(w)}[\boldsymbol{\mathcal{H}}_s^v(\pi_{v \leftarrow w}(\boldsymbol{z}))]$. By convexity of influences (see Proposition 3.0.13), if $\mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}_s^w) \geqslant \tau$ then

$$\mathop{\mathbb{E}}_{v \in N(w)}[\mathrm{Inf}_{\pi_{v \leftarrow w}(\ell)}(\boldsymbol{\mathcal{H}}_s^v)] \geqslant \tau$$

Since the range of the function $\boldsymbol{\mathcal{H}}_s^v$ is $\blacktriangle_q$, we have $\mathrm{Inf}_\ell(\boldsymbol{\mathcal{H}}_s^v) \leqslant 2$ for all v, ℓ. Hence for at least $\tau/3$ fraction of neighbors $v \in N(w)$, $\mathrm{Inf}_{\pi_{v \leftarrow w}(\ell)}(\boldsymbol{\mathcal{H}}_s^v) \geqslant \frac{\tau}{3}$. Thus for at least $\tau/3$ fraction of the neighbors $v \in N(w)$, there exists $\ell \in [R]$ such that $\ell \in \mathsf{L}(w)$ and $\pi_{v \leftarrow w}(\ell) \in \mathsf{L}(v)$. For every such neighbor v, the edge constraint $\pi_{v \leftarrow w}$ is satisfied with probability at least $\frac{1}{|\mathsf{L}(v)||\mathsf{L}(w)|}$.

From Lemma 3.0.2, each function $\boldsymbol{\mathcal{H}}_s^w$ can have at most $1/\tau\varepsilon$ influential coordinates. Thus the maximum size of the label set $\mathsf{L}(w)$ is $k/\tau\varepsilon$. In conclusion, the expected fraction of Unique Games constraints satisfied is at least $\eta \times \beta \times \frac{\tau}{3} \times k^{-2}\tau^2\varepsilon^2 = \tau^3\varepsilon^2\eta\beta k^2/3$. This implies that $\mathrm{opt}(\Phi) > \tau^3\varepsilon^2\eta\beta k^2/100$, and the conclusion follows by setting $\tau^3\varepsilon^2\eta\beta k^2 = \delta$.

7.6 Optimal Algorithm

Theorem 7.7. *Let* $\mathsf{Round}_\eta(\boldsymbol{V}, \boldsymbol{\mu})$ *denote the expected value of assignment returned by the rounding scheme* Round_η *on a SDP solution* $(\boldsymbol{V}, \boldsymbol{\mu})$ *for a* GCSP *instance* $\Im$*. For every* $\eta > 0$*, given an instance* $\Im'$ *of the* GCSP Λ*, it is UG hard to distinguish whether,*

$$\mathrm{opt}(\Im') \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 3\eta \qquad \textit{OR} \qquad \mathrm{opt}(\Im') \leqslant \mathsf{Round}_\eta(\boldsymbol{V}, \boldsymbol{\mu}) + 3\eta$$

Proof. By Lemma 7.4.1, the α-smooth SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$ satisfies $\mathrm{val}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - \eta$. Let $\mathsf{DICT}_\Im = \mathsf{DICT}^\varepsilon_{\boldsymbol{V}^*, \boldsymbol{\mu}^*}$ be the dictatorship test obtained from the α-smooth SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$.

The completeness of $\mathsf{DICT}_\Im$ is at least $\mathrm{val}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) - \eta \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\eta$ by Lemma 7.4.2.

Among $(\beta, \tau, \varepsilon)$-quasirandom functions, let $\boldsymbol{\mathcal{F}}^*$ be the function that achieves the optimal expected payoff under $\mathsf{DICT}_\Im$, i.e., $\mathsf{DICT}_\Im(\boldsymbol{\mathcal{F}}^*) = \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}_\Im)$. By Corollary 7.4.3, and the choices of β, τ, we have $\mathsf{Round}^\varepsilon_{\boldsymbol{\mathcal{F}}^*}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) \geqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}_\Im) - \eta$.

By definition of S_κ, there exists some $\boldsymbol{\mathcal{F}}_i \in S_\kappa$ such that $\sup_{\boldsymbol{x} \in [q]^R} \|\boldsymbol{\mathcal{F}}^*(\boldsymbol{x}) - \boldsymbol{\mathcal{F}}_i(\boldsymbol{x})\| \leqslant \kappa$. By Lemma 7.7.1, we have $\mathsf{Round}^\varepsilon_{\boldsymbol{\mathcal{F}}_i}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) \geqslant \mathsf{Round}_{\boldsymbol{\mathcal{F}}^*}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) - \eta \geqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}_\Im) - 2\eta$. Consequently, we get

$$\mathsf{Round}_\eta(\boldsymbol{V}, \boldsymbol{\mu}) \geqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}_\Im) - 2\eta\,.$$

Summarizing the above argument, the completeness of $\mathsf{DICT}_\Im$ is at least $\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\eta$, while the soundness is given by $\mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}_\Im) \leqslant \mathsf{Round}_\eta(\boldsymbol{V}, \boldsymbol{\mu}) + 2\eta$. By applying the UG-hardness reduction (Theorem 7.6) starting from $\mathsf{DICT}_\Im$, the proof is complete. ■

Round_η Algorithm

Input: An SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ for the GCSP instance $\Im$ and a parameter $\eta > 0$.

Parameters: Set $\alpha = \eta/q^k$, $\varepsilon = \eta/4k$. Fix the value of $\tau = \exp(-O(\frac{\log^2 \eta}{\eta}))$, $\beta = \eta/100C$ such that the error $C(\beta + \tau^{K\varepsilon/\log(1/\alpha)})$ in Corollary 7.4.3 is less than $\eta/10$. Fix $R = \exp(O(\frac{1}{\eta^3}))$ to ensure that the quantitative version of **UGC** can have completeness $1 - \frac{\eta}{10}$ and soundness $\eta\beta\varepsilon^2\tau^3k^2/100$. Let $\kappa = \frac{\eta}{8C_0(q,k)qk}$ where $C_0(q,k)$ is the smoothness parameter of the payoffs P (Property II 7.4.1).
Smoothing: Using the transformation in Lemma 7.4.1, construct an α-smooth SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$.
Rounding $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$: Let $S_\kappa = \{\boldsymbol{\mathcal{F}}_1, \ldots, \boldsymbol{\mathcal{F}}_M\}$ be a set of functions such that for every $\boldsymbol{\mathcal{F}} : [q]^R \to \blacktriangle_q$ there exists $\boldsymbol{\mathcal{F}}_i \in S_\kappa$ satisfying $\sup_{\boldsymbol{x}\in[q]^R} \|\boldsymbol{\mathcal{F}}_i(\boldsymbol{x}) - \boldsymbol{\mathcal{F}}(\boldsymbol{x})\|_2 \leqslant \frac{\eta}{8C_0(q,k)qk}$.

- For each function $\boldsymbol{\mathcal{F}}_i \in S_\kappa$, run the subroutine $\mathsf{Round}^\varepsilon_{\boldsymbol{\mathcal{F}}_i}$ on the SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$
- Output the assignment obtained with the largest objective value.

Figure 7.2: Round_η Algorithm

Lemma 7.7.1. *Let $C_0(q,k)$ denote the smoothness of the payoffs P in Property II (7.4.1). For two functions $\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}' : [q]^R \to \blacktriangle_q$ and an SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ to a* GCSP *instance $\Im$, we have*

$$\|\mathsf{Round}^\varepsilon_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) - \mathsf{Round}^\varepsilon_{\boldsymbol{\mathcal{F}}'}(\boldsymbol{V}, \boldsymbol{\mu})\| \leqslant 8C_0(q,k)qk \sup_{\boldsymbol{x}\in[q]^R} \|\boldsymbol{\mathcal{F}}(\boldsymbol{x}) - \boldsymbol{\mathcal{F}}'(\boldsymbol{x})\|_2$$

where the constant in O depends on the GCSP *problem Λ.*

Proof. For each variable $s \in \mathcal{V}$, let $\boldsymbol{\mathcal{F}}_s, \boldsymbol{\mathcal{F}}'_s$ denote the functions $\boldsymbol{\mathcal{F}}$, $\boldsymbol{\mathcal{F}}'$ thought of as functions over the probability space Ω_s. Let $\boldsymbol{H}_s$ and $\boldsymbol{H}'_s$ denote the multilinear polynomials representing $T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_s$ and $T_{1-\varepsilon}\boldsymbol{\mathcal{F}}'_s$ respectively. For a vector $\boldsymbol{g}^R \in \mathbb{R}^R$, let $\boldsymbol{H}^*_s(\boldsymbol{g}^R)$ and $\boldsymbol{H}'^*_s(\boldsymbol{g}^R)$ denote the rounding of $\boldsymbol{H}_s(\boldsymbol{g}^R), \boldsymbol{H}'_s(\boldsymbol{g}^R)$ to $\blacktriangle_q$. Then we have,

$$\mathsf{Round}_{\boldsymbol{\mathcal{F}}'}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathcal{G}^R} \left[P\Big(\boldsymbol{H}'^*_{s_1}(\boldsymbol{g}^R_{s_1}), \ldots, \boldsymbol{H}'^*_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\right]$$
$$\mathsf{Round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathcal{G}^R} \left[P\Big(\boldsymbol{H}^*_{s_1}(\boldsymbol{g}^R_{s_1}), \ldots, \boldsymbol{H}^*_{s_k}(\boldsymbol{g}^R_{s_k})\Big)\right]$$

Fix a payoff $P \in \mathcal{P}$. For the sake of convenience, we can rename the indices so that $\mathcal{V}(P) = \{1, \ldots, k\}$. From the smoothness (Property II 7.4.1) of the payoff P we see that,

$$\begin{aligned}
&\Big|\mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}_1^*(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k^*(\boldsymbol{g}_k^R)\Big)\Big]-\mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}_1'^*(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k'^*(\boldsymbol{g}_k^R)\Big)\Big]\Big| \\
&\qquad\leqslant C_0\sum_{i=1}^{k}\mathbb{E}\Big[\|\boldsymbol{H}_i'^*(\boldsymbol{g}_i^R)-\boldsymbol{H}_i{}^*(\boldsymbol{g}_i^R)\|_2\Big] \\
&\qquad\leqslant C_0\sum_{i=1}^{k}\Big(\mathbb{E}\Big[\|\boldsymbol{H}_i'^*(\boldsymbol{g}_i^R)-\boldsymbol{H}_i{}^*(\boldsymbol{g}_i^R)\|_2^2\Big]\Big)^{\frac{1}{2}} \qquad \text{(Cauchy-Schwartz inequality)} \quad (7.11)
\end{aligned}$$

From Claim 7.4.1, for each i that $\sum_{j\in[q]}\boldsymbol{H}_j(\boldsymbol{g}_i^R)=\sum_{j\in[q]}\boldsymbol{H}_j'(\boldsymbol{g}_i^R)=1$. By the second part of Claim 7.4.2, for all random choices of $\boldsymbol{g}_i^R$,

$$\|\boldsymbol{H}_i^*(\boldsymbol{g}_i^R)-\boldsymbol{H}_i'^*(\boldsymbol{g}_i^R)\|^2\leqslant(2q+2)\|\boldsymbol{H}_i(\boldsymbol{g}_i^R)-\boldsymbol{H}_i'(\boldsymbol{g}_i^R)\|^2$$

Substituting in inequality 7.11 we get,

$$\begin{aligned}
&\Big|\mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}_1^*(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k^*(\boldsymbol{g}_k^R)\Big)-P\Big(\boldsymbol{H}_1'^*(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k'^*(\boldsymbol{g}_k^R)\Big)\Big]\Big| \\
&\qquad\leqslant C_0(2q+2)^{\frac{1}{2}}\sum_{i=1}^{k}\Big(\mathop{\mathbb{E}}_{\boldsymbol{g}_i^R}\big[\|\boldsymbol{H}_i(\boldsymbol{g}_i^R)-\boldsymbol{H}_i'(\boldsymbol{g}_i^R)\|^2\big]\Big)^{\frac{1}{2}} \qquad (7.12)
\end{aligned}$$

As the ensembles $\boldsymbol{g}_i^R$ and ℓ_i^R have matching moments up to order 2, it is easy to see that

$$\mathop{\mathbb{E}}_{\boldsymbol{g}_i^R}\big[\|\boldsymbol{H}_i(\boldsymbol{g}_i^R)-\boldsymbol{H}_i'(\boldsymbol{g}_i^R)\|^2\big]=\mathop{\mathbb{E}}_{\ell_i^R}\big[\|\boldsymbol{H}_i(\ell_i^R)-\boldsymbol{H}_i'(\ell_i^R)\|^2\big] \qquad (7.13)$$

Recall that $\boldsymbol{H}_i$ is a representation of the function $T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_i:\Omega^i\to\blacktriangle_q$, over the ensemble $\boldsymbol{\ell}_i^R$. Therefore, we see that,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\ell_i^R}\big[\|\boldsymbol{H}_i(\ell_i^R)-\boldsymbol{H}_i'(\ell_i^R)\|_2^2\big]&=\mathop{\mathbb{E}}_{\boldsymbol{x}\in\Omega_i^R}\big[\|T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_i(\boldsymbol{x})-T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_i'(\boldsymbol{x})\|_2^2\big] \\
&\leqslant\mathop{\mathbb{E}}_{\boldsymbol{x}\in\Omega_i^R}\big[\|\boldsymbol{\mathcal{F}}_i(\boldsymbol{x})-\boldsymbol{\mathcal{F}}_i'(\boldsymbol{x})\|_2^2\big] \qquad (T_{1-\varepsilon}\text{ is contractive for all }\Omega_i) \\
&\leqslant\sup_{\boldsymbol{x}\in[q]^R}\|\boldsymbol{\mathcal{F}}_i(\boldsymbol{x})-\boldsymbol{\mathcal{F}}_i'(\boldsymbol{x})\|_2^2 \qquad (7.14)
\end{aligned}$$

From Equations 7.12,7.13 and 7.14, we see that for the payoff $P\in\mathcal{P}$ with $\mathcal{V}(P)=\{1,\ldots,k\}$,

$$\begin{aligned}
&\Big|\mathop{\mathbb{E}}_{\mathcal{G}^R}\Big[P\Big(\boldsymbol{H}_1^*(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k^*(\boldsymbol{g}_k^R)\Big)-P\Big(\boldsymbol{H}_1'^*(\boldsymbol{g}_1^R),\ldots,\boldsymbol{H}_k'^*(\boldsymbol{g}_k^R)\Big)\Big]\Big| \\
&\qquad\leqslant C_0(2q+2)^{\frac{1}{2}}k\sup_{\mathbf{x}\in[q]^R}\|\boldsymbol{\mathcal{F}}(\boldsymbol{x})-\boldsymbol{\mathcal{F}}'(\boldsymbol{x})\|_2
\end{aligned}$$

Averaging the above inequality over all $P\in\mathcal{P}$, the result follows.

■

7.7 From UG-hardness to SDP integrality gaps

The reduction from UG-hardness results to SDP integrality gaps can be implemented for the class of generalized constraint satisfaction problems. Specifically, the following holds:

Theorem 7.8. *For all $\gamma, \delta > 0$, given a dictatorship test DICT over $\{\pm 1\}^R$ for $R \geqslant (\frac{1}{\delta})^{\frac{100}{\gamma}}$, there exists a Λ instance Ψ such that* $\mathrm{sdp}(\Psi) \geqslant (1-\gamma)\mathsf{Completeness}(\mathit{DICT}) - 2\gamma$ *and* $\mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathit{DICT}) + \frac{100\delta}{k^2\beta\varepsilon^2\tau^3}$.

The idea of the proof is to execute the UG-hardness reductions starting from an integrality gap instance $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi, [R])$ for the LH_r relaxation (LH_r) of Unique Games for $r = k+1$. The existence of such strong integrality gaps for Unique Games is shown in Chapter 12.

Let Ψ denote the instance generated by executing the UG-hardness reduction on Φ. By the soundness analysis of the UG-hardness reduction (Theorem 7.6), we have $\mathrm{opt}(\Psi) \leqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT})$.

It remains to argue that the $\mathrm{sdp}(\Psi)$ is at least $\mathsf{Completeness}(\mathsf{DICT}) - 2\gamma$. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote a feasible solution to the LH_r relaxation of the unique games instance $\mathfrak{I}$.

The vertices of Ψ consist of $(v, \boldsymbol{x})$ for some $\boldsymbol{x} \in [q]^R$. Define the SDP vectors associated with vertex $(v, \boldsymbol{x})$ are given as,

$$\boldsymbol{V}_i^{v,\boldsymbol{x}} = \sum_{x^{(\ell)}=i} \boldsymbol{b}_{v,\ell} \qquad \forall v \in \mathcal{V}_\Phi, \boldsymbol{x} \in [q]^R, i \in [q]\,.$$

Fix a payoff P in the instance Ψ. Let $\mathcal{V}(P) = \{(v_1, \boldsymbol{z}_1)(v_2, \boldsymbol{z}_2), \ldots, (v_k, \boldsymbol{z}_k)\}$. Let $S = \{v_1, \ldots, v_k\}$, and let μ_S denote the corresponding local distribution from LC relaxation. Thus μ_S is a probability distribution over $[R]^S$. Define the local distributions $\{\mu_P | P \in \mathcal{P}\}$ on $[q]^{\mathcal{V}(P)}$ as,

$$\mu_P(x) = \mathop{\mathbb{P}\mathrm{r}}_{\boldsymbol{\ell} \in \mu_S} \left[z_1^{(\ell_1)} = x_1 \wedge \cdots \wedge z_k^{(\ell_k)} = x_k \right]$$

It is fairly straightforward and by now standard, but notationally intense to argue that the SDP solution as constructed above is a feasible SDP solution for Ψ, with value at least $\mathsf{Completeness}(\mathsf{DICT})$. Most of the ideas of the proof have been discussed in the proof of Theorem 6.7. Hence, we omit the formal proof of the above claim from the thesis.

7.8 Implications

In this section, we will use the reductions between dictatorship tests, UG-hardness results and SDP integrality gaps to derive the results stated in Section 7.1. First, we describe the setting of parameters used in the reductions to derive the implications.

Parameters Set $\alpha = \eta/q^k$, $\varepsilon = \eta/4k$. Fix the value of $\tau = \exp(-O(\frac{\log^2 \eta}{\eta}))$, $\beta = \eta/100C$ such that the error $C(\beta + \tau^{K\varepsilon/\log(1/\alpha)})$ in Corollary 7.4.3 is less than $\eta/10$. Fix $R = \exp(O(\frac{1}{\eta^3}))$ to ensure that the quantitative version of **UGC** can have completeness

$1-\gamma = 1-\frac{\eta}{10}$ and soundness $\delta = \eta\beta\varepsilon^2\tau^3k^2/1000$. Let $\kappa = \frac{\eta}{8C_0(q,k)qk}$ where $C_0(q,k)$ is the smoothness parameter of the payoffs P (Property II, 7.4.1).

Optimal UG-hardness (Proof of Theorem 7.1) To show the result we compose the conversion from SDP integrality gaps to dictatorship tests (Theorem 7.5) and dictatorship tests to UG-hardness results (Theorem 7.6). First, we state and prove the composed reduction from integrality gaps to UG-hardness result.

Theorem 7.9 (Integrality Gaps ⇒ UG-hardness)**.** *Let $\Im$ be any instance of* GCSP Λ*. For every $\eta > 0$, given an instance $\Im'$ of the* GCSP Λ*, it is UG hard to distinguish whether,*

$$\mathrm{opt}(\Im') \geqslant \mathit{LC}(\Im) - \eta \qquad \mathit{OR} \qquad \mathrm{opt}(\Im') \leqslant \mathrm{opt}(\Im) + \eta$$

Proof. Set $\alpha = \eta/q^k$, $\varepsilon = \eta/2k$. Let C, K be the constants in Theorem 7.5. Define $\beta = \eta/10C$ and $\tau = \exp(-O(\frac{\log^2(1/\eta)}{\eta}))$ such that $C\tau^{K\varepsilon/\log(1/\alpha)} \leqslant \eta/10$.

Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote an optimal SDP solution to the LC relaxation of $\Im$. Hence we have $\mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathsf{LC}(\Im)$. By Lemma 7.4.1, there exists an α-smooth SDP solution $(\boldsymbol{V}^*, \boldsymbol{\mu}^*)$ with $\mathrm{val}(\boldsymbol{V}^*, \boldsymbol{\mu}^*) \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - \eta = \mathsf{LC}(\Im) - \eta$.

Let us denote $\mathsf{DICT}_\Im = \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V}^*, \boldsymbol{\mu}^*}$. By Theorem 7.5, the dictatorship test $\mathsf{DICT}_\Im$ satisfies the following properties,

$$\mathsf{Completeness}(\mathsf{DICT}_\Im) \geqslant \mathsf{LC}(\Im) - 2\eta \qquad \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT}_\Im) \leqslant \mathrm{opt}(\Im) + \eta\,.$$

Using the dictatorship test $\mathsf{DICT}_\Im$ into the UG hardness reduction (Theorem 7.6), we obtain a UG-hardness for GCSP Λ of distinguishing between $\mathsf{LC}(\Im) - 4\eta$ vs $\mathrm{opt}(\Im) + 2\eta$. Since η can be made arbitrarily small, the result follows. ■

Proof of Theorem 7.1. Let $\Im$ be an instance of Λ-GCSP such that $\mathrm{sdp}(\Im) = c$ while $\mathrm{opt}(\Im) \leqslant \mathsf{Gap}_\Lambda(c) + \frac{\eta}{4}$. Applying Theorem 7.9 on the instance $\Im$ yields the required result. ■

Optimal Rounding Scheme (Proof of Theorem 7.2)

Proof. Consider the Round_η rounding scheme presented in Section 7.6. Let $\mathsf{RoundingCurve}(c)$ denote the curve associated with the rounding scheme Round_η. By Theorem 7.7, clearly we have,

$$\mathsf{RoundingCurve}(c) \geqslant \mathsf{UGhard}_\Lambda(c-\eta) - \eta\,.$$

By the conversion from UG-hardness results to SDP integrality gaps (Theorem 7.8), we have

$$\mathsf{Gap}_\Lambda(c-\eta) \leqslant \mathsf{UGhard}_\Lambda(c) + \eta\,,$$

for all $\eta > 0$. The result follows from the above two inequalities.

A κ-net of functions over $[q]^R$ has about $(\frac{1}{\kappa})^{q^R} = \exp(\exp(\exp(O(1/\eta^3))))$ different functions. Hence, the running time of the algorithm is as stated.

■

Computing Integrality Gaps (Proof of Theorem 7.3) In Theorem 7.5 and Theorem 7.8, we have effectively established an equivalence between SDP integrality gaps and dictatorship tests for arbitrary GCSPs. The integrality gap curve $\mathsf{Gap}_\Lambda(c)$ is the worst case value of the optimum over all instances with SDP value c. As there are infinitely many instances of all sizes with SDP value c, it is unclear how $\mathsf{Gap}_\Lambda(c)$ can be computed in finite time.

The crucial observation is that the set of all dictatorship tests over $[q]^R$ is a compact set, that can be easily discretized. Recall that a dictatorship test over $[q]^R$ is nothing but an instance of Λ-GCSP over $[q]^R$.

Define $\mathsf{Soundness}_{\beta,\tau,\varepsilon}(c)$ as follows:

$$\mathsf{Soundness}_{\beta,\tau,\varepsilon}(c) = \inf_{\substack{\mathsf{DICT}-\text{ a dictatorship test over } [q]^R \\ \mathsf{Completeness}(\mathsf{DICT})=c}} \mathsf{Soundness}_{\beta,\tau,\varepsilon}(\mathsf{DICT})$$

By Theorem 7.8, there exists a Λ-GCSP instance $\Im'$ with $\mathrm{sdp}(\Im') \geqslant (1-\gamma)c - 2\gamma \geqslant c - \eta$ and $\mathrm{opt}(\Im') \leqslant \mathsf{Soundness}_{\beta,\tau,\varepsilon}(c) + \eta$. Therefore we have

$$\mathsf{Soundness}_{\beta,\tau,\varepsilon}(c) \geqslant \mathsf{Gap}_\Lambda(c-\eta) - \eta\,.$$

Furthermore by Theorem 7.5, we have

$$\mathsf{Soundness}_{\beta,\tau,\varepsilon}(c) \leqslant \mathsf{Gap}_\Lambda(c+\eta) + \eta$$

Observe that for any $\eta > 0$, by iterating over a sufficiently fine κ-net over the space of dictatorship tests, the value $\mathsf{Soundness}_{\beta,\tau,\varepsilon}(c)$ can be computed within an accuracy of η. This completes the proof of Theorem 7.3.

Chapter 8

METRIC LABELING PROBLEMS

8.1 Introduction

The metric labeling problem falls under the class of edge deletion problems along with many other classic optimization problems. In an edge deletion problem, given an undirected graph $G = (V, E)$ and a non-negative weight function w on E, the goal is to find a minimum weight set of edges E' such that $G' = (V, E - E')$ satisfies certain properties. A special case is when the set of deleted edges forms a *cut*. The simplest and probably most familiar problem in this class is the minimum (s, t) cut problem. Given two terminals s and t, the goal is to find a minimum weight cut that separates s and t. This problem can be solved precisely in polynomial time following the classic work of Ford and Fulkerson.

The Multiway Cut problem is a natural generalization of the minimum (s, t) cut problem when more than two terminals are involved. The input is a set of q terminals $\mathsf{L} \subseteq V$ and the goal is to find a minimum weight set of edges that separates every pair of terminals. The problem is NP-hard and the best known approximation algorithm uses a geometric relaxation by Calinescu et.al [46].

The Zero-Extension problem [95, 96] is a generalization of the Multiway Cut problem in which a metric d is defined on the terminal set L. The goal is to assign to each vertex $v \in V$ a terminal $t(v)$ in L, while minimizing the total cost given by $\sum_{u,v \in E} w(u, v) d(t(u), t(v))$. Notice that in case the metric on the terminals L is the uniform metric (all distances equal to 1), the problem reduces to the Multiway Cut problem.

Generalizing the Zero-Extension problem further, one defines the Metric Labeling problem as follows: The input consists of a metric space (L, d) of *labels* and a non-negative cost function c on vertex-label pairs. The objective is to find an assignment of labels to the vertices minimizing $\sum_{v \in V} c(v, t(v)) + \mathbb{E}_{(u,v) \in E}\, w(u, v)\; d(t(u), t(v))$. The Zero-Extension problem is the special case where the assignment costs are all zero.

Inspired by the geometric relaxation for the Multiway Cut problem, Chekuri et al. [39] proposed an earthmover metric linear relaxation for the Metric Labeling and Zero-Extension problems. The best known approximation ratios [46, 30, 53, 107, 71, 39, 6] for all the above labelling problems, are achieved using linear programs that are either equivalent or strictly weaker than the earth-mover linear program. Nevertheless, the hardness results [45, 92, 47] known for the above described problems do not match the best known approximation algorithms. For instance, while Multiway Cut is known to be approximable within a factor roughly 1.3438, nothing better than APX-hardness [47] is known for the problem.

In the above discussion, an intriguing possibility that remains open is the use of semidefinite programming (SDP) to obtain better approximation factors for Metric Labeling . Even for the case of Multiway Cut , obtaining a better approximation using semidefinite programming has not been ruled out.

8.1.1 Results

In this chapter, we further develop the integrality gap to UG-hardness paradigm that has been the subject of the previous two chapters by extending it to the class of Metric Labeling problems. The main result of this chapter is a black-box reduction from integrality gaps to UG-hardness result for Multiway Cut , Zero-Extension and the class of Met-

RIC LABELING problems. More precisely, starting from an integrality gap instance $\Im$ with ratio α for the so called *earthmover linear program*, the reduction shows that it is UG-hard to approximate the problem to a factor better than α. Roughly speaking, this implies that the earthmover linear program (EM-LP) yields the best approximation computable in polynomial time for each of the problems MULTIWAY CUT ,ZERO-EXTENSION and METRIC LABELING , assuming the Unique Games Conjecture.

The precise statement of the reduction from integrality gaps to UG hardness result is as follows:

Theorem 8.1. *For the* MULTIWAY CUT *, the* ZERO-EXTENSION *, and the* METRIC LABELING *problems, the following holds: Given an instance $\Im$ with integrality gap α for the earthmover linear program (EM-LP), it is UG-hard to approximate the problem to a factor better than α. Further, the instances produced by the UG-hardness reduction have the same set of labels as $\Im$.*

The UG-hardness reduction stated above produces instances whose size is at least doubly exponential in the size of the integrality gap instance $\Im$. Therefore, the above theorem is to be applied with an integrality gap instance $\Im$ of fixed constant size, with a constant integrality gap α.

As the reduction always produces an instance with the same set of labels, the following stronger result holds:

Theorem 8.2. *It is UG-hard to approximate the* METRIC LABELING *and* ZERO-EXTENSION *problems with any finite metric (L, d) to a factor better than the integrality gap of the earthmover linear program on (L, d).*

Note that determining the exact value of the earthmover linear program integrality gap for these problems is not always easy. The following table shows the earthmover linear program gaps and the best known approximation factors.

Problem	Integrality Gap	App. Factor
3-WAY CUT	12/11 [91]	12/11 [91]
ZERO-EXTENSION	$\Omega\big((\log \lvert\mathsf{L}\rvert)^{\frac{1}{2}}\big)$ [92]	$O\big(\frac{\log \lvert\mathsf{L}\rvert}{\log\log \lvert\mathsf{L}\rvert}\big)$ [53]
METRIC LABELING	$\Omega(\log \lvert\mathsf{L}\rvert)$ [92]	$O(\log \lvert\mathsf{L}\rvert)$ [107]
UNIFORM METRIC LABELING	$2 - \frac{2}{\lvert\mathsf{L}\rvert}$ [107]	2 [107]

Interestingly, the reductions in this paper would apply even if the distance function between the labels does not satisfy triangle inequality. In particular, it is enough that $d(x, x) = 0$ and $d(x, y) \neq 0$ for $x \neq y$.

Using the connection from the UG-hardness results to SDP integrality gaps (Section 6.7), the above results imply that even certain strong semidefinite programming relaxations does not yield better approximation factors than the earthmover linear program. Specifically, the following theorem holds:

Theorem 8.3. *For the* MULTIWAY CUT *, the* ZERO-EXTENSION *, and the* METRIC LABELING *problems, the integrality gap of the following strong SDP relaxations is equal to the integrality gap of the earthmover linear program.*

– LH_r *hierarchy up to* $r = 2^{O((\log\log n)^{\frac{1}{4}})}$ *number of rounds.*

– SA_r *hierarchy up to* $r = O((\log\log n)^{\frac{1}{4}})$ *number of rounds.*

We refer the reader to Section 4.7 for formal definitions of the LH_r and SA_r hierarchies.

8.1.2 Labelling Problems : Prior Work

While the minimum (s,t)-cut problem is solvable in polynomial time, the Multiway Cut - a close generalization turns out to be NP-hard. Using the (s,t)-cut algorithm as a subroutine, a $(2-\frac{2}{k})$-approximation algorithm was proposed in [47]. Based on a novel geometric relaxation, Calinescu et al. [46] obtained a $\frac{3}{2}-\frac{1}{k}$ approximation for the problem. Roughly speaking, the algorithm of Calinescu et al. [46] finds an embedding of the graph on the simplex with the terminals on its corners. A Multiway Cut solution can be extracted out of the embedding by randomly partitioning the simplex. Continuing this line of work, Karger et al. [91] obtained tight integrality gaps for the case $k = 3$, and improved approximation factors for general q (about 1.3438).

For the Zero-Extension problem, Calinescu et al. [30] obtained an $O(\log|\mathsf{L}|)$ approximation algorithm, where L is the set of terminals. The approximation factor was improved to $O(\log|\mathsf{L}|/\log\log|\mathsf{L}|)$ in [53] using a better analysis. The ideas from the Zero-Extension problem [30, 53] have found further applications in metric embeddings [108] and in analysis [113].

Motivated by applications in computer vision Kleinberg et al. [107] introduced the Metric Labeling problem. Using an approximate representation of metrics as a combination of dominating tree metrics [21], Kleinberg et al. [107] also gave an approximation algorithm for Metric Labeling . Its approximation factor can be shown to be $O(\log|\mathsf{L}|)$ using the later improvement of [53] in embedding metrics into dominating tree metrics. A special case of Metric Labeling that is of particular interest is the Uniform Metric Labeling (UML) problem. Here the distance metric d on the labels L is just the uniform metric, i.e., $d(\ell_1,\ell_2) = 1$ for all labels $\ell_1 \neq \ell_2$. For Uniform Metric Labeling a factor 2 approximation algorithm is known [107]. Constant factor approximation algorithms [107, 71, 39, 6] are known for several other special cases of metrics.

Inspired by the geometric relaxation for the Multiway Cut problem, Chekuri et al. [39] proposed an earthmover metric linear relaxation for the Metric Labeling and Zero-Extension problems. They also showed that the integrality gap of the earthmover relaxation is at least as good as the approximation factor of the Kleinberg-Tardos algorithm [107] for general metrics. Archer et al. [6] gave an earthmover relaxation based Metric Labeling algorithm whose performance depends on the decomposability of the metric d. However, even the earthmover linear relaxation proved unsuccessful in obtaining approximation factors better than $O(\log|\mathsf{L}|)$ for Metric Labeling . In fact, for the problems of Metric Labeling and Zero-Extension , integrality gaps of $\Omega(\log|\mathsf{L}|)$ and $\Omega((\log|\mathsf{L}|)^{\frac{1}{2}})$, respectively, were shown for the earthmover relaxation [92].

On the hardness side, the Multiway Cut problem was shown to be APX-hard in [47]. A strong inapproximability result for the Metric Labeling problem was first proven

by Chuzhoy and Naor [45]. Specifically, they showed that for any $\varepsilon > 0$, there is no polynomial time algorithm that approximates the METRIC LABELING problem within a factor of $O((\log |\mathsf{L}|)^{\frac{1}{2}-\varepsilon})$, unless $NP \subseteq DTIME(n^{poly(\log n)})$. Building on this work, Karloff et al.[92] showed that there is no polynomial time algorithm that approximates ZERO-EXTENSION within a factor of $O((\log |\mathsf{L}|)^{\frac{1}{4}-\varepsilon})$, unless $NP \subseteq DTIME(n^{poly(\log n)})$.

We wish to point out that the general conversion from SDP gaps to UG-hardness in [136] applies to the problems METRIC LABELING , ZERO-EXTENSION and MULTIWAY CUT . However, the reduction in [136] makes crucial use of the SDP vectors, and thus would not apply to linear programming integrality gaps. Although both [136] and our work proceed by converting integrality gaps to hardness results, the soundness proofs are very different. For all the problems in this work, the objective is to minimize the number of edges cut. Hence, along the lines of many other UG-hardness results for cut problems [104, 102], the proof uses noise stability of functions.

8.2 Proof Overview

To illustrate the main ideas, we outline the reduction for the 3-way cut problem. Let $G = (V, E)$ be a 3-way cut instance with terminals $\{t_1, t_2, t_3\}$. Here we recall the simplex based linear program (Simplex), which is equivalent to the earthmover linear program EM-LP for MULTIWAY CUT problems.

$$\text{Minimize} \quad \frac{1}{2} \sum_{e=(u,v)\in E} \|X_u - X_v\|_1 \tag{8.1}$$

$$\text{subject to:} \quad X_u^{(1)} + X_u^{(2)} + X_u^{(3)} = 1 \quad \forall u \in V \tag{8.2}$$

$$X_u^{(i)} \geqslant 0 \tag{8.3}$$

$$X_{t_1} = (1,0,0), X_{t_2} = (0,1,0), X_{t_3} = (0,0,1) \tag{8.4}$$

As seen in previous sections, a crucial ingredient in all UG-hardness reductions is a dictatorship test. Recall that a function $\mathsf{F} : \{1,2,3\}^R \to \{1,2,3\}$ is said to be a *dictator* if the function is given by $\mathsf{F}(x) = x_i$ for some fixed i. The input to a dictatorship test consists of a function $\mathsf{F} : \{1,2,3\}^R \to \{1,2,3\}$. The objective is to query the function F at a few locations, and distinguish whether the function is a dictator or far from every dictator. Given a dictatorship test, the UG-hardness reduction usually follows by standard techniques. Roughly speaking, one introduces a vertex for every point in $\{1,2,3\}^R$ and translates the queries made by the dictatorship test in to constraints between these vertices. Therefore, we shall now describe the long code gadget used as part of our reduction. Actually, we convert an integrality gap instance for the Earthmover LP in to a long code gadget.

Let us suppose $G = (V, E)$ is an integrality gap instance for the above linear program. Let $\mathsf{EM\text{-}LP}(G)$ and $\mathrm{opt}(G)$ denote the optimal LP and integral values, respectively. The LP solution associates each vertex v in V with a point X_v on the 3-dimensional simplex. The coordinates of X_v can be thought of as probabilities of assigning the corresponding labels.

From G, we shall construct a 3-way cut instance G' such that :

– There exist special 3-way cuts in G' whose cost equals the linear programming optimum $\mathsf{EM\text{-}LP}(G)$. These cuts will be referred to as *dictator* cuts.

– A 3-way cut solution in G' which is far from every dictator cut pays at least the integral optimum $\mathrm{opt}(G)$.

The vertices of G' are as follows : For each vertex v of G introduce a group Ω_v^R of 3^R vertices. The vertices in Ω_v^R are indexed by vectors $\{1,2,3\}^R$. It is useful to think of Ω_v^R as having a product probability distribution X_v^R on it.

For example, consider the terminal t_1 of the 3-way cut instance G. The corresponding LP assignment X_{t_1} is a corner of the simplex $e_1 = (1,0,0)$. Hence the probability distribution $X_{t_1}^R$ is non-zero on a single vertex $(1,1,1,\ldots,1)$. Similarly for each t_i, the probability distribution $X_{t_i}^R$ on $\Omega_{t_i}^R$ is nonzero only at $(i,i,\ldots,i)$. These special vertices are the terminals of G'. More precisely, the terminals of G' are the vertices $(i,i,\ldots,i) \in \Omega_{t_i}^R$.

A 3-way cut solution assigns to each vertex a label from the set $\{1,2,3\}$. Thus a 3-way cut solution to G' consists of a set of functions $\mathsf{F}_v : \Omega_v^R \to \{1,2,3\}$, one for each vertex $v \in G$. There are two special 3-way cut solutions that will be of interest:

– The set of functions $\mathsf{F}_v(x) = x_i$ for some i. These functions form a feasible 3-cut solution, since they assign different labels to all the terminals. We shall refer to these solutions as *dictator* cuts.

– Each function F_v is a constant function. These solutions will be referred to as *integral cuts*, since they assign a single label to all the 3^R vertices corresponding to a vertex v.

For an edge $e = (v,w)$ in the graph G, we will introduce edges between groups Ω_v^R and Ω_w^R. The edges introduced are such that the *dictator* cuts have a cost close to $\mathsf{EM\text{-}LP}(G)$. We illustrate the basic idea with an example. Let $e = (v,w)$ be an edge in G, with $X_v = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$ $X_w = (\frac{1}{6}, \frac{1}{2}, \frac{1}{3})$. The edges between groups Ω_v^R and Ω_w^R are given by a joint distribution over pairs $x \in \Omega_v^R$, $y \in \Omega_w^R$. Generate each coordinate of x according to the probability distribution X_v. To generate y, we shall mimic the flow of probability mass required to convert distribution X_v into X_w. Specifically, the i^{th} coordinate y_i is generated from x_i using the following distribution:

> If $x_i = 1$, then $y_i = 1$ with probability $\frac{1}{2}$ and $y_i = 3$ with the remaining probability. If $x_i = 2$ or 3, then $y_i = x_i$.

It is easy to check that if x_i is generated according to distribution $X_v = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$, then the distribution of y_i is same as $X_w = (\frac{1}{6}, \frac{1}{2}, \frac{1}{3})$.

Consider a *dictator* cut given by functions $F_v(x) = x_1$ and $F_w(y) = y_1$. The cost of the cut is equal to the probability that $x_1 \neq y_1$ when x, y are generated as above. But this is exactly equal to the total probability mass that *flows* so as to change distribution X_v to X_w. In this case, the probability of $x_1 \neq y_1$, is $\frac{1}{6} = \frac{1}{2}\|X_v - X_w\|_1$. Consequently, the *dictator* cuts pay exactly the LP value $\mathsf{EM\text{-}LP}(G)$.

In an *integral* cut, the group of $[3]^R$ vertices corresponding to a vertex v all have the same label. Intuitively, an *integral* 3-way cut is assigning a label to the vertex v in the original graph G. In fact, an *integral* 3-way cut of G' corresponds to a 3-way cut of G. Thus, if all the functions F_v were constant functions, then the cost of the cut is at least the minimum cost $\mathrm{opt}(G)$ of a 3-way cut of G.

We need to ensure that the functions which are far from a *dictator* function have a cost at least the integral optimum $\mathrm{opt}(G)$. Towards this end, we shall introduce *noise sensitivity* edges. Inside each group Ω_v^R we will introduce edges between pairs (x, y) where x is from the distribution X_v^R, and y is generated by perturbing the coordinates of x. By an appropriate choice of parameters, the total cost of *noise sensitivity* edges overwhelms the remaining edges. Using results on noise stability, if a function $\mathsf{F}_v : \Omega_v^R \to \{1, 2, 3\}$ cuts a small fraction of the *noise sensitivity* edges, then either:

- The function F_v is close to a dictator function (more precisely, it has an influential variable).
- Function F_v is close to a constant function.

Hence, either we obtain a function F_v with an influential variable, or the cost of the cut is $\mathrm{opt}(G)$. Using standard techniques, such a gadget can be used to obtain a UG-hardness result.

Organization The formal definitions of Metric Labeling , Zero-Extension and Multiway Cut are presented in Section 8.3, followed by the definition of the earthmover linear program. The UG-hardness reduction is described in its full entirety in Section 8.4

Mathematical Tools This chapter uses multilinear expansion of functions over product spaces and associated notions of influences and noise operators (Section 3.4). It also crucially uses noise stability bounds for low influence functions on product spaces (Section 3.7).

8.3 Preliminaries

For a positive integer q, $\blacktriangle_q$ denotes the q dimensional simplex. The notation $[q]$ refers to the set $\{1, \ldots, q\}$.

All graphs considered in this chapter are weighted graphs, whose edge-weights sum up to 1. Thus a weighted graph is given by $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ denotes the set of vertices, while $\mathcal{E}$ is a probability distribution over pairs in $\mathcal{V} \times \mathcal{V}$. In particular, the notation $\mathbb{E}_{e \in \mathcal{E}}$ denotes expectation over a random edge chosen from the distribution $\mathcal{E}$.

8.3.1 Problem Definitions

In general, an instance of the Metric Labeling problem is a weighted graph, $\Im = (\mathcal{V}, \mathcal{E})$, a set of labels L and two cost functions,

- **(Assignment Cost)** For each vertex $v \in \mathcal{V}$ and a label ℓ there is a non-negative cost $C(v, \ell)$ of assigning ℓ to vertex v, i.e., the assignment costs are specified by a map $C : \mathcal{V} \times \mathsf{L} \to \mathbb{R}^+$.

– **(Separation Cost)** The labels L have a metric d defined on them and the *separation cost* of assigning labels ℓ_1, ℓ_2 to a pair of vertices u_1, u_2 is simply the distance $d(\ell_1, \ell_2)$ between the labels.

The total cost of a labeling $\mathcal{L} : \mathcal{V} \to \mathsf{L}$, $\text{val}_{\Im}(\mathcal{L})$, is given by

$$\text{val}_{\Im}(\mathcal{L}) = \sum_{v \in \mathcal{V}} C\big(v, \mathcal{L}(v)\big) + \mathop{\mathbb{E}}_{(u,v)=e \in \mathcal{E}} d(\mathcal{L}(u), \mathcal{L}(v)) \,.$$

The objective is to minimize the total cost of the labelling.

In [39], the authors exhibit an approximation-preserving reduction from Metric Labeling to the *restricted* Metric Labeling where all assignment costs are either zero or infinity. Therefore, for the sake of constructing approximation algorithms and hardness results for Metric Labeling , it is enough to consider the special case of *restricted* Metric Labeling . For the sake of simplicity, we will always use Metric Labeling to refer to the restricted version of the problem.

Formally, define the Metric Labeling problem over a finite metric as follows.

Definition 8.3.1. A Metric Labeling problem is specified as $\Lambda = (\mathsf{L}, d)$ where d is a metric over the set of labels L.

We will use q to denote the number of labels $|\mathsf{L}|$

Definition 8.3.2 (Λ-Metric Labeling)**.** An instance $\Im = (\mathcal{V}, \mathcal{E}, \{\mathsf{L}(v)\}_{v \in \mathcal{V}})$ of the Λ-Metric Labeling problem consists of a set of vertices $\mathcal{V}$, a probability distribution $\mathcal{E}$ over pairs from $\mathcal{V} \times \mathcal{V}$ (equivalent to edges with weights) and a family of subsets $\{\mathsf{L}(v)\}_{v \in \mathcal{V}}$ of L. A valid labeling is a mapping $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each vertex, $v \in \mathcal{V}$, $\mathcal{L}(v)$ belongs to $\mathsf{L}(v)$. The cost of a labeling $\mathcal{L}$, $\text{val}_{\Im}(\mathcal{L})$, is

$$\mathop{\mathbb{E}}_{(u,v)=e \in \mathcal{E}} d(\mathcal{L}(u), \mathcal{L}(v)).$$

The optimum value of the instance, $\text{opt}(\Im)$, is the minimum cost labeling for the instance.

An important special case of the Λ-Metric Labeling problem is the Λ-Zero-Extension problem defined below.

Definition 8.3.3 (Λ-Zero-Extension)**.** An instance $\Im = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of Λ-Zero-Extension problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of terminals $\mathsf{L} \subset \mathcal{V}$ with a metric d on them. The objective is to assign each vertex v a terminal $\mathcal{L}(v) \in \mathsf{L}$ such that the following cost is minimized:

$$\mathop{\mathbb{E}}_{(u,v)=e \in \mathcal{E}} d(\mathcal{L}(u), \mathcal{L}(v)).$$

The value of the instance, $\text{opt}(\Im)$ is the minimum cost labeling for the instance.

Observe that a valid solution to the above problem consists of a labeling $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each terminal $t \in \mathsf{L}$, $\mathcal{L}(t) = t$. This corresponds to Λ-Metric Labeling over the

graph $(\mathcal{V}, \mathcal{E})$ with the family of sets $\{\mathsf{L}(v)\}_{v \in \mathcal{V}}$ defined as,

$$\mathsf{L}(v) = \begin{cases} \{v\} & \text{if } v \in \mathsf{L} \\ \mathsf{L} & \text{otherwise} \end{cases}$$

Definition 8.3.4 (Multiway Cut). An instance $\mathfrak{I} = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of Multiway Cut problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of terminals $\mathsf{L} \subset \mathcal{V}$. The objective is to delete a set of edges of minimum weight so as to separate every pair of terminals.

The Multiway Cut problem can be formulated as a labeling problem (with a uniform metric) as follows: A valid multiway cut corresponds to a labeling $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each terminal $t \in \mathsf{L}$, $\mathcal{L}(t) = t$. The cost of such a labeling $\mathcal{L}$, $\mathrm{val}_{\mathfrak{I}}(\mathcal{L})$ is given by $\mathbb{E}_{(u,v)\in\mathcal{E}}\left[\mathbf{1}[\mathcal{L}(u) \neq \mathcal{L}(v)]\right]$. The optimum value of the instance $\mathrm{opt}(\mathfrak{I})$ is the minimum cost labeling for the instance.

A special case of Multiway Cut problem is the q-WayCut for a positive integer q.

Definition 8.3.5 (q-WayCut). An instance $\mathfrak{I} = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of q-WayCut problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of q terminals $\mathsf{L} \subset \mathcal{V}$. The objective is to delete a set of edges of minimum weight so as to separate every pair of terminals.

8.3.2 Earthmover Linear Program for Metric Labeling

The Earthmover linear programming (EM-LP) relaxation for Metric Labeling was introduced by [39]. Let $\mathfrak{I} = (\mathcal{V}, \mathcal{E}, \{\mathsf{L}(v)\}_{v\in\mathcal{V}})$ be an instance of metric labeling. Intuitively, the EM-LP program finds an embedding of the vertices $\mathcal{V}$ on the q dimensional simplex $\blacktriangle_q$. For every vertex v, there is a variable $X_v = (X_{v,\ell})_{\ell\in[q]}$ which is a point on the q-ary simplex $\blacktriangle_q$. The point X_v represents the probability distribution of each label being assigned to v. For example, each corner of the simplex represents a particular label.

The labeling constraint $\mathcal{L}(v) \in \mathsf{L}(v)$ is enforced by a linear constraint on the probability distribution X_v. Specifically, one can include the following constraints,

$$X_{v,\ell} = 0 \qquad \text{for all } \ell \notin \mathsf{L}(v)\,.$$

These labeling constraints force the point X_v to lie in the face containing the allowed labels $\mathsf{L}(v)$, denoted by $\blacktriangle(\mathsf{L}(v))$. The objective is to minimize the weighted sum of the *earthmover* distance between adjacent vertices.

Definition 8.3.6 (Earthmover Distance). Given two points $X, Y \in \blacktriangle_q$, and a metric $d(i,j)$ on $[q]$, the earthmover distance, $d_{\bowtie}(X,Y)$ is given by the optimal value of the following LP:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i,j\in[q]} d(i,j)\mu_{ij} \\ \text{s.t.} \quad & \sum_i \mu_{ij} = Y_j \qquad \sum_j \mu_{ij} = X_i \qquad \forall i,j \in [q] \\ & \mu_{ij} \geqslant 0 \end{aligned}$$

In other words, the earthmover distance is the minimum cost of moving the probability mass from distribution X to Y, given the distance metric d on the labels. It is easy to see that this defines a metric on the simplex $\blacktriangle_q$. Thus, the earthmover distance generalizes a metric on q points to a metric on $\blacktriangle_q$ such that the distance between corner points is the same as the original metric. In this notation, the linear program of [39] is simply:

$$\begin{array}{lll} \text{Minimize} & \mathop{\mathbb{E}}_{(u,v)\in\mathcal{E}} d_{\bowtie}(X_u, X_v) & \text{(EM-LP)} \\ \text{s.t.} & X_u \in \blacktriangle(\mathsf{L}(u)) & \forall u \in \mathcal{V} \end{array}$$

Definition 8.3.7. A feasible solution $(\boldsymbol{X}, \boldsymbol{\mu})$ to the EM-LP relaxation is said to be α-smooth if $\alpha = \min_{v\in\mathcal{V}, X_{v,i}\neq 0} X_{v,i}$

8.3.3 Analytic Notions

In this chapter, we will require notions of influence and noise stability for functions over product spaces. We refer the reader to Section 3.4 for an introduction to these notions. Furthermore, we will be using the following noise stability bound which is a consequence of the invariance principle (Section 3.7).

We recall the noise stability bound (Theorem 3.6) for the convenience of the reader here. The Gaussian noise stability Γ_ρ is defined as follows:

Definition 8.3.8. Given $\mu \in [0,1]$, let $t = \Phi^{-1}(\mu)$ where Φ denotes the distribution function of the standard Gaussian. Then,

$$\Gamma_\rho(\mu) = \mathbb{P}\mathrm{r}[X \leqslant t, Y \leqslant t],$$

where (X, Y) is a two-dimensional Gaussian vector with covariance matrix $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

Theorem 8.4. *Let Ω be a finite probability space with the least non-zero probability of an atom at least α. For every $\mu, \varepsilon, \gamma, \eta > 0$ there exists τ such that the following holds: For every function $\mathcal{F} : \Omega^R \to [0,1]$ with $\mu = \mathbb{E}[\mathcal{F}]$ and $\mathrm{Inf}_\ell(\mathrm{T}_{1-\gamma}\mathcal{F}) < \tau$ for all $\ell \in [R]$,*

$$\langle \mathcal{F}, \mathrm{T}_{1-\varepsilon}\mathcal{F}\rangle = \mathop{\mathbb{E}}_{\boldsymbol{z}\in\Omega^R} [\mathcal{F}(\boldsymbol{z})\mathrm{T}_{1-\varepsilon}\mathcal{F}(\boldsymbol{z})] \leqslant \Gamma_{1-\varepsilon}(\mu) + \eta\,.$$

8.4 The Reduction

In this section, we shall describe the reduction from Unique Games to Metric Labeling . The same reduction applies with minor changes for the Multiway Cut and Zero-Extension problems.

Let $\Im = (\mathcal{V}, \mathcal{E}, \{\mathsf{L}(v)\}_{v\in\mathcal{V}})$ be an instance of a Λ-Metric Labeling problem for a finite metric $\Lambda = (\mathsf{L}, d)$. Without loss of generality, we may assume that the set of labels

$\mathsf{L} = [q]$. Let $(\boldsymbol{X}, \boldsymbol{\mu})$ denote an α-smooth feasible solution to EM-LP relaxation of $\Im$. For each vertex $v \in \mathcal{V}$, let Ω_v denote the probability space over the set of atoms are $[q]$ given by, $\mathbb{P}\mathrm{r}_{a \in \Omega_a}[a = i] = X_{a,i}$

Let $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi, [R])$ be a UG instance. We construct a Λ-Metric Labeling instance $\Im(\Phi)$ as in Figure 8.1.

Proof. (of Theorem 8.1) For Metric Labeling the proof directly follows from Theorems 8.5 and 8.6. As stated, the instance produced by the reduction also has the same set of labels L.

For Zero-Extension and Multiway Cut , the instances produced by the reduction have too many terminals. Specifically, for every vertex $v \in \mathcal{V}_\Phi$, terminal $t \in \mathsf{L}$ and $\mathbf{x} \in [q]^R$ there are q^R vertices of the form $(v, t, \mathbf{x}) \in V(\Im(\Phi))$. For every vertex $(v, t, \mathbf{x})$, the set of allowed labels in $\Im(\Phi)$ is just $\{t\}$. Using standard techniques the graph $\Im(\Phi)$ can be modified into $\Im(\Phi)'$ with the correct set of terminals.

Introduce a new vertex in $V(\Im(\Phi)')$ for each label in $t \in \mathsf{L}$. These new vertices are the terminals for $\Im(\Phi)'$. For every vertex $(w, t, \mathbf{x})$ with $t \in \mathsf{L}$, introduce an edge of infinite(sufficiently high) cost between t and $(w, t, \mathbf{x})$. A solution to the instance $\Im(\Phi)'$ will not cut any of the edges of infinite cost. This simulates the constraint that $(w, t, \mathbf{x})$ is assigned label t. ■

8.4.1 Completeness

Theorem 8.5. *For every $\varepsilon, \delta > 0$, given a UG instance Φ that is $1 - \delta$ strongly satisfiable and an integrality gap instance Ψ, the value of the metric labeling instance $(\Im(\Phi), \mathsf{L}, d)$ obtained from the reduction is at most $(1 - \delta)(\varepsilon^{7/8}$ EM-LP$(\Im) + \varepsilon) + \delta$.*

Proof. Let $\mathcal{A} : \mathcal{W}_\Phi \to [R]$ denote an assignment to the UG instance Φ. Consider the labeling $\mathcal{L}$ to $\Im(\Phi)$ that sets $\mathcal{L}(u, a, \mathbf{x}) = x_{\mathcal{A}(u)}$. It is easy to check that $\mathcal{L}$ is a valid labeling for the instance $\Im(\Phi)$. Then, the cost of the labeling $\mathcal{L}$ is:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{w,v_1,v_2}\Bigg[\varepsilon^{7/8} \cdot \mathop{\mathbb{E}}_{(a,b)\in\mathcal{E}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}}\Big[d(\mathcal{L}(v_1, a, \pi_1(\mathbf{x})), \mathcal{L}(v_2, b, \pi_2(\mathbf{y})))\Big] \\
&\qquad\qquad + (1 - \varepsilon^{7/8}) \cdot \mathop{\mathbb{E}}_{a\in\mathcal{V}} \mathop{\mathbb{E}}_{\mathbf{x}\sim_{1-\varepsilon}\mathbf{y}}\Big[d(\mathcal{L}(v_1, a, \pi_1(\mathbf{x})), \mathcal{L}(v_2, a, \pi_2(\mathbf{y})))\Big]\Bigg] \\
= \mathop{\mathbb{E}}_{w,v_1,v_2}&\Bigg[\varepsilon^{7/8} \cdot \mathop{\mathbb{E}}_{(a,b)\in\mathcal{E}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}}\Big[d(x_{\pi_1(\mathcal{A}(v_1))}, y_{\pi_2(\mathcal{A}(v_2))})\Big] \\
&\qquad\qquad + \ (1 - \varepsilon^{7/8}) \cdot \mathop{\mathbb{E}}_{a\in\mathcal{V}} \mathop{\mathbb{E}}_{\mathbf{x}\sim_{1-\varepsilon}\mathbf{y}}\Big[d(x_{\pi_1(\mathcal{A}(v_1))}, y_{\pi_2(\mathcal{A}(v_2))})\Big]\Bigg].
\end{aligned}$$

With probability $1 - \delta$ over the choice of vertex w, the Unique Games assignment $\mathcal{A}$ satisfies all the edges incident at w. Let us refer to these vertices w as *good* vertices. For a *good* w, for all choices of v_1, v_2, $\pi_1(\mathcal{A}(v_1)) = \pi_2(\mathcal{A}(v_2)) = \mathcal{A}(w)$. Thus the expected cost for

The vertices of $\Im(\Phi)$ are $V(\Im(\Phi)) = \mathcal{V}_\Phi \times \mathcal{V} \times [q]^R$.
The set of allowed labels for a vertex $(v, a, \mathbf{x}) \in V(\Im(\Phi))$ is $\mathsf{L}(a)$.
The weight of an edge is the probability it is output by the following test:

- Pick $w \in \mathcal{W}_\Phi$ at random, and two of its neighbors v_1, v_2 independently at random. Let us denote by π_1, π_2, the permutations $\pi_{v_1 \leftarrow w}, \pi_{v_2 \leftarrow w}$ on edges (w, v_1) and (w, v_2) respectively.
- With probability $\varepsilon^{7/8}$ perform the edge test, otherwise (with probability $1 - \varepsilon^{7/8}$) perform the vertex test.

Edge test: (with probability $\varepsilon^{7/8}$)

- Sample an edge $e = (a, b) \in \mathcal{V} \times \mathcal{V}$ from the probability distribution $\mathcal{E}$.
- Sample $\mathbf{x}, \mathbf{y} \in [q]^R$, by generating each coordinate $(x^{(i)}, y^{(i)}) \in [q]^2$ from the distribution μ_e. Clearly, we have $\mathbf{x} \in \Omega_a^R$, $\mathbf{y} \in \Omega_b^R$.
- Output the edge $(v_1, a, \pi_1(\mathbf{x})) \leftrightarrow (v_2, b, \pi_2(\mathbf{y}))$.

Vertex test: (with probability $1 - \varepsilon^{7/8}$)

- Pick a uniformly random vertex $a \in \mathcal{V}$.
- Sample $\mathbf{x} \in \Omega_a^R$. Sample $\mathbf{y} \in [q]^R$ as follows:

$$y_i = \begin{cases} x_i & \text{with probability } 1 - \varepsilon\,, \\ \text{a new sample from } \Omega_a & \text{with probability } \varepsilon\,. \end{cases} \tag{8.5}$$

- Output the edge $(v_1, a, \pi_1(\mathbf{x})) \leftrightarrow (v_2, a, \pi_2(\mathbf{y}))$.

Figure 8.1: The reduction.

a *good* vertex is given by

$$\delta \cdot \mathop{\mathbb{E}}_{(a,b)\in\mathcal{E}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \left[d(x_{\mathcal{A}(w)}, y_{\mathcal{A}(w)})\right] + (1-\delta) \cdot \mathop{\mathbb{E}}_{a\in\mathcal{V}} \mathop{\mathbb{E}}_{\mathbf{x}\sim_{1-\varepsilon}\mathbf{y}} \left[d(x_{\mathcal{A}(w)}, y_{\mathcal{A}(w)})\right] \leqslant \delta \cdot \mathsf{EM\text{-}LP}(\Im) + (1-\delta)\cdot\varepsilon.$$

For an arbitrary vertex $w \in \mathcal{W}_\Phi$, the expected cost is always bounded by 1 since all the distances are bounded by 1. Thus, the total cost of the labeling $\mathcal{L}$ is at most $(1-\delta)\cdot\left(\varepsilon^{7/8}\mathsf{EM\text{-}LP}(\Im)+\varepsilon\right)+\delta$. ■

Corollary 8.5.1. *For every $\eta > 0$, there exists $\varepsilon, \delta > 0$ such that, the value of the metric labeling instance $\Im(\Phi)$ obtained as in Theorem 8.5 is at most $\varepsilon^{7/8}$EM-LP$(\Im)(1+\eta)$.*

Proof. Setting $\varepsilon < (\eta\mathsf{EM\text{-}LP}(\Im)/2)^8$ and $\delta < \varepsilon^{7/8}\mathsf{EM\text{-}LP}(\Im)\eta/4$ in Theorem 8.5 gives the required result. ■

8.4.2 Soundness

Let $\mathcal{L}$ be a labeling of the instance $\Im(\Phi)$. Let ε be as defined in the reduction. For each $v \in \mathcal{V}_\Phi$, $a \in \mathcal{V}$, define q functions $\mathcal{F}^i_{v,a} : [q]^R \to [0,1]$ as follows,

$$\mathcal{F}^i_{v,a}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathcal{L}(v,a,\mathbf{x}) = i\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

For each $w \in \mathcal{W}_\Phi$, $a \in \mathcal{V}$, define q functions $\mathcal{G}^i_{w,a} : [q]^R \to [0,1]$ as follows

$$\mathcal{G}^i_{w,a}(\mathbf{x}) = \mathop{\mathbb{E}}_{v\in N(w)} \left[\mathcal{F}^i_{v,a}(\pi_{v\leftarrow w}(\mathbf{x}))\right].$$

Observe that for any $\mathbf{x} \in [q]^R$ and $w \in \mathcal{W}_\Phi$,

$$\sum_{i=1}^{q} \mathcal{G}^i_{w,a}(\mathbf{x}) = \sum_{i=1}^{q} \mathop{\mathbb{E}}_{v\in N(w)} \left[\mathcal{F}^i_{v,a}(\pi_{v\leftarrow w}(\mathbf{x}))\right] = \mathop{\mathbb{E}}_{v\in N(w)} \left[\sum_{i=1}^{q} \mathcal{F}^i_{v,a}(\pi_{v\leftarrow w}(\mathbf{x}))\right] = 1\,.$$

Define $\theta^i_{w,a} = \mathbb{E}_{\mathbf{x}}[\mathcal{G}^i_{w,a}(\mathbf{x})]$ where $\mathbf{x}$ is distributed according to the probability distribution of Ω^R_a. Further, define $\theta_{w,a} = (\theta^1_{w,a}, \ldots, \theta^q_{w,a})$. Hence, for all w, a we have

$$\sum_{i=1}^{q} \theta^i_{w,a} = \sum_{i=1}^{q} \mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i_{w,a}(\mathbf{x})] = \mathop{\mathbb{E}}_{\mathbf{x}} \left[\sum_{i=1}^{q} \mathcal{G}^i_{w,a}(\mathbf{x})\right] = 1.$$

Thus, $\theta_{w,a} \in \blacktriangle_q$, i.e., it defines an embedding in the simplex. We will drop w and a when they are clear from the context.

For a vertex $w \in \mathcal{W}_\Phi$, let $\mathrm{val}^{\mathsf{Edge}}_w(\mathcal{L})$ and $\mathrm{val}^{\mathsf{Vertex}}_w(\mathcal{L})$ denote the expected cost incurred by the edge and vertex tests respectively when the verifier chooses vertex w. We can write the cost of the labeling $\mathcal{L}$ as $\mathbb{E}_w\left[\varepsilon^{7/8}\,\mathrm{val}^{\mathsf{Edge}}_w(\mathcal{L}) + (1-\varepsilon^{7/8})\mathrm{val}^{\mathsf{Vertex}}_w(\mathcal{L})\right]$.

We will show that for most choices $w \in \mathcal{W}_\Phi$, $a \in \mathcal{V}$, either the functions $\mathcal{G}^i_{w,a}$ have an influential variable or they are close to constant functions. More precisely, we show that if the functions are neither constant nor have influential variables, then the cost of the vertex test on w, a is overwhelmingly large.

Lemma 8.5.1. *Fix $w \in \mathcal{W}_\Phi$, $a \in \mathcal{V}$ and let $\{\mathcal{G}^i\}_{i\in[q]}$ denote the family of functions associated with (w, a). For every $\varepsilon > 0$ we have*

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim_{(1-\varepsilon)}\mathbf{y}}\Big[\sum_{i,j\in\mathsf{L}} d(i,j)\mathcal{G}^i(\mathbf{x})\mathcal{G}^j(\mathbf{y})\Big] \geqslant \beta \sum_i \Big(\mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i] - \mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i T_{1-\varepsilon}(\mathcal{G}^i)]\Big),$$

where $\beta = \min_{i\neq j} d(i,j)$. Further, for all $\varepsilon, \zeta > 0$, there exists γ, τ such that if $\operatorname{Inf}_j(T_{1-\gamma}\mathcal{G}^i) < \tau$ for all $i \in [q], j \in [R]$, then

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim_{1-\varepsilon}\mathbf{y}}\Big[\sum_{i,j\in\mathsf{L}} d(i,j)\mathcal{G}^i(\mathbf{x})\mathcal{G}^j(\mathbf{y})\Big] \geqslant \beta \sum_i \Big(\theta^i - \Gamma_{1-\varepsilon}(\theta^i)\Big) - \zeta.$$

Proof. Since $\sum_{j\neq i}\mathcal{G}^j(\mathbf{x}) = 1 - \mathcal{G}^i(\mathbf{x})$, we get

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{x}\sim_{1-\varepsilon}\mathbf{y}}\Big[\sum_{i,j\in\mathsf{L}} d(i,j)\mathcal{G}^i(\mathbf{x})\mathcal{G}^j(\mathbf{y})\Big] &\geqslant \beta \mathop{\mathbb{E}}_{\mathbf{x}\sim_{1-\varepsilon}\mathbf{y}}\Big[\sum_i \mathcal{G}^i(\mathbf{x})\big(1-\mathcal{G}^i(\mathbf{y})\big)\Big]\\ &= \beta\sum_i\Big(\mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i] - \mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i T_{1-\varepsilon}(\mathcal{G}^i)]\Big).\end{aligned}$$

To derive the second part of the lemma, apply Theorem 8.4 on each of the functions $\mathcal{G}^i$ with the error term ζ/q instead of ζ:

$$\mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i] - \mathop{\mathbb{E}}_{\mathbf{x}}[\mathcal{G}^i T_{1-\varepsilon}(\mathcal{G}^i)] \geqslant \Big(\theta^i - \Gamma_{1-\varepsilon}(\theta^i)\Big) - \zeta/q.$$

Summing up over all i, we obtain the desired result. ∎

The following lemma lower bounds the cost of the vertex test, when none of the functions $\mathcal{G}^i$ are neither constant nor have an influential variable.

Lemma 8.5.2. *There exists an ε_0 such that for all $\varepsilon < \varepsilon_0$, for all $(\theta_1, \theta_2, \ldots, \theta_q) \in \blacktriangle_q$ such that $\max_i \theta_i < 1 - \varepsilon^{1/4}$,*

$$\sum_i [\theta_i - \Gamma_{1-\varepsilon}(\theta_i)] = \Omega(\varepsilon^{3/4}).$$

Proof. Let $\theta = \max_i \theta_i$, then we have $1 - \varepsilon^{1/4} > \theta$. By choosing $\varepsilon < \frac{1}{q^4}$, one can ensure that $\theta > \varepsilon^{1/6}$. Observe that Γ satisfies, $\Gamma_{1-\varepsilon}(\mathbf{x}) \leqslant x$ for all $\mathbf{x} \in [0, 1]$. Thus we can write,

$$\sum_i [\theta_i - \Gamma_{1-\varepsilon}(\theta_i)] > \theta - \Gamma_{1-\varepsilon}(\theta).$$

Using known estimates, (see Corollary 10.4 in [99]), we have:

$$\Gamma_{1-\varepsilon}(\theta) \leqslant \theta \left[1 - \sqrt{\Omega(\varepsilon \log(1/\theta))}\ \right] + o(\theta).$$

Thus, setting $\theta > \varepsilon^{1/4}$, we have the required result:

$$\sum_i [\theta_i - \Gamma_\rho(\theta_i)] \geqslant \Omega(\theta\sqrt{\varepsilon \log(1/\theta)}\ ) \geqslant \Omega(\varepsilon^{3/4}).$$

■

Lemma 8.5.3. *For any vertex $w \in \mathcal{W}_\Phi$,*

$$\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) \geqslant \mathop{\mathbb{E}}_{e=(a,b)\in\mathcal{E}} d_{\bowtie}(\theta_{w,a}, \theta_{w,b}).$$

Proof. Fix an edge $e = (a, b) \in \mathcal{E}$. Define a probability distribution $\mu'_e \in [q]^2$ as follows,

$$\mu'_e(i, j) = \mathop{\mathbb{E}}_{\mathbf{x}\sim_e\mathbf{y}} [\mathcal{G}^i_{w,a}(\mathbf{x})\mathcal{G}^j_{w,b}(\mathbf{y})].$$

Then, $\sum_i \mu'_e(i, j) = \theta^j_{w,b}$; $\sum_j \mu'_e(i, j) = \theta^i_{w,b}$. From Definition 8.3.6 we have

$$\sum_{i,j} d(i, j)\mu'_e(i, j) \geqslant d_{\bowtie}(\theta_{w,a}, \theta_{w,b}).$$

Recall that $\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L})$ is given by

$$\begin{aligned}\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) \ &= \ \mathop{\mathbb{E}}_{e=(a,b)\in\mathcal{E}} \sum_{ij} d(i, j) \mathop{\mathbb{E}}_{\mathbf{x}\sim_e\mathbf{y}} [\mathcal{G}^i_{w,a}(\mathbf{x})\mathcal{G}^j_{w,b}(\mathbf{y})] \\ = \ \mathop{\mathbb{E}}_{e=(a,b)\in\mathcal{E}} \sum_{ij} d(i, j)\mu'_e(i, j) \ &\geqslant \ \mathop{\mathbb{E}}_{e=(a,b)\in\mathcal{E}} d_{\bowtie}(\theta_{w,a}, \theta_{w,b}).\end{aligned}$$

■

Lemma 8.5.4. *There exists $\varepsilon_1 > 0$, such that for every $\varepsilon < \varepsilon_1$, there exist τ, γ, such that for all $w \in \mathcal{W}_\Phi$, if $\mathrm{Inf}_j(T_{1-\gamma}\mathcal{G}^i_{w,a}) < \tau$ for all i, j, a, then one of the following inequalities holds:*

$$\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) \geqslant \mathrm{opt}(\Im)(1 - 4\varepsilon^{1/8})$$

or

$$\mathrm{val}_w^{\mathsf{Vertex}}(\mathcal{L}) \geqslant (\beta\varepsilon^{3/4} - \varepsilon)/m.$$

Proof. Let γ, τ be as obtained from Lemma 8.5.1 by setting $\zeta = \varepsilon$. Since w is fixed we shall denote $\mathcal{G}^i_{w,a}$ by $\mathcal{G}^i_a$. Then, there are two possibilities:

Case 1: For all a, the functions $\mathcal{G}^i_a$ are near constant, i.e there is a labeling function $\mathcal{L} : \mathcal{V} \to [q]$ such that $\theta^{\mathcal{L}(a)}_{w,a} > 1 - \varepsilon^{1/4}$ for all a.

Set $\theta = \varepsilon^{1/8}$. A simple averaging argument shows that for every a, $\mathcal{G}_a^{\mathcal{L}(a)}(\mathbf{x}) > 1-\theta$ for a $1-\theta$ fraction of $\mathbf{x}$. By a union bound, if $\mathbf{x}, \mathbf{y}$ are generated from $\mathbf{x} \sim_e \mathbf{y}$, both $\mathcal{G}_a^{\mathcal{L}(a)}(\mathbf{x}), \mathcal{G}_b^{\mathcal{L}(b)}(\mathbf{y})$ are greater than $1-\theta$ with probability $1-2\theta$. Thus the cost of the edge test is:

$$\begin{aligned}\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) &= \mathop{\mathbb{E}}_{(a,b)\in\mathcal{E}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \Big[\sum_{i,j\in\mathsf{L}} d(i,j)\mathcal{G}^i_{w,a}(\mathbf{x})\mathcal{G}^j_{w,b}(\mathbf{y}) \Big] \\ &\geqslant (1-2\theta)(1-\theta)^2 \mathop{\mathbb{E}}_{(a,b)\in\mathcal{E}} [d(\mathcal{L}(a),\mathcal{L}(b))]\,.\end{aligned}$$

It is easy to check that the labeling $\mathcal{L}$ is a valid metric labeling solution for $\Im$. Hence we have,

$$\mathop{\mathbb{E}}_{(a,b)\in\mathcal{E}} [d(\mathcal{L}(a),\mathcal{L}(b))] \geqslant \mathrm{opt}(\Im).$$

Substituting we get $\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) \geqslant \mathrm{opt}(\Im)(1-4\theta)$.

Case 2: There exists $b \in \mathcal{V}$ such that for all i, $\theta^i_{w,b} \leqslant 1-\varepsilon^{1/4}$. Then, the vertex cost is:

$$\begin{aligned}\mathrm{val}_w^{\mathsf{Vertex}}(\mathcal{L}) &= \mathop{\mathbb{E}}_{a\in\mathcal{V}} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \sum_{i,j\in\mathsf{L}} \Big[d(i,j)\mathcal{G}^i_{w,a}(\mathbf{x})\mathcal{G}^j_{w,a}(\mathbf{y}) \Big] \\ &\geqslant \frac{1}{m} \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \sum_{i,j\in\mathsf{L}} \Big[d(i,j)\mathcal{G}^i_{w,b}(\mathbf{x})\mathcal{G}^j_{w,b}(\mathbf{y}) \Big] \\ &\geqslant \frac{1}{m} \left(\beta \sum_i (\theta^i_{w,b} - \Gamma_{1-\varepsilon}(\theta^i_{w,b})) - \varepsilon \right) \\ &\geqslant \frac{1}{m} \left(\beta\varepsilon^{3/4} - \varepsilon \right).\end{aligned}$$

■

Theorem 8.6. *For every $\eta > 0$, for sufficiently small $\varepsilon, \delta > 0$, if the UG instance Φ is at most δ-satisfiable, then the optimum assignment to $\Im(\Phi)$ has value at least $\varepsilon^{7/8}\mathrm{opt}(\Im)(1-\eta)$, i.e.,*

$$\mathrm{opt}(\Phi) \leqslant \delta \qquad \Rightarrow \qquad \mathrm{opt}(\Im(\Phi)) \geqslant \varepsilon^{7/8}\mathrm{opt}(\Im)(1-\eta)\,.$$

Proof. Set $\varepsilon \leqslant \min\{(\eta/12)^8, (\beta/4m\mathrm{opt}(\Im))^8, \varepsilon_1\}$. Let γ, τ be as obtained from Lemma 8.5.4. For every vertex $w \in \mathcal{W}_\Phi$, one of the following is true:

- There exists $a \in \mathcal{V}$, $i \in [q]$, $j \in [R]$ such that $\mathrm{Inf}_j(T_{1-\gamma}\mathcal{G}^i_{w,a}) \geqslant \tau$.
- $\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) \geqslant \mathrm{opt}(\Im)(1-4\varepsilon^{1/8}) \geqslant \mathrm{opt}(\Im)(1-\eta/3)$.
- $\mathrm{val}_w^{\mathsf{Vertex}}(\mathcal{L}) \geqslant (\beta\varepsilon^{3/4} - \varepsilon)/m \geqslant \varepsilon^{7/8}\mathrm{opt}(\Im)$.

Thus, if none of the functions $\{\mathcal{G}^i_{w,a}\}_{a\in\mathcal{V},i\in[q]}$ associated with a particular $w \in \mathcal{W}_\Phi$ have any influential coordinate,

$$\varepsilon^{7/8}\mathrm{val}_w^{\mathsf{Edge}}(\mathcal{L}) + (1-\varepsilon^{7/8})\mathrm{val}_w^{\mathsf{Vertex}}(\mathcal{L}) \geqslant \varepsilon^{7/8}\mathrm{opt}(\Im)(1-\eta/3).$$

Call a vertex $w \in \mathcal{W}_\Phi$ *good* if at least one of the functions associated with it has an influential variable. More precisely, if there exists a, i, j such that $\mathrm{Inf}_j(T_{1-\gamma}\mathcal{G}^i_{w,a}) \geqslant \tau$. If $\mathrm{val}_{\mathcal{L}}(\Im(\Phi)) \leqslant \varepsilon^{7/8}\mathrm{opt}(\Im)(1-\eta)$, then at least $\eta/2$ fraction of the vertices are *good*.

We will define a labelling $\mathcal{A}$ for vertices of the UG instance as follows:

$$\mathsf{L}(w) = \{j \in [R] \;\mid\; \exists\, i, a;\ \mathrm{Inf}_j(T_{1-\gamma}\mathcal{G}^i_{w,a}) \geqslant \tau\} \qquad \text{(for every } w \in \mathcal{W}_\Phi\text{)},$$
$$\mathsf{L}(v) = \{j \in [R] \;\mid\; \exists\, i, a;\ \mathrm{Inf}_j(T_{1-\gamma}\mathcal{F}^i_{v,a}) \geqslant \tau/2\} \qquad \text{(for every } v \in \mathcal{V}_\Phi\text{)}.$$

For each $u \in \mathcal{W}_\Phi \cup \mathcal{V}_\Phi$, assign a label uniformly at random from $\mathsf{L}(u)$.

We will analyze the fraction of edges in the UG instance satisfied in expectation by the UG assignment $\mathcal{A}$. Fix a *good* vertex $w \in \mathcal{W}_\Phi$ with the corresponding a, i, j satisfying $\mathrm{Inf}_j(T_{1-\gamma}\mathcal{G}^i_{w,a}) \geqslant \tau$. By convexity of influences (Fact 3.0.13), we have

$$\mathop{\mathbb{E}}_{v\in N(w)}\left[\mathrm{Inf}_{\pi^{-1}_{v\leftarrow w}(j)}(T_{1-\gamma}\mathcal{F}^i_{v,a})\right] \geqslant \mathrm{Inf}_j(T_{1-\gamma}\mathcal{G}^i_{w,a}) \geqslant \tau$$

For a good vertex w and a label $\ell \in \mathsf{L}(w)$, for at least $\tau/2$ fraction of the neighbors $v \in N(w)$, we will have $\pi^{-1}_{v\leftarrow w}(\ell) \in \mathsf{L}(v)$. Thus, for a good vertex w, for at least $\tau/2$ fraction of its neighbors $v \in N(w)$, $\mathcal{A}$ satisfies the edge (v, w) with probability at least $\frac{1}{|\mathsf{L}(v)||\mathsf{L}(w)|}$. Using Lemma 3.0.2, the sizes of the label sets $\mathsf{L}(u)$ are at most $2qm/\gamma\tau$. Thus, such an edge (v, w) is satisfied with probability at least $\gamma^2\tau^2/4q^2m^2$. The expected weight of edges in Φ satisfied by the assignment $\mathcal{A}$ is at least $(\eta/2)(\tau/2)(\tau^2\gamma^2/4q^2m^2) \geqslant \frac{\gamma^2\eta\tau^3}{16m^2q^2}$. Choosing $\delta < \frac{\gamma^2\eta\tau^3}{16m^2q^2}$ gives the required result.

■

Chapter 9

ORDERING CONSTRAINT SATISFACTION PROBLEMS

9.1 Introduction

Given a directed acyclic graph G, one can efficiently order ("topological sort") its vertices so that all edges go forward from a lower ranked vertex to a higher ranked vertex. But what if a few, say fraction ε, of edges of G are reversed? Can we detect these "errors" and find an ordering with few back edges? Formally, given a directed graph whose vertices admit an ordering with many, i.e., $1-\varepsilon$ fraction, forward edges, can we find a good ordering with fraction α of forward edges (for some $\alpha \to 1$)? This is equivalent to finding a subgraph of G that is acyclic and has many edges, and hence this problem is called the Maximum Acyclic Subgraph (Maximum Acyclic Subgraph) problem.

It is trivial to find an ordering with fraction $1/2$ of forward edges: take the better of an arbitrary ordering and its reverse. This gives a factor $1/2$ approximation algorithm for Maximum Acyclic Subgraph. (This is also achieved by picking a *random* ordering of the vertices.) Despite much effort, no efficient ρ-approximation algorithm for a constant $\rho > 1/2$ has been found for Maximum Acyclic Subgraph. The existence of such an algorithm has been a long-standing and central open problem in the theory of approximation algorithms. In this chapter, we prove a strong hardness result that rules out the existence of such an approximation algorithm assuming the Unique Games Conjecture. Formally, we show the following:

Theorem 9.1. *For every constant $\gamma > 0$, given a directed graph G with m edges, it is UG-hard to distinguish between the following two cases:*

1. *There is an ordering of the vertices of G with at least $(1-\gamma)m$ forward edges (or equivalently, G has an acyclic subgraph with at least $(1-\gamma)m$ edges).*

2. *For every ordering of the vertices of G, there are at most $(1/2+\gamma)m$ forward edges (or equivalently, every subgraph of G with more than $(1/2+\gamma)m$ edges contains a directed cycle).*

To the best of our knowledge, the above is the first tight hardness of approximation result for an ordering/permutation problem. As an immediate consequence, we obtain the following hardness result for the complementary problem of Min Feedback Arc Set, where the objective is to minimize the number of back edges.

Corollary 9.1.1. *For every $C > 0$, it is UG-hard to find a C-approximation to the* Min Feedback Arc Set *problem.*

Combining the unique game integrality gap instance of Khot-Vishnoi [104] along with the UG reduction, we obtain SDP integrality gaps for Maximum Acyclic Subgraph problem. Our integrality gap instances also apply to a related SDP relaxation studied by Newman [130]. This SDP relaxation was shown to obtain an approximation better than half on random graphs which were previously used to obtain integrality gaps for a natural linear program [129].

Building on these techniques and the ideas from Chapter 7, we obtain UGC based hardness results for the class of Ordering Constraint Satisfaction Problems (OCSP). An OCSP

Λ with arity k is specified by a family of predicates over the set of permutations on k elements. An instance of the Λ-OCSP consists of a set of variables $\mathcal{V}$ and a set of ordering constraints on them. Each ordering constraint consists of a predicate from the family Λ applied to an ordered tuple of variables from $\mathcal{V}$. The objective is to find an ordering of the variables $\mathcal{V}$ that satisfies the maximum number of constraints. Our results hold in a more general setting where the predicates are replaced by bounded payoff functions which could take both positive and negative values.

In order to state the result for general OCSPs, we present the following definition.

Definition 9.1.1. For an Ordering Constraint Satisfaction problem Λ, define $\mathsf{Gap}_\Lambda(c), \mathsf{UGhard}_\Lambda(c)$ as follows:
$\mathsf{Gap}_\Lambda(c)$- The minimum value of the integral optimum over all instances $\Im$ with SDP value c.
$\mathsf{UGhard}_\Lambda(c)$- The minimum value it is UG-hard to distinguish between instances with objective value c from an instance with objective value $\mathsf{UGhard}_\Lambda(c)$.

With these definitions in place, we can state our UG hardness result for OCSPs as follows,

Theorem 9.2. *(UGC Hardness) For every constant $\eta > 0$, and every Ordering CSP Λ:*

$$\mathsf{UGhard}_\Lambda(c) \leqslant \mathsf{Gap}_\Lambda(c+\eta) + \eta \qquad \forall c \in (-1, 1)$$

9.1.1 Related work

Maximum Acyclic Subgraph is a classic optimization problem, figuring in Karp's early list of NP-hard problems [94]; the problem remains NP-hard on graphs with maximum degree 3, when the in-degree plus out-degree of any vertex is at most 3. Maximum Acyclic Subgraph is also complete for the class of permutation optimization problems, MAX SNP[π], defined in [134], that can be approximated within a constant factor. It is shown in [129] that Maximum Acyclic Subgraph is NP-hard to approximate within a factor greater than $\frac{65}{66}$.

Turning to algorithmic results, the problem is known to be efficiently solvable on planar graphs [119, 89] and reducible flow graphs [138]. Berger and Shor [24] gave a polynomial time algorithm with approximation ratio $1/2 + \Omega(1/\sqrt{d_{\max}})$ where $d_{\max}$ is the maximum vertex degree in the graph. When $d_{\max} = 3$, Newman [129] gave a factor 8/9 approximation algorithm.

The complementary objective of minimizing the number of back edges, or equivalently deleting the minimum number of edges in order to make the graph a DAG, leads to the Min Feedback Arc Set (FAS) problem. This problem admits a factor $O(\log n \log\log n)$ approximation algorithm [148] based on bounding the integrality gap of the natural covering linear program for FAS; see also [52]. Using this algorithm, one can get an approximation ratio of $\frac{1}{2} + \Omega(1/(\log n \log\log n))$ for Maximum Acyclic Subgraph.

Recently, Charikar, Makarychev, and Makarychev [33] gave a factor $(1/2 + \Omega(1/\log n))$-approximation algorithm for Maximum Acyclic Subgraph, where n is the number of vertices. In fact, their algorithm is stronger: given a digraph with an acyclic subgraph

consisting of a fraction $(1/2 + \delta)$ of edges, it finds a subgraph with at least a fraction $(1/2+\Omega(\delta/\log n))$ of edges. This algorithm, and in particular an instance showing tightness of its analysis from [33], plays a crucial role in our work.

Apart from Maximum Acyclic Subgraph, the other OCSP that has received some attention is the Betweenness problem. Betweenness is an OCSP where all the constraints are of the form "X appears between Y and Z" for variables X, Y and Z. In [44], a $\frac{1}{2}$-approximation algorithm is presented for Betweenness on instances that are promised to be perfectly satisfiable. Building on the techniques in this chapter, Charikar .et.al. [31] show that for every OCSP of arity 3, it is UGC-hard to obtain an approximation better than one attained by a random ordering.

9.1.2 Organization

We begin with an outline of the key ideas of the proof in Section 9.2. In Section 9.3, we review the definitions of influences, noise operators and restate the unique games conjecture. The groundwork for the reduction is laid in Section 9.4 and Section 9.5, where we define influences for orderings, and multiscale gap instances respectively. We present the dictatorship test in Section 9.6, and convert it to a UG hardness result in Section 9.7.

Towards generalizing these hardness results, we begin with formal definition of OCSP s and the natural semidefinite program for OCSP s in Section Section 9.8. The construction of dictatorship tests from SDP integrality gaps for an OCSP is presented in Section 9.9. Finally, in Section 9.10, we sketch the component of the soundness analysis for Maximum Acyclic Subgraph and OCSP hardness results, that is mostly borrowed from [136].

9.2 Proof Overview

At the heart of all UG-hardness results lies a dictatorship testing result for an appropriate class of functions. For sake of brevity, let us denote $[m] = \{1, \ldots, m\}$. A function $\mathcal{F} : [m]^R \to [m]$ is said to be a *dictator* if $\mathcal{F}(x) = x_i$ for some fixed i. A dictatorship test (DICT) is a randomized algorithm such that, given a function $\mathcal{F} : [m]^R \to [m]$, it makes a few queries to the values of $\mathcal{F}$ and distinguishes between whether $\mathcal{F}$ is a dictator or *far* from every dictator. While Completeness of the test refers to the probability of acceptance of a dictator function, Soundness is the maximum probability of acceptance of a function *far* from a dictator. The approximation problem one is showing UG hardness for determines the nature of the dictatorship test needed for the purpose.

Unlike the case of functions, it is unclear as to what is the right notion of *Dictators* for orderings. For every ordering $\mathcal{O}$ of $[m]^R$, define m^{2R} functions $\mathcal{F}^{[s,t]} : [m]^R \to \{0, 1\}$ as follows:

$$\mathcal{F}^{[s,t]}(x) = \begin{cases} 1 & \text{if } s \leqslant \mathcal{O}(x) \leqslant t \\ 0 & \text{otherwise} \end{cases}$$

The i^{th} coordinate is said to be *influential* if it has a large influence ($> \tau$) on any of the functions $\mathcal{F}^{[s,t]}$. Here influence refers to the natural notion of influence for real valued functions on $[m]^R$ (see Section 9.3). An ordering $\mathcal{O}$ is said to be τ-pseudorandom (*far* from a dictator) if it has no influential coordinates ($> \tau$). For this notion to be useful, it is

necessary that a given ordering $\mathcal{O}$ does not have too many *influential* coordinates. Towards this, in Lemma 9.4.3 we show that the number of influential coordinates is bounded (after certain smoothening). Further this notion of influence is well suited to deal with orderings of multiple long codes instead of one - a crucial requirement in translating dictatorship tests to UG hardness.

Maximum Acyclic Subgraph Now we shall describe the proof strategy for the UG hardness of Maximum Acyclic Subgraph. Given an ordering $\mathcal{O}$ of the vertices of a directed graph $G = (V, E)$, let $\mathsf{Val}(\mathcal{O})$ refer to the fraction of the edges E that are oriented correctly in $\mathcal{O}$.

Designing the appropriate dictatorship test for Maximum Acyclic Subgraph amounts to the following: Construct a directed graph over the set of vertices $V = [m]^R$ such that:

- For a *Dictator* ordering $\mathcal{O}$ of V, $\mathsf{Val}(\mathcal{O}) \approx 1$
- For any ordering $\mathcal{O}$ which is *far from a dictator*, $\mathsf{Val}(\mathcal{O}) \approx \frac{1}{2}$.

Recall that our definition of influential coordinates for orderings can be used to formalize the notion - "far from dictator". Under this definition, we obtain a directed graph on $[m]^R$ (a dictatorship test) for which the following holds:

Theorem 9.3. *(Soundness) If $\mathcal{O}$ is any τ-pseudorandom ordering of $[m]^R$, then* $\mathsf{Val}(\mathcal{O}) \leqslant \frac{1}{2} + o_\tau(1)$.

This dictatorship test yields tight UG hardness for the Maximum Acyclic Subgraph problem. Using the Khot-Vishnoi [104] SDP gap instance for unique games, we obtain an SDP integrality gap for the same.

Now we describe the design of the dictatorship test in greater detail. At the outset, the approach is similar to recent work on Constraint Satisfaction Problems(CSPs) [136]. Fix a constraint satisfaction problem Λ. Starting with an integrality gap instance $\Im$ for the natural semi-definite program for Λ, [136] constructs a dictatorship test $\mathsf{DICT}_\Im$. The $\mathsf{Completeness}$ of $\mathsf{DICT}_\Im$ is equal to the SDP value $\mathsf{SDP}(\Im)$, while the $\mathsf{Soundness}$ is close to the integral value $\mathsf{INT}(\Im)$.

Since the result of [136] applies to arbitrary CSPs, a natural direction would be to pose the Maximum Acyclic Subgraph as a CSP. Maximum Acyclic Subgraph is fairly similar to a CSP, with each vertex being a variable taking values in domain $[n]$ and each directed edge a constraint between 2 variables. However, the domain, $[n]$, of the CSP is not fixed, but grows with input size. We stress here that this is not a superficial distinction but an essential characteristic of the problem. For instance, if Maximum Acyclic Subgraph was reducible to a 2-CSP over a domain of fixed size, then we could obtain a approximation ratio better than a random assignment [81].

Towards using techniques from the CSP result, we define the following variant of Maximum Acyclic Subgraph:

Definition 9.2.1. A q-ordering of a directed graph $G = (V, E)$ consists of a map $\mathcal{O} : V \to [q]$. The value of a q-ordering $\mathcal{O}$ is given by

$$\mathrm{val}_q(\mathcal{O}) = \mathop{\mathbb{P}\mathrm{r}}_{(u,v)\in E}\Big(\mathcal{O}(u) < \mathcal{O}(v)\Big) + \frac{1}{2}\mathop{\mathbb{P}\mathrm{r}}_{(u,v)\in E}\Big(\mathcal{O}(u) = \mathcal{O}(v)\Big)$$

In the q-Order problem, the objective is to find an q-ordering of the input graph G with maximum value.

On the one hand, the q-Order problem is a CSP over a fixed domain that is similar to Maximum Acyclic Subgraph. However, to the best of our knowledge, for the q-Order problem, there are no known SDP gaps, which constitute the starting point for results in [136]. For any fixed constant q, Charikar, Makarychev and Makarychev [33] construct directed acyclic graphs (i.e., with value of the best ordering equal to 1), while the value of any q-ordering of G is close to $\frac{1}{2}$. For the rest of the discussion, let us fix one such graph G on m vertices. Notice that the graph G does not serve as SDP gap example for either the Maximum Acyclic Subgraph or the q-Order problem.

As the graph G has only m vertices, and an ordering of value ≈ 1, it has a good q-ordering for $q = m$. Viewing G as an instance of the m-Order CSP (corresponding to predicate $<$), we obtain a directed graph, $\mathcal{G}$, on $[m]^R$. As a m-order CSP, the dictator m-orderings yield value ≈ 1 on $\mathcal{G}$. In turn, this implies that the *Dictator* orderings have value ≈ 1 on $\mathcal{G}$. Turning to the soundness proof, consider a τ-pseudorandom ordering $\mathcal{O}$. Obtain a q-ordering $\mathcal{O}^*$ by the following *coarsening* process: Divide the ordering $\mathcal{O}$ into q equal blocks, and map the vertices in the i^{th} block to value i. The crucial observation relating $\mathcal{O}$ and $\mathcal{O}^*$ is as follows:

> **Coarsening Observation:** "For a τ-pseudorandom ordering $\mathcal{O}$, $\mathrm{val}_q(\mathcal{O}^*) \approx \mathrm{val}(\mathcal{O})$."

Clearly, $\mathrm{val}(\mathcal{O}) - \mathrm{val}_q(\mathcal{O}^*)$ is bounded by the fraction of edges whose both endpoints fall in the same block, during the coarsening. We use the Gaussian noise stability bounds of [124], to bound the fraction of such edges. From the above observation, in order to prove that $\mathrm{val}(\mathcal{O}) \approx \frac{1}{2}$, it is enough to bound $\mathrm{val}_q(\mathcal{O}^*)$. Notice that $\mathcal{O}^*$ is a solution to q-order problem - a CSP over finite domain. Consequently, the soundness analysis of [136] can be used to show that $\mathrm{val}_q(\mathcal{O}^*)$ is at most the value of the best q-ordering for G, which is close to $\frac{1}{2}$.

Summarizing the key ideas, we define the notion of influential coordinates for orderings, and then use it to construct a dictatorship test for orderings. Using Gaussian noise stability bounds, we relate the value of a pseudorandom ordering to a related CSP, and then apply techniques from [136].

Ordering Constraint Satisfaction Problem The techniques developed in the case of Maximum Acyclic Subgraph, along with ideas from Chapter 7, immediately yield UG hardness results for general ordering CSPs.

First, as in the case of Maximum Acyclic Subgraph, for every OCSP Λ, it is possible to define a related CSP Λ_q over the domain $[q]$ for every positive integer q. Roughly speaking, the CSP Λ_q consists of the problem of finding the q-Order that satisfies the maximum number

of constraints. For a q-Order $\mathcal{O}$ of an instance $\Im$ of a Λ-OCSP we will use $\mathrm{val}_q(\mathcal{O})$ to denote the number of objective value. Further, let $\mathrm{val}_q(\Im)$ denote the optimum value of a q-Order for the instance $\Im$.

The statement of Theorem 9.2 relates the Unique Games hardness threshold of a OCSP Λ to the integrality gap of the natural SDP for the problem. However, constructing integrality gap instances for OCSPs is in itself a challenging task. In this light, we will show a stronger result than Theorem 9.2. Specifically, we will exhibit a black-box reduction to UG hardness result starting from what we will refer to as a *coarsening gap* instance.

Definition 9.2.2. An instance $\Im$ of a Λ-OCSP is a (q, c, s)-coarsening gap instance if $\mathrm{sdp}(\Im) \geqslant c$ and $\mathrm{val}_q(\Im) \leqslant s$.

We will show that an integrality gap instance $\Im$ with $\mathrm{sdp}(\Im) = c$ and $\mathrm{opt}(\Im) = s$, is a (q, c, s)-coarsening gap instance for all q (see Claim 9.8.1). Hence, clearly a *coarsening gap* instance is a weaker notion than a integrality gap instance. Furthermore, constructing *coarsening gap* instances has proved to be an easier task in the case of Maximum Acyclic Subgraph and ordering 3-CSPs [31].

Theorem 9.4. *Given a (q, c, s)-coarsening gap instance $\Im$ of a OCSP Λ, for every constant $\eta > 0$ we have*

$$\mathsf{UGhard}_\Lambda(c) \leqslant \mathsf{Gap}_\Lambda(c + \eta) + \eta + O(q^{-\eta})$$

Fix an OCSP Λ. Let $\Im$ be an instance of Λ with SDP value $c + \eta$ and optimum value $S_\Lambda(c + \eta)$. To show Theorem 9.2, we obtain a black box reduction that converts the integrality gap instance $\Im$ with SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ into a dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ with completeness c and soundness at most $S_\Lambda(c+\eta)+\eta$. Further all the predicates checked by the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ belong to the family of predicates corresponding to Λ.

Let m denote the number of variables in the instance $\Im$. The dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is constructed by viewing the instance $\Im$ as a CSP over a domain of size m. Specifically $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is an instance of Λ-OCSP over the set of variables indexed by $[m]^R$ for any integer R. By virtue of the construction in [136], the m-Orders of $[m]^R$ given by the dictator functions, have an objective value equal to the SDP value ($c + \eta$ in this case). To perform the soundness analysis, we appeal to the coarsening observation above. By using this observation, we can relate the value of an ordering $\mathcal{O}$ of $\Im$, to the value of the q-Order $\mathcal{O}_q$ obtained by coarsening $\mathcal{O}$. Finally, using a proof strategy along the lines of Section 7.4 we relate the value $\mathrm{val}_q(\mathcal{O}_q)$ of the q-Order $\mathcal{O}_q$ of $[m]^R$, to $\mathrm{val}_q(\Im)$ - the optimum q-Order value of the instance $\Im$!

While it is not trivial to obtain a UGC based hardness result for OCSP Λ starting from the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$, it follows entirely along the lines of Maximum Acyclic Subgraph. Therefore, we omit the proof of the UG hardness result from this presentation.

9.3 Preliminaries

For a positive integer q, Δ_q denotes the set of corners of the q dimensional simplex, i.e., $\Delta_q = \{\mathbf{e}_i | i \in [q]\}$. Let $\blacktriangle_q$ denote the convex hull of the set Δ_q, in other words $\blacktriangle_q$ is the

q-dimensional simplex. We will use boldface letters $\mathbf{z}$ to denote vectors $\mathbf{z} = (z^{(1)}, \ldots, z^{(R)})$. Let $o_\tau(1)$ denote a quantity that tends to zero as $\tau \to 0$, while keeping all other parameters fixed. A q-ordering $\mathcal{O}$ of the graph G consists of a map $\mathcal{O} : V \to [q]$. Note that the map $\mathcal{O}$ need not be injective or surjective. If the map $\mathcal{O}$ is a injection, then it corresponds to an ordering of the vertices V. In a q-ordering $\mathcal{O}$, an edge $e = (u, v)$ is a *forward* edge if $\mathcal{O}(u) < \mathcal{O}(v)$.

Observation 9.3.1. *For all directed graphs G, and integers $q \leqslant q'$, $\mathrm{val}_q(G) \leqslant \mathrm{val}_{q'}(G) \leqslant \mathrm{val}(G)$*

While the first part of the inequality is trivial, we will elaborate on the latter half. Given a q'-ordering, construct a full ordering $\mathcal{O}^*$ by using a random permutation of the elements within each of the q' blocks, while retaining the natural order between the blocks. The expected value of the random ordering $\mathcal{O}^*$ is exactly equal to the value of $\mathrm{val}(\mathcal{O})$, thus proving the latter half of the inequality.

9.3.1 Noise Operators and Influences

Let Ω denote the finite probability space corresponding to the uniform distribution over $[m]$. Let $\{\chi_0 = 1, \chi_1, \chi_2, \ldots, \chi_{m-1}\}$ be an orthonormal basis for the space $L_2(\Omega)$. For $\sigma \in [m]^R$, define $\chi_\sigma(\mathbf{z}) = \prod_{k \in [R]} \chi_{\sigma_i}(z^{(k)})$. Every function $\mathcal{F} : \Omega^R \to \mathbb{R}$ can be expressed as a multilinear polynomial as $\mathcal{F}(\mathbf{z}) = \sum_\sigma \hat{\mathcal{F}}(\sigma)\chi_\sigma(\mathbf{z})$. The L_2 norm of $\mathcal{F}$ in terms of the coefficients of the multilinear polynomial is $||\mathcal{F}||_2^2 = \sum_\sigma \hat{\mathcal{F}}^2(\sigma)$

Definition 9.3.1. For a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, define $\mathrm{Inf}_k(\mathcal{F}) = \mathbb{E}_{\mathbf{z}}[\mathbf{Var}_{z^{(k)}}[\mathcal{F}]] = \sum_{\sigma_k \neq 0} \hat{\mathcal{F}}^2(\sigma)$.
Here $\mathbf{Var}_{z^{(k)}}[\mathcal{F}]$ denotes the variance of $\mathcal{F}(\mathbf{z})$ over the choice of the k^{th} coordinate $z^{(k)}$.

Definition 9.3.2. For a function $\mathcal{F} : \Omega^R \to \mathbb{R}$, define the function $T_\rho\mathcal{F}$ as follows:

$$T_\rho\mathcal{F}(\mathbf{z}) = \mathbb{E}[\mathcal{F}(\tilde{\mathbf{z}}) \mid \mathbf{z}] = \sum_{\sigma \in [m]^R} \rho^{|\sigma|}\hat{\mathcal{F}}(\sigma)\chi_\sigma(\mathbf{z})$$

where each coordinate $\tilde{z}^{(k)}$ of $\tilde{\mathbf{z}} = (\tilde{z}^{(1)}, \ldots, \tilde{z}^{(R)})$ is equal to $z^{(k)}$ with probability ρ and with the remaining probability, $\tilde{z}^{(k)}$ is a random element from the distribution Ω.

Lemma 9.4.1. *Consider two functions $\mathcal{F}, \mathcal{G} : [m]^R \to [0, 1]$ with $\mathbb{E}[\mathcal{F}] = \mathbb{E}[\mathcal{G}] = \mu$, and $\mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}), \mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{G}) \leqslant \tau$ for all k. Let $\mathbf{x}, \mathbf{y}$ be random vectors in $[m]^R$ whose marginal distributions are uniform over $[m]^R$ but are arbitrarily correlated. For every $\varepsilon > 0$, there exists a $\mu_0 > 0$ such that if $\mu < \mu_0$ then*

$$\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}}[T_{1-2\varepsilon}\mathcal{F}(\mathbf{x})T_{1-2\varepsilon}\mathcal{G}(\mathbf{y})] \leqslant \mu^{1+\varepsilon/2} + o_\tau(1)$$

Proof. The lemma essentially follows from the Majority is Stablest theorem (see Theorem

4.4 in [125]). We bound each factor individually as follows:

$$||T_{1-2\varepsilon}\mathcal{F}||_2^2 = \sum_{\sigma\in[k]^R}(1-2\varepsilon)^{2|\sigma|}\hat{\mathcal{F}}^2(\sigma) \leqslant \sum_{\sigma\in[k]^R}(1-\varepsilon)^{|\sigma|}\hat{\mathcal{F}}(\sigma)(1-\varepsilon)^{2|\sigma|}\hat{\mathcal{F}}(\sigma)$$
$$\leqslant \mathbb{E}[(T_{1-\varepsilon}\mathcal{F})(\mathbf{x})T_{1-\varepsilon}(T_{1-\varepsilon}\mathcal{F})(\mathbf{x})]\,.$$

Since the influences of $\mathrm{T}_{1-\varepsilon}\mathcal{F}$ are low, we can apply Theorem 3.6 to bound the last expression by noise stability in Gaussian space $\Gamma_{1-\varepsilon}(\mu)$.

$$\mathbb{E}[(\mathrm{T}_{1-\varepsilon}\mathcal{F})\mathrm{T}_{1-\varepsilon}(\mathrm{T}_{1-\varepsilon}\mathcal{F})] \leqslant \Gamma_{1-\varepsilon}(\mu) + o_\tau(1)$$

Using Theorem 3.5, $\Gamma_{1-\varepsilon}(\mu)$ is bounded by $\mu^{1+\varepsilon/2}$ for μ small enough compared to ε. Applying a similar bound for $\mathcal{F}'$ and applying Cauchy-Schwartz gives the result:

$$\mathop{\mathbb{E}}_{x}[\mathrm{T}_{1-2\varepsilon}\mathcal{F}(x)\mathrm{T}_{1-2\varepsilon}\mathcal{F}'(y)] \leqslant \sqrt{||\mathrm{T}_{1-2\varepsilon}\mathcal{F}||_2^2||\mathrm{T}_{1-2\varepsilon}\mathcal{F}'||_2^2} \leqslant \mu^{1+\varepsilon/2}+\eta$$
(for μ small enough)

■

Here we recall the following lemma bounding the sum of influences, for the sake of convenience.

Lemma 9.4.2 (Sum of Influences Lemma)**.** *Given a function $\mathcal{F} : [m]^R \to [0,1]$, if $\mathcal{H} = T_{1-\varepsilon}\mathcal{F}$ then $\sum_{k=1}^R \mathrm{Inf}_k(\mathcal{H}) \leqslant \frac{1}{2e\ln 1/(1-\varepsilon)} \leqslant \frac{1}{\varepsilon}$*

9.3.2 Semidefinite Program

For the sake of convenience, we recall the LC relaxation here. A solution the SDP consists of a set of vectors $\boldsymbol{V} = \{\boldsymbol{b}_{u,i} | u \in V, i \in [n]\}$, n-orthogonal vectors for each vertex, and a set of distributions $\boldsymbol{\mu} = \{\mu_e | e \in E\}$ over $[n]^2$ one for each edge. The formal statement of the SDP relaxation is as follows.

LC Relaxation

$$\text{maximize} \quad \mathop{\mathbb{E}}_{e=(u,v)\sim E}\left[\Pr_{(x_u,x_v)\in\mu_e}\left\{x_u < x_v\right\} + \frac{1}{2}\Pr_{(x_u,x_v)\in\mu_e}\left\{x_u = x_v\right\}\right] \tag{LC}$$

$$\text{subject to} \quad \langle \boldsymbol{b}_{u,i}, \boldsymbol{b}_{v,j}\rangle = \Pr_{(x_u,x_v)\in\mu_e}\left\{x_u = i, x_v = j\right\} \qquad (e=(u,v)\in E,\ \ i,j\in[n])\,.$$
$$\mu_e \in \blacktriangle([n]^2) \qquad \forall e \in E$$

9.4 Orderings

In this section, we develop the notions of influences for orderings and prove some basic results about them.

Definition 9.4.1. Given an ordering $\mathcal{O}$ of vertices V, its q-coarsening is a q-ordering $\mathcal{O}^*$ obtained by dividing $\mathcal{O}$ into q-contiguous blocks, and assigning label i to vertices in the i^{th} block. Formally, if $M = |V|/q$ then

$$\mathcal{O}^*(u) = \left\lfloor \frac{\mathcal{O}(u)}{M} \right\rfloor + 1$$

For an ordering $\mathcal{O}$ of points in $[m]^R$, Define functions $\mathcal{F}_{\mathcal{O}}^{[s,t]} : [m]^R \to \{0,1\}$ for integers s, t as follows:

$$\mathcal{F}_{\mathcal{O}}^{[s,t]}(x) = \begin{cases} 1 & \text{if } \mathcal{O}(x) \in [s,t] \\ 0 & \text{otherwise} \end{cases}$$

We will omit the subscript and write $\mathcal{F}^{[s,t]}$ instead of $\mathcal{F}_{\mathcal{O}}^{[s,t]}$, when it is clear.

Definition 9.4.2. For an ordering $\mathcal{O}$ of $[m]^R$, define the set of influential coordinates $\mathsf{L}_\tau(\mathcal{O})$ as follows:

$$\mathsf{L}_\tau(\mathcal{O}) = \{k \mid \mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}^{[s,t]}) \geqslant \tau \text{ for some } s,t \in \mathbb{Z}\}$$

An ordering $\mathcal{O}$ is said to be τ-pseudorandom if $\mathsf{L}_\tau(\mathcal{O})$ is empty.

Lemma 9.4.3. *(Few Influential Coordinates) For any ordering $\mathcal{O}$ of $[m]^R$, we have $|\mathsf{L}_\tau(\mathcal{O})| \leqslant \frac{400}{\varepsilon\tau^3}$*

Proof. For integers s, t, δ_1, δ_2 such that $|\delta_i| < \frac{\tau}{8}m^R$, let $f = T_{1-\varepsilon}\mathcal{F}^{[s,t]}$ and $g = T_{1-\varepsilon}\mathcal{F}^{[s+\delta_1,t+\delta_2]}$. Now,

$$\mathrm{Inf}_k(f-g) \leqslant ||f-g||_2^2 \leqslant ||\mathcal{F}^{[s,t]} - \mathcal{F}^{[s+\delta_1,t+\delta_2]}||_2^2 = \Pr_{\mathbf{z}}[\mathcal{F}^{[s,t]}(\mathbf{z}) \neq \mathcal{F}^{[s+\delta_1,t+\delta_2]}(\mathbf{z})] \leqslant \tau/4$$

Hence, using $a^2 \leqslant 2(b^2 + (a-b)^2)$, we get:

$$\mathrm{Inf}_k(f) = \sum_{\sigma_k \neq 0} \hat{f}^2(\sigma) \leqslant 2\left[\sum_{\sigma_k \neq 0} \hat{g}^2(\sigma) + \sum_{\sigma_k \neq 0} \left(\hat{f}(\sigma) - \hat{g}(\sigma)\right)^2\right] \leqslant 2\mathrm{Inf}_k(g) + \tau/2$$

Thus, if $\mathrm{Inf}_k(f) \geqslant \tau$, then $\mathrm{Inf}_k(g) \geqslant \tau/4$. It is easy to see that there is a set $N = \{\mathcal{F}^{[s,t]}\}$ of size at most $100/\tau^2$ such that for every $\mathcal{F}^{[s,t]}$ there is a $\mathcal{F}^{[s',t']} \in N$ such that $\max|s-s'|, |t-t'| < \frac{\tau m^R}{8}$. Further, by Lemma 9.4.2, the functions $T_{1-\varepsilon}\mathcal{F}^{[s',t']}$ have at most $\frac{4}{\varepsilon\tau}$ coordinates with influence more than $\tau/4$. Hence, $|\mathsf{L}_\tau(\mathcal{O})| \leqslant \frac{400}{\varepsilon\tau^3}$. ■

Claim 9.4.1. *For any τ-pseudorandom ordering $\mathcal{O}$ of $[m]^R$, its q-coarsening $\mathcal{O}^*$ is also τ-pseudorandom.*

Proof. Since the functions $\{\mathcal{F}_{\mathcal{O}^*}^{[\cdot,\cdot]}\}$ are a subset of the functions $\{\mathcal{F}_{\mathcal{O}}^{[\cdot,\cdot]}\}$, $S_\tau(\mathcal{O}^*) \subseteq S_\tau(\mathcal{O})$. ■

9.5 Multiscale Gap Instances

In this section, we will construct acyclic directed graphs with no good q-ordering. These graphs will be crucial in designing the dictatorship test (Section 9.6).

Definition 9.5.1. For $\eta > 0$ and a positive integer q, a (η, q)-*Multiscale Gap* instance is a weighted directed graph $G = (V, E)$ with the following properties:

- $\text{val}(G) = 1$ and $\text{val}_q(G) \leqslant \frac{1}{2} + \eta$
- There exists a solution $\{\boldsymbol{b}_{u,i} \,|\, u \in V, i \in [|V|]\}$ to LC relaxation with objective value at least $1 - \eta$ such that for all $u \in V$ and $1 \leqslant i \leqslant |V|$, we have $\|\boldsymbol{b}_{u,i}\|_2^2 = \frac{1}{|V|}$.

Clearly, if $\text{val}(G) = 1$, then the SDP value of G is at least 1. Hence, by definition we have,

Observation 9.5.1. *An (η, q)-multiscale gap instance is a $(q, 1, \frac{1}{2} + \eta)$-coarsening gap instance for* Maximum Acyclic Subgraph.

The cut norm of a directed graph, G, represented by a skew-symmetric matrix W is: $||G||_C = \max_{x_i, y_j \in \{0,1\}} \sum_{ij} x_i y_j w_{ij}$

We will need the following theorem from [33] relating the cut norm of a directed graph G to $\text{val}(G)$.

Theorem 9.5 (Theorem 3.1, [33])**.** *If a directed graph G on n vertices has a maximum acyclic subgraph with at least a $\frac{1}{2} + \delta$ fraction of the edges, then, $||G||_C \geqslant \Omega\left(\frac{\delta}{\log n}\right)$.*

The following lemma and its corollary construct Multiscale Gap instances starting from graphs that are the "tight cases" of the above theorem.

Lemma 9.5.1. *Given $\eta > 0$ and a positive integer q, for every sufficiently large n, there exists a directed graph $G = (V, E)$ on n vertices such that $\text{val}(G) = 1$, $\text{val}_q(G) \leqslant \frac{1}{2} + \eta$.*

Proof. Charikar et al (Section 4, [33]) construct a directed graph, $G = (V, E)$, on n vertices whose cut norm is bounded by $O\left(1/\log n\right)$. The graph is represented by the skew-symmetric matrix W, where $w_{ij} = \sum_{k=1}^{n} \sin \frac{\pi(j-i)k}{n+1}$. It is easy to verify that for every $0 < q < n$, $\sum_{k=1}^{n} \sin\left(\frac{\pi q k}{n+1}\right) \geqslant 0$. Thus, $w_{ij} \geqslant 0$ whenever $i < j$, implying that the graph is acyclic (in other words, $\text{val}(G) = 1$).

We bound $\text{val}_q(G)$ as follows. Let $\text{val}_q(G) = \frac{1}{2} + \delta$ and let $\mathcal{O} : V \to [q]$ be the optimal q-ordering. Construct a graph H on q vertices with a directed edge from $\mathcal{O}(u)$ to $\mathcal{O}(v)$ for every edge $(u, v) \in E$ with $\mathcal{O}(u) \neq \mathcal{O}(v)$. Now, using Theorem 9.5, the cut norm of H is bounded from below by $\Omega\left(\frac{\delta}{\log q}\right)$. Moreover, since $\mathcal{O}$ is a partition of V, the cut norm of G is at least the cut norm of H. Thus, $\Omega\left(\frac{\delta}{\log q}\right) \leqslant ||H||_C \leqslant ||G||_C \leqslant O\left(1/\log n\right)$Thus, $\delta \leqslant O\left(\frac{\log q}{\log n}\right)$ implying that $\text{val}_q(G) \leqslant \frac{1}{2} + O\left(\frac{\log q}{\log n}\right)$. Choosing n to be a sufficiently gives the required result. ■

Corollary 9.5.1. *For every $\eta > 0$ and positive integer q, there exists a (η, q)-Multiscale Gap instance with a corresponding SDP solution $\{\boldsymbol{b}_{u,i} | u \in V, i \in [|V|]\}$ and $\boldsymbol{\mu} = \{\mu_e | e \in E\}$ satisfying $\|\boldsymbol{b}_{u,i}\|_2^2 = 1/|V|$ for all $u \in V, i \in [|V|]$.*

Proof. Let $G = (V, E)$ be the graph obtained by taking $\lceil 1/\eta \rceil$ disjoint copies of the graph guaranteed by Lemma 9.5.1 and let $m = |V|$. Note that the graph still satisfies the required properties: $\mathrm{val}(G) = 1$, $\mathrm{val}_q(G) \leqslant \frac{1}{2} + \eta$. Let $\mathcal{O}$ be the ordering of $[m]$ that satisfies every edge of G. Let D denote the distribution over labellings obtained by shifting $\mathcal{O}$ by a random offset cyclically. For every $u \in V, i \in [m]$, $\mathbb{P}\mathrm{r}[D(u) = i] = 1/m$. Further, every directed edge is satisfied with probability at least $1 - \eta$. Being a distribution over integral labellings, D gives raise to a set of vectors satisfying the constraints in Definition 9.5.1. G along with these vectors form the required (η, q)-multiscale gap instance. ■

9.6 Dictatorship Test

Let $G = (V, E)$ be a (η, q)-multiscale gap instance on m vertices, where m is divisible by q. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote the corresponding SDP solution. Using the multiscale gap instance G, construct a dictatorship test $\mathsf{DICT}^{\varepsilon}_{G}$ on orderings $\mathcal{O}$ of $[m]^R$ as follows:

$\mathsf{DICT}^{\varepsilon}_{G}$ Test:

- Pick an edge $e = (u, v) \in E$ at random from the Multiscale gap instance G.
- Sample $\mathbf{z}_e = \{\mathbf{z}_u, \mathbf{z}_v\}$ from the product distribution μ_e^R, i.e. For each $1 \leqslant k \leqslant R$, $z_e^{(k)} = \{z_u^{(k)}, z_v^{(k)}\}$ is sampled using the distribution μ_e.
- Obtain $\tilde{\mathbf{z}}_u, \tilde{\mathbf{z}}_v$ by perturbing each coordinate of $\mathbf{z}_u$ and $\mathbf{z}_v$ independently. Specifically, sample the k^{th} coordinates $\tilde{z}_u^{(k)}, \tilde{z}_v^{(k)}$ as follows: With probability $(1-2\varepsilon)$, $\tilde{z}_u^{(k)} = z_u^{(k)}$, and with the remaining probability $\tilde{z}_u^{(k)}$ is a new sample from Ω.
- Introduce a directed edge $\tilde{\mathbf{z}}_u \to \tilde{\mathbf{z}}_v$. (alternatively test if $\mathcal{O}(\tilde{\mathbf{z}}_u) < \mathcal{O}(\tilde{\mathbf{z}}_v)$)

Theorem 9.6. *(Soundness Analysis) For every $\varepsilon > 0$, there exists sufficiently large m, q such that: For any τ-pseudorandom ordering $\mathcal{O}$ of $[m]^R$,*

$$\mathrm{val}(\mathcal{O}) \leqslant \mathrm{val}_q(G) + O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$$

where $o_\tau(1) \to 0$ as $\tau \to 0$ keeping all other parameters fixed.

Let $\mathcal{F}^{[s,t]} : [m]^R \to \{0, 1\}$ denote the functions associated with the q-ordering $\mathcal{O}^*$. For the sake of brevity, we shall write $\mathcal{F}^i$ for $\mathcal{F}^{[i,i]}$. The result follows from Lemma 9.6.2 and Lemma 9.6.1 shown below.

Lemma 9.6.1. *For every $\varepsilon > 0$, there exists sufficiently large m, q such that: For any τ-pseudorandom ordering $\mathcal{O}$ of $[m]^R$*

$$\mathrm{val}(\mathcal{O}) \leqslant \mathrm{val}_q(\mathcal{O}^*) + O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$$

where $\mathcal{O}^*$ is the q-coarsening of $\mathcal{O}$.

Proof. As $\mathcal{O}^*$ is a coarsening of $\mathcal{O}$, clearly $\mathrm{val}(\mathcal{O}) \geqslant \mathrm{val}_q(\mathcal{O}^*)$. Note that the loss due to coarsening, is because for some edges $e = (\mathbf{z}, \mathbf{z}')$ which are oriented correctly in $\mathcal{O}$, fall into same block during coarsening, i.e. $\mathcal{O}^*(z) = \mathcal{O}^*(z')$. Thus we can write

$$\mathrm{val}(\mathcal{O}) \leqslant \mathrm{val}_q(\mathcal{O}^*) + \frac{1}{2}\mathbb{P}\mathrm{r}\left(\mathcal{O}^*(\tilde{\mathbf{z}}_u) = \mathcal{O}^*(\tilde{\mathbf{z}}_v)\right)$$

$$\begin{aligned}\mathbb{P}\mathrm{r}\left(\mathcal{O}^*(\tilde{\mathbf{z}}_u) = \mathcal{O}^*(\tilde{\mathbf{z}}_v)\right) &= \sum_{i\in[q]} \mathop{\mathbb{E}}_{e=(u,v)} \mathop{\mathbb{E}}_{\mathbf{z}_u,\mathbf{z}_v} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_u,\tilde{\mathbf{z}}_v} \left[\mathcal{F}^i(\tilde{\mathbf{z}}_u)\cdot\mathcal{F}^i(\tilde{\mathbf{z}}_v)\right] \\ &= \sum_{i\in[q]} \mathop{\mathbb{E}}_{e=(u,v)} \mathop{\mathbb{E}}_{\mathbf{z}_u,\mathbf{z}_v} \left[T_{1-2\varepsilon}\mathcal{F}^i_u(\mathbf{z}_u)\cdot T_{1-2\varepsilon}\mathcal{F}^i_v(\mathbf{z}_v)\right]\end{aligned}$$

As $\mathcal{O}$ is a q-coarsening of $\mathcal{O}$, for each value $i \in [q]$, there are exactly $\frac{1}{q}$ fraction of $\mathbf{z}$ for which $\mathcal{O}^*(\mathbf{z}) = i$. Hence for each $i \in [q]$, $\mathbb{E}_{\mathbf{z}}[\mathcal{F}^i_u(\mathbf{z}) = \frac{1}{q}]$. Further, since the ordering $\mathcal{O}^*$ is τ-pseudorandom, for every $k \in [R]$ and $i \in [q]$, $\mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}^i_a) \leqslant \tau$. Hence using Lemma 9.4.1, for sufficiently large q, the above probability is bounded by $q\cdot q^{-1-\frac{\varepsilon}{2}} + q\cdot o_\tau(1) = O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$. ■

Lemma 9.6.2. *For every choice of m, q, ε, and any τ-pseudorandom q-ordering $\mathcal{O}^*$ of $[m]^R$, $\mathrm{val}_q(\mathcal{O}^*) \leqslant \mathrm{val}_q(G) + o_\tau(1)$.*

Proof. The q-ordering problem is a CSP over a finite domain, and is thus amenable to techniques of [136]. Specifically, consider the payoff function $P : [q]^2 \to [0,1]$ defined by: $P(i,j) = 1$ for $i < j$, $P(i,j) = 0$ for $i > j$ and $P(i,j) = \frac{1}{2}$ otherwise. The q-ordering problem is a Generalized CSP(see Definition 2.4.1) with the payoff function P.

For the sake of exposition, let us pretend that $q = m$. In this case, the vectors $\{u_i | u \in V, i \in [m]\}$ form a feasible SDP solution for the q-ordering instance G. Let $\mathsf{DICT}^{\varepsilon}_G$ denote the dictatorship test obtained by running the reduction presented in Chapter 7 on this SDP solution for q-ordering instance G. $\mathsf{DICT}^{\varepsilon}_G$ is an instance of the q-ordering problem, over the set of vertices $[q]^R$. A q-ordering solution $\mathcal{O}^*$ for $\mathsf{DICT}^{\varepsilon}_G$ corresponds naturally to a function $\mathcal{F} : [q]^R \to \Delta_q$. Now we make the following observations:

- The q-ordering instance $\mathsf{DICT}^{\varepsilon}_G$ is identical to the dictatorship test described in this section when $q = m$.
- For a τ-pseudorandom q-ordering $\mathcal{O}^*$, for every $k \in [R]$ and $i \in [q]$, the corresponding function $\mathcal{F}$ satisfies $\mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}^i) \leqslant \tau$. In the terminology of Chapter 7 (Definition 7.3.3), this is equivalent to the function $\boldsymbol{\mathcal{F}} = (\mathcal{F}^1, \ldots, \mathcal{F}^q)$ being "(β,τ)-pseudorandom" with $\beta = 0$.
- By the soundness analysis of the dictatorship test from Chapter 7 (Theorem 7.5), for a (γ,τ)-pseudorandom function $\boldsymbol{\mathcal{F}}$, its probability of acceptance on the dictatorship test is at most $\mathrm{val}_q(G) + o_{\beta,\tau}(1)$.

Hence the above lemma is just a restatement of Theorem 7.5 for the specific generalized CSP: q-Ordering, albeit in the language of τ-pseudorandom orderings.

Recall that the actual case of interest here satisfies $q < m$. Unfortunately, in this case, a black box application of the result from Theorem 7.5 does not suffice. However, the proof of Theorem 7.5 can be easily adopted without any new technical ideas. In fact, many of the technical difficulties encountered in proving Theorem 7.5 can be avoided here. For instance, the SDP solution associates with each vertex u, the uniform probability distribution over $\{1 \dots m\}$, unlike in Chapter 7 where there are several arbitrary probability distributions to deal with. For the sake of completeness, we include a sketch of the above soundness analysis in the more general setting of ordering constraint satisfaction problem in Section 9.10. ■

Let $\mathcal{F}^{[s,t]} : [m]^R \to \{0,1\}$ denote the functions associated with the q-ordering $\mathcal{O}^*$. For the sake of brevity, we shall write $\mathcal{F}^i$ for $\mathcal{F}^{[i,i]}$, and $\mathcal{F} = (\mathcal{F}^1, \dots, \mathcal{F}^q)$. Arithmetizing $\mathrm{val}_q(\mathcal{O}^*)$ in terms of functions $\mathcal{F}^i$ we get:

$$\mathrm{val}_q(\mathcal{O}^*) = \mathbb{E}\left[\frac{1}{2}\sum_{i=j} \mathcal{F}^i(\tilde{\mathbf{z}}_u) \cdot \mathcal{F}^j(\tilde{\mathbf{z}}_v) + \sum_{i<j} \mathcal{F}^i(\tilde{\mathbf{z}}_u) \cdot \mathcal{F}^j(\tilde{\mathbf{z}}_v)\right]$$

where the expectation is over the edge $e = (u,v)$, $\mathbf{z}_u$, $\mathbf{z}_v$, $\tilde{\mathbf{z}}_u$, and $\tilde{\mathbf{z}}_v$. Lemma 9.6.2 asserts that the above expectation is bounded by $\mathrm{val}_q(G) + o_\tau(1)$ for all functions $\boldsymbol{\mathcal{F}} = (\mathcal{F}^1, \dots, \mathcal{F}^q)$ that correspond to a q-ordering. Specifically, for each $\mathbf{z} \in [m]^R$, $\mathcal{F}(\mathbf{z})$ is a corner of the simplex ($\mathcal{F}(\mathbf{z}) \in \Delta_q$).

For the Unique Games hardness reduction, we need the above lemma to hold for the more general class of functions that take values in $\blacktriangle_q$ - the q-dimensional simplex. The following stronger claim also immediately follows from the proof of Lemma 9.6.2.

Claim 9.6.1. *For a function $\boldsymbol{\mathcal{F}} : [m]^R \to \blacktriangle_q$ satisfying $\mathrm{Inf}_k(T_{1-\varepsilon}\boldsymbol{\mathcal{F}}) \leqslant \tau$ for all $k \in [R]$,*

$$\mathbb{E}\left[\frac{1}{2}\sum_{i=j} \mathcal{F}^i(\tilde{\mathbf{z}}_u)\mathcal{F}^j(\tilde{\mathbf{z}}_u) + \sum_{i<j} \mathcal{F}^i(\tilde{\mathbf{z}}_u)\mathcal{F}^j(\tilde{\mathbf{z}}_u)\right] \leqslant \mathrm{val}_q(G) + o_\tau(1)$$

where the expectation is over the edge $e = (u,v)$, $\mathbf{z}_u$, $\mathbf{z}_v$, $\tilde{\mathbf{z}}_u$, and $\tilde{\mathbf{z}}_v$.

We will sketch the proof of the above claim in the more general setting (see Lemma 9.9.3) of OCSP's in Section 9.10.

9.7 Hardness Reduction

Let $G = (V,E)$ be a (η, q)-Multiscale gap instance, and let $m = |V|$. Further let $\boldsymbol{V} = \{\boldsymbol{b}_{v,i}\}$ and $\boldsymbol{\mu} = \{\mu_e | e \in E\}$ denote the corresponding SDP solution. Let $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi = \{\pi_e : [R] \to [R] | e \in E\}, [R])$ be a bipartite Unique Games instance. Towards constructing a Maximum Acyclic Subgraph instance $\Psi = (\mathcal{V}, \mathcal{E})$ from Φ, we shall introduce a long code for each vertex in $\mathcal{V}_\Phi$. Specifically, the set of vertices $\mathcal{V}$ of the directed graph Ψ is indexed by $\mathcal{V}_\Phi \times [m]^R$.

Hardness Reduction:
Input: Unique Games instance $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi = \{\pi_e : [R] \to [R] | e \in E\}, [R])$ and a (η, q) Multiscale gap instance $G = (V, E)$.
Output : Directed graph $\Psi = (\mathcal{V}, \mathcal{E})$ with set of vertices : $\mathcal{V} = \mathcal{V}_\Phi \times [m]^R$ and edges $\mathcal{E}$ given by the following verifier:

- Pick a random vertex $a \in \mathcal{W}_\Phi$. Choose two neighbours $b, b' \in \mathcal{V}_\Phi$ independently at random. Let π, π' denote the permutations on the edges (a, b) and (a, b').
- Pick an edge $e = (u, v) \in E$ at random from the Multiscale gap instance G.
- Sample $\mathbf{z}_e = \{\mathbf{z}_u, \mathbf{z}_v\}$ from the product distribution μ_e^R, i.e. For each $1 \leqslant k \leqslant R$, $z_e^{(k)} = \{z_u^{(k)}, z_v^{(k)}\}$ is sampled using the distribution $\mu_e(i, j) = u_i \cdot v_j$.
- Obtain $\tilde{\mathbf{z}}_u, \tilde{\mathbf{z}}_v$ by perturbing each coordinate of $\mathbf{z}_u$ and $\mathbf{z}_v$ independently. Specifically, sample the k^{th} coordinates $\tilde{z}_u^{(k)}, \tilde{z}_v^{(k)}$ as follows: With probability $(1 - 2\varepsilon)$, $\tilde{z}_u^{(k)} = z_u^{(k)}$, and with the remaining probability $\tilde{z}_u^{(k)}$ is a new sample from Ω.
- Introduce a directed edge $(b, \pi(\tilde{\mathbf{z}}_u)) \to (b', \pi'(\tilde{\mathbf{z}}_v))$.

Theorem 9.7. *For every $\eta > 0$, there exists choice of parameters ε, q, δ such that:*

- Completeness: *If Φ is a $(1-\delta)$-strongly satisfiable instance of* Unique Games*, then there is an ordering $\mathcal{O}$ for the graph Ψ with value at least $(1-5\eta)$. i.e.* $\mathrm{val}(\Psi) \geqslant 1-5\eta$.
- Soundness: *If Φ is not δ-satisfiable, then no ordering to Ψ has value more than $\frac{1}{2} + 4\eta$, i.e.* $\mathrm{val}(\Psi) \leqslant \frac{1}{2} + 4\eta$.

In the rest of the section, we will present the proof of the above theorem. To begin with, we fix the parameters of the reduction.
Parameters : Fix $\varepsilon = \eta/100$. Let τ, q be the constants obtained from Theorem 9.8. Finally, let us choose $\delta = \min\{\eta/4, \eta\varepsilon^2\tau^8/10^9\}$.

9.7.1 Completeness

In order to show that $\mathrm{val}(\Psi) \geqslant 1 - 5\eta$, we will instead show that $\mathrm{val}_m(\Psi) \geqslant 1 - 5\eta$. From Observation 9.3.1, this will imply the required result.

By assumption, there exists labellings to the Unique Game instance Φ such that for $1-\delta$ fraction of the vertices $a \in \mathcal{W}_\Phi$ all the edges (a, b) are satisfied. Let $\mathcal{A} : \mathcal{V}_\Phi \cup \mathcal{W}_\Phi \to [R]$ denote one such labelling. Define an m-ordering of Ψ as follows:

$$\mathcal{O}(a, \mathbf{z}) = z^{(\mathcal{A}(a))} \qquad \forall a \in \mathcal{W}_\Phi, \mathbf{z} \in [m]^R$$

Clearly the mapping $\mathcal{O} : \mathcal{V} \to [m]$ defines an m-ordering of the vertices $\mathcal{V} = \mathcal{V}_\Phi \times [m]^R$. To determine $\mathrm{val}_m(\mathcal{O})$, let us compute the probability of acceptance of a verifier that follows

the above procedure to generate an edge in $\mathcal{E}$ and then checks if the edge is satisfied. Arithmetizing this probability, we can write

$$\mathrm{val}_m(\mathcal{O}) = \frac{1}{2}\,\mathbb{P}\mathrm{r}\left(\mathcal{O}(b,\pi(\tilde{\mathbf{z}}_u)) = \mathcal{O}(b',\pi'(\tilde{\mathbf{z}}_v))\right) + \mathbb{P}\mathrm{r}\left(\mathcal{O}(b,\pi(\tilde{\mathbf{z}}_u)) < \mathcal{O}(b',\pi'(\tilde{\mathbf{z}}_v))\right)$$

With probability at least $(1-\delta)$, the verifier picks a vertex $a \in \mathcal{W}_\Phi$ such that the assignment $\mathcal{A}$ satisfies all the edges (a,b). In this case, for all choices of $b, b' \in N(a)$, $\pi(\mathcal{A}(a)) = \mathcal{A}(b)$ and $\pi'(\mathcal{A}(a)) = \mathcal{A}(b')$. Let us denote $\mathcal{A}(a) = l$. By definition of the m-ordering $\mathcal{O}$, we get $\mathcal{O}(b,\pi(\mathbf{z})) = (\pi(\mathbf{z}))^{(\mathcal{A}(b))} = z^{(\pi^{-1}(\mathcal{A}(b)))} = z^{(l)}$ for all $\mathbf{z} \in [m]^R$. Similarly for b', $\mathcal{O}(b',\pi'(\mathbf{z})) = z^{(l)}$ for all $\mathbf{z} \in [m]^R$. Thus we get

$$\mathrm{val}_m(\mathcal{O}) \geqslant (1-\delta)\cdot\left(\frac{1}{2}\,\mathbb{P}\mathrm{r}\left(\tilde{z}_u^{(l)} = \tilde{z}_v^{(l)}\right) + \mathbb{P}\mathrm{r}\left(\tilde{z}_u^{(l)} < \tilde{z}_v^{(l)}\right)\right)$$

With probability at least $(1-2\varepsilon)^2$, for both $\tilde{\mathbf{z}}_u$ and $\tilde{\mathbf{z}}_v$ we have $\tilde{z}_u^{(l)} = z_u^{(l)}$ and $\tilde{z}_v^{(l)} = z_v^{(l)}$. Further, note that each coordinate $z_u^{(l)}, z_v^{(l)}$ is generated according to the local distribution μ_e for the edge $e = (u,v)$. Substituting in the expression for $\mathrm{val}_m(\mathcal{O})$ we get,

$$\mathrm{val}_m(\mathcal{O}) \geqslant (1-\delta)(1-2\varepsilon)^2 \mathop{\mathbb{E}}_{e=(u,v)}\left[\mathop{\mathbb{P}\mathrm{r}}_{(x_u,x_v)\in\mu_e}\left\{x_u < x_v\right\} + \frac{1}{2}\mathop{\mathbb{P}\mathrm{r}}_{(x_u,x_v)\in\mu_e}\left\{x_u = x_v\right\}\right]$$

Recall that the SDP solution $(\boldsymbol{V},\boldsymbol{\mu})$ have an objective value at least $(1-\eta)$. Thus for small enough choice of δ and ε , we have $\mathrm{val}_m(\mathcal{O}) \geqslant 1 - 5\eta$.

9.7.2 Soundness

Let $\mathcal{O}$ be an ordering of Ψ with $\mathrm{val}(\mathcal{O}) \geqslant \frac{1}{2} + 4\eta$. Using the ordering, we will obtain a labelling $\mathcal{A}$ for the Unique Games instance Φ. Towards this, we shall build machinery to deal with multiple long codes. For $b \in \mathcal{V}_\Phi$, define $\mathcal{O}_b$ as the restriction of the map $\mathcal{O}$ to vertices corresponding to the long code of b. Formally, $\mathcal{O}_b$ is a map $\mathcal{O}_b : [m]^R \to \mathbb{Z}$ given by $\mathcal{O}_b(\mathbf{z}) = \mathcal{O}(b,\mathbf{z})$. Similarly, for a vertex $a \in \mathcal{W}_\Phi$, let $\mathcal{O}_a$ denote the restriction of the map $\mathcal{O}$ to the vertices $N(a)\times[m]^R$, i.e. $\mathcal{O}_a(b,\mathbf{z}) = \mathcal{O}(b,\mathbf{z})$.

Multiple Long Codes

Throughout this section, we shall fix a vertex $a \in \mathcal{W}_\Phi$ and analyze the long codes corresponding to all neighbours of a. For a neighbour $b \in N(a)$, we shall use π_b to denote the permutation along the edge (a,b). Let $\mathcal{F}_b^{[s,t]}$ denote the functions associated with the ordering $\mathcal{O}_b$. Define functions $\mathcal{F}_a^{[s,t]} : [m]^R \to \mathbb{R}$ as follows:

$$\mathcal{F}_a^{[s,t]}(\mathbf{z}) = \mathop{\mathbb{P}\mathrm{r}}_{b\in N(a)}\left(\mathcal{O}_a(b,\pi_b(\mathbf{z})) \in [s,t]\right) = \mathop{\mathbb{E}}_{b\in N(a)}[\mathcal{F}_b^{[s,t]}(\pi_b(\mathbf{z}))]$$

Definition 9.7.1. Define the set of influential coordinates $\mathsf{L}_\tau(\mathcal{O}_a)$ as follows:

$$\mathsf{L}_\tau(\mathcal{O}_a) = \{k | \mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}_a^{[s,t]}) \geqslant \tau \text{ for some } s,t \in \mathbb{Z}\}$$

An ordering $\mathcal{O}_a$ is said to be τ-pseudorandom if $\mathsf{L}_\tau(\mathcal{O}_a)$ is empty.

Lemma 9.7.1. *For any influential coordinate $k \in \mathsf{L}_\tau(\mathcal{O}_a)$, for at least $\frac{\tau}{2}$ fraction of $b \in N(a)$, $\pi_b(k)$ is influential on $\mathcal{O}_b$. More precisely, $\pi_b(k) \in \mathsf{L}_{\tau/2}(\mathcal{O}_b)$.*

Proof. As the coordinate k is influential on $\mathcal{O}_a$, there exists s, t such that $\mathrm{Inf}_k(\mathcal{F}_a^{[s,t]}) \geqslant \tau$. Recall that $\mathcal{F}_a^{[s,t]}(\mathbf{z}) = \mathbb{E}_{b \in N(a)}[\mathcal{F}_b^{[s,t]}(\pi_b(\mathbf{z}))]$. Using convexity of Inf this implies, $\mathbb{E}_{b \in N(a)}[\mathrm{Inf}_{\pi_b(k)}(\mathcal{F}_b^{[s,t]})] \geqslant \tau$. All the influences $\mathrm{Inf}_{\pi_b(k)}(\mathcal{F}_b^{[s,t]})$ are bounded by 1, since each of the functions $\mathcal{F}_b^{[s,t]}$ take values in the range $[0, 1]$. Therefore for at least $\tau/2$ fraction of vertices $b \in N(a)$, we have $\mathrm{Inf}_{\pi_b(k)}(\mathcal{F}_b^{[s,t]}) \geqslant \tau/2$. This concludes the proof. ∎

Lemma 9.7.2. *For any vertex $a \in \mathcal{W}_\Phi$, $|\mathsf{L}_\tau(\mathcal{O}_a)| \leqslant 800/\varepsilon\tau^4$.*

Proof. From Lemma 9.7.1, for each coordinate $k \in \mathsf{L}_\tau(\mathcal{O}_a)$ there is a corresponding coordinate $\pi_b(k)$ in $\mathsf{L}_{\tau/2}(\mathcal{O}_b)$ for at least $\tau/2$ fraction of the neighbours b. Further from Lemma 9.4.3, the size of each set $\mathsf{L}_{\tau/2}(\mathcal{O}_b)$ is at most $400/\varepsilon\tau^3$. By double counting, we get that $|\mathsf{L}_\tau(\mathcal{O}_a)|$ is at most $800/\varepsilon\tau^4$. ∎

Theorem 9.8. *For all $\varepsilon, \eta > 0$, there exists constants $q, \tau > 0$ such that for any vertex $a \in \mathcal{W}_\Phi$, if $\mathcal{O}_a$ is τ-pseudorandom then $\mathrm{val}(\mathcal{O}_a) \leqslant \mathrm{val}_q(G) + \eta/4$.*

Proof. The proof outline is similar to that of Theorem 9.6. Let $\mathcal{O}_a^*$ denote the q-coarsening of $\mathcal{O}_a$. Then we can write,

$$\mathrm{val}(\mathcal{O}_a) \leqslant \mathrm{val}_q(\mathcal{O}_a^*) + \frac{1}{2}\mathbb{P}\mathrm{r}\left(\mathcal{O}_a^*(b, \pi_b(\tilde{\mathbf{z}}_u)) = \mathcal{O}_a^*(b', \pi_{b'}(\tilde{\mathbf{z}}_v))\right)$$

The q-coarsening $\mathcal{O}_a^*$ is obtained by dividing the order $\mathcal{O}_a$ into q-blocks. Let $[p_1+1, p_2], [p_2+1, p_3], \ldots, [p_q+1, p_{q+1}]$ denote the q blocks. For the sake of brevity, let us denote $\mathcal{F}_a^i = \mathcal{F}_a^{[p_i+1, p_{i+1}]}$ and $\mathcal{F}_b^i = \mathcal{F}_b^{[p_i+1, p_{i+1}]}$. In this notation, we can write:

$$\begin{aligned}
\mathbb{P}\mathrm{r}\left(\mathcal{O}_a^*(b, \pi_b(\tilde{\mathbf{z}}_u)) = \mathcal{O}_a^*(b', \pi_{b'}(\tilde{\mathbf{z}}_v))\right) &= \sum_{i \in [q]} \mathop{\mathbb{E}}_{e=(u,v)} \mathop{\mathbb{E}}_{b,b'} \mathop{\mathbb{E}}_{\mathbf{z}_u, \mathbf{z}_v, \tilde{\mathbf{z}}_u, \tilde{\mathbf{z}}_v} \left[\mathcal{F}_b^i(\pi_b(\tilde{\mathbf{z}}_u)) \cdot \mathcal{F}_{b'}^i(\pi_{b'}(\tilde{\mathbf{z}}_v))\right] \\
&= \sum_{i \in [q]} \mathop{\mathbb{E}}_{e=(u,v)} \mathop{\mathbb{E}}_{\mathbf{z}_u, \mathbf{z}_v} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_u, \tilde{\mathbf{z}}_v} \left[\mathcal{F}_a^i(\tilde{\mathbf{z}}_u) \cdot \mathcal{F}_a^i(\tilde{\mathbf{z}}_v)\right] \\
&= \sum_{i \in [q]} \mathop{\mathbb{E}}_{e=(u,v)} \mathop{\mathbb{E}}_{\mathbf{z}_u, \mathbf{z}_v} \left[T_{1-2\varepsilon}\mathcal{F}_a^i(\mathbf{z}_u) \cdot T_{1-2\varepsilon}\mathcal{F}_a^i(\mathbf{z}_v)\right]
\end{aligned}$$

As the ordering $\mathcal{O}_a$ is τ-pseudorandom, for every $k \in [R]$ and $i \in [q]$, $\mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}_a^i) \leqslant \tau$. Hence by Lemma 9.4.1, the above value is less than $O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$.

Now we shall bound the value of $\mathrm{val}_q(\mathcal{O}_a^*)$. In terms of the functions $\mathcal{F}_b^i$, the expression

for $\mathrm{val}_q(\mathcal{O}_a^*)$ is as follows:

$$\begin{aligned}\mathrm{val}_q(\mathcal{O}_a^*) &= \mathbb{E}\left[\frac{1}{2}\sum_{i=j}\mathcal{F}_b^i(\pi_b(\tilde{\mathbf{z}}_u))\cdot\mathcal{F}_{b'}^j(\pi_{b'}(\tilde{\mathbf{z}}_v)) + \sum_{i<j}\mathcal{F}_b^i(\pi_b(\tilde{\mathbf{z}}_u))\cdot\mathcal{F}_{b'}^j(\pi_{b'}(\tilde{\mathbf{z}}_v))\right] \\ &= \mathbb{E}\left[\frac{1}{2}\sum_{i=j}\mathcal{F}_a^i(\tilde{\mathbf{z}}_u)\cdot\mathcal{F}_a^j(\tilde{\mathbf{z}}_v) + \sum_{i<j}\mathcal{F}_a^i(\tilde{\mathbf{z}}_u)\cdot\mathcal{F}_a^j(\tilde{\mathbf{z}}_v)\right]\end{aligned}$$

Again, since the ordering $\mathcal{O}_a$ is τ-pseudorandom, for every $k \in [R]$ and $i \in [q]$, $\mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{F}_a^i) \leqslant \tau$. Hence by Claim 9.6.1, the above value is bounded by $\mathrm{val}_q(G) + o_\tau(1)$. From the above inequalities, we get $\mathrm{val}(\mathcal{O}_a) \leqslant \mathrm{val}_q(G) + O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$, which finishes the proof. ■

Defining a Labelling

Define the labelling $\mathcal{A}$ for the Unique Games instance Φ as follows: For each $a \in \mathcal{W}_\Phi$, $\mathcal{A}(a)$ is a uniformly random element from $\mathsf{L}_\tau(\mathcal{O}_a)$ if it is non-empty, and a random label otherwise. Similarly for each $b \in \mathcal{V}_\Phi$, assign $\mathcal{A}(b)$ to be a random element of $\mathsf{L}_{\tau/2}(\mathcal{O}_b)$ if it is nonempty, else an arbitrary label.

If $\mathrm{val}(\mathcal{O}) = \mathbb{E}_{a\in\mathcal{W}_\Phi}[\mathrm{val}(\mathcal{O}_a)] \geqslant \frac{1}{2}+4\eta$, then for at least 2η fraction of vertices $a \in \mathcal{W}_\Phi$, we have $\mathrm{val}(\mathcal{O}_a) \geqslant \frac{1}{2}+2\eta$. Let us refer to these vertices a as *good* vertices. From Theorem 9.8, for every *good* vertex the order $\mathcal{O}_a$ is not τ-pseudorandom. In other words, for every *good* vertex a, the set $\mathsf{L}_\tau(\mathcal{O}_a)$ is non-empty. Further by Lemma 9.7.1 for every label $l \in \mathsf{L}_\tau(\mathcal{O}_a)$, for at least $\tau/2$ fraction of the neighbours $b \in N(a)$, $\pi_b(l)$ belongs to $\mathsf{L}_{\tau/2}(\mathcal{O}_b)$. For every such b, the edge (a,b) is satisfied with probability at least $1/|\mathsf{L}_\tau(\mathcal{O}_a)| \times 1/|\mathsf{L}_{\tau/2}(\mathcal{O}_b)|$. By Lemma 9.4.3 and Lemma 9.7.2, this probability is at least $\varepsilon\tau^4/800 \times \varepsilon\tau^3/3200$. Summarizing the argument, the expected fraction of edges satisfied by the labelling $\mathcal{A}$ is at least $\eta\varepsilon^2\tau^8/10240000$. By a small enough choice of δ, this yields the required result.

9.8 Ordering CSP

In this section, we will state a general UG hardness result for Ordering Constraint Satisfaction Problems (OCSP) and outline the central ideas of the proof. To this end, we begin by defining ordering constraint satisfaction problems.

Definition 9.8.1. An Ordering Constraint Satisfaction Problem (OCSP) Λ is specified by a family of *payoff functions* $P : \Pi_k \to [-1,1]$ on the set Π_k of permutations on k elements. The integer k is referred to as the arity of the OCSP Λ.

Notice that every payoff $P \in \Lambda$ is assumed to be on the set of permutations of exactly k elements. However, there is no loss of generality since for every $q \leqslant k$, a payoff on set Π_q of permutations on q elements can be expressed as a payoff on Π_k by including dummy variables.

For $m \geqslant k$, let $\Pi_{k\to\mathbb{N}}$ denote the set of one to one maps from $[k] \to \mathbb{N}$. The domain of a payoff function P can be extended naturally from the set of permutations Π_k to $\Pi_{k\to\mathbb{N}}$. In particular, an injective map $f \in \Pi_{k\to\mathbb{N}}$, along with the ordering on the range $\mathbb{N}$ induces a permutation π_f on $[k]$. To extend the payoff, just define $P(f) = P(\pi_f)$ for all $f \in \Pi_{k\to\mathbb{N}}$.

Definition 9.8.2 (Λ-OrderingConstraintSatisfactionProblem (OCSP))**.** An instance $\Im$ of Ordering Constraint Satisfaction Problem Λ is given by $\Im = (\mathcal{V}, \mathcal{P})$ where

- $\mathcal{V} = \{y_1, \ldots, y_m\}$ is the set of variables that need to be ordered. Thus an ordering $\mathcal{O}$ is a one to one map from $\mathcal{V}$ to natural numbers $\mathbb{N}$.

- $\mathcal{P}$ is a probability distribution over constraints/payoffs applied to subsets of at most k variables from $\mathcal{V}$. More precisely, a sample $P \sim \mathcal{P}$ would be a payoff function from Λ, applied on a sequence of variables $y_S = (y_{s_1}, \ldots, y_{s_k})$. If $\mathcal{O}_{|S}$ denotes the injective map from $y_S \to \mathbb{N}$ obtained by restricting $\mathcal{O}$ to y_S, then the payoff returned is $P(\mathcal{O}_{|S})$.

For a payoff $P \in \mathcal{P}$, we define $\mathcal{V}(P) \in \mathcal{V}$ to denote the set of variables on which P is applied. The objective is to find an ordering $\mathcal{O}$ of the variables that maximizes the total weighted payoff/expected payoff, i.e.,

$$\mathop{\mathbb{E}}_{P \sim \mathcal{P}} \left[P(\mathcal{O}_{|P}) \right]$$

Here $\mathcal{O}_{|P}$ denotes the ordering $\mathcal{O}$ restricted to the variables in $\mathcal{V}(P)$. We define the value $\mathrm{opt}(\mathcal{P})$ as

$$\mathrm{opt}(\Im) \stackrel{\text{def}}{=} \max_{\mathcal{O}:\Pi_{\mathcal{V}\to\mathbb{N}}} \mathop{\mathbb{E}}_{P \sim \mathcal{P}} P(\mathcal{O}_{|P}) \,.$$

Observe that if the payoff functions P are predicates, then maximizing the payoff amounts to maximizing the number of constraints satisfied.

We will use Λ to denote both the OCSP and the family of payoffs associated with it. The notions "payoff function" and "constraint" will be used interchangeably.

9.8.1 Relation to CSPs

An ordering $\mathcal{O}$ can be thought of as an assignment of values from $\{1, \ldots, m\}$ to each variable y_i such that $y_i \neq y_j$ for all $i \neq j$. By suitably extending the payoff functions $P \in \Lambda$, it is possible to eliminate the "one to one" condition ($y_i \neq y_j$ whenever $i \neq j$). More precisely, we shall extend the domain of payoff functions $P \in \Lambda$ from $\Pi_{k\to[m]}$ to $\mathbb{N}^{[k]}$ - the set of all maps from $[k]$ to $\mathbb{N}$.

Given an arbitrary function $f : [k] \to \mathbb{N}$, define a probability distribution P_f on the set of permutations Π_k by the following random procedure: 1) For each $j \in \mathbb{N}$ with $f^{-1}(j) \neq \phi$, pick a uniform random permutation π_j of elements in $f^{-1}(j)$. 2) Concatenate the permutations π_j in the natural ordering on $j \in \mathbb{N}$ to obtain the permutation $\pi \in \Pi_k$. For a payoff $P \in \Lambda$, define

$$P(f) = \mathop{\mathbb{E}}_{\pi \sim P_f} [P(\pi)]$$

With this extension of payoff functions, the following lemma shows that optimizing over all orderings is equivalent to optimizing over all assignments of values in $[m]$ to variables $\{y_1, \ldots, y_m\}$.

Lemma 9.8.1. *For an instance $\Im = (\mathcal{V}, \mathcal{P})$ of a Λ-OCSP with $|\mathcal{V}| = m$, we have*

$$\max_{\mathcal{O}\in\Pi_{\mathcal{V}\to\mathbb{N}}} \mathop{\mathbb{E}}_{P\in\mathcal{P}} P(\mathcal{O}_{|P}) = \max_{f\in[m]^{\mathcal{V}}} \mathop{\mathbb{E}}_{P\in\mathcal{P}} P(f_{|P})$$

Here $[m]^{\mathcal{V}}$ denotes the set of all functions from $\mathcal{V}$ to $[m]$.

Proof. For every injective map $\mathcal{O} : \mathcal{V} \to \mathbb{N}$, there is an injective map $\mathcal{O}' : \mathcal{V} \to [m]$ corresponding to the permutation induced by $\mathcal{O}$. Clearly, the objective value of $\mathcal{O}$ is the same as $\mathcal{O}'$. Since $\mathcal{O}' \in [m]^{\mathcal{V}}$, we have

$$\max_{\mathcal{O} \in \Pi_{\mathcal{V} \to \mathbb{N}}} \mathop{\mathbb{E}}_{P \in \mathcal{P}} P(\mathcal{O}_{|P}) \leqslant \max_{f \in [m]^{\mathcal{V}}} \mathop{\mathbb{E}}_{P \in \mathcal{P}} P(f_{|P})$$

Given an arbitrary function $f : \mathcal{V} \to [m]$, define a probability distribution D_f on the orderings $\mathcal{O} \in \Pi_{\mathcal{V} \to [m]}$ by the following random procedure: 1) For each $j \in [m]$ with $f^{-1}(j) \neq \phi$, pick a uniform random permutation π_j of elements in $f^{-1}(j)$. 2) Concatenate the permutations π_j in the natural ordering on $j \in \mathbb{N}$ to obtain the ordering $\mathcal{O} \in \Pi_{\mathcal{V} \to [m]}$. By our definition of extended payoffs P, it easily follows that,

$$\mathop{\mathbb{E}}_{P \in \mathcal{P}} P(f_{|P}) = \mathop{\mathbb{E}}_{\mathcal{O} \in D_f} \left[\mathop{\mathbb{E}}_{P \in \mathcal{P}} P(\mathcal{O}_{|P}) \right].$$

In turn, this implies that

$$\max_{\mathcal{O} \in \Pi_{\mathcal{V} \to \mathbb{N}}} \mathop{\mathbb{E}}_{P \in \mathcal{P}} P(\mathcal{O}_{|P}) \geqslant \max_{f \in [m]^{\mathcal{V}}} \mathop{\mathbb{E}}_{P \in \mathcal{P}} P(f_{|P}),$$

thus finishing the proof. ■

By virtue of Lemma 9.8.1, the Λ-OCSP instance $\Im = (\mathcal{V}, \mathcal{P})$ is transformed into a constraint satisfaction problem over variables $\mathcal{V}$, albeit over a domain $[m]$ whose size is not fixed. Specifically, the problem of finding an optimal ordering $\mathcal{O}$ for the Λ-OCSP instance can be reformulated as computing

$$\mathrm{val}(\Im) = \max_{y \in [m]^m} \mathop{\mathbb{E}}_{P \in \mathcal{P}} \left[P(y_{\mathcal{V}(P)}) \right] \tag{9.1}$$

Here we are slightly abusing notation to denote the payoff P to be a function over the assignment $y_{\mathcal{V}(P)}$ itself. For the sake of convenience, we will use y_P to denote $y_{\mathcal{V}(P)}$.

Taking the analogy with CSPs a step further, one can define a CSP Λ_q for every positive integer $q > 0$. Given an instance $\Im = (\mathcal{V}, \mathcal{P})$ of Λ-OCSP, the corresponding Λ_q problem is to find a q-ordering that maximizes the expected payoff. Formally, the goal of the Λ_q-CSP instance $\Im$ is to compute an assignment $y \in [q]^m$ that is the maximizes the following:

$$\mathrm{val}_q(\Im) = \max_{y \in [q]^m} \mathop{\mathbb{E}}_{P \in \mathcal{P}} \left[P(y_P) \right] \tag{9.2}$$

The following claim is an easy consequence of the above definitions:

Claim 9.8.1. *For every Λ-OCSP instance $\Im = (\mathcal{V}, \mathcal{P})$, and integers $q \leqslant q'$*

$$\mathrm{val}_q(\Im) \leqslant \mathrm{val}_{q'}(\Im) \leqslant \mathrm{val}(\Im),$$

Further, if $|\mathcal{V}| = m$ then $\mathrm{val}_m(\Im) = \mathrm{val}(\Im)$.

9.8.2 SDP Relaxation

LC Relaxation (Equivalent Version)

$$\begin{aligned}
\text{maximize} \quad & \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \\
\text{subject to} \quad & \langle \boldsymbol{b}_{s,a}, \boldsymbol{b}_{s',b}\rangle = \Pr_{x\sim\mu_P}\left\{x_s = a, x_{s'} = b\right\} && (P\in\operatorname{supp}(\mathcal{P}),\ \ s,s'\in\mathcal{V}(P),\ \ a,b\in[m])\,. \\
& && (9.3)\\
& \langle \boldsymbol{b}_{s,a}, \boldsymbol{b}_0\rangle = \|\boldsymbol{b}_{s,a}\|_2^2 && \forall s\in\mathcal{V}, a\in[m]\,, && (9.4)\\
& \|\boldsymbol{b}_0\|_2^2 = 1 && && (9.5)\\
& \mu_P \in \blacktriangle([q]^{\mathcal{V}(P)}) && \forall P\in\operatorname{supp}(\mathcal{P})
\end{aligned}$$

A *coarsening gap* instance has much weaker properties than a *integrality gap* instance, thus making it easier to construct.

Definition 9.8.3. An instance $\Im$ of a Λ-OCSP is a (q,c,s)-coarsening gap instance if $\operatorname{sdp}(\Im)\geqslant c$ and $\operatorname{val}_q(\Im)\leqslant s$.

Smoothing Coarsening Gaps

Definition 9.8.4. For $\alpha>0$, a (q,c,s)-coarsening gap instance $\Im=(\mathcal{V},\mathcal{P})$ over m variables is said to be α-smooth if for every $P\in\mathcal{P}$ and $x\in[m]^k$, $\mu_{P,x}\geqslant\alpha$.

Here we will outline a transformation on coarsening gap instance $\Im^*$, to another coarsening gap instance $\Im$ with certain special properties including α-smoothness. Note that the smoothness parameter of the resulting solutions is $\alpha=\frac{\eta}{10m^k}$.

Lemma 9.8.2. *For all $\eta>0$ the following holds, given a (q,c,s)-coarsening gap instance $\Im^*=(\mathcal{V}^*,\mathcal{P}^*)$ of a Λ-OCSP, for large enough m, there exists a $(q,c-\eta/5,s+\eta/5)$-coarsening gap instance $\Im=(\mathcal{V},\mathcal{P})$ on m variables, an SDP solution $\{\boldsymbol{v}_{i,a}\}_{i\in\mathcal{V},a\in[M]},\{\mu_P\}_{P\in\operatorname{supp}(\mathcal{P})}$ and a vector $\boldsymbol{b}_0$ satisfying*

$$\begin{aligned}
\langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{i,a}\rangle &= \frac{1}{m} && \forall i\in\mathcal{V}, a\in[m]\,, && (9.6)\\
\mu_{P,x} &\geqslant \frac{\eta}{10m^k} && \forall P\in\mathcal{P}, x\in[m]^k\,, && (9.7)
\end{aligned}$$

and

$$\mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x)\geqslant c-\frac{\eta}{5} \qquad \operatorname{val}_q(\Im)\leqslant s+\frac{\eta}{5}$$

Proof. Intuitively, the SDP solution corresponding to instance $\Im$ assigns each of the variables $y_i\in\mathcal{V}$ each of the locations in $[m]$ with equal probability. $\Im$ is constructed by taking many copies of $\Im^*$ and joining them side by side such that cyclic shifts of orderings obtain around the same payoff.

More formally, let $L = \lceil \frac{20}{\eta} \rceil$ and set $\mathcal{V} = \mathcal{V}^* \times [L]$. The distribution $\mathcal{P}$ is obtained by simply the product distribution of $\mathcal{P}^*$ and the uniform distribution over $[L]$. That is, for every $p = (y_1, y_2, \ldots y_k)$ in the support of $\mathcal{P}^*$ and for every $l \in [L]$, $\mathbb{P}\mathrm{r}_{\mathcal{P}}((y_1, l), (y_2, l), \ldots, (y_k, l)) = \mathbb{P}\mathrm{r}_{\mathcal{P}^*}(p)/L$.

Let $\mathcal{O}$ be an optimal ordering for $\Im$. Let $m = |\mathcal{V}| = L|\mathcal{V}^*|$. For every $i \in [m]$, define ordering $\mathcal{O}^*_{(i)} : \mathcal{V} \to [m]$ to be $\mathcal{O}^*(v, k) = i + k|\mathcal{V}| + \mathcal{O}(v)$ (addition modulo m). Since except for at most one copy of $\mathcal{P}^*$, every other constraint is ordered as in $\mathcal{O}$, the payoff of $\mathcal{O}^*_{(i)}$ is at least $c - \eta/20$.

Further, since the q-ordering value of $\mathcal{P}$ is simply the average of the q-ordering values of the individual pieces, $\mathrm{val}_q(\mathcal{P}) \leqslant s$.

To construct the vectors, we consider the distribution over assignments obtained by taking, with probability $1 - \eta/10$, one of $\mathcal{O}^*_{(i)}$ with equal probability and taking a completely random assignment with probability $\eta/10$. It is easy to see that the probability $y \in \mathcal{V}$ is assigned $a \in [m]$ is exactly $1/m$. Further, since we take a completely random assignment with probability $\eta/10$, for any constraints $p \in \mathcal{P}$ and $x \in [m]^k$, the distribution assigns x to p with probability at least $\frac{\eta}{10m^k}$. The payoff obtained by this distribution is at least $(1 - \eta/10)(c - \eta/20) \geqslant c - \eta/5$. The distribution over assignments naturally gives vectors satisfying the required constraints. ■

Abusing notation, henceforth we shall use $\Im$ to denote the smoothed coarsening gap instance $\Im^*$.

9.9 Dictatorship Test for OCSP

In this section, we will construct a dictatorship test for an OCSP Λ starting with a coarsening gap instance $\Im$ for the problem. Formally, let $\Im^* = (\mathcal{V}^*, \mathcal{P}^*)$ be a (q, c, s) coarsening gap instance with $|\mathcal{V}| = m$. Let $\Im = (\mathcal{V}, \mathcal{P})$ denote the $(q, c - \frac{\eta}{5}, s + \frac{\eta}{5})$-coarsening gap instance, which is $\alpha = \eta/10m^k$-smooth, obtained from Lemma 9.8.2. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote the SDP solution associated with the instance $\Im$. Define a dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ on orderings $\mathcal{O}$ of $[m]^R$ as follows:

$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ Test
Let $\Im^* = (\mathcal{V}^*, \mathcal{P}^*)$ be a (q, c, s) coarsening gap instance with $|\mathcal{V}| = m$. Let $\Im = (\mathcal{V}, \mathcal{P})$ denote the $(q, c - \frac{\eta}{5}, s + \frac{\eta}{5})$-coarsening gap instance, which is $\alpha = \eta/10m^k$-smooth, obtained from Lemma 9.8.2. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote the SDP solution associated with the instance $\Im$.

- Sample a payoff P from the distribution $\mathcal{P}$. Let $\mathcal{V}(P) = S = \{s_1, s_2, \ldots, s_k\}$.
- Sample $\mathbf{z}_S = \{\mathbf{z}_{s_1}, \ldots, \mathbf{z}_{s_k}\}$ from the product distribution μ_P^R, i.e. For each $1 \leqslant j \leqslant R$, $z_S^{(j)} = \{z_{s_1}^{(j)}, \ldots, z_{s_k}^{(j)}\}$ is sampled using the local distribution μ_P on $[m]^{\mathcal{V}(P)}$.
- For each $s_i \in S$ and each $1 \leqslant j \leqslant R$, sample $\tilde{z}_{s_i}^j$ as follows: With probability $(1 - \varepsilon)$, $\tilde{z}_{s_i}^{(j)} = z_{s_i}^{(j)}$, and with the remaining probability $\tilde{z}_{s_i}^{(j)}$ is a uniform random element from $[m]$.
- Query the ordering values $\mathcal{O}(\tilde{\mathbf{z}}_{s_1}), \ldots, \mathcal{O}(\tilde{\mathbf{z}}_{s_q})$.
- Return the Pay-Off : $P\Big(\mathcal{O}\big(\tilde{\mathbf{z}}_{s_1}\big), \ldots, \mathcal{O}\big(\tilde{\mathbf{z}}_{s_k}\big)\Big)$

Completeness It is fairly simple to check that the completeness of the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is close to the SDP value of $\Im$. Specifically, we will now show,

Lemma 9.8.3.

$$\mathsf{Completeness}(\mathit{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\varepsilon k = c - \frac{\eta}{5} - 2\varepsilon k$$

Proof. A dictator "m-ordering" $\mathcal{O}$ is given by $\mathcal{O}(\mathbf{z}) = z^{(j)}$. The expected payoff returned by the verifier $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ on $\mathcal{O}$ is given by

$$\mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z_S}} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_S} \Big[P\Big(\mathcal{O}\big(\tilde{\mathbf{z}}_{s_1}\big), \ldots, \mathcal{O}\big(\tilde{\mathbf{z}}_{s_k}\big)\Big)\Big] = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z_S}} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_S} \Big[P_S\Big(\tilde{z}_{s_1}^{(j)}, \ldots, \tilde{z}_{s_k}^{(j)}\Big)\Big]$$

With probability $(1 - \varepsilon)^k$, $\tilde{z}_{s_i}^{(j)} = z_{s_i}^{(j)}$ for each $s_i \in S$. Further the payoff functions $P \in \mathcal{P}$ take values in $[-1, 1]$. Hence a lower bound for the expected payoff is given by

$$\mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z_S}} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_S} \Big[P\Big(\mathcal{O}\big(\tilde{z}_{s_1}\big), \ldots, \mathcal{O}\big(\tilde{z}_{s_q}\big)\Big)\Big] \geqslant (1-\varepsilon)^k \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z_S}} \Big[P\Big(z_{s_1}^{(j)}, \ldots, z_{s_q}^{(j)}\Big)\Big] + (1 - (1-\varepsilon)^k) \cdot (-1)$$

The j^{th} coordinates $\mathbf{z}_S^{(j)} = \{z_{s_1}^{(j)}, \ldots, z_{s_q}^{(j)}\}$ are generated from the local probability distribution μ_P. Thus we get,

$$\mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z_S}} \Big[P\Big(z_{s_1}^{(j)}, \ldots, z_{s_q}^{(j)}\Big)\Big] = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{x\in\mu_P} \Big[P(x)\Big] = \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) \tag{9.8}$$

The expected payoff is at least $(1 - \varepsilon)^k \cdot \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - (1 - (1 - \varepsilon)^k) \geqslant \mathrm{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\varepsilon k$. ■

Soundness The following soundness claim is an immediate consequence of Lemma 9.9.2 and Lemma 9.9.1.

Theorem 9.9. *(Soundness Analysis) For every $\varepsilon > 0$, for any τ-pseudorandom ordering $\mathcal{O}$ of $[m]^R$,*

$$\mathrm{val}(\mathcal{O}) \leqslant \mathrm{val}_q(\Im) + O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$$

where $o_\tau(1) \to 0$ as $\tau \to 0$ keeping all other parameters fixed.

Lemma 9.9.1. *For every $\varepsilon > 0$, for any τ-pseudorandom ordering $\mathcal{O}$ of $[m]^R$*

$$\mathrm{val}(\mathcal{O}) \leqslant \mathrm{val}_q(\mathcal{O}^*) + \binom{k}{2} q^{-\frac{\varepsilon}{2}} + o_\tau(1)$$

where $\mathcal{O}^*$ is the q-coarsening of $\mathcal{O}$ and k denotes the arity of the OCSP Λ.

Proof. Let $\mathcal{F}^{[s,t]} : [m]^R \to \{0,1\}$ denote the functions associated with the q-ordering $\mathcal{O}^*$. For the sake of brevity, we shall write $\mathcal{F}^i$ for $\mathcal{F}^{[i,i]}$.

Note that the loss due to coarsening, is because for some payoffs P the k variables in $\mathcal{V}(P)$ do not fall into distinct bins during coarsening. Let us upper bound the probability that some two of the variables queried $\tilde{\mathbf{z}}_{s_i}, \tilde{\mathbf{z}}_{s_j}$ fall into same block during coarsening, i.e. $\mathcal{O}^*(\tilde{\mathbf{z}}_{s_i}) = \mathcal{O}^*(\tilde{\mathbf{z}}_{s_j})$. Observe that,

$$\begin{aligned}
\mathbb{P}\mathrm{r}\left(\mathcal{O}^*(\tilde{\mathbf{z}}_{s_i}) = \mathcal{O}^*(\tilde{\mathbf{z}}_{s_j})\right) &= \sum_{i\in[q]} \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_{s_i},\mathbf{z}_{s_j}} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_{s_i},\tilde{\mathbf{z}}_{s_j}} \left[\mathcal{F}^i(\tilde{\mathbf{z}}_{s_i}) \cdot \mathcal{F}^i(\tilde{\mathbf{z}}_{s_j})\right] \\
&= \sum_{i\in[q]} \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_{s_i},\mathbf{z}_{s_j}} \left[T_{1-2\varepsilon}\mathcal{F}^i(\mathbf{z}_{s_i}) \cdot T_{1-2\varepsilon}\mathcal{F}^i(\mathbf{z}_{s_j})\right]
\end{aligned}$$

As $\mathcal{O}$ is a q-coarsening of $\mathcal{O}$, for each value $i \in [q]$, there are exactly $\frac{1}{q}$ fraction of $\mathbf{z}$ for which $\mathcal{O}^*(\mathbf{z}) = i$. Hence for each $i \in [q]$, $\mathbb{E}_{\mathbf{z}}[\mathcal{F}^i(\mathbf{z}) = \frac{1}{q}]$. Further, since the ordering $\mathcal{O}^*$ is τ-pseudorandom, for every $j \in [R]$ and $i \in [q]$, $\mathrm{Inf}_j(T_{1-\varepsilon}\mathcal{F}^i) \leqslant \tau$. Hence using Lemma 9.4.1, for sufficiently large q, the above probability is bounded by $q \cdot q^{-1-\frac{\varepsilon}{2}} + q \cdot o_\tau(1)$. By a simple union bound, the probability that two of the queried values fall in the same bin is at most $\binom{k}{2}\left(q \cdot q^{-1-\frac{\varepsilon}{2}} + q \cdot o_\tau(1)\right)$ As all the payoffs are bounded by 1 in absolute value, we can write

$$\begin{aligned}
\mathrm{val}(\mathcal{O}) &\leqslant \mathrm{val}_q(\mathcal{O}^*) + \mathbb{P}\mathrm{r}\left(\exists i, j \in [k] \text{ such that } \mathcal{O}^*(\tilde{\mathbf{z}}_{s_i}) = \mathcal{O}^*(\tilde{\mathbf{z}}_{s_j})\right) \\
&\leqslant \mathrm{val}_q(\mathcal{O}^*) + \binom{k}{2} q^{-\frac{\varepsilon}{2}} + o_\tau(1)
\end{aligned}$$

■

Lemma 9.9.2. *For every choice of m, q, ε, and any τ-pseudorandom q-ordering $\mathcal{O}^*$ of $[m]^R$, $\mathrm{val}_q(\mathcal{O}^*) \leqslant \mathrm{val}_q(\Im) + o_\tau(1)$.*

Proof. Let $\mathcal{F}^{[s,t]} : [m]^R \to \{0,1\}$ denote the functions associated with the q-ordering $\mathcal{O}^*$. For the sake of brevity, we shall write $\mathcal{F}^i$ for $\mathcal{F}^{[i,i]}$, and $\boldsymbol{\mathcal{F}} = (\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(q)})$. The expected

payoff returned by the verifier in the dictatorship test $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is given by,

$$\mathrm{val}_q(\mathcal{O}^*) = \mathop{\mathbb{E}}_{P\in\mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_S} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_S} \left[P\Big(\mathcal{F}\big(\tilde{\mathbf{z}}_{s_1}\big), \ldots, \mathcal{F}\big(\tilde{\mathbf{z}}_{s_k}\big) \Big) \right].$$

Further, since the ordering $\mathcal{O}^*$ is τ-pseudorandom, for every $j \in [R]$ we have $\mathrm{Inf}_j(T_{1-\varepsilon}\mathcal{F}^i) \leqslant \tau$. The proof follows from Lemma 9.9.3. ■

9.10 Soundness Analysis for q-Orderings

In this section, we will sketch the proof of Lemma 9.9.2 and Lemma 9.6.2. As Lemma 9.6.2 is a special case of Lemma 9.9.2, we will restrict ourselves to the proof of Lemma 9.9.2. The proof of Lemma 9.9.2 closely resembles the soundness analysis of dictatorship tests for the case of GCSPs (Theorem 7.5) . However, the lemma is not an direct consequence of Theorem 7.5. This is because, in the soundness analysis of Theorem 7.4 assumes that the domain of the function is $[q]^R$ while the output is also in $\blacktriangle_q$ for some q. For the sake of completeness we include a sketch of the proof here.

9.10.1 Payoff Functions

For the sake of the proof, we will extend the payoff functions P corresponding to the CSP Λ_q to *smooth* real valued functions on $\mathbb{R}^{tk}$. The details of the extension are identical to the case of GCSPs (Subsection 7.4.1).

9.10.2 Local and Global Distributions

Now, we shall describe two ensembles of random variables, namely the local integral ensembles $\mathcal{L}_P$ for each payoff P, and a global Gaussian ensemble $\mathcal{G}$.

Definition 9.10.1. For every payoff $P \in \mathcal{P}$ of size at most k, the Local Distribution μ_P is a distribution over $[m]^{\mathcal{V}(P)}$. In other words, the distribution μ_P is a distribution over assignments to the CSP variables in set $\mathcal{V}(P)$. The corresponding *Local Integral Ensemble* is a set of random variables $\mathcal{L}_P = \{\ell_{s_1}, \ldots, \ell_{s_k}\}$ each taking values in Δ_m.

Definition 9.10.2. The *Global* Ensemble $\mathcal{G} = \{\boldsymbol{g}_s | s \in \mathcal{V}, j \in [m]\}$ are generated by setting $\boldsymbol{g}_s = \{g_{s,1}, \ldots, g_{s,m}\}$ where

$$g_{s,j} = \langle \boldsymbol{b}_0, \boldsymbol{b}_{s,j} \rangle + \langle (\boldsymbol{b}_{s,j} - (\langle \boldsymbol{b}_0, \boldsymbol{b}_{s,j} \rangle)\boldsymbol{b}_0), \zeta \rangle$$

and ζ is a normal Gaussian random vector of appropriate dimension.

It is easy to see that the local and global integral ensembles have matching moments up to degree two.

Observation 9.10.1. *For any set $P \in \mathcal{P}$, the global ensemble $\mathcal{G}$ matches the following*

moments of the local integral ensemble $\mathcal{L}_P$

$$\mathbb{E}[g_{s,j}] = \mathbb{E}[\ell_{s,j}] = \langle \boldsymbol{b}_0, \boldsymbol{b}_{s,j} \rangle \qquad \mathbb{E}[g_{s,j}^2] = \mathbb{E}[\ell_{s,j}^2] = \langle \boldsymbol{b}_0, \boldsymbol{b}_{s,j} \rangle$$
$$\mathbb{E}[g_{s,j} g_{s,j'}] = \mathbb{E}[\ell_{s,j} \ell_{s,j'}] = 0 \qquad \forall j \neq j', s \in \mathcal{V}(P)$$

9.10.3 Putting It All Together

Finally, we will now show the following lemma which forms the core of the soundness argument in Lemma 9.9.2 and is a generalization of the Claim 9.6.1.

Lemma 9.9.3. *For a function* $\boldsymbol{\mathcal{F}} : [m]^R \to \blacktriangle_q$ *satisfying* $\mathrm{Inf}_j(T_{1-\varepsilon}\boldsymbol{\mathcal{F}}) \leqslant \tau$ *for all* $j \in [R]$,

$$\mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_S} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_S} \Big[P\Big(\boldsymbol{\mathcal{F}}\big(\tilde{\mathbf{z}}_{s_1}\big), \ldots, \boldsymbol{\mathcal{F}}\big(\tilde{\mathbf{z}}_{s_k}\big) \Big) \Big] \leqslant \mathrm{val}_q(\Im) + o_\tau(1)$$

Here $o_\tau(1) \to 0$ *as* $\tau \to 0$ *while all other parameters are fixed.*

Proof. Let us denote $\boldsymbol{\mathcal{H}} = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}$. Let $\mathbf{F}(\mathbf{x}), \boldsymbol{H}(\mathbf{x})$ denote the multilinear polynomials corresponding to functions $\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{H}}$ respectively. Let us denote,

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) = \mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_S} \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_S} \Big[P\Big(\boldsymbol{\mathcal{F}}\big(\tilde{\mathbf{z}}_{s_1}\big), \ldots, \boldsymbol{\mathcal{F}}\big(\tilde{\mathbf{z}}_{s_k}\big) \Big) \Big]$$

Each vector $\mathbf{z}_{s_i}$ is independently perturbed to obtain $\tilde{\mathbf{z}}_{s_i}$. The payoff functions P are multilinear when restricted to the domain $\blacktriangle_q$. Consequently, we can write

$$\begin{aligned} \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) &= \mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_S} \Big[P\Big(\mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_{s_1}} [\boldsymbol{\mathcal{F}}\big(\tilde{\mathbf{z}}_{s_1}\big) | \mathbf{z}_{s_1}], \ldots, \mathop{\mathbb{E}}_{\tilde{\mathbf{z}}_{s_1}} [\boldsymbol{\mathcal{F}}\big(\tilde{\mathbf{z}}_{s_q} | \mathbf{z}_{s_k}\big]) \Big) \Big] \\ &= \mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_S} \Big[P\Big(\boldsymbol{\mathcal{H}}\big(\mathbf{z}_{s_1}\big), \ldots, \boldsymbol{\mathcal{H}}\big(\mathbf{z}_{s_k}\big) \Big) \Big] \end{aligned}$$

The last equality is due to the fact $\mathbb{E}_{\tilde{\mathbf{z}}_{s_i}}[\boldsymbol{\mathcal{F}}_{s_i}(\tilde{\mathbf{z}}_{s_i}) | \mathbf{z}_{s_i}] = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}_{s_i}(\mathbf{z}_{s_i}) = \boldsymbol{\mathcal{H}}_{s_i}(z_{s_i})$. For each $s \in S$, the coordinates of $\mathbf{z}_s$ are generated by the distribution μ_P. Therefore the above expectation can be written in terms of the polynomial $\boldsymbol{H}$ applied integral ensemble $\mathcal{L}_P$. Specifically, we can write

$$\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) = \mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathbf{z}_S} \Big[P\Big(\boldsymbol{\mathcal{H}}\big(\mathbf{z}_{s_1}\big), \ldots, \boldsymbol{\mathcal{H}}\big(\mathbf{z}_{s_k}\big) \Big) \Big] = \mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathcal{L}_P^R} \Big[P\Big(\boldsymbol{H}\big(\ell_{s_1}^R\big), \ldots, \boldsymbol{H}\big(\ell_{s_k}^R\big) \Big) \Big] \tag{9.9}$$

The following procedure $\mathsf{Round}_{\boldsymbol{\mathcal{F}}}$ returns an ordering for the original Λ-OCSP instance $\Im$.

$\mathsf{Round}_{\boldsymbol{\mathcal{F}}}$ Scheme

Input : A Λ_q-OCSP instance $\Im = (\mathcal{V}, \mathcal{P})$ with a SDP solution $\{\boldsymbol{b}_{v,i}\}, \{\mu_P\}$.

Sample R vectors $\zeta^{(1)}, \ldots, \zeta^{(R)}$ with each coordinate being i.i.d normal random variable. For each $s \in \mathcal{V}$ do

- For all $1 \leqslant j \leqslant R$ and $c \in [m]$, compute the projection $g^{(j)}_{s,c}$ of the vector $\boldsymbol{b}_{s,c}$ as follows:

$$g^{(j)}_{s,c} = \langle \boldsymbol{b}_0, \boldsymbol{b}_{s,c} \rangle + \left[\langle (\boldsymbol{b}_{s,c} - (\langle \boldsymbol{b}_0, \boldsymbol{b}_{s,c} \rangle) \boldsymbol{b}_0), \zeta^{(j)} \rangle \right]$$

- Evaluate the function $\boldsymbol{\mathcal{H}} = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}$ with $g^{(j)}_{s,c}$ as inputs. In other words, compute $\mathbf{p}_s = (p_{s,1}, \ldots, p_{s,q})$ as follows:

$$\mathbf{p}_s = \boldsymbol{H}(\mathbf{g_s})$$

- Round $\mathbf{p}_s$ to $\mathbf{p}^*_s \in \blacktriangle_q$ using the following procedure.

$$f_{[0,1]}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leqslant x \leqslant 1 \\ 1 & \text{if } x > 1 \end{cases} \qquad \text{Scale}(x_1, x_2, \ldots, x_q) = \begin{cases} \frac{1}{\sum_i x_i}(x_1, \ldots, x_q) \text{if } \sum_i x_i \neq 0 \\ (1, 0, 0, \ldots, 0) \text{ if } \sum_i x_i = 0 \end{cases}$$

$$\mathbf{p}^*_s = \text{Scale}(f_{[0,1]}(p_{s,1}), \ldots, f_{[0,1]}(p_{s,q}))$$

- Assign the Λ-OCSP variable $y_s \in \mathcal{V}$ the value $j \in [q]$ with probability $p^*_{s,j}$.

Let $\mathsf{Round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu})$ denote the expected payoff of the ordering returned by the rounding scheme $\mathsf{Round}_{\boldsymbol{\mathcal{F}}}$ on the SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$. By definition, we have:

$$\mathsf{Round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) \leqslant \text{val}_q(\Im) \tag{9.10}$$

In the remainder of the proof, we will show the following inequality:

$$\mathsf{Round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) \leqslant \mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) + o_\tau(1)$$

Along with Equation 9.10, this would imply that $\mathsf{DICT}^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}})$ is less than $\text{val}_q(\Im) + o_\tau(1)$, thus showing the required claim. To this end, we will arithmetize the value of $\mathsf{Round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu})$. Notice that the $\boldsymbol{g}_i$ are nothing but samples of the Global Ensemble $\mathcal{G}$ associated with $\Im$. Further, let us denote by $\boldsymbol{H}^*(\boldsymbol{g}_s)$, the rounded value $\mathbf{p}^*_s$ of $\mathbf{p}_s = \boldsymbol{H}(\boldsymbol{g}_s)$. By definition, the expected payoff is given by

$$\mathsf{Round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathop{\mathbb{E}}_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{\mathcal{G}^R_P} \left[P\Big(\boldsymbol{H}^*\big(\mathbf{g}^R_{s_1}\big), \ldots, \boldsymbol{H}^*\big(\mathbf{g}^R_{s_k}\big) \Big) \right] \tag{9.11}$$

We will show that the quantities in Equation 9.9, 9.11 are roughly equal. Fix a payoff $P \in \mathcal{P}$. Apply the assertion 1 of the invariance principle (Theorem 3.2) with the ensembles $\mathcal{L}_P$, $\mathcal{G}_P$, smooth function P_S and the vector of kt multilinear polynomials given by $(\boldsymbol{H}, \boldsymbol{H}, \ldots, \boldsymbol{H})$ where $\boldsymbol{H} = (H_1, \ldots, H_q)$. As a consequence, we get

$$\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\left[P\Big(\boldsymbol{H}\big(\ell_{s_1}^R\big), \ldots, \boldsymbol{H}\big(\ell_{s_k}^R\big)\Big)\right] = \mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[P\Big(\boldsymbol{H}\big(\mathbf{g}_{s_1}^R\big), \ldots, \boldsymbol{H}\big(\mathbf{g}_{s_k}^R\big)\Big)\right] + o_\tau(1) \tag{9.12}$$

To show that the quantities in Equation 9.11 and 9.12 are roughly equal, we will appeal again to the invariance principle (Theorem 3.2). ■

9.10.4 Bounding the Rounding Error

The following claim bounds the loss incurred in the payoff due to rounding the assignment from $\mathbf{p}_s$ to $\mathbf{p}_s^*$.

Claim 9.9.1. *Let $\boldsymbol{\mathcal{F}} : [m]^R \to \blacktriangle_q$ be a function satisfying $\mathrm{Inf}_j(T_{1-\varepsilon}\boldsymbol{\mathcal{F}}) \leqslant \tau$ for all $j \in [R]$, and let $\boldsymbol{\mathcal{H}} = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}$. Let $\boldsymbol{H}$ denote the multilinear extension (a polynomial representing $\boldsymbol{\mathcal{H}}$), and let $\boldsymbol{H}^*$ denote the function $\boldsymbol{H}$ rounded to $\blacktriangle_q$.*

$$\left|\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[P\Big(\boldsymbol{H}^*\big(\mathbf{g}_{s_1}^R\big), \ldots, \boldsymbol{H}^*\big(\mathbf{g}_{s_q}^R\big)\Big)\right] - \mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[P\Big(\boldsymbol{H}_{s_1}\big(\mathbf{g}_{s_1}^R\big), \ldots, \boldsymbol{H}_{s_q}\big(\mathbf{g}_{s_q}^R\big)\Big)\right]\right| \leqslant o_\tau(1)$$

Proof. Intuitively, the invariance principle (Theorem 3.2) asserts that the distribution of the random variable $\boldsymbol{H}(\boldsymbol{g}_s^R)$ is roughly the same as that of $\boldsymbol{H}(\ell_s^R)$. Observe that on inputs from the local distribution ℓ_s^R, the value $\boldsymbol{H}(\ell_s^R)$ is always contained in $\blacktriangle_q$. This suggests that the random variables $\boldsymbol{H}(\boldsymbol{g}_s^R)$ is *nearly always* close to $\blacktriangle_q$. For a point $\mathbf{p} \in \blacktriangle_q$, its rounded value $\mathbf{p}^* = \mathbf{p}$. Thus, the rounding of $\boldsymbol{H}(\boldsymbol{g}_s^R)$ only slightly changes its value, i.e., $\boldsymbol{H}(\boldsymbol{g}_s^R) \approx \boldsymbol{H}^*(\boldsymbol{g}_s^R)$. Recall that the payoff functions P are smooth in that they satisfy Property II (Subsection 7.4.1). Therefore if $\boldsymbol{H}(\boldsymbol{g}_s^R) \approx \boldsymbol{H}^*(\boldsymbol{g}_s^R)$ for all s, the two quantities in the above claim are approximately equal.

Without loss of generality, we may assume that S on which the payoff P applies is $\{1, \ldots, k\}$. From Property II that the payoff functions satisfy,

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[P\Big(\boldsymbol{H}^*\big(\mathbf{g}_1^R\big), \ldots, \boldsymbol{H}^*\big(\mathbf{g}_k^R\big)\Big)\right] - \mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[P\Big(\boldsymbol{H}\big(\mathbf{g}_1^R\big), \ldots, \boldsymbol{H}\big(\mathbf{g}_k^R\big)\Big)\right]\right| \\ \leqslant C_0 \sum_{i=1}^{k}\Big(\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\|\boldsymbol{H}^*(\boldsymbol{g}_i^R) - \boldsymbol{H}(\boldsymbol{g}_i^R)\|_2^2\right]\Big)^{\frac{1}{2}}\end{aligned} \tag{9.13}$$

Using the same argument as Claim 7.4.1, it is easy to observe that

$$\sum_{j\in[q]} H_j(\boldsymbol{g}_s^R) = 1\,.$$

Now apply Fact 7.4.2 to conclude,

$$\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\|\boldsymbol{H}^*(\mathbf{g}_i^R)-\boldsymbol{H}(\mathbf{g}_i^R)\|_2^2\right]\leqslant(2q+2)\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\xi(\boldsymbol{H}(\mathbf{g}_i^R))\right].$$

Since $\boldsymbol{H}(\ell_i^R)\in\blacktriangle_q$ we have $\mathbb{E}_{\mathcal{G}_P^R}\left[\xi(\boldsymbol{H}(\ell_i^R))\right]=0$. Rewriting the above inequality,

$$\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\|\boldsymbol{H}^*(\mathbf{g}_i^R)-\boldsymbol{H}(\mathbf{g}_i^R)\|_2^2\right]\leqslant(2q+2)\left|\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\xi(\boldsymbol{H}(\mathbf{g}_i^R))\right]-\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\left[\xi(\boldsymbol{H}(\ell_i^R))\right]\right|$$

Using Assertion 2 of the invariance principle (Theorem 3.2) with the ensembles $\mathcal{L}_P$, $\mathcal{G}_P$ and the vector of qk multilinear polynomials given by $(\boldsymbol{H},\boldsymbol{H},\ldots,\boldsymbol{H})$ where $\boldsymbol{H}=(H_1,\ldots,H_q)$.

$$\left|\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\left[\xi(\boldsymbol{H}(\ell_i^R))\right]-\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\xi(\boldsymbol{H}(\mathbf{g}_i^R))\right]\right|\leqslant o_\tau(1)$$

Consequently,

$$\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\|\boldsymbol{H}^*(\mathbf{g}_i^R)-\boldsymbol{H}(\mathbf{g}_i^R)\|_2^2\right]\leqslant(2q+2)\left|\mathop{\mathbb{E}}_{\mathcal{G}_P^R}\left[\xi(\boldsymbol{H}(\mathbf{g}_i^R))\right]-\mathop{\mathbb{E}}_{\mathcal{L}_P^R}\left[\xi(\boldsymbol{H}(\ell_i^R))\right]\right|\leqslant o_\tau(1)$$

Substituting in 9.13 we get the required result. ■

Chapter 10

GROTHENDIECK INEQUALITY

10.1 Introduction

The Grothendieck inequality states that for every $n \times m$ matrix $A = (a_{ij})$ and every choice of unit vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$, there exists a choice of signs $x_1, \ldots, x_n, y_1, \ldots, y_m \in \{\pm 1\}$ such that

$$\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \leqslant K_G \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_i y_j \,,$$

where K_G is a universal constant. The smallest value of K_G for which the inequality holds, is referred to as the Grothendieck constant. Since the inequality was first discovered [69], the inequality has not only undergone various restatements under different frameworks of analysis (see [115]), it has also found numerous applications in functional analysis.

In recent years, the Grothendieck's inequality has found algorithmic applications in efficient construction of Szemeredi partitions of graphs and estimation of cut norms of matrices [4], in turn leading to efficient approximation algorithms for dense graph problems [63]. The inequality has also proved useful in certain lower bound techniques for communication complexity [116]. Among its various applications, here we shall elaborate on the $K_{N,N}$-QuadraticProgramming problem. In this problem, the objective is to maximize the following quadratic program given as input the matrix $A = (a_{ij})$.

$$\text{Maximize} \qquad \sum_{i,j} a_{ij} x_i y_j \qquad \text{Subject to: } x_i, y_j \in \{\pm 1\}$$

Alternatively, the problem amounts to computing the norm $\|A\|_{\infty \to 1}$ of the matrix A. The $K_{N,N}$-QuadraticProgramming problem is a special case of the correlation clustering problem with two clusters, where the underlying graph is bipartite. The following natural SDP relaxation to the problem is obtained by relaxing the variables x_i, y_j to unit vectors.

$$\text{Maximize} \qquad \sum_{i,j} a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \qquad \text{Subject to: } \|\boldsymbol{u}_i\| = \|\boldsymbol{v}_j\| = 1$$

The Grothendieck constant K_G is precisely the integrality gap of this SDP relaxation for the $K_{N,N}$-QuadraticProgramming problem.

Despite several proofs and reformulations, the value of the Grothendieck constant K_G still remains unknown. In his original work, Grothendieck showed that $\frac{\pi}{2} \leqslant K_G \leqslant 2.3$. The upper bound has been later improved to $\pi/2 \log(1+\sqrt{2}) \approx 1.78$ by Krivine [109], while the best known lower bound is roughly 1.67 [142]. More importantly, very little seems to be known about the matrices A for which the inequality is tight.

10.1.1 Results

In this chapter, we will apply the connections between SDP integrality gaps, dictatorship tests and UG-hardness results to the $K_{N,N}$-QuadraticProgramming problem. First, we obtain the following UGC-based hardness result for $K_{N,N}$-QuadraticProgramming.

Theorem 10.1. *It is* Unique Games*-hard to approximate* $K_{N,N}$-QuadraticProgramming *by any constant factor smaller than the Grothendieck constant* K_G*.*

Note that $K_{N,N}$-QuadraticProgramming is a Generalized Constraint Satisfaction Problem. However, the above result does not immediately follow from Theorem 7.1, since the reduction does not preserve bipartiteness. The main technical hurdle in obtaining a bipartiteness-preserving reduction, is to give a stronger analysis of the dictatorship test so as to guarantee a common influential variable. This is achieved using a standard truncation argument as outlined in [124].

On the other hand, neither the generic rounding scheme outlined in Chapter 5 nor the rounding scheme in Chapter 7 directly translate in to an algorithm for $K_{N,N}$-QuadraticProgramming. The main issue is the constant additive error term incurred by both algorithms. For a CSP, the objective function is guaranteed to be at least a fixed constant fraction (say 0.5), and hence the additive constant error is negligible. In case of $K_{N,N}$-QuadraticProgramming, the value of the optimum solution could be $1/\log n$, in which case an additive constant error destroys the approximation ratio.

To obtain better bound on the error, we use a bootstrapping argument similar to the Gaussian Hilbert space approach to Grothendieck inequality [29] (this approach is used for algorithmic purposes in [4, 3, 105]). Using ideas from the proof of Grothendieck inequality, we perform a tighter analysis of the reduction from integrality gaps to dictatorship tests (outlined in Chapter 6) for the special case of $K_{N,N}$-QuadraticProgramming. This tighter analysis yields the following new results:

Theorem 10.2. *For every* $\varepsilon > 0$*, there is an efficient algorithm that achieves an approximation ratio* $K_G - \varepsilon$ *for* $K_{N,N}$-QuadraticProgramming *running in time* $F(\varepsilon) \cdot poly(n)$ *where* $F(\varepsilon) = exp(exp(O(1/\varepsilon^3)))$*.*

Theorem 10.3. *For every* $\varepsilon > 0$*, the Grothendieck constant* K_G *can be computed within an error* ε *in time proportional to* $exp(exp(O(1/\varepsilon^3)))$*.*

A tighter running time analysis could improve the $O(1/\varepsilon^3)$, but reducing the number of exponentiations seems to require new ideas.

10.1.2 Prior Work

The general Grothendieck problem on a graph G amounts to maximizing a quadratic polynomial $\sum_{ij} a_{ij} x_i x_j$ over $\{\pm 1\}$ values, where a_{ij} is non zero only for edges (i, j) in G. $K_{N,N}$ Quadratic programming is the special case where the graph G is the complete bipartite graph.

The Grothendieck problem on a complete graph admits a $O(\log n)$ approximation [126, 123, 36] and has applications in correlation clustering [36]. For the Grothendieck problem on general graphs, [3] obtain an approximation that depends on the Lovasz θ number of the graph.

In an alternate direction, the Grothendieck problem has been generalized to the L_p-Grothendieck problem where the L_p norm of the assignment is bounded by 1. The traditional Grothendieck corresponds to the case when $p = \infty$. In a recent work, [105] obtain UGC hardness results and approximation algorithms for the L_p-Grothendieck problem.

On the hardness side, [9] show a $O(\log^c n)$-NP hardness for the Grothendieck problem on the complete graph for some fixed constant $c < 1$. Integrality gaps for the Grothendieck problem on complete graphs were exhibited in [101, 3]. For the $K_{N,N}$-Quadratic Programming problem, a UGC-based hardness of roughly 1.67 was shown in [101]. The reduction uses the explicit operator constructed in the proof of 1.67 lower bound [142] for the Grothendieck constant.

Organization The next section is devoted to formal definitions of the $K_{N,N}$-QuadraticProgramming problem, and certain analytic notions like common influences. This is followed by three sections that outline the three reductions between dictatorship tests, UG-hardness results and SDP integrality gaps. Among these, the reduction in Section 10.3 is the central contribution of the chapter, while the other two reductions follow easily from existing work. Finally, in Section 10.6 we use the reductions to obtain the optimal algorithm for $K_{N,N}$-QuadraticProgramming, and the algorithm to compute the Grothendieck constant.

Mathematical Tools This chapter relies on the harmonic analysis of boolean functions (Section 3.3), associated notions of influence and noise stability, Gaussian random variables (Section 3.5) and the associated noise operator. The chapter also makes use of a simple version of the invariance principle (Section 3.6), which is stated here for the sake of completeness.

10.2 Preliminaries

Problem 10 ($K_{N,N}$-QuadraticProgramming)**.** Given an $m \times n$ matrix $A = (a_{ij})$, compute the optimal value of the following optimization problem,

$$\mathrm{opt}(A) \stackrel{\mathrm{def}}{=} \max \sum_{ij} a_{ij} x_i y_j \,,$$

where the maximum is over all $x_1, \ldots, x_m \in [-1, 1]$ and $y_1, \ldots, y_n \in [-1, 1]$. Note that the optimum value $\mathrm{opt}(A)$ is always attained for numbers with $|x_i| = |y_j| = 1$.

The following is the natural semidefinite relaxation of $K_{N,N}$-QuadraticProgramming.

Problem 11 ($K_{N,N}$-SemidefiniteProgramming)**.** Given an $m \times n$ matrix $A = (a_{ij})$, compute the optimal value of the following optimization problem,

$$\mathrm{sdp}(A) \stackrel{\mathrm{def}}{=} \max \sum_{ij} a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \,,$$

where the maximum is over all vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m \in B^{(d)}$ and all vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in B^{(d)}$. Here $B^{(d)}$ denotes the unit ball in $\mathbb{R}^d$ and we choose $d \geqslant m + n$. Note that the optimum value $\mathrm{sdp}(A)$ is always attained for vectors with $\|\boldsymbol{u}_i\| = \|\boldsymbol{v}_j\| = 1$.

Remark 10.2.1. Since $K_{N,N}$-QuadraticProgramming is a GCSP, the LC relaxation would be the canonical relaxation to consider. However, the LC relaxation is equivalent to the above semidefinite program. The proof of equivalence follows along the lines of

Lemma 4.1.3 that proves a similar equivalence for the case of MAX CUT with the Goemans-Williamson SDP.

Definition 10.2.1. The Grothendieck constant K_G is the supremum of $\mathrm{sdp}(A)/\mathrm{opt}(A)$ over all matrices A.

For $\mathcal{F}, \mathcal{G} \in L_2(\Omega)$, we denote $\langle \mathcal{F}, \mathcal{G} \rangle \stackrel{\text{def}}{=} \mathbb{E}\, \mathcal{F}\mathcal{G}$, $\|\mathcal{F}\| \stackrel{\text{def}}{=} \sqrt{\mathbb{E}\, \mathcal{F}^2}$, and $\|\mathcal{F}\|_\infty \stackrel{\text{def}}{=} \sup_{\boldsymbol{x} \in \Omega} \mathcal{F}(\boldsymbol{x})$. We have $\|\mathcal{F}\| \leqslant \|\mathcal{F}\|_\infty$.

Lemma 10.3.1. *Given an operator A on $L_2(\Omega^R)$, and functions $\mathcal{F}, \mathcal{G}, \mathcal{F}', \mathcal{G}' \in L_2^{(d)}(\Omega^R)$ satisfying $\|\mathcal{F}\|, \|\mathcal{G}\|, \|\mathcal{F}'\|, \|\mathcal{G}'\| \leqslant 1$, then*

$$|\langle \mathcal{F}, A\mathcal{G} \rangle - \langle \mathcal{F}', A\mathcal{G}' \rangle| \leqslant \|A\| (\|\mathcal{F} - \mathcal{F}'\| + \|\mathcal{G} - \mathcal{G}'\|)\,.$$

Lemma 10.3.2 (Bootstrapping Lemma)**.** *Given an $m \times n$ matrix $A = (a_{ij})$, and vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$, then*

$$\sum_{ij} a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle \leqslant \big(\max_i \|\boldsymbol{u}_i\|\big)\big(\max_j \|\boldsymbol{v}_j\|\big) \cdot \mathrm{sdp}(A) \leqslant 2\big(\max_i \|\boldsymbol{u}_i\|\big)\big(\max_j \|\boldsymbol{v}_j\|\big) \cdot \mathrm{opt}(a)$$

Proof. On scaling the vectors $\boldsymbol{u}_i$, $\boldsymbol{v}_i$ by $(\max_i \|\boldsymbol{u}_i\|)$ and $(\max_i \|\boldsymbol{v}_i\|)$ respectively, the resulting vectors have lengths bounded by 1. Therefore, the scaled vectors form a feasible SDP solution. Therefore we get,

$$\sum_{ij} a_{ij} \Big\langle \frac{\boldsymbol{u}_i}{\max_i \|\boldsymbol{u}_i\|}, \frac{\boldsymbol{v}_j}{\max_i \|\boldsymbol{v}_i\|} \Big\rangle \leqslant \mathrm{sdp}(A)\,.$$

■

Common Influences. For a function $\mathcal{F} \in L_2(\{\pm 1\}^R)$, we define $\mathrm{Inf}_i \mathcal{F} = \sum_{\sigma \ni i} \hat{\mathcal{F}}_\sigma^2$, where $\hat{\mathcal{F}}$ is the Fourier-transform of $\mathcal{F}$. Let us denote $\mathrm{MaxInf}\, \mathcal{F} \stackrel{\text{def}}{=} \max_{i \in [R]} \mathrm{Inf}_i \mathcal{F}$. For a pair of functions $\mathcal{F}, \mathcal{G} \in L_2(\{\pm 1\}^R)$, we define $\mathrm{MaxComInf}(\mathcal{F}, \mathcal{G}) \stackrel{\text{def}}{=} \max_{i \in [R]} \min\{\mathrm{Inf}_i \mathcal{F}, \mathrm{Inf}_i \mathcal{G}\}$ to be the *maximum common influence*.

Lemma 10.3.3. *Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ be two unit vectors, and $\mathcal{F}, \mathcal{G} \in L_2(\{\pm 1\}^R)$. Then,*

$$\mathop{\mathbb{E}}_{\Phi} F(\Phi \boldsymbol{u}) G(\Phi \boldsymbol{v}) = \langle \mathcal{F}, T_{\langle \boldsymbol{u}, \boldsymbol{v} \rangle} \mathcal{G} \rangle$$

where Φ is a $R \times d$ Gaussian matrix, that is, the entries of Φ are mutually independent normal variables with standard deviation $\frac{1}{\sqrt{d}}$.

Proof. It suffices to show the lemma for the case that both $\mathcal{F}$ and $\mathcal{G}$ are the same multilinear monomial. Since the variables are independent, one may assume that the monomial has degree 1. For this case, it is trivial. ■

Here we state a version of the invariance principle suited for the application at hand.

Truncation of Low-influence Functions. For $\mathcal{F}\colon \mathbb{R}^R \to \mathbb{R}$, let $\operatorname{trunc}\mathcal{F}\colon \mathbb{R}^R \to [-1,1]$ denote the function

$$\operatorname{trunc}\mathcal{F}(\boldsymbol{x}) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \mathcal{F}(\boldsymbol{x}) > 1\,, \\ \mathcal{F}(\boldsymbol{x}) & \text{if } -1 < \mathcal{F}(\boldsymbol{x}) < 1\,, \\ -1 & \text{if } \mathcal{F}(\boldsymbol{x}) < -1\,. \end{cases}$$

Theorem 10.4 (Invariance Principle, [125])**.** *There is a universal constant C such that, for all $\rho = 1-\varepsilon \in (0,1)$ the following holds: Let $\mathcal{F} \in L_2(\{\pm 1\}^R)$ with $\|\mathcal{F}\|_\infty \leqslant 1$ and $\operatorname{Inf}_i(T_\rho \mathcal{F}) \leqslant \tau$ for all $i \in [R]$. Then,*

$$\|T_\rho F - \operatorname{trunc} T_\rho F\| \leqslant \tau^{C\cdot\varepsilon}$$

where $F \in L_2(\boldsymbol{\mathcal{G}}^R)$ denotes the (unique) multilinear extension of $\mathcal{F}$ to $\mathbb{R}^R$.

10.2.1 Dictatorship Tests

In the current context, a dictatorship test can be defined succinctly as follows.

Definition 10.2.2. A dictatorship test B is an operator on $L_2(\{\pm 1\}^R)$ of the following form:

$$B = \sum_{d=0}^{R} \lambda_d P_d$$

where P_d is the projection operator on to the degree d part, and $|\lambda_1| \geqslant |\lambda_d|$ for all d. We define two parameter of B:

$$\mathsf{Completeness}(B) \stackrel{\text{def}}{=} \inf_i \langle \chi_i, B\chi_i \rangle\,, \text{ where } \chi_i(\boldsymbol{x}) = x_i \text{ is the } i^{th} \text{ dictator function.}$$

$$\mathsf{Soundness}_{\varepsilon,\tau}(B) \stackrel{\text{def}}{=} \sup_{\substack{\mathcal{F},\mathcal{G}:\{\pm1\}^R\to[-1,1],\\ \operatorname{MaxComInf}(T_\rho\mathcal{F},T_\rho\mathcal{G})\leqslant\tau}} \langle \mathcal{F}, B\mathcal{G} \rangle\,, \text{ where } \rho = 1-\varepsilon.$$

10.3 From Integrality Gaps to Dictatorship Tests

In the first step, we describe a reduction from a matrix A of arbitrary size, to a dictatorship test $\mathsf{DICT}^\varepsilon_A$ on $L_2(\{\pm 1\}^R)$ for a constant R independent of the size of A.

Towards this, let us set up some notation. Let $A = (a_{ij})$ be an $m \times n$ matrix with SDP value $\operatorname{sdp}(A)$. Let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m \in B^{(d)}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in B^{(d)}$ be an SDP solution such that

$$\sum_{ij} a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle = \operatorname{sdp}(A)\,.$$

In general, an optimal SDP solution $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ might not be unique. In the following, we will however assume that for every instance A we can uniquely associate an optimal SDP solution, e.g., the one computed by a given implementation of the ellipsoid method. With this notation, we are ready to define the dictatorship test $\mathcal{D}_A$.

Definition 10.3.1. For $d \in \mathbb{N}$, let us define coefficients $\lambda_d \in \mathbb{R}$,

$$\lambda_d \stackrel{\text{def}}{=} \sum_{ij} a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle^d .$$

Define linear operator $\mathcal{D}_A$ $\mathsf{DICT}^{\varepsilon}_A$ on $L_2(\{\pm 1\}^R)$ as follows:

$$\mathcal{D}_A \stackrel{\text{def}}{=} \sum_{d=0}^{R} \lambda_d P_d \,, \qquad\qquad \mathsf{DICT}^{\varepsilon}_A \stackrel{\text{def}}{=} T_\rho \mathcal{D}_A T_\rho = \sum_{d=0}^{R} \rho^{2d} \lambda_d P_d \,,$$

where $\rho = 1 - \varepsilon$.

By the definition of $\mathsf{Completeness}(\mathsf{DICT}^{\varepsilon}_A)$, we have:

Lemma 10.4.1. *For all matrices A,* $\mathsf{Completeness}(\mathit{DICT}^{\varepsilon}_A) = \lambda_1 \rho^2 \geqslant \mathrm{sdp}(A)(1 - 2\varepsilon)$.

10.3.1 A Rounding Scheme

Towards bounding $\mathsf{Soundness}_{\varepsilon,\tau}(\mathsf{DICT}^{\varepsilon}_A)$, we define a rounding scheme $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}$ for every pair of functions $\mathcal{F}, \mathcal{G} \in L_2(\{\pm 1\}^R)$ and $\rho < 1$. The rounding scheme $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}$ is an efficient randomized procedure that takes as input the optimal SDP solution for A, and outputs a solution $x_1, \ldots, x_m, y_1, \ldots, y_n \in \{\pm 1\}$.

For functions $\mathcal{F}, \mathcal{G} \in L_2(\{\pm 1\}^R)$, define the rounding procedure $\mathsf{Round}_{\mathcal{F},\mathcal{G}}$ as follows:

$\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}$
Input : An $m \times n$ matrix $A = (a_{ij})$ with SDP solution $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m\}, \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\} \subset B^{(d)}$

- Let $T_\rho F, T_\rho G$ denote the multilinear polynomials obtained by Fourier expansion of functions $T_\rho \mathcal{F}, T_\rho \mathcal{G}$. Compute $T_\rho F$ and $T_\rho G$ explicitly.
- Generate $R \times d$ matrix Φ all of whose entries are mutually independent normal variables of standard deviation $1/\sqrt{d}$.
- Assign $x_i = \mathrm{trunc}\, T_\rho F(\Phi \boldsymbol{u}_i)$ and $y_j = \mathrm{trunc}\, T_\rho G(\Phi \boldsymbol{v}_j)$ for all $i \in [m], j \in [n]$.

The expected value of the solution returned $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A)$ is given by:

$$\mathsf{Round}_{\mathcal{F},\mathcal{G}}(A) = \mathop{\mathbb{E}}_{\Phi} \textstyle\sum_{ij} a_{ij} \mathrm{trunc}\, T_\rho F(\Phi \boldsymbol{u}_i) \mathrm{trunc}\, T_\rho G(\Phi \boldsymbol{v}_j)$$

Definition 10.3.2. $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A)$ is the expected value of the solution returned by the randomized rounding procedure $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}$ on the input A.

In this section, we will show following relationship between performance of rounding schemes and soundness of the dictatorship test [125].

Theorem 10.5. *Let A be a matrix. For functions $\mathcal{F}, \mathcal{G} \in L_2(\{\pm 1\}^R)$ satisfying $\|\mathcal{F}\|_\infty, \|\mathcal{G}\|_\infty \leqslant 1$ and $\mathrm{MaxComInf}(T_\rho \mathcal{F}, T_\rho \mathcal{G}) \leqslant \tau$ for $\rho = 1 - \varepsilon$, there exists functions $\mathcal{F}', \mathcal{G}' \in L_2(\{\pm 1\}^R)$ such that*

$$\langle \mathcal{F}, \mathit{DICT}^\varepsilon_A \mathcal{G} \rangle \leqslant \mathsf{Round}_{\varepsilon,\mathcal{F}',\mathcal{G}'}(A) + \left(10\tau^{C\varepsilon/8}/\sqrt{\varepsilon}\right) \cdot \mathrm{sdp}(A)\,.$$

Further the functions $\mathcal{F}', \mathcal{G}'$ satisfy $\min \mathrm{Inf}_i T_\rho \mathcal{F}', \mathrm{Inf}_i T_\rho \mathcal{G}' \leqslant \tau$ for all i.

By taking the supremum on both sides of the above inequality over all low influence functions, one obtains the following corollary:

Corollary 10.5.1. *For every matrix A and $\varepsilon > 0$,*

$$\mathsf{Soundness}_{\varepsilon,\tau}(\mathit{DICT}^\varepsilon_A) \leqslant \left(\sup_{\substack{\mathcal{F},\mathcal{G} \in L_2(\{\pm 1\}^R), \\ \mathrm{MaxComInf}(T_\rho \mathcal{F}, T_\rho \mathcal{G}) \leqslant \tau}} \mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A) \right) + \left(10\tau^{C\varepsilon/8}/\sqrt{\varepsilon}\right) \cdot \mathrm{sdp}(A)\,,$$

where $\rho = 1 - \varepsilon$.

As $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}$ is the expected value of a $\{\pm 1\}$ solution, it is necessarily smaller than $\mathrm{opt}(A)$. Further by Grothendieck's inequality, $\mathrm{sdp}(A)$ and $\mathrm{opt}(A)$ are within constant factor of each other. Together, these facts immediately imply the following corollary:

Corollary 10.5.2. *For a fixed $\varepsilon > 0$, if $\tau \leqslant 2^{-100 \log \varepsilon / C\varepsilon}$ then, then for all matrices A,*

$$\mathsf{Soundness}_{\varepsilon,\tau}(\mathit{DICT}^\varepsilon_A) \leqslant \mathrm{opt}(A)(1 + \varepsilon)$$

10.3.2 Relaxed Influence Condition

The following lemma shows that we could replace the condition $\mathrm{MaxComInf}(T_\rho \mathcal{F}, T_\rho \mathcal{G}) \leqslant \tau$ in Definition 10.2.2 by the condition $\mathrm{MaxInf}\, T_\rho \mathcal{F}, \mathrm{MaxInf}\, T_\rho \mathcal{G} \leqslant \sqrt{\tau}$ with a small loss in the soundness.

Lemma 10.5.1. *Let A be a dictatorship test on $L_2(\{\pm 1\}^k)$, and let $\mathcal{F}, \mathcal{G}$ be a pair of functions in $L_2(\{\pm 1\}^R)$ with $\|\mathcal{F}\|_\infty, \|\mathcal{G}\|_\infty \leqslant 1$ and $\mathrm{MaxComInf}(T_\rho \mathcal{F}, T_\rho \mathcal{G}) \leqslant \tau$ for $\rho = 1 - \varepsilon$. Then for every $\tau' > 0$, there are functions $\mathcal{F}', \mathcal{G}' \in L_2(\{\pm 1\}^R)$ with $\|\mathcal{F}'\|_\infty, \|\mathcal{G}'\|_\infty \leqslant 1$ and $\mathrm{MaxInf}\, T_\rho \mathcal{F}', \mathrm{MaxInf}\, T_\rho \mathcal{G}' \leqslant \tau'$ such that*

$$\langle T_\rho \mathcal{F}', A T_\rho \mathcal{G}' \rangle \geqslant \langle T_\rho \mathcal{F}, A T_\rho \mathcal{G} \rangle - 2\|A\| \sqrt{\tau/\tau' \varepsilon}\,.$$

Proof. Let J denote the set of variables i with $\mathrm{Inf}_i T_\rho \mathcal{F} > \tau'$. Since the total influence of $T_\rho \mathcal{F}$ is bounded by $1/\varepsilon$ (see Lemma 3.0.2), the set J has cardinality at most $1/\varepsilon\tau'$. Let M_J be the orthogonal projector on the space of functions that do not depend on any variable in J. We define $\mathcal{F}' = M_J \mathcal{F}$ and $\mathcal{G}' = M_J \mathcal{G}$. We still have $\|\mathcal{F}'\|_\infty, \|\mathcal{G}'\| \leqslant 1$. Note that $\mathrm{Inf}_i T_\rho \mathcal{G} \leqslant \tau$ for every $i \in J$. Hence, $\|T_\rho \mathcal{G} - T_\rho \mathcal{G}'\|^2 \leqslant |J|\tau$. Now,

$$\begin{aligned} \langle T_\rho \mathcal{F}, A\, T_\rho \mathcal{G} \rangle - \langle T_\rho \mathcal{F}, A T_\rho \mathcal{G}' \rangle &= \langle T_\rho \mathcal{F}, A\, T_\rho (\mathcal{G} - \mathcal{G}') \rangle \\ &\leqslant \|A\| \|T_\rho \mathcal{F}\| \cdot \|T_\rho (\mathcal{G} - \mathcal{G}')\| \\ &\leqslant \sqrt{|J|\tau}\,. \end{aligned}$$

On the other hand, $\langle T_\rho \mathcal{F}, AT_\rho \mathcal{G}' \rangle = \langle T_\rho \mathcal{F}', AT_\rho \mathcal{G}' \rangle$, because

$$\langle T_\rho(\mathcal{F}-\mathcal{F}'), AT_\rho \mathcal{G}' \rangle = \langle T_\rho \mathcal{F}, (I-M_J)AM_J T_\rho \mathcal{G} \rangle = \langle T_\rho \mathcal{F}, A(I-M_J)M_J T_\rho \mathcal{G} \rangle = 0\,,$$

where we used the fact that the operators A and M_J commute (as both are diagonalized by the Fourier transform), and that $(I-M_J)M_J = 0$.

We repeat the same argument with the set K of variables i with $\mathrm{Inf}_i T_\rho \mathcal{G} > \tau'$. Again, projecting on M_K changes the value of $\langle T_\rho \mathcal{F}', A\, T_\rho \mathcal{G}' \rangle$ by at most $\sqrt{|J|\tau}$. ■

With this background, we now present the soundness analysis.

10.3.3 Proof of Theorem 10.5

Proof. By Lemma 10.5.1, there exists function $\mathcal{F}', \mathcal{G}' \in L_2(\{\pm 1\}^R)$ with $\|\mathcal{F}'\|_\infty, \|\mathcal{G}'\|_\infty \leqslant 1$ and $\mathrm{MaxInf}\, T_\rho \mathcal{F}', \mathrm{MaxInf}\, T_\rho \mathcal{G}' \leqslant \sqrt{\tau}$ such that

$$\langle \mathcal{F}', \mathsf{DICT}^\varepsilon_A \mathcal{G}' \rangle \geqslant \langle \mathcal{F}, \mathsf{DICT}^\varepsilon_A \mathcal{G} \rangle - 2\|\mathcal{D}_A\| \cdot \tau^{1/4}/\sqrt{\varepsilon} \geqslant \langle \mathcal{F}, \mathsf{DICT}^\varepsilon_A \mathcal{G} \rangle - 4\mathrm{opt}(A) \cdot \tau^{1/4}/\sqrt{\varepsilon}\,.$$

On the other hand, we have

$$\langle \mathcal{F}', \mathsf{DICT}^\varepsilon_A \mathcal{G}' \rangle = \sum_{ij} \sum_{d=0}^{R} \langle T_\rho \mathcal{F}', a_{ij} \langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle^d P_d(T_\rho \mathcal{G}') \rangle = \sum_{ij} a_{ij} \langle T_\rho \mathcal{F}', T_{\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle}(T_\rho \mathcal{G}') \rangle \quad (10.1)$$

We can assume that all vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ have unit norm. By Lemma 10.3.3, we have

$$\mathop{\mathbb{E}}_{\Phi} \sum_{ij} a_{ij} T_\rho F'(\Phi \boldsymbol{u}_i) T_\rho G'(\Phi \boldsymbol{v}_j) = \sum_{ij} a_{ij} \langle T_\rho \mathcal{F}', T_{\langle \boldsymbol{u}_i, \boldsymbol{v}_j \rangle}(T_\rho \mathcal{G}') \rangle \quad (10.2)$$

From the above equations we have

$$\langle \mathcal{F}, \mathsf{DICT}^\varepsilon_A \mathcal{G} \rangle = \mathop{\mathbb{E}}_{\Phi} \sum_{ij} a_{ij} T_\rho F'(\Phi \boldsymbol{u}_i) T_\rho G'(\Phi \boldsymbol{v}_j) \quad (10.3)$$

By the invariance principle (Theorem 10.4), we have

$$\|T_\rho F' - \mathrm{trunc}\, T_\rho F'\| \leqslant \tau^{C\varepsilon/2} \qquad \text{and } \|T_\rho G' - \mathrm{trunc}\, T_\rho G'\| \leqslant \tau^{C\varepsilon/2}\,. \quad (10.4)$$

Now we shall apply the simple yet powerful bootstrapping trick. Let us define new vectors in $L_2(\boldsymbol{\mathcal{G}}^{R\times d})$,

$$\boldsymbol{u}'_i = T_\rho F'(\Phi \boldsymbol{u}_i) \qquad \boldsymbol{v}'_j = T_\rho G'(\Phi \boldsymbol{v}_j)$$

and

$$\boldsymbol{u}''_i = \mathrm{trunc}\, T_\rho F'(\Phi \boldsymbol{u}_i) \qquad \boldsymbol{v}''_j = \mathrm{trunc}\, T_\rho G'(\Phi \boldsymbol{v}_j)$$

Equation (10.4) implies that $\|\boldsymbol{u}'_i - \boldsymbol{u}''_i\| \leqslant \tau^{C\varepsilon/2}$ and $\|\boldsymbol{v}'_j - \boldsymbol{v}''_j\| \leqslant \tau^{C\varepsilon/2}$. Using the bootstrapping argument (Lemma 10.3.2), we finish the proof

$$\begin{aligned}\mathsf{Round}_{\varepsilon,\mathcal{F}',\mathcal{G}'}(A) = \sum_{ij} a_{ij}\langle \boldsymbol{u}''_i, \boldsymbol{v}''_j\rangle &= \sum_{ij} a_{ij}\langle \boldsymbol{u}'_i, \boldsymbol{v}'_j\rangle - \sum_{ij} a_{ij}\langle \boldsymbol{u}'_i - \boldsymbol{u}''_i, \boldsymbol{v}'_j\rangle - \sum_{ij} a_{ij}\langle \boldsymbol{u}''_i, \boldsymbol{v}'_j - \boldsymbol{v}''_j\rangle \\ &\overset{(10.4)}{\geqslant} \sum_{ij} a_{ij}\langle \boldsymbol{u}'_i, \boldsymbol{v}'_j\rangle - 2\tau^{C\varepsilon}\mathrm{opt}(A) - 2\tau^{C\varepsilon}\mathrm{opt}(A) \\ &\geqslant \langle \mathcal{F}, \mathsf{DICT}^{\varepsilon}_{A}\mathcal{G}\rangle - 4\tau^{C\varepsilon/2}\mathrm{opt}(A) - 4\tau^{1/4}\mathrm{opt}(A)/\sqrt{\varepsilon}\,. \quad (10.5)\end{aligned}$$

■

10.4 From Dictatorship Tests to UG-hardness

Although, $K_{N,N}$-QuadraticProgramming is a GCSP, the generic UG-hardness reduction in Chapter 7 does not apply directly, since the resulting instance is required to be bipartite. Towards achieving bipartiteness, the soundness analysis of the dictatorship test (Theorem 10.5) has been strengthened in that it yields common influential coordinates between the functions involved. Armed with this stronger soundness condition, the Unique Games-hardness reduction for GCSP (Theorem 7.1) can be modified easily to yield bipartite instances. For the sake of completeness, we include a sketch of the UG-hardness reduction below.

Lemma 10.5.2. *Given a dictatorship test A with completeness c and soundness s, and a* Unique Games *instance Φ, it is possible to efficiently construct an operator $\Phi \otimes_\rho A$ satisfies the following to two conditions:*

1. *if* $\mathrm{val}(\Phi) \geqslant 1-\gamma$, *then* $\mathrm{opt}(\Phi \otimes_\rho A) \geqslant c(1 - o_{\varepsilon,\gamma\to 0}(1))$,
2. *if* $\mathrm{val}(\Phi) < \delta$, *then* $\mathrm{opt}(\Phi \otimes_\rho A) < s(1 + o_{\varepsilon,\delta\to 0}(1))$,

The **UGC** asserts that for every $\varepsilon > 0$, there is a R such that for a unique game Φ on alphabet $[R]$ it is hard to distinguish between $\mathrm{val}(\Phi) \geqslant 1-\varepsilon$ and $\mathrm{val}(\Phi) < \varepsilon$.

Formally, we will represent a unique game Φ on alphabet $[R]$ as a distribution over triples (v, w, π), where $v \in \mathcal{V}_\Phi$ and $w \in \mathcal{W}_\Phi$ are vertices, and π is a permutation of $[R]$. Here we can and will assume that the game is bipartite, i.e., $\mathcal{V}_\Phi$ and $\mathcal{W}_\Phi$ are disjoint.

Let $A = \sum_{d\in[R]} \lambda_d P_d$ be a dictatorship test on $L_2(\{\pm 1\}^R)$. For $\rho = 1-\varepsilon$, we define a linear operator $\Phi \otimes_\varepsilon A$ on $\left(L_2(\{\pm 1\}^R)\right)^{|\mathcal{V}_\Phi|+|\mathcal{W}_\Phi|}$ as follows:

$$\langle \boldsymbol{\mathcal{F}}, (\Phi \otimes_\varepsilon A)\boldsymbol{\mathcal{G}}\rangle \overset{\mathrm{def}}{=} \mathop{\mathbb{E}}_{(v,w,\pi)\sim\Phi} \langle T_\rho(\pi.\mathcal{F}_v), AT_\rho\mathcal{G}_w\rangle\,,$$

where $\boldsymbol{\mathcal{F}} = (\mathcal{F}_v)_{v\in\mathcal{V}_\Phi}$, $\boldsymbol{\mathcal{G}} = (\mathcal{G}_w)_{w\in\mathcal{W}_\Phi}$, and $\pi.\mathcal{F}_v$ denotes the function $\mathcal{F}_v(x_{\pi(1)}, \ldots, x_{\pi(R)})$.

We claim the following properties of the reduction $\Phi \mapsto \Phi \otimes_\varepsilon A$. This claim implies Lemma 10.5.2.

Claim 10.5.1. *For $\tau, \varepsilon \in [0,1]$, $\rho = 1-\varepsilon$, and every unique game Φ, we have*

1. *If* $\mathrm{val}(\Phi) \geqslant 1-\varepsilon$ *then* $\mathrm{opt}(A \otimes_\varepsilon \Phi) \geqslant \mathsf{Completeness}(A)(1 - O(\varepsilon+\varepsilon))$.

2. *If* $\mathrm{val}(\Phi) < (\tau\varepsilon)^3$ *then* $\mathrm{opt}(A \otimes_\varepsilon \Phi) \leqslant \mathsf{Soundness}_{\varepsilon,\tau}(T_\rho A T_\rho) + O(\tau\varepsilon)\mathsf{Completeness}(A)$.

Proof. By scaling[1], we may assume $\lambda_1 = 1$ and $\lambda_d \in [-1,1]$ for all $d \in [R]$, where $A = \sum_d \lambda_d P_d$. Note that $\mathsf{Completeness}(A) = \lambda_1 = 1$.

Suppose that $\mathrm{val}(\Phi) \geqslant 1-\varepsilon$. Then there exists a labeling $\ell\colon \mathcal{V}_\Phi \cup \mathcal{W}_\Phi \to [R]$ such that

$$\Pr_{(v,w,\pi)\sim\Phi}\Big\{\pi(\ell(v)) = \ell(w)\Big\} \geqslant 1-\varepsilon\,.$$

We choose $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ such that $\mathcal{F}_v(\boldsymbol{x}) = x_{\ell(v)}$ and $\mathcal{G}_w(\boldsymbol{x}) = x_{\ell(w)}$ are dictator functions. If $\pi(\ell(v)) = \ell(w)$, then $\pi.\mathcal{F}_v = \mathcal{G}_w$. Hence, $\langle T_\rho \pi.\mathcal{F}_v, A T_\rho \mathcal{G}_w\rangle = \rho^2\lambda_1 = \rho^2$. On the other hand, if $\pi(\ell(v)) \neq \ell(w)$, then clearly $|\langle T_\rho \pi.\mathcal{F}_v, A T_\rho \mathcal{G}_w\rangle| \leqslant 1$. Thus,

$$\mathop{\mathbb{E}}_{(v,w,\pi)\sim\Phi}\langle T_\rho(\pi.\mathcal{F}_v), A T_\rho \mathcal{G}_w\rangle \geqslant (1-\varepsilon)\cdot\rho^2 - \varepsilon \geqslant 1 - 2\varepsilon - 2\varepsilon\,.$$

It follows that $\mathrm{opt}(\Phi \otimes_\varepsilon A) \geqslant c - o_{\varepsilon\to 0}(1)$ for any game Φ with $\mathrm{val}(\Phi) \geqslant 1-\varepsilon$.

Now suppose that $\mathrm{opt}(\Phi \otimes_\varepsilon A) \geqslant \mathsf{Soundness}_{\varepsilon,\tau}(A) + \delta$, where $\delta = \tau\varepsilon$. In this case, we want to show that $\mathrm{val}(\Phi) \geqslant \varepsilon$ for $\varepsilon = \tau\varepsilon^3$. Let $\boldsymbol{\mathcal{F}} = (\mathcal{F}_v)$ and $\boldsymbol{\mathcal{G}} = (\mathcal{G}_w)$ be vectors with $\|\mathcal{F}_v\|_\infty, \|\mathcal{G}_w\|_\infty \leqslant 1$ that achieve

$$\mathop{\mathbb{E}}_{(v,w,\pi)\sim\Phi}\langle T_\rho(\pi.\mathcal{F}_v), A T_\rho \mathcal{G}_w\rangle \geqslant \mathsf{Soundness}_{\varepsilon,\tau}(A) + \delta\,. \tag{10.6}$$

In hindsight, let us define a set of candidate labels for vertices $v \in \mathcal{V}_\Phi$ and $w \in \mathcal{W}_\Phi$,

$$\mathsf{L}(v) = \{i \mid \mathrm{Inf}_i T_\rho(\mathcal{F}_v) > \tau\} \quad \text{and} \quad \mathsf{L}(w) = \{i \mid \mathrm{Inf}_i T_\rho(\mathcal{G}_w) > \tau\}\,.$$

Since $\rho = 1-\varepsilon$, we have $|\mathsf{L}(v)|, |\mathsf{L}(w)| \leqslant 1/\varepsilon\tau$ (see Lemma 3.0.2). Since A is contracting, we get from equation (10.6) that

$$\Pr_{(v,w,\pi)\sim\Phi}\Big\{\langle T_\rho \pi.\mathcal{F}_v, A T_\rho \mathcal{G}_w\rangle > \mathsf{Soundness}_{\varepsilon,\tau}(A)\Big\} \geqslant \delta\,.$$

The situation $\langle T_\rho \pi.\mathcal{F}_v, A T_\rho \mathcal{G}_w\rangle > \mathsf{Soundness}_{\varepsilon,\tau}(A)$ implies that $\mathrm{Inf}_i T_\rho(\pi.\mathcal{F}_v) > \tau$ and $\mathrm{Inf}_i T_\rho \mathcal{G}_w > \tau$ for some $i \in [R]$. Of course, $\mathrm{Inf}_i T_\rho(\pi.\mathcal{F}_v) > \tau$ just means that variable $\pi^{-1}(i)$ has influence $\mathrm{Inf}_{\pi^{-1}(i)} T_\rho(\mathcal{F}_v) > \tau$. It follows that $i \in \mathsf{L}(w)$ and $\pi(i) \in \mathsf{L}(v)$. Thus,

$$\Pr_{(v,w,\pi)\sim\Phi}\Big\{\exists i \in [R].\ i \in \mathsf{L}(v) \text{ and } \pi(i) \in \mathsf{L}(w)\Big\} \geqslant \delta\,.$$

---

[1]Note that scaling A by a factor α, scales $\mathrm{opt}(\Phi \otimes_\varepsilon A)$, $\mathsf{Completeness}(A)$, and $\mathsf{Soundness}_{\varepsilon,\tau}(A)$ by the same factor α

Hence, if we choose a random element of $\mathsf{L}(v)$ as the label $\ell(v)$ and a random element of $\mathsf{L}(w)$ as the label $\ell(w)$, we have

$$\Pr_{(v,w,\pi)\sim\Phi}\left\{\exists i \in [R].\ \ell(v) = i \text{ and } \ell(w) = \pi(i)\right\} \geqslant \delta \cdot (\tau\varepsilon)^2\,,$$

where we use the fact that $|\mathsf{L}(v)||\mathsf{L}(w)| \leqslant 1/\tau^2\varepsilon^2$. We can conclude that $\mathrm{val}(\Phi) \geqslant \delta\tau^2\varepsilon^2$ for every unique game Φ with $\mathrm{opt}(\Phi \otimes_\varepsilon A) \geqslant \mathsf{Soundness}_{\varepsilon,\tau}(A) + \delta$. By our choice of τ and δ, we have $\delta\tau^2\varepsilon^2 = \varepsilon$. Hence, we get $\mathrm{val}(\Phi) \geqslant (\tau\varepsilon)^3$ for every Φ with $\mathrm{opt}(\Phi \otimes_\varepsilon A) \geqslant \mathsf{Soundness}_{\varepsilon,\tau}(A) + \delta = \mathsf{Soundness}_{\varepsilon,\tau}(A) + \tau\varepsilon$.

■

10.4.1 Proof of Theorem 10.1

Let A be a finite matrix for which the ratio of $\mathrm{sdp}(A)/\mathrm{opt}(A) \geqslant K_\Phi - \varepsilon$. Consider the dictatorship test $\mathsf{DICT}^\varepsilon_A$ obtained from the matrix A. By Lemma 10.4.1, the completeness of $\mathsf{DICT}^\varepsilon_A$ is $\mathrm{sdp}(A)(1-\varepsilon)$. Further by Corollary 10.5.2, the soundness is at most $\mathrm{opt}(A)(1+\varepsilon)$ for sufficiently small choice of τ. Plugging this dictatorship test $\mathsf{DICT}^\varepsilon_A$ in to the above lemma, we obtain a UG-hardness of $(K_\Phi - \varepsilon)(1-\varepsilon)/(1+\varepsilon) \geqslant K_\Phi - 5\varepsilon$. Since ε can be made arbitrarily small, the proof is complete.

10.5 From UG-hardness to Integrality Gaps

Completing the cycle of reductions, here we show that a dictatorship test can be used to construct an integrality gap, such that the ratio between the completeness and soundness of the dictatorship translates in to the integrality gap ratio. More precisely, we will show

Theorem 10.6. *For all $\varepsilon > 0$, there exists R, τ such that following holds: For any dictatorship test B on $L_2(\{\pm 1\}^R)$, there exists an instance B' of $K_{N,N}$-QuadraticProgramming such that,*

$$\mathrm{sdp}(B') \geqslant \mathsf{Completeness}(B)\,(1 - 5\varepsilon) \tag{10.7}$$

$$\mathrm{opt}(B') \leqslant \mathsf{Soundness}_{\varepsilon,\tau}(B)\,(1+\varepsilon) + \varepsilon\mathsf{Completeness}(B) \tag{10.8}$$

In particular, the choices $\tau = O(2^{-100/\varepsilon^3})$ and $R = \Omega(2^{200/\varepsilon^3})$ suffice.

Proof Sketch: As in the case of Max Cut Section 6.7 and other GCSPs Section 7.7, the idea is to execute the UG-hardness reductions starting with an SDP integrality gap for Unique Games. More precisely, if Red denotes the hardness reduction that maps instances of Unique Games to an instance of $K_{N,N}$-QuadraticProgramming. Starting with an SDP integrality gap Φ as input to Red, the resulting instance can be shown to be an SDP integrality gap for $K_{N,N}$-QuadraticProgramming.

For each vertex $u \in \mathcal{V}_\Phi \cup \mathcal{W}_\Phi$, there are R vectors $\{\boldsymbol{b}_{u,\ell}\}$ in the Unique Games SDP solution. As in the case of Max Cut, the SDP vector corresponding to a vertex $(u, \boldsymbol{x}) \in$

$(\mathcal{V}_\Phi \cup \mathcal{W}_\Phi) \times \{\pm 1\}^R$ is given by,

$$\boldsymbol{V}_{u,\boldsymbol{x}} = \sum_{\ell \in [R]} x_\ell \, \boldsymbol{b}_{u,\ell}$$

It is easy to check that the vectors $\boldsymbol{V}_{v,\boldsymbol{x}}$ are unit vectors, and their objective value is at least $\mathsf{Completeness}(B)\,(1-5\varepsilon)$. In particular, the proof follows almost along the lines of the corresponding reduction for Max Cut (presented in Section 6.7). We omit the details of the proof from the thesis.

By definition, the Grothendieck constant K_G is the maximum possible integrality gap for $K_{N,N}$-QuadraticProgramming. Therefore, we have the following corollary of the above theorem.

Corollary 10.6.1. *Fix $R = \Omega(2^{200/\varepsilon^3})$ and $\tau = O(2^{-100/\varepsilon^3})$. Then for any dictatorship test over $\{\pm 1\}^R$ we have*

$$\mathsf{Soundness}_{\varepsilon,\tau}(B) \geqslant \left(\frac{1}{K_G} - 6\varepsilon\right) \mathsf{Completeness}(B)$$

10.6 Implications

In this section, we follow the implications of the reductions between dictatorship tests, integrality gaps and UG-hardness results in order to prove the main theorems of this chapter.

10.6.1 Proof of Theorem 10.2

Consider the following idealized algorithm for the $K_{N,N}$-Quadratic Programming problem

- Find the optimal SDP solution $\boldsymbol{u}_i, \boldsymbol{v}_j$
- Fix $R = 2^{200/\varepsilon^3}$ and $\tau = 2^{-100/\varepsilon^3}$. For every function $\mathcal{F}, \mathcal{G} \in L_2(\{\pm 1\}^R)$ with $\|\mathcal{F}\|, \|\mathcal{G}\| \leqslant 1$, run the rounding scheme $\mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A)$ to obtain a $\{\pm 1\}$ solution. Output the solution with the largest value.

The value of the solution obtained is given by $\sup_{\mathcal{F},\mathcal{G} \in L_2(\{\pm 1\}^R)} \mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A)$. From Corollary 10.5.1 we have

$$\begin{aligned} \sup_{\mathcal{F},\mathcal{G} \in L_2(\{\pm 1\}^R), \|\mathcal{F}\|, \|\mathcal{G}\| \leqslant 1} \mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A) &\geqslant \sup_{\substack{\mathcal{F},\mathcal{G} \in L_2(\{\pm 1\}^R), \\ \mathrm{MaxComInf}(T_\rho \mathcal{F}, T_\rho \mathcal{G}) \leqslant \tau}} \mathsf{Round}_{\varepsilon,\mathcal{F},\mathcal{G}}(A) \\ &\geqslant \mathsf{Soundness}_{\varepsilon,\tau}(\mathsf{DICT}^\varepsilon_A) - \left(10\tau^{C\varepsilon/8}/\sqrt{\varepsilon}\right) \cdot \mathrm{sdp}(A) \end{aligned} \tag{10.9}$$

From Lemma 10.4.1, we know $\mathsf{Completeness}(\mathsf{DICT}^\varepsilon_A) = \mathrm{sdp}(A)(1-\varepsilon)$. By the choice of R, τ, we can apply Corollary 10.6.1 on $\mathsf{DICT}^\varepsilon_A$ to conclude

$$\mathsf{Soundness}_{\varepsilon,\tau}(\mathsf{DICT}^\varepsilon_A) \geqslant \mathsf{Completeness}(\mathsf{DICT}^\varepsilon_A)\left(\frac{1}{K_G} - \varepsilon\right) \geqslant \mathrm{sdp}(A)\left(\frac{1}{K_G} - \varepsilon\right)(1-\varepsilon) \tag{10.10}$$

From Equations 10.9 and 10.10, we conclude that the value returned by the algorithm is at least

$$\mathrm{sdp}(A)\left(\left(\frac{1}{K_G}-\varepsilon\right)(1-\varepsilon)-10\tau^{C\varepsilon/8}/\sqrt{\varepsilon}\right),$$

which by the choice of τ is at least $\mathrm{sdp}(A)(1/K_G-4\varepsilon)$.

In order to implement the idealized algorithm, we discretize the unit ball in space $L_2(\{\pm1\}^R)$ using a κ-net in the L_2 norm. As R is a fixed constant depending on ε, there is a finite κ-net that would serve the purpose. To finish the argument, one needs to show that the value of the solution returned is not affected by the discretization. This follows from the following lemma:

Lemma 10.6.1. *For $\mathcal{F},\mathcal{G},\mathcal{F}',\mathcal{G}'\in L_2(\{\pm1\}^R)$ with $\|\mathcal{F}\|,\|\mathcal{G}\|,\|\mathcal{F}'\|,\|\mathcal{G}'\|\leqslant 1$,*

$$|\mathsf{Round}_{\mathcal{F},\mathcal{G}}(A)-\mathsf{Round}_{\mathcal{F}',\mathcal{G}'}(A)|\leqslant \mathrm{sdp}(A)(\|\mathcal{F}-\mathcal{F}'\|+\|\mathcal{G}-\mathcal{G}'\|)$$

Proof. Define $\boldsymbol{u}_i'=\mathrm{trunc}\,T_\rho F(\Phi\boldsymbol{u}_i)$, $\boldsymbol{v}_j'=\mathrm{trunc}\,T_\rho G(\Phi\boldsymbol{v}_j)$ and $\boldsymbol{u}_i''=\mathrm{trunc}\,T_\rho F'(\Phi\boldsymbol{u}_i)$, $\boldsymbol{v}_j''=\mathrm{trunc}\,G'(\Phi\boldsymbol{v}_j)$. Substituting we get,

$$\mathsf{Round}_{\mathcal{F},\mathcal{G}}(A)-\mathsf{Round}_{\mathcal{F}',\mathcal{G}'}=\sum_{ij}a_{ij}\langle\boldsymbol{u}_i'-\boldsymbol{u}_i'',\boldsymbol{v}_j'\rangle+\sum_{ij}a_{ij}\langle\boldsymbol{u}_i'',\boldsymbol{v}_j'-\boldsymbol{v}_j''\rangle$$

As trunc and T_ρ are contractive operators, $\|\boldsymbol{u}_i'\|,\|\boldsymbol{u}_i''\|,\|\boldsymbol{v}_j'\|,\|\boldsymbol{v}_j''\|\leqslant 1$. Further, observe that $\|\boldsymbol{u}_i'-\boldsymbol{u}_i''\|\leqslant\|\mathcal{F}-\mathcal{F}'\|$ and $\|\boldsymbol{v}_j'-\boldsymbol{v}_j''\|\leqslant\|\mathcal{G}-\mathcal{G}'\|$, since for all x, $|\mathrm{trunc}\,\mathcal{F}(x)-\mathrm{trunc}\,\mathcal{F}'(x)|\leqslant|\mathcal{F}(x)-\mathcal{F}'(x)|$. Substituting in the above equation, we get the required result. ■

10.6.2 Proof of Theorem 10.3

A naive approach to compute the Grothendieck constant, is to iterate over all matrices A and compute the largest possible value of $\mathrm{sdp}(A)/\mathrm{opt}(A)$. However, the set of all matrices is an infinite set, and there is no guarantee on when to terminate.

As there is a conversion from integrality gaps to dictatorship tests and vice versa, instead of searching for the matrix with the worst integrality gap, we shall find the dictatorship test with the worst possible ratio between completeness and soundness. Recall that a dictatorship test is an operator on $L_2(\{\pm1\}^R)$ for a finite R depending only on ε the error incurred in the reductions. In principle, this already shows that the Grothendieck constant is computable up to an error ε in time depending only on ε.

Define K as follows

$$\frac{1}{K}=\inf_{\substack{\lambda_1=1,\\ \lambda_d\in[-1,1]\forall 0\leqslant d\leqslant R}}\ \sup_{\substack{\mathcal{F},\mathcal{G}\in L_2(\{\pm1\}^R),\\ \mathrm{MaxComInf}(T_\rho\mathcal{F},T_\rho\mathcal{G})\leqslant\tau\\ \|\mathcal{F}\|,\|\mathcal{G}\|\leqslant 1}}\langle\mathcal{F},\sum_{d=0}^{R}\rho^{2d}\lambda_dQ_d\mathcal{G}\rangle,\qquad\text{where }\rho=1-\varepsilon.$$

Let $\mathcal{P}$ denote the space of all pairs of functions $\mathcal{F},\mathcal{G}\in L_2(\{\pm1\}^R)$ with $\mathrm{MaxComInf}(T_\rho\mathcal{F},T_\rho\mathcal{G})\leqslant\tau$ and $\|\mathcal{F}\|,\|\mathcal{G}\|\leqslant 1$.. Since $\mathcal{P}$ is a compact set, there exists an ε-net of pairs of functions $\mathcal{F}=\{(\mathcal{F}_1,\mathcal{G}_1),\ldots,(\mathcal{F}_N,\mathcal{G}_N)\}$ such that : For every point

$(\mathcal{F}, \mathcal{G}) \in \mathcal{P}$, there exists $\mathcal{F}_i, \mathcal{G}_i \in \mathcal{F}$ satisfying $\|\mathcal{F} - \mathcal{F}'\| + \|\mathcal{G} - \mathcal{G}'\| \leqslant \varepsilon$. The size of the ε-net is a constant depending only on R and ε (note: R depends only on ε).

The constant K can be expressed using the following finite linear program:

$$\begin{aligned}
&\text{Minimize } \frac{1}{K} = \mu \\
&\text{Subject to } \mu \geqslant \sum_{d=0}^{R} \lambda_d \cdot \langle \mathcal{F}, \sum_{d=0}^{R} \rho^{2d} Q_d \mathcal{G} \rangle && \text{for all functions } \mathcal{F}, \mathcal{G} \in \mathcal{F} \\
&\qquad \lambda_i \in [-1, 1] && \text{for all } 0 \leqslant i \leqslant R \\
&\qquad \lambda_1 = 1
\end{aligned}$$

Chapter 11

HARDEST CSP TO APPROXIMATE?

11.1 Introduction

In the MAX κ-CSP problem, the input consists of a set of variables taking values over a domain(say $\{0,1\}$), and a set of constraints with each acting on k of the variables. The objective is to find an assignment of values to the variables that maximizes the number of constraints satisfied. Several classic optimization problems like MAX-3SAT, MAXCUT fall in to the general framework of CSPs.

Apart from its natural appeal, the study of approximability of MAX κ-CSP problem is interesting for yet another reason. The best approximation ratio achievable for MAX κ-CSP equals the optimal soundness of a **PCP** verifier making at most k queries. In fact, inapproximability results for MAX κ-CSP have often been accompanied by corresponding developments in analysis of linearity testing.

Over the boolean domain, the problem of MAX κ-CSP has been studied extensively. For a boolean predicate $P : \{0,1\}^k \to \{0,1\}$, the MAX κ-CSP (P) problem is the special case of MAX κ-CSP where all the constraints are of the form $P(l_1, l_2, \ldots, l_k)$ with each literal l_i being either a variable or its negation. For many natural boolean predicates P, approximation algorithms and matching NP-hardness results are known for MAX κ-CSP (P)[86]. For the general MAX κ-CSP problem over boolean domain, the best known algorithm yields a ratio of $\Omega(\frac{k}{2^k})$ [32], while any ratio better than $2^{\sqrt{2k}}/2^k$ is known to be NP-hard to achieve [51]. Further it is UG-hard to approximate MAX κ-CSP problem to a factor better than $\frac{2k}{2^k}$ [144].

In this chapter, we study the approximability of the MAX κ-CSP problem over non-boolean domains, more specifically over $\{0,1,\ldots,q-1\}$ for some integer q, obtaining a near-tight hardness result under the UGC. Specifically, we extend the techniques of [144] to obtain a UGC hardness result when q is a prime. More precisely, assuming the Unique Games Conjecture, we show that it is **NP**-hard to approximate the problem to a ratio greater than q^2k/q^k. Except for constant factors depending on q, the algorithm and the UG-hardness result have the same dependence on of the arity k. Independent of this work, Austrin and Mossel [19] obtain a more general UG-hardness result using entirely different techniques. Technically, the proof presented here extends the Gowers Uniformity based approach of Samorodnitsky and Trevisan [144] to correlations on q-ary cubes instead of the binary cube. This is related to the detection of multidimensional arithmetic progressions by a Gowers norm of appropriately large degree.

11.1.1 Related Work

The simplest algorithm for MAX κ-CSP over boolean domain is to output a random assignment to the variables, thus achieving an approximation ratio of $\frac{1}{2^k}$. The first improvement over this trivial algorithm, a ratio of $\frac{2}{2^k}$ was obtained by Trevisan [154]. Hast [79] proposed an approximation algorithm with a ratio of $\Omega(\frac{k}{\log k 2^k})$, which was later improved to the current best known algorithm achieving an approximation factor of $\Omega(\frac{k}{2^k})$ [32].

On the hardness side, MAX κ-CSP over the boolean domain was shown to be NP-hard to approximate to a ratio greater than $\Omega(2^{2\sqrt{k}}/2^k)$ by Samorodnitsky and Trevisan [143]. The result involved an analysis of a graph-linearity test which was simplified subsequently

by Håstad and Wigderson [84]. Later, using the machinery of multi-layered **PCP** developed in [49], the inapproximability factor was improved to $O(2^{\sqrt{2k}}/2^k)$ in [51].

A predicate P is *approximation resistant* if the best optimal approximation ratio for MAX k-CSP (P) is given by the random assignment. While no predicate over 2 variables is approximation resistant, a predicate over 3 variables is approximation resistant if and only if it is implied by the XOR of 3 variables [86, 163]. Almost all predicates on 4 variables were classified with respect to approximation resistance in [80].

Assuming the Unique Games Conjecture, a tight inapproximability of $\Theta\left(\frac{k}{2^k}\right)$ for the MAX k-CSP problem over the boolean domain was shown in [144]. The proof relies on the analysis of a hypergraph linearity test using the Gowers uniformity norms. Building on this work, Hastad showed that if UGC is true, then as k increases, nearly every predicate P on k variables is *approximation resistant* [83].

Subsequently, it was shown in [136] that for every CSP over an arbitrary finite domain, the best possible approximation ratio is equal to the integrality gap of a well known Semidefinite program. Further the same work also obtains an algorithm that achieves the best possible approximation ratio assuming UGC. Although the results of [136] apply to non-boolean domains, they do not determine the value of the approximation factor explicitly, but only show that it is equal to the integrality gap of an SDP. Further the algorithm proposed in [136] does not yield any approximation guarantee for MAX k-CSP unconditionally. Thus neither the inapproximability nor the algorithmic results of this work are subsumed by [136].

Austrin and Mossel [19] obtain a sufficient condition for a predicate P to be approximation resistant. Through this sufficiency condition, they obtain strong UG-hardness results for MAX k-CSP problem over the domain $\{1, \ldots, q\}$ for arbitrary k and q. For the case when q is a prime power, their results imply a UG-hardness of $kq(q-1)/q^k$. The hardness results in this work and [19] were obtained independently and use entirely different techniques.

11.1.2 Organization of the Chapter

We begin with background on the Gowers norm and influence of variables in Section 11.2. In Section 11.3, we present a linearity test that forms the core of the UG-hardness reduction. We prove our inapproximability result (for the case when q is a prime) by a reduction from Unique Games in Section 11.4. The proof uses a technical step bounding a certain expectation by an appropriate Gowers norm; this step is proved in Section 11.5.

11.2 Preliminaries

For a positive integer n, we use the notation $[n]$ for the ring $\mathbb{Z}/(n) = \{0, 1, \ldots, n-1\}$.

11.2.1 Gowers uniformity norm and influence of variables

We now recall the definition of the Gowers uniformity norm. For an integer $d \geqslant 1$ and a complex-valued function $f : G \to \mathbb{C}$ defined on an abelian group G (whose group operation

we denote by $+$), the d'th uniformity norm $U_d(f)$ is defined as

$$U^d(\mathcal{F}) := \mathop{\mathbb{E}}_{x,y_1,y_2,\ldots,y_d} \left[\prod_{\substack{S\subseteq\{1,2,\ldots,d\} \\ |S| \text{ even}}} \mathcal{F}\left(x + \sum_{i\in S} y_i\right) \prod_{\substack{S\subseteq\{1,2,\ldots,d\} \\ |S| \text{ odd}}} \overline{\mathcal{F}\left(x + \sum_{i\in S} y_i\right)} \right] . \qquad (11.1)$$

where the expectation is taken over uniform and independent choices of $x, y_0, \ldots, y_{d-1}$ from the group G. Note that $U^1(\mathcal{F}) = \left(\mathbb{E}_x[\mathcal{F}(x)]\right)^2$.

We will be interested in the case when the group G is $[q]^n$ for positive integers q, n, with group addition being coordinate-wise addition modulo q. G is also closed under coordinate-wise multiplication modulo q by scalars in $[q]$, and thus has a $[q]$-module structure. For technical reasons, we will restrict attention to the case when q is prime and thus our groups will be vector spaces over the field $\mathbb{F}_q$ of q elements. For a vector $\mathbf{a} \in [q]^k$, we denote by $a_1, a_2, \ldots, a_k$ its k coordinates. We will use $\mathbf{1}, \mathbf{0}$ to denote the all 1's and all 0's vectors respectively (the dimension will be clear from the context). Further denote by $\boldsymbol{e}_i$ the ith basis vector with 1 in the ith coordinate and 0 in the remaining coordinates. As we shall mainly be interested in functions over $[q]^n$ for a prime q, we make our further definitions in this setting. Firstly, every function $\mathcal{F} : [q]^n \to \mathbb{C}$ has a Fourier expansion given by:

$$\mathcal{F}(x) = \sum_{\sigma\in[q]^n} \hat{\mathcal{F}}_\sigma \chi_\sigma(x)$$

where $\hat{\mathcal{F}}_\sigma = \mathbb{E}_{x\in[q]^n}[\mathcal{F}(x)\chi_\sigma(x)]$ and $\chi_\sigma(x) = \prod_{i=1}^n \omega^{\sigma_i x_i}$ for a q^{th} root of unity ω.

The central lemma in the hardness reduction relates a large Gowers norm for a function f, to the existence of an influential coordinate.

The following well known result relates influences to the Fourier spectrum of the function.

Fact. For a function $f : [q]^n \to \mathbb{C}$ and a coordinate $i \in \{1, 2, \ldots, R\}$,

$$\mathrm{Inf}_i(f) = \sum_{\sigma_i\neq 0, \sigma\in[q]^n} |\hat{\mathcal{F}}_\sigma|^2 .$$

The following lemma is a restatement of Theorem 12 in [144].

Lemma 11.0.2. *There exists an absolute constant C such that, if $\mathcal{F} : [q]^m \to \mathbb{C}$ is a function satisfying $|\mathcal{F}(x)| \leqslant 1$ for every x then for every $d \geqslant 1$,*

$$U^d(\mathcal{F}) \leqslant U^1(\mathcal{F}) + 2^{Cd} \max_i \mathrm{Inf}_i(\mathcal{F})$$

11.2.2 Noise Operator

Like many other UG-hardness results, one of the crucial ingredients of our reduction will be a noise operator on functions over $[q]^n$. We define the noise operator $T_{1-\varepsilon}$ formally below.

Definition 11.2.1. For $0 \leqslant \varepsilon \leqslant 1$, define the operator $T_{1-\varepsilon}$ on functions $f : [q]^n \to \mathbb{C}$ as :

$$T_{1-\varepsilon} f(\mathbf{x}) = \mathop{\mathbb{E}}_{\eta}[\mathcal{F}(\mathbf{x} + \eta)]$$

where each coordinate η_i of η is 0 with probability $1 - \varepsilon$ and a random element from $[q]$ with probability ε. The Fourier expansion of $T_{1-\varepsilon} f$ is given by

$$T_{1-\varepsilon}\mathcal{F}(\mathbf{x}) = \sum_{\sigma \in [q]^n} (1-\varepsilon)^{|\sigma|} \hat{\mathcal{F}}_\sigma \chi_\sigma(x)$$

11.3 Linearity Tests and MAX κ-CSP *Hardness*

The best approximation ratio possible for MAX κ-CSP is identical to the best soundness of a **PCP** verifier for **NP** that makes k queries. This follows easily by associating the proof locations to CSP variables, and the tests of the verifier to k-ary constraints on the locations. In this light, it is natural that the hardness results of [143, 51, 144] are all associated with a linearity test with a strong soundness. The hardness result in this work is obtained by extending the techniques of [144] from binary to q-ary domains. In this section, we describe the test of [144] and outline the extension to it.

For the sake of simplicity, let us consider the case when $k = 2^d - 1$ for some d. In [144], the authors propose the following linearity test for functions $\mathsf{F} : \{0,1\}^n \to \{0,1\}$.

Complete Hypergraph Test (F, d)

- Pick $x_1, x_2, \ldots, x_d \in \{0,1\}^n$ uniformly at random.
- Accept if for each $S \subseteq [r]$,

$$\mathsf{F}(\sum_{i \in S} x_i) = \sum_{i \in S} \mathsf{F}(x_i)$$

The test reads the value of the function F at $k = 2^d - 1$ points of a random subspace(spanned by $x_1, \ldots, x_d$) and checks that F agrees with a linear function on the subspace. Note that a random function F would pass the test with probability $2^d/2^k$, since there are 2^d different satisfying assignments to the k binary values queried by the verifier. The following result is a special case of a more general result by Samorodnitsky and Trevisan [144].

Theorem 11.1. *[144] If a function $\mathsf{F} : \{0,1\}^n \to \{0,1\}$ passes the Complete Hypergraph Test with probability greater than $2^d/2^k + \gamma$, then the function $\mathcal{F}(x) = (-1)^{\mathsf{F}(x)}$ has a large d^{th} Gowers norm. Formally, $U^d(\mathcal{F}) \geqslant C(\gamma, k)$ for some fixed function C of γ, k.*

Towards extending the result to the domain $[q]$, we propose a somewhat similar linearity test. Again for convenience, let us assume $k = q^d$ for some d. Given a function $\mathsf{F} : [q]^n \to [q]$, the test proceeds as follows:

Affine Subspace Test (F, d)

- Pick $\mathbf{x}, \mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_d} \in [q]^n$ uniformly at random.
- Accept if for each $\mathbf{a} \subseteq [q]^d$,

$$\mathsf{F}\Big(\mathbf{x} + \sum_{i=1}^{d} a_i \mathbf{y_i}\Big) = \Big(1 - \sum_{i=1}^{d} a_i\Big)\mathsf{F}(\mathbf{x}) + \sum_{i=1}^{d} a_i \mathsf{F}\Big(\mathbf{x} + \mathbf{y_i}\Big)$$

Essentially, the test queries the values along a randomly chosen affine subspace, and tests if the function F agrees with an affine function on the subspace. Let ω denote a q'th root of unity. From Theorem 11.4 presented in Section 11.5, the following result can be shown:

Theorem 11.2. *If a function $\mathsf{F} : [q]^n \to [q]$ passes the Affine Subspace Test with probability greater than $q^{d+1}/q^k + \gamma$, then for some q'th root of unity $\omega \neq 1$, the function $\mathcal{F}(x) = \omega^{\mathsf{F}(x)}$ has a large dq'th Gowers norm. Formally, $U^{dq}(\mathcal{F}) \geqslant C(\gamma, k)$ for some fixed function C of γ, k.*

The above result follows easily from Theorem 11.4 using techniques of [144], and the proof is ommited here. The Affine Subspace Test forms the core of the UG-hardness reduction presented in Section 11.4.

11.4 Hardness reduction from Unique Games

In this section, we will prove a hardness result for approximating MAX k-CSP over a domain of size q when q is prime for every $k \geqslant 2$. Let d be such that $q^{d-1} + 1 \leqslant k \leqslant q^d$. Let us consider the elements of $[q]$ to have a natural order defined by $0 < 1 < \ldots < q-1$. This extends to a lexicographic ordering on vectors in $[q]^d$. Denote by $[q]^d_{<k}$ the set consisting of the k lexicographically smallest vectors in $[q]^d$. We shall identify the set $\{1, \ldots, k\}$ with set of vectors in $[q]^d_{<k}$. Specifically, we shall use $\{1, \ldots, k\}$ and vectors in $[q]^d_{<k}$ interchangeably as indices to the same set of variables. For a vector $\mathbf{x} \in [q]^n$ and a permutation π of $[n]$, define $\pi(x) \in [q]^n$ defined by $(\pi(x))_i = x_{\pi(i)}$.

Let $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi = \{\pi_e \ : \ [n] \to [n] | e \in E\}, [n])$ be a bipartite Unique Games instance. Towards constructing a k-CSP instance $\Im$ from Φ, we shall introduce a long code for each vertex in $\mathcal{V}_\Phi$. Specifically, the set of variables for the k-CSP $\Im$ is indexed by $\mathcal{V}_\Phi \times [q]^n$. Thus a solution to $\Im$ consists of a set of functions $\mathsf{F}_w : [q]^n \to [q]$, one for each $w \in \mathcal{V}_\Phi$.

Similar to several other long code based hardness results, we shall assume that the long codes are *folded*. More precisely, we shall use *folding* to force the functions F_w to satisfy $\mathsf{F}_w(\mathbf{x} + \mathbf{1}) = \mathsf{F}(\mathbf{x}) + 1$ for all $\mathbf{x} \in [q]^n$. The k-ary constraints in the instance $\Im$ are specified by the following verifier. The verifier uses an additional parameter ε that governs the level of noise in the noise operator.

- Pick a random vertex $w \in \mathcal{W}_\Phi$. Pick k vertices $\{v_{\mathbf{a}} | \mathbf{a} \in [q]^d_{<k}\}$ from $N(w) \subset \mathcal{V}_\Phi$ uniformly at random independently. Let $\pi_{\mathbf{a}}$ denote the permutation $\pi_{v_{\mathbf{a}} \leftarrow w}$ on the edge $(w, v_{\mathbf{a}})$.
- Sample $\mathbf{x}, \mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_d} \in [q]^n$ uniformly at random. Sample vectors $\eta_{\mathbf{a}} \in [q]^n$ for each $\mathbf{a} \in [q]^d_{<k}$ from the following distribution: With probability $1-\varepsilon$, $(\eta_{\mathbf{a}})_j = 0$ and with the remaining probability, $(\eta_{\mathbf{a}})_j$ is a uniformly random element from $[q]$.
- Query $\mathsf{F}_{v_{\mathbf{a}}}\Big(\pi_{\mathbf{a}}(\mathbf{x} + \sum_j a_j \mathbf{y_j} + \eta_{\mathbf{a}})\Big)$ for each $\mathbf{a} \in [q]^d_{<k}$. Accept if the following equality holds for each $\mathbf{a} \in [q]^d_{<k}$.

$$\mathsf{F}_{v_{\mathbf{a}}}\Big(\pi_{\mathbf{a}}(\mathbf{x}+\sum_{j=1}^{d} a_j \mathbf{y_j} + \eta_{\mathbf{a}})\Big) = \Big(1 - \sum_{j=1}^{d} a_j\Big)\mathsf{F}_{v_{\mathbf{0}}}\Big(\pi_{\mathbf{0}}(\mathbf{x}+\eta_{\mathbf{0}})\Big) + \sum_{j=1}^{d} a_j \mathsf{F}_{v_{\mathbf{e_j}}}\Big(\pi_{\mathbf{e_j}}(\mathbf{x} + \mathbf{y_j} + \eta_{\mathbf{e_j}})\Big) \,.$$

Theorem 11.3. *For all primes q, positive integers d, k satisfying $q^{d-1} < k \leqslant q^d$, and every $\gamma > 0$, there exists small enough $\delta, \varepsilon > 0$ such that*

- Completeness: *If Φ is a $(1-\delta)$-perfectly satisfiable instance of* Unique Games, *then* $\mathrm{opt}(\Im) \geqslant (1-\gamma)$
- Soundness: $\mathrm{opt}(\Phi) \leqslant \delta \Rightarrow \mathrm{opt}(\Im) \leqslant \frac{q^{d+1}}{q^k} + \gamma$.

Proof. We begin with the completeness claim, which is straightforward.

Completeness: There exists labellings to the Unique Game instance Φ such that for $1-\delta$ fraction of the vertices $w \in \mathcal{W}_\Phi$ all the edges (v, w) are satisfied. Let $\mathcal{A} : \mathcal{W}_\Phi \cup \mathcal{V}_\Phi \to [n]$ denote one such labelling. Define an assignment to the k-CSP instance by $\mathsf{F}_w(\mathbf{x}) = x_{A(w)}$ for all $w \in \mathcal{V}_\Phi$.

With probability at least $(1-\delta)$, the verifier picks a vertex $w \in \mathcal{W}_\Phi$ such that the assignment $\mathcal{A}$ satisfies all the edges $(w, v_{\mathbf{a}})$. In this case for each $\mathbf{a}$, $\pi_{\mathbf{a}}(\mathcal{A}(v)) = \mathcal{A}(v_{\mathbf{a}})$. Let us denote $\mathcal{A}(v) = \ell$. By definition of the functions F_w, we get $\mathsf{F}_{v_{\mathbf{a}}}(\pi_{\mathbf{a}}(x)) = (\pi_{\mathbf{a}}(x))_{\mathcal{A}(v_{\mathbf{a}})} = x_{\pi_{\mathbf{a}}^{-1}(\mathcal{A}(v_{\mathbf{a}}))} = x_\ell$ for all $x \in [q]^n$. With probability at least $(1-\varepsilon)^k$, each of the vectors $\eta_{\mathbf{a}}$ have their ℓth component equal to zero, i.e $(\eta_{\mathbf{a}})_\ell = 0$. In this case, it is easy to check that all the constraints are satisfied. In conclusion, the verifier accepts the assignment with probability at least $(1-\delta)(1-\varepsilon)^k$. For small enough δ, ε, this quantity is at least $(1-\gamma)$.

Soundness: Suppose there is an assignment given by functions F_v for $v \in \mathcal{V}_\Phi$ that the verifier accepts with probability greater than $\frac{q^{d+1}}{q^k} + \gamma$.

Let $z_1, z_2, \ldots, z_k$ be random variables denoting the k values read by the verifier. Thus $z_1, \ldots, z_k$ take values in $[q]$. Let $P : [q]^k \to \{0,1\}$ denote the predicate on k variables that represents the acceptance criterion of the verifier. Essentially, the value of the predicate $P(z_1, \ldots, z_k)$ is 1 if and only if $z_1, \ldots, z_k$ values are consistent with some affine function. By definition,

$$\Pr[\text{ Verifier Accepts }] = \mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{v_{\mathbf{a}}\in N(w)}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\mathop{\mathbb{E}}_{\eta_{\mathbf{a}}}\Big[P(z_1,\ldots,z_k)\Big] \geqslant \frac{q^{d+1}}{q^k}+\gamma$$

Let ω denote a q^{th} root of unity. The Fourier expansion of the function $P:[q]^k\to\mathbb{C}$ is given by

$$P(z_1,\ldots,z_k)=\sum_{\sigma\in[q]^k}\hat{P}_\sigma\chi_\sigma(z_1,\ldots,z_k)$$

where $\chi_\sigma(z_1,\ldots,z_k)=\prod_{i=1}^k\omega^{\sigma_i z_i}$ and $\hat{P}_\sigma=\mathbb{E}_{z_1,\ldots,z_k}[P(z_1,\ldots,z_k)\chi_\sigma(z_1,\ldots,z_k)]$. Notice that for $\sigma=\mathbf{0}$, we get $\chi_\sigma(z_1,\ldots,z_k)=1$. Further,

$$\hat{P}_{\mathbf{0}}=\mathop{\mathbb{E}}_{z_1,\ldots,z_k}[P(z_1,\ldots,z_k)]=\Pr[\text{ random assignment to } z_1,z_2,\ldots,z_k \text{ satisfies } P]=\frac{q^{d+1}}{q^k}$$

Substituting the Fourier expansion of P, we get

$$\Pr[\text{ Verifier Accepts }]=\frac{q^{d+1}}{q^k}+\sum_{\sigma\neq\mathbf{0}}\hat{P}_\sigma\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{v_{\mathbf{a}}\in N(w)}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\mathop{\mathbb{E}}_{\eta_{\mathbf{a}}}\Big[\chi_\sigma(z_1,\ldots,z_k)\Big]$$

Recall that the probability of acceptance is greater than $\frac{q^{d+1}}{q^k}+\gamma$. Further $|\hat{P}_\sigma|\leqslant 1$ for all $\sigma\in[q]^k$. Thus there exists $\sigma\neq 0$ such that,

$$\Big|\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{v_{\mathbf{a}}\in N(w)}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\mathop{\mathbb{E}}_{\eta_{\mathbf{a}}}\Big[\chi_\sigma(z_1,\ldots,z_k)\Big]\Big|\geqslant\frac{\gamma}{q^k}$$

For each $w\in\mathcal{V}_\Phi, t\in[q]$, define the function $\mathcal{F}_w^{(t)}:[q]^d\to\mathbb{C}$ as $\mathcal{F}_w^{(t)}(x)=\omega^{t\mathsf{F}_w(x)}$. For convenience we shall index the vector σ with the set $[q]^d_{<k}$ instead of $\{1,\ldots,k\}$. In this notation,

$$\Big|\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{v_{\mathbf{a}}\in N(w)}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\mathop{\mathbb{E}}_{\eta_{\mathbf{a}}}\Big[\prod_{\mathbf{a}\in[q]^d_{<k}}\mathcal{F}_{v_{\mathbf{a}}}^{(\sigma_{\mathbf{a}})}\big(\pi_{\mathbf{a}}(\mathbf{x}+\sum_{i=1}^d a_i\mathbf{y_i}+\eta_{\mathbf{a}})\big)\Big]\Big|\geqslant\frac{\gamma}{q^k}$$

Let $\mathcal{G}_w^{(t)}:[q]^d\to\mathbb{C}$ denote the *smoothened* version of function $\mathcal{F}_w^{(t)}$. Specifically, let $\mathcal{G}_w^{(t)}(x)=T_{1-\varepsilon}\mathcal{F}_w^{(t)}(x)=\mathbb{E}_\eta[\mathcal{F}_w^{(t)}(x+\eta)]$ where η is generated from ε-noise distribution. Since each $\eta_{\mathbf{a}}$ is independently chosen, we can rewrite the above expression,

$$\Big|\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{v_{\mathbf{a}}\in N(w)}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\Big[\prod_{\mathbf{a}\in[q]^d_{<k}}\mathcal{G}_{v_{\mathbf{a}}}^{(\sigma_{\mathbf{a}})}\big(\pi_{\mathbf{a}}(\mathbf{x}+\sum_{i=1}^d a_i\mathbf{y_i})\big)\Big]\Big|\geqslant\frac{\gamma}{q^k}\,.$$

For each $w\in\mathcal{W}_\Phi, t\in[q]$, define the function $\mathcal{G}_v^{(t)}:[q]^d\to\mathbb{C}$ as $\mathcal{G}_v^{(t)}(x)=\mathbb{E}_{w\in N(w)}[\mathcal{G}_w^{(t)}(\pi_{v\leftarrow w}(x))]$.

As the vertices $v_{\mathbf{a}}$ are chosen independent of each other,

$$\Big|\mathop{\mathbb{E}}_{w\in\mathcal{W}_\Phi}\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\Big[\prod_{\mathbf{a}\in[q]^d_{<k}}\mathcal{G}_v^{(\sigma_{\mathbf{a}})}\big(\mathbf{x}+\sum_{i=1}^d a_i\mathbf{y_i}\big)\Big]\Big|\geqslant\frac{\gamma}{q^k}\,.$$

As $\sigma\neq 0$, there exists an index $\mathbf{b}\in[q]^d_{<k}$ such that $\sigma_{\mathbf{b}}\neq 0$. For convenience let us denote $c=\sigma_{\mathbf{b}}$. Define

$$\kappa=2^{-Cdq}\left(\frac{\gamma}{2q^k}\right)^{2^{dq}}$$

where C is the absolute constant defined in Lemma 11.0.2.

For each $w\in\mathcal{W}_\Phi$, define the set of labels $\mathsf{L}(w)=\{\ell\in[n]\ :\ \mathrm{Inf}_\ell(\mathcal{G}_w^c)\geqslant\kappa\}$. Similarly for each $v\in\mathcal{V}_\Phi$, let $\mathsf{L}(v)=\{\ell\in[n]\ :\ \mathrm{Inf}_\ell(\mathcal{G}_v^c)\geqslant\kappa/2\}$. Obtain a labelling $\mathcal{A}$ to the Unique Games instance Φ as follows : For each vertex $u\in\mathcal{W}_\Phi\cup\mathcal{V}_\Phi$, if $\mathsf{L}(u)\neq\phi$ then assign a randomly chosen label from $\mathsf{L}(u)$, else assign a uniformly random label from $[n]$.

The functions $\mathcal{G}_w^{(c)}$ are given by $\mathcal{G}_w^{(c)}=T_{1-\varepsilon}\mathcal{F}_w^{(c)}$ where $\mathcal{F}_w^{(c)}$ is bounded in absolute value by 1. By Lemma 3.0.2, the sum of its influences is bounded by $\frac{1}{\varepsilon}$. Consequently, for all $v\in\mathcal{V}_\Phi$ the size of the label set $\mathsf{L}(v)$ is bounded by $\frac{2}{\kappa\varepsilon}$. Applying a similar argument to $w\in\mathcal{W}_\Phi$, $|\mathsf{L}(w)|\leqslant\frac{1}{\kappa\varepsilon}$.

For at least $\gamma/2q^k$ fraction of vertices $w\in\mathcal{W}_\Phi$ we have,

$$\Big|\mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y_1},\ldots,\mathbf{y_d}}\Big[\prod_{\mathbf{a}\in[q]^d_{<k}}\mathcal{G}_v^{(\sigma_{\mathbf{a}})}\big(\mathbf{x}+\sum_{i=1}^d a_i\mathbf{y_i}\big)\Big]\Big|\geqslant\frac{\gamma}{2q^k}$$

We shall refer to these vertices as *good* vertices. Fix a *good* vertex w.

Observe that for each $u\in\mathcal{W}_\Phi\cup\mathcal{V}_\Phi$ the functions $\mathcal{G}_u^{(t)}$ satisfy $|\mathcal{G}_u^{(t)}(x)|\leqslant 1$ for all x. Now we shall apply Theorem 11.4 to conclude that the functions $\mathcal{G}_w^{(t)}$ have a large Gowers norm. Specifically, consider the collection of functions given by $\mathcal{F}_{\mathbf{a}}=\mathcal{G}_w^{(\sigma_{\mathbf{a}})}$ for $\mathbf{a}\in[q]^d_{<k}$, and $\mathcal{F}_{\mathbf{a}}=1$ for all $\mathbf{a}\notin[q]^d_{<k}$. From Theorem 11.4, we get

$$\min_{\mathbf{a}}U^{dq}(\mathcal{G}_w^{(\sigma_{\mathbf{a}})})\geqslant\left(\frac{\gamma}{2q^k}\right)^{2^{dq}}$$

In particular, this implies $U^{dq}(\mathcal{G}_w^{(c)})\geqslant\left(\frac{\gamma}{2q^k}\right)^{2^{dq}}$. Now we shall use Lemma 11.0.2 to conclude that the function $\mathcal{G}_w$ has influential coordinates. Towards this, observe that the functions $\mathcal{F}_v^{(t)}$ satisfy $\mathcal{F}_v^{(t)}(x+1)=\mathcal{F}_v^{(t)}(x)\cdot\omega^t$ due to folding. Thus for all $t\neq 0$ and all $v\in\mathcal{V}_\Phi$, $\mathbb{E}_x[\mathcal{F}_v^{(t)}(x)]=0$. Specifically for $c\neq 0$,

$$U^1(\mathcal{G}_w^{(c)})=\Big(\mathop{\mathbb{E}}_x[\mathcal{G}_w^{(c)}(x)]\Big)^2=\Big(\mathop{\mathbb{E}}_{v\in N(w)}\mathop{\mathbb{E}}_\eta\mathop{\mathbb{E}}_x[\mathcal{F}_v^{(c)}(x+\eta)]\Big)^2=0$$

Hence it follows from Lemma 11.0.2 that there exists influential coordinates i with $\mathrm{Inf}_i(\mathcal{G}_w^{(c)})\geqslant 2^{-Cdq}\left(\frac{\gamma}{2q^k}\right)^{2^{dq}}=\kappa$. In other words, $\mathsf{L}(w)$ is non-empty. Observe that, due

to convexity of influences,

$$\mathrm{Inf}_\ell(\mathcal{G}_w^{(c)}) = \mathrm{Inf}_\ell(\mathop{\mathbb{E}}_{v\in N(w)}[\mathcal{G}_v^{(c)}]) \leqslant \mathop{\mathbb{E}}_{v\in N(w)} \mathrm{Inf}_{\pi_{v\leftarrow w}(\ell)}([\mathcal{G}_v^{(c)}(x)])$$

If the coordinate ℓ has influence at least κ on $\mathcal{G}_w^{(c)}$, then the coordinate $\pi_{v\leftarrow w}(\ell)$ has an influence of at least $\kappa/2$ for at least $\kappa/2$ fraction of neighbors $v \in N(w)$. The edge $\pi_{v\leftarrow w}$ is satisfied if ℓ is assigned to w, and $\pi_{v\leftarrow w}(\ell)$ is assigned to v. This event happens with probability at least $\frac{1}{|\mathsf{L}(u)||\mathsf{L}(v)|} \geqslant \kappa^2\varepsilon^2/2$ for at least $\kappa/2$ fraction of the neighbors $v \in N(w)$. As there are at least $(\gamma/2q^k)$ fraction of *good* vertices w, the assignment satisfies at least $(\gamma/2q^k)\cdot\kappa^2\varepsilon^2\cdot\kappa/4$ fraction of the UNIQUE GAMES constraints. By choosing δ smaller than this fraction, the proof is complete. ■

Since each test performed by the verifier involve k variables, by the standard connection between hardness of MAX k-CSP and k-query **PCP** verifiers, we get the following hardness result conditioned on the UGC.

Corollary 11.3.1. *For every prime q, it is UG-hard to approximate* MAX k-CSP *over domain size q within a factor that is greater than q^2k/q^k.*

Using the reduction of [120], the above UG-hardness result can be extended from primes to arbitrary composite number q.

Corollary 11.3.2. *[120] For every positive integer q, it is UG-hard to approximate* MAX k-CSP *over domain size q within a factor that is greater than $q^2k(1+o(1))/q^k$.*

11.5 Gowers Norm and Multidimensional Arithmetic Progressions

The following theorem forms a crucial ingredient in the soundness analysis in the proof of Theorem 11.3.

Theorem 11.4. *Let $q \geqslant 2$ be a prime and G be a $\mathbb{F}_q$-vector space. Then for all positive integers $\ell \leqslant q$ and d, and all collections $\{\mathcal{F}_{\mathbf{a}} : G \to \mathbb{C}\}_{\mathbf{a}\in[\ell]^d}$ of ℓ^d functions satisfying $|\mathcal{F}_{\mathbf{a}}(x)| \leqslant 1$ for every $x \in G$ and $\mathbf{a} \in [\ell]^d$, the following holds:*

$$\left|\mathop{\mathbb{E}}_{x,y_1,y_2,\ldots,y_d}\left[\prod_{\mathbf{a}\in[\ell]^d}\mathcal{F}_{\mathbf{a}}(x+a_1y_1+a_2y_2+\cdots+a_dy_d)\right]\right| \leqslant \min_{\mathbf{a}\in[\ell]^d}\left(U^{d\ell}(\mathcal{F}_{\mathbf{a}})\right)^{1/2^{d\ell}} \tag{11.2}$$

The proof of the above theorem is via double induction on d, ℓ. We first prove the theorem for the one-dimensional case, i.e., $d = 1$ and every ℓ, $1 \leqslant \ell < q$ (Lemma 11.4.1). This will be done through induction on ℓ. We will then prove the result for arbitrary d by induction on d.

Remark 11.5.1. Green and Tao, in their work [68] on configurations in the primes, isolate and define a property of a system of linear forms that ensures that the degree t Gowers norm is sufficient to analyze patterns corresponding to those linear forms, and called this

property *complexity* (see Definition 1.5 in [68]). Gowers and Wolf [67] later coined the term Cauchy-Schwartz (CS) complexity to refer to this notion of complexity. For example, the CS-complexity of the q linear forms $x, x+y, x+2y, \ldots, x+(q-1)y$ corresponding to a q-term arithmetic progression equals $q-2$, and the U^{q-1} norm suffices to analyze them. It can similarly be shown that the CS-complexity of the d-dimensional arithmetic progression (with q^d linear forms as in (11.2)) is at most $d(q-1)-1$. In our application, we need a "multi-function" version of these statements, since we have a different function $\mathcal{F}_{\mathbf{a}}$ for each linear form $x + \mathbf{a}\cdot\mathbf{y}$. We therefore work out a self-contained proof of Theorem 11.4 in this setting.

Lemma 11.4.1. *Let $q \geqslant 2$ be prime and ℓ, $1 \leqslant \ell \leqslant q$, be an integer, and G be a $\mathbb{F}_q$-vector space. Let $\{\mathcal{H}_\sigma : G \to \mathbb{C}\}_{\sigma\in[\ell]}$ be a collection of ℓ functions such that $|\mathcal{H}_\sigma(x)| \leqslant 1$ for all $\sigma \in [\ell]$ and $x \in G$. Then*

$$\left|\mathop{\mathbb{E}}_{x,y_1}\left[\prod_{\sigma\in[\ell]}\mathcal{H}_\sigma(x+\sigma y_1)\right]\right| \leqslant \min_{\sigma\in[\ell]}\left(U^\ell(\mathcal{H}_\sigma)\right)^{\frac{1}{2^\ell}} . \tag{11.3}$$

Proof. The proof is by induction on ℓ. For $\ell = 1$, the LHS of (11.3) equals $|\mathbb{E}_x[\mathcal{H}_0(x)]|$, and the RHS equals $\sqrt{U^1(\mathcal{H}_0)}$. By definition $U^1(\mathcal{H}_0) = \mathbb{E}_{x,y_1}[\mathcal{H}_0(x)\overline{\mathcal{H}_0(x+y_1)}] = |\mathbb{E}_x[\mathcal{H}_0(x)]|^2$.

Now consider ℓ satisfying $1 < \ell \leqslant q$. By a change of variables it suffices to upper bound the LHS of (11.3) by $\left(U^\ell(\mathcal{H}_{\ell-1})\right)^{1/2^\ell}$. We have

$$\begin{aligned}
\left|\mathop{\mathbb{E}}_{x,y_1}\left[\prod_{\sigma\in[\ell]}\mathcal{H}_\sigma(x+\sigma y_1)\right]\right|^2 &\leqslant \mathop{\mathbb{E}}_{x}[|\mathcal{H}_0(x)|^2]\cdot\mathop{\mathbb{E}}_{x}\left[\left|\mathop{\mathbb{E}}_{y_1'}\left[\prod_{\sigma\in\{1,\ldots,\ell-1\}}\mathcal{H}_\sigma(x+\sigma y_1')\right]\right|^2\right]\\
&\leqslant \mathop{\mathbb{E}}_{x,y_1',z_1'}\left[\prod_{\sigma\in\{1,\ldots,\ell-1\}}\mathcal{H}_\sigma(x+\sigma y_1')\overline{\mathcal{H}_\sigma(x+\sigma z_1')}\right]\\
&= \mathop{\mathbb{E}}_{x,y_1,z_1}\left[\prod_{\sigma\in\{0,1,\ldots,\ell-2\}}\mathcal{H}_{\sigma+1}(x+\sigma y_1)\overline{\mathcal{H}_{\sigma+1}(x+\sigma y_1+(\sigma+1)z_1)}\right]\\
&= \mathop{\mathbb{E}}_{z_1}\left[\mathop{\mathbb{E}}_{x,y_1}\left[\prod_{\sigma\in[\ell-1]}\tilde{h}_\sigma^{z_1}(x+\sigma y_1)\right]\right]
\end{aligned} \tag{11.4}$$

where we define $\tilde{h}_\sigma^{z_1}(t) := \mathcal{H}_{\sigma+1}(t)\overline{\mathcal{H}_{\sigma+1}(t+(\sigma+1)z_1)}$. By induction hypothesis, the inner expectation in (11.4) satisfies

$$\mathop{\mathbb{E}}_{x,y_1}\left[\prod_{\sigma\in[\ell-1]}\tilde{h}_\sigma^{z_1}(x+\sigma y_1)\right] \leqslant \left(U^{\ell-1}(\tilde{h}_{\ell-2}^{z_1})\right)^{\frac{1}{2^{\ell-1}}} . \tag{11.5}$$

Now,

$$\begin{aligned}\left(\mathop{\mathbb{E}}_{z_1}\left[U^{\ell-1}(\tilde{h}^{z_1}_{\ell-2})^{\frac{1}{2^{\ell-1}}}\right]\right)^{2^{\ell-1}} &\leqslant \mathop{\mathbb{E}}_{z_1}\left[U^{\ell-1}(\tilde{h}^{z_1}_{\ell-2})\right] \\ &= \mathop{\mathbb{E}}_{z_1}\mathop{\mathbb{E}}_{x,z_2,\ldots,z_\ell}\Bigg[\prod_{\substack{S\subseteq\{2,3,\ldots,d\}\\ |S| \text{ even}}} \mathcal{H}_{\ell-1}(x+\sum_{i\in S} z_i)\overline{\mathcal{H}_{\ell-1}(x+(\ell-1)z_1+\sum_{i\in S} z_i)} \\ &\qquad \prod_{\substack{S\subseteq\{2,3,\ldots,d\}\\ |S| \text{ odd}}} \overline{\mathcal{H}_{\ell-1}(x+\sum_{i\in S} z_i)}\mathcal{H}_{\ell-1}(x+(\ell-1)z_1+\sum_{i\in S} z_i)\Bigg] \\ &= U^\ell(\mathcal{H}_{\ell-1})\ , \end{aligned} \tag{11.6}$$

where the last step uses the fact that for a random choice of z_1, $(\ell-1)z_1$ is distributed uniformly in G (this is why we need q to be a prime). Combining (11.4), (11.5), and (11.6), we obtain our desired conclusion $\left|\mathbb{E}_{x,y_1}\left[\prod_{\sigma\in[\ell]}\mathcal{H}_\sigma(x+\sigma y_1)\right]\right| \leqslant \left(U^\ell(\mathcal{H}_{\ell-1})\right)^{1/2^\ell}$. ■

Proof of Theorem 11.4: Fix an arbitrary ℓ, $1 \leqslant \ell \leqslant q$. We will prove the result by induction on d. The base case $d = 1$ is the content of Lemma 11.4.1, so it remains to consider the case $d > 1$.

By a change of variables, it suffices to upper bound the LHS of (11.2) by $\left(U^{d\ell}(\mathcal{F}_{(\ell-1)\mathbf{1}})\right)^{1/2^{d\ell}}$, and this is what we will prove.

For $\sigma \in [\ell]$, and $y_2, y_3, \ldots, y_d \in G$, define the function

$$\mathcal{G}^{y_2,\ldots,y_d}_\alpha(x) = \prod_{\mathbf{b}=(b_2,b_3,\ldots,b_d)\in[\ell]^{d-1}} \mathcal{F}_{(\alpha,\mathbf{b})}(x+b_2y_2+\cdots+b_dy_d)\ . \tag{11.7}$$

The LHS of (11.2), raised to the power $2^{d\ell}$, equals

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{y_2,\ldots,y_d}\mathop{\mathbb{E}}_{x,y_1}\left[\prod_{\alpha\in[\ell]}\mathcal{G}^{y_2,\ldots,y_d}_\alpha(x+\alpha y_1)\right]\right|^{2^{d\ell}} &\leqslant \left(\mathop{\mathbb{E}}_{y_2,\ldots,y_d}\left[\ \left|\mathop{\mathbb{E}}_{x,y_1}\prod_{\alpha\in[\ell]}\mathcal{G}^{y_2,\ldots,y_d}_\alpha(x+\alpha y_1)\right|^{2^\ell}\right]\right)^{2^{(d-1)\ell}} \\ &\leqslant \left|\mathop{\mathbb{E}}_{y_2,\ldots,y_d} U^\ell(\mathcal{G}^{y_2,\ldots,y_d}_{\ell-1})\right|^{2^{(d-1)\ell}} \qquad \text{(using Lemma 11.4.1)} \\ &= \left|\mathop{\mathbb{E}}_{y_2,\ldots,y_d}\mathop{\mathbb{E}}_{x,z_1,\ldots,z_\ell}\left[\prod_{S\subseteq\{1,2,\ldots,\ell\}}\mathcal{G}^{y_2,\ldots,y_d}_{\ell-1}\Big(x+\sum_{i\in S} z_i\Big)\right]\ \right|^{2^{(d-1)\ell}}\end{aligned}$$

Defining the function

$$\mathcal{P}^{z_1,\ldots,z_\ell}_{\mathbf{b}}(t) := \prod_{S\subseteq\{1,2,\ldots,\ell\}} \mathcal{F}_{(\ell-1,\mathbf{b})}\Big(t+\sum_{i\in S} z_i\Big) \tag{11.8}$$

for every $\mathbf{b} \in [\ell]^{d-1}$ and $z_1, \ldots, z_\ell \in G$, the last expression equals

$$\left| \mathop{\mathbb{E}}_{z_1,\ldots,z_\ell} \mathop{\mathbb{E}}_{x,y_2,\ldots,y_d} \left[\prod_{\mathbf{b}=(b_2,\ldots,b_d)\in[\ell]^{d-1}} \mathcal{P}_{\mathbf{b}}^{z_1,\ldots,z_\ell}\Big(x + b_2y_2 + \cdots + b_dy_d\Big) \right] \right|^{2^{(d-1)\ell}}$$

which is at most

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_\ell} \left[\left| \mathop{\mathbb{E}}_{x,y_2,\ldots,y_d} \left[\prod_{\mathbf{b}=(b_2,\ldots,b_d)\in[\ell]^{d-1}} \mathcal{P}_{\mathbf{b}}^{z_1,\ldots,z_\ell}\Big(x + b_2y_2 + \cdots + b_dy_d\Big) \right] \right|^{2^{(d-1)\ell}} \right] . \tag{11.9}$$

By the induction hypothesis, (11.9) is at most

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_\ell} \left[U^{(d-1)\ell}\Big(\mathcal{P}_{(\ell-1)\mathbf{1}}^{z_1,\ldots,z_\ell}\Big) \right]$$

Recalling the definition of $\mathcal{P}_{\mathbf{b}}^{z_1,\ldots,z_\ell}$ from (11.8), the above expectation equals

$$\mathop{\mathbb{E}}_{z_1,\ldots,z_\ell} \mathop{\mathbb{E}}_{\substack{x,\{z'_j\} \\ 1\leqslant j\leqslant(d-1)\ell}} \left[\prod_{\substack{S\subseteq\{1,2,\ldots,\ell\} \\ T\subseteq\{1,2,\ldots,(d-1)\ell\}}} \mathcal{F}_{(\ell-1)\mathbf{1}}\Big(x + \sum_{i\in S} z_i + \sum_{j\in T} z'_j\Big) \right]$$

which clearly equals $U^{d\ell}(\mathcal{F}_{(\ell-1)\mathbf{1}})$.

11.6 Extending the CMM Algorithm for Non-Boolean CSPs

In this section, we outline a reduction from Non-boolean CSPs to boolean CSPs which in conjunction with CMM algorithm yields another algorithm for MAX κ-CSP .

Consider the case when $q = 2^t$ for an integer t. Given a CSP instance $\mathfrak{S} = (\mathcal{V}, \mathcal{P})$, for each variable $x \in \mathcal{V}$ in the q-ary CSP, introduce t boolean variables corresponding to the encoding of the value x in to binary. Every constraint on k variables in $\mathcal{V}$, translate in to a constraint on kt of the boolean variables. Using the algorithm from [32] on the boolean CSP instance, it is possible to obtain an assignment of value at least $0.44\frac{kt}{2^{kt}}\mathrm{opt}(\mathfrak{S}) > 0.44\frac{k}{q^k}\mathrm{opt}(\mathfrak{S})$.

Set $r = 2^{\lfloor \log q \rfloor}$. Then we have $q \geqslant r > q/2$. We will reduce the q-ary CSP instance $\mathfrak{S} = (\mathcal{V}, \mathcal{P})$ to a CSP $\mathfrak{S}' = (\mathcal{V}', \mathcal{P}')$ over a domain of size r. The variables of $\mathfrak{S}'$ are the same as that of $\mathfrak{S}$, i.e $\mathcal{V} = \mathcal{V}'$. For each variable $x \in \mathcal{V}'$, its domain(set of allowed values) is a randomly chosen subset of size r of the set $[q]$. Thus the domain of each variable in $\mathcal{V}'$ is a set of size r. The constraints in $\mathcal{P}'$ are the same as the constraints in $\mathcal{P}$ in the following sense: For any constraint $c \in \mathcal{P}$, there is a corresponding constraint $c' \in \mathcal{P}'$ whose satisfying assignments are the same as that of c.

Consider the optimal assignment $\mathcal{A}$ to the instance $\mathfrak{S}$. We will obtain an assignment to the instance $\mathfrak{S}'$ as follows: For a variable $x \in \mathcal{V}$, if the assignment $\mathcal{A}(x)$, is not an allowed value for x in $\mathfrak{S}'$, just reassign a random allowed value to x. For any given k-tuple of variables $(x_1, \ldots, x_k)$, with probability at least $(r/q)^k$ over the choice of the random restriction, all

the values $\mathcal{A}(x_1), \mathcal{A}(x_2), \ldots, \mathcal{A}(x_k)$ are allowed. Thus the expected number of constraints $\mathcal{P}'$ satisfied by the optimal assignment $\mathcal{A}$ is at least $\left(\frac{r}{q}\right)^k \cdot \text{opt}(\Im)$. Running the CMM algorithm on the instance $\Im'$ produces an assignment that satisfies at least $0.44\frac{k}{r^k} \cdot \left(\frac{r}{q}\right)^k \cdot \text{opt}(\Im) \geqslant 0.44\frac{k}{q^k} \cdot \text{opt}(\Im)$.

Part III

UNCONDITIONAL LOWER BOUNDS

Chapter 12

LIMITS OF SEMIDEFINITE PROGRAMMING

12.1 Introduction

Irrespective of the truth of UGC, it is now clear that UGC precisely identifies an algorithmic barrier reached by existing work on approximation algorithms. In particular, the results of Chapters 7, 8, 9 and 10 demonstrate this for many large classes of problems. A natural question that arises is whether stronger semidefinite programming relaxations are sufficient to breach this barrier and disprove the UGC? or does disproving UGC warrant the use of a new technique different from semidefinite programming?

Unfortunately, progress towards answering this compelling question has been slow and difficult. In the influential paper of Khot–Vishnoi [104], the authors construct an integrality gap instance for a simple SDP relaxation of Unique Games. To the best of our knowledge, this is the sole SDP gap construction for Unique Games that appears in literature. On one hand, this leaves out the possibility that strong SDPs disprove UGC. More alarmingly, except in a few cases, most UG-hardness results could possibly be falsified using a strong SDP relaxation. Except for Vertex Cover [64], and k-CSPs [145, 157], in all other cases, there are no strong SDP gaps supporting a UG-hardness result.

Obtaining strong SDP gap that support a UG-hardness result has been a difficult exercise. In fact, the work of [104] stemmed out of an effort in this direction for the Sparsest Cut problem. Specifically, the Goemans-Linial conjecture regarding embeddability of L_2^2 metrics into L_1 was refuted in [104] by constructing a SDP gap supporting the UGC based hardness for sparsest cut.

The following possibility is entirely consistent with the existing literature: Even for the Max Cut problem which is fairly well studied [99, 132], including an extra inequality on every set of 5 variables into the Goemans-Williamson semidefinite program (GW) yields a better approximation, thus disproving UGC.

12.1.1 Results

In this chapter, we exhibit an integrality gap for certain strong SDP relaxations of Unique Games. More precisely, we consider two strong hierarchies of SDP relaxations $\{\mathsf{LH}_r\}_{r\in\mathbb{N}}$ and $\{\mathsf{SA}_r\}_{r\in\mathbb{N}}$ defined in Chapter 4. We recall the rough definition of these relaxations here for the convenience of the reader. Formal definitions are reproduced in Section 12.3. The r^{th} level relaxation LH_r consists of the following: 1) SDP vectors for every vertex of the unique game, 2) All valid constraints on vectors corresponding to at most r vertices. Equivalently, the LH_r relaxation consists of SDP vectors and local distributions μ_S over integral assignments to sets S of at most r variables, such that the second moments of local distributions μ_S match the corresponding inner products of SDP vectors.

The SA_r relaxation is a strengthening of LH_r with the additional constraint that for two sets S, T of size at most r, the corresponding local distribution over integral assignments μ_S, μ_T must have the same marginal distribution over $S \cap T$. The SA_r relaxation corresponds to simple SDP relaxation strengthened by r^{th} round of Sherali-Adams hierarchy [150]. Let $\mathsf{LH}_r(\Phi)$ and $\mathsf{SA}_r(\Phi)$ denote the optimum value of the corresponding SDP relaxations on the instance Φ. Further, let $\mathrm{opt}(\Phi)$ denote the value of the optimum labeling for Φ. For the LH and SA hierarchies, we show:

Theorem 12.1. *For all constants $\eta > 0$, there exists a* Unique Games *instance Φ on N vertices such that $\mathsf{LH}_r(\Phi) \geqslant 1 - \eta$ and $\mathrm{opt}(\Phi) \leqslant \eta$ for $r = O(2^{(\log\log N)^{\frac{1}{4}}})$*

Theorem 12.2. *For all constants $\eta > 0$, there exists a* Unique Games *instance Φ on N vertices such that $\mathsf{SA}_r(\Phi) \geqslant 1 - \eta$ and $\mathrm{opt}(\Phi) \leqslant \eta$ for $r = O((\log\log N)^{\frac{1}{4}})$*

Demonstrated for the first time in [104], and used in numerous later works [34, 146, 157, 136, 73, 121], it is by now well known that integrality gaps can be composed with hardness reductions. A reduction from UG-hardness results to SDP integrality gaps was presented in great detail in Section 6.7. Such a reduction were also utilized in Chapters 7 and 10.

In particular, given a reduction Red from Unique Games to a certain problem Λ, on starting the reduction with a integrality gap instance Φ for Unique Games, the resulting instance $\mathsf{Red}(\Phi)$ is a corresponding integrality gap for Λ. Composing the integrality gap instance for LH_r or SA_r relaxation of Unique Games, along with UG reductions in [99, 17, 136, 73, 121, 137], one can obtain integrality gaps for LH_r and SA_r relaxations for several important problems. For the sake of succinctness, we will state the following general theorem:

Theorem 12.3. *Let Λ denote a problem in one of the following classes:*

- *A Generalized Constraint Satisfaction Problem (Definition 2.4.1).*
- *An Ordering Constraint Satisfaction Problem (Definition 9.8.2).*

Let LC denote the SDP relaxation that yields the optimal approximation ratio for Λ under UGC. Then the following holds: Given an instance $\Im$ of the problem Λ, with $\mathsf{LC}(\Im) \geqslant c$ and $\mathrm{opt}(\Im) \leqslant s$, for every constant $\eta > 0$, there exists an instance Ψ_η over N variables such that:

- $\mathsf{LH}_r(\Psi_\eta) \geqslant c - \eta$ *and* $\mathrm{opt}(\Psi_\eta) \leqslant s + \eta$ *with* $r = O(2^{(\log\log N)^{1/4}})$.
- $\mathsf{SA}_r(\Psi_\eta) \geqslant c - \eta$ *and* $\mathrm{opt}(\Psi_\eta) \leqslant s + \eta$ *with* $r = O((\log\log N)^{1/4})$.

The O notation in the number of rounds hides a constant depending on η.

The classes of problems for which the above result holds include Max Cut [99], Max 2-Sat [17], Grothendieck Problem(also called $K_{N,N}$-QuadraticProgramming) [137] k-way Cut [121] and Maximum Acyclic Subgraph [73]. Notable exceptions that do not directly fall under this framework are Vertex Cover and Sparsest Cut.

Reductions from Unique Games to Sparsest Cut have been exhibited in [104] and [38]. With the integrality gap for LH_r relaxation of Unique Games (Theorem 12.1), these reductions imply a corresponding LH_r integrality gap for Sparsest Cut. Integrality gaps for Sparsest Cut are directly related to lower bounds for distortion required to embed a given metric into L_1 metric. Here the L_1 metric consists of points in $\mathbb{R}^d$ for arbitrarily large d, and the distance between two points $(\mathbf{x}, \mathbf{y})$ is $\|\mathbf{x} - \mathbf{y}\|_1$. An L_2^2 metric consists of a set of points in $\mathbb{R}^d$ such that the squares of the distances between them also form a metric (satisfy triangle inequality). Restated in this language, the SDP vectors of the Sparsest Cut integrality gap that we construct, yield the following result:

Theorem 12.4. *For some absolute constants $\gamma, \delta > 0$, there exists an N-point L_2^2 metric that requires distortion at least $\Omega(\log\log N)^\delta$ to embedd into L_1, while every set of size at most $O(2^{(\log\log N)^\gamma})$ embedds isometrically into L_1.*

The Uniform Sparsest Cut problem is among the many important problems for which no Unique Games reduction is known. In [48], the techniques of [104] were extended to obtain an integrality gap for Uniform Sparsest Cut for the SDP with triangle inequalities. Roughly speaking, the SDP gap construction in [48] consists of the hypercube with its vertices identified by certain symmetries such as cyclic shift of the coordinates. Using the techniques from this chapter, the following SDP integrality gap for the Balanced Separator problem can be exhibited. The details of the proof of this theorem are omitted from the thesis.

Theorem 12.5. *For some absolute constants $\gamma, \delta > 0$, there exists an instance G on N vertices of* Balanced Separator *such that the ratio* $\mathrm{opt}(G)/\mathsf{LH}_r(G) \geqslant \Omega(\log\log N)^\delta$ *for* $r = O(\log\log N)^\gamma$.

12.1.2 Related Work

In a breakthrough result, Arora et al. [16] used a strong semidefinite program with triangle inequalities to obtain $O(\sqrt{\log n})$ approximation for the Sparsest Cut problem. Inspired by this work, stronger semidefinite programs have been utilized to obtain better approximation algorithms for certain graph coloring problems [40, 12, 42]. We wish to point out that the work of Chlamtac and Singh [42] uses the SA_r hierarchy to obtain approximation algorithms for the hypergraph coloring problem.

In this light, hierarchies of stronger SDP relaxations such as Lovász–Schriver [118], Lasserre [110], and Sherali–Adams hierarchies [150] (See [112] for a comparison) have emerged as possible avenues to obtain better approximation ratios.

Considerable progress has been made in understanding the limits of linear programming hierarchies. Building on a sequence of works [10, 11, 152, 153], Schoenebeck et al. [147] obtained a $2-\varepsilon$-factor integrality gap for $\Omega(n)$ rounds of Lovász–Schriver LS hierarchy. More recently, Charikar et al. [34] constructed integrality gaps for $\Omega(n^\delta)$ rounds of Sherali–Adams hierarchy for several problems like Max Cut, Minimum Vertex Cover, Sparsest Cut and Maximum Acyclic Subgraph. Furthermore, the same work also exhibits $\Omega(n^\delta)$-round Sherali–Adams integrality gap for Unique Games, in turn obtaining a corresponding gap for every problem to which Unique Games is reduced to.

Lower bound results of this nature are fewer in the case of semidefinite programs. A $\Omega(n)$ LS_+ round lower bound for proving unsatisfiability of random 3-SAT formulae was obtained in [28, 1]. In turn, this leads to $\Omega(n)$-round LS_+ integrality gaps for problems like Set Cover, Hypergraph Vertex Cover where a matching **NP**-hardness result is known. Similarly, the $\frac{7}{6}$-integrality gap for $\Omega(n)$ rounds of LS_+ in [146] falls in a regime where a matching **NP**-hardness result has been shown to hold. A significant exception is the result of Georgiou et al. [64] that exhibited a $2-\varepsilon$-integrality gap for $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ rounds of LS_+ hierarchy. More recently, building on the beautiful work of [145] on Lasserre integrality gaps

for Random 3-SAT, Tulsiani [157] obtained a $\Omega(n)$-round Lasserre integrality gap matching the corresponding UG-hardness for k-CSP [144].

12.1.3 Overview of the Technique

In this section, we will present a brief overview of the techniques and a roadmap for the rest of the chapter.

The overall strategy in this work to construct SDP integrality gaps is along the lines of Khot–Vishnoi [104]. Let us suppose we wish to construct a SDP integrality gap for a problem Λ (say Max Cut). Let Red_Λ be a reduction from Unique Games to the problem Λ. The idea is to construct a SDP integrality gap Φ for Unique Games, and then execute the reduction Red_Λ on the instance Φ, to obtain the SDP gap construction $\mathsf{Red}_\Lambda(\Phi)$. Surprisingly, as demonstrated in [104], the SDP vector solution for Φ can be transformed through the reduction to obtain the SDP solution for $\mathsf{Red}_\Lambda(\Phi)$.

Although this technique has been used extensively in numerous works [34, 147, 157, 136, 73, 121] since [104], there is a crucial distinction between [104] and later works. In all other works, starting with an SDP gap Φ for Unique Games, one obtains an integrality gap for an SDP relaxation that is no stronger. For instance, starting with a integrality gap for 10-rounds of a SDP hierarchy, the resulting SDP gap instance satisfies at most 10 rounds of the same hierarchy.

The surprising aspect of [104], is that it harnesses the UG reduction Red_Λ to obtain an integrality gap for a "stronger" SDP relaxation than the one which it stared with. Specifically, starting with an integrality gap Φ for a simple SDP relaxation of Unique Games, [104] exhibit an SDP gap for Max Cut which obeys all valid constraints on 3 variables. The proof of this fact (the triangle inequality) is perhaps the most technical and least understood aspect about [104]. One of the main contributions of this chapter is to conceptualize and simplify this aspect of [104]. Armed with the understanding of [104], we then develop the requisite machinery to extend it to a strong SDP integrality gap for Unique Games.

To obtain strong SDP gaps for Unique Games, we will apply the above strategy on the reduction from Unique Games to $\mathsf{E2Lin}_q$ obtained in [99]. Note that $\mathsf{E2Lin}_q$ is a special case of Unique Games. Formally, we show the following reduction from a *weak gap* instance for Unique Games over a large alphabet to a integrality gap for a strong SDP relaxation of $\mathsf{E2Lin}_q$.

Theorem 12.6. *(Weak Gaps for* Unique Games $\Longrightarrow$ *Strong gaps for* $\mathsf{E2Lin}_q$*)*
For a positive integer q, let $\mathsf{Red}_{\mathsf{E2Lin}_q}$ denote the reduction from Unique Games *to* $\mathsf{E2Lin}_q$*. Given a $(1-\eta,\delta)$*-weak gap *instance Φ for* Unique Games*, the $\mathsf{E2Lin}_q$ instance $\mathsf{Red}_{\mathsf{E2Lin}_q}(\Phi)$ is a $(1-2\gamma, 1/q^{\gamma/2}+o_\delta(1))$ SDP gap for the relaxation LH_r for $r = 2^{O(1/\eta^{1/4})}$. Further, $\mathsf{Red}_{\mathsf{E2Lin}_q}(\Phi)$ is a $(1-\gamma,\delta)$ SDP gap for the relaxation SA_r for $r = O(1/\eta^{1/4})$.*

Using the weak gap for Unique Games constructed in [104], along with the above theorem, implies Theorems 12.1 and 12.2. As already pointed out, by now it is fairly straightforward to compose an r-round integrality gap for Unique Games, with reductions to obtain a r round integrality gaps for other problems. Hence, Theorem 12.3 is a fairly straightforward consequence of Theorems 12.1 and 12.2.

12.1.4 Organization

In the next section, we present a detailed proof overview that describes the entire integrality gap construction restricted to the case of Max Cut. The formal definitions of the SDP hierarchies LH_r,SA_r and their robustness are presented in Section 12.3. We formally define weak gap instances for Unique Games in Section 12.4. We also outline an alternate integrality gap for a very minimal SDP relaxation of Unique Games in the same section. This section is followed by the description of the integrality gap instance for $\mathsf{E2Lin}_q$ obtained by reduction of Khot et al. [99]. In the rest of the chapter, we construct SDP vectors and local distributions to show that this is an integrality gap for the strong SDP relaxations – LH_r and SA_r. The two subsequent sections are devoted to developing the requisite machinery of integral vectors, their tensor products and local distributions for Unique Games. The SDP vectors and local distributions for the integrality gap instance described in Section 12.5 are exhibited in Sections Section 12.8 and Section 12.8.2.

12.2 Proof Overview

For the sake of exposition, we will describe the construction of an SDP integrality gap for Max Cut. To further simplify matters, we will exhibit an integrality gap for the basic Goemans-Williamson relaxation, augmented with the triangle inequalities on every three vectors. While an integrality gap of this nature is already part of the work of Khot–Vishnoi [104], our proof will be conceptual and amenable to generalization.

Let Φ be a SDP integrality gap for Unique Games on an alphabet $[R]$. For each vertex B in Φ, the SDP solution associates R orthogonal unit vectors $B = \{b_1, \cdots, b_R\}$. For the sake of clarity, we will refer to a vertex B in Φ and the set of vectors $B = \{b_1, \ldots, b_R\}$ associated with it as a "cloud". The clouds satisfy the following properties:

- (Matching Property) For every two clouds A, B, there is a unique matching $\pi_{B \leftarrow A}$ along which the inner product of vectors between A and B is maximized. Specifically, if $\rho(A, B) = \max_{a \in A, b \in B} \langle a, b \rangle$, then for each vector a in A, we have $\langle a, \pi_{B \leftarrow A}(a) \rangle = \rho(A, B)$.

- (High objective value) For most edges $e = (A, B)$ in the Unique Games instance Φ, the maximal matching $\pi_{A \leftarrow B}$ is the same as the permutation π_e corresponding to the edge, and $\rho(A, B) \approx 1$.

Let $\mathsf{Red}_{\text{Max Cut}}(\Phi)$ be the Max Cut instance obtained by executing the reduction in [99] on Φ. The reduction $\mathsf{Red}_{\text{Max Cut}}$ in [99] introduces a long code (2^R vertices indexed by $\{-1, 1\}^R$) for every cloud in Φ. Hence the vertices of $\mathsf{Red}_{\text{Max Cut}}(\Phi)$ are given by pairs (B, x) where B is a cloud in Φ and $x \in \{-1, 1\}^R$.

The SDP vectors we construct for the integrality gap instance resemble (somewhat simpler in this work) the vectors in [104]. Roughly speaking, for a vertex (B, x), we associate an SDP vector $\boldsymbol{V}^{B,x}$ defined as follows:

$$\boldsymbol{V}^{B,x} = \frac{1}{\sqrt{R}} \sum_{i \in [R]} x_i b_i^{\otimes t}$$

The point of departure from [104] is the proof that the vectors form a feasible solution for the stronger SDP. Instead of directly showing that the inequalities hold for the vectors, we exhibit a distribution over integral assignments whose second moments match the inner products. Specifically, to show that triangle inequality holds for three vertices $S = \{(A, x), (B, y), (C, z)\}$, we will exhibit a μ_S distribution over $\{\pm 1\}$ assignments to the three vertices, such that

$$\mathop{\mathbb{E}}_{\{Y^{A,x}, Y^{B,y}, Y^{C,z}\} \sim \mu_S} [Y^{A,x} Y^{B,y}] = \langle \boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y} \rangle$$

The existence of an integral distribution matching the inner products shows that the vectors satisfy all valid inequalities on the three variables, including the triangle inequality. We shall construct the distribution μ_S over local assignments in three steps,

Local Distributions over Labelings for Unique Games For a subset of clouds $\mathcal{S}$ within the Unique Games instance Φ, we will construct a distribution $\mu_{\mathcal{S}}$ over labelings to the set $\mathcal{S}$. The distribution $\mu_{\mathcal{S}}$ over $[R]^{\mathcal{S}}$ will be "consistent" with the SDP solution to Φ. More precisely, if two clouds A and B are *highly correlated* ($\rho(A, B) \approx 1$), then when the distribution $\mu_{\mathcal{S}}$ assigns label ℓ to A, with high probability it assigns the corresponding label $\pi_{B \leftarrow A}(\ell)$ to B. Recall that $\rho(A, B)$ was defined as $\max_{a \in A, b \in B} \langle a, b \rangle$.

Consider a set $\mathcal{S}$ where every pair of clouds A, B are *highly correlated* ($\rho(A, B) \geqslant 0.9$). We will refer to such a set of clouds as *Consistent*. For a *Consistent* set $\mathcal{S}$,assigning a label ℓ for a cloud A in $\mathcal{S}$, forces the label of every other cloud B to $\pi_{B \leftarrow A}(\ell)$. Furthermore, it is easy to check that the resulting labeling satisfies consistency for every pair of clouds in $\mathcal{S}$. (see Lemma 12.9.2 for details) Hence, in this case, the distribution $\mu_{\mathcal{S}}$ could be simply obtained by picking the label ℓ for an arbitrary cloud in $\mathcal{S}$ uniformly at random, and assigning every other cloud the induced label.

Now consider a set $\mathcal{S}$ which is not consistent. Here the idea is to decompose the set of clouds $\mathcal{S}$ into clusters, such that each cluster is consistent. Given a decomposition, for each cluster the labeling can be independently generated as described earlier. In this chapter, we will use a geometric decomposition to decompose the set of clouds $\mathcal{S}$ into clusters. The crucial observation is that the correlations $\rho(A, B)$ for clouds $A, B \in \mathcal{S}$, can be approximated well by a certain L_2^2 metric. More precisely, for each cloud A, we can associate a unit vector $\boldsymbol{v}_A = \sum_{a \in A} a^{\otimes s}$ such that the L_2^2 distance between $\boldsymbol{v}_A, \boldsymbol{v}_B$ is a good approximation of the quantity $1 - \rho(A, B)$.

By using t random halfspace cuts on this geometric representation, we obtain a partition into 2^t clusters. A pair of clouds A, B that are not highly correlated ($\rho(A, B) < 1 - 1/16$), are separated by the halfspaces with probability at least $1 - (3/4)^t$. Hence for a large enough t, all resulting clusters are consistent with high probability. (see Lemma 12.9.3).

A useful feature of the geometric clustering is that for two subsets $\mathcal{T} \subset \mathcal{S}$, the distribution over labelings $\mu_{\mathcal{T}}$ is equal to the marginal of the distribution $\mu_{\mathcal{S}}$ on $\mathcal{T}$. To see this, observe that the distribution over clusterings depends solely on the geometry of the associated vectors. On the downside, the geometric clustering produces inconsistent clusters with a very small but non-zero probability. (see Corollary 12.9.2).

The details of the construction of local distributions to Unique Games are presented

in 12.7.

Constructing Approximate Distributions Fix a set $S \subseteq \mathcal{S} \times \{\pm 1\}^R$ of vertices in the Max Cut instance $\mathsf{Red}_{\text{Max Cut}}(\Phi)$. We will now describe the construction of the local integral distribution μ_S.

In the reduction $\mathsf{Red}_{\text{Max Cut}}$, the labeling ℓ to a cloud B in the Unique Games instance is encoded as choosing the ℓth dictator cut in the long code corresponding to cloud B. Specifically, assigning the label ℓ to a cloud B should translate into assigning x_ℓ for every vertex (B, x) in the long code of B. Hence, a straightforward approach to define the distribution μ_S would be the following:

- Sample a labeling $\ell : \mathcal{S} \to [R]$ from the distribution $\mu_{\mathcal{S}}$,
- For every vertex $(B, x) \in \mathcal{S}$, assign $x_{\ell(B)}$.

Although inspired by this, our actual construction of μ_S is slightly more involved. First, we make the following additional assumption regarding the Unique Games instance Φ:

> *Assumption*: All the SDP vectors for the integrality gap instance Φ are $\{\pm 1\}$-vectors (have all their coordinates from $\{\pm 1\}$).

The SDP gap instance for Unique Games constructed in [104] satisfies this additional requirement. Furthermore, we outline a generic transformation to convert an arbitrary Unique Games SDP gap into one that satisfies the above property (see Observation 12.6.1). A $\{\pm 1\}$-vector is to be thought of as a distribution over $\{\pm 1\}$ assignments. It is easy to see that tensored powers of $\{\pm 1\}$-vectors yield $\{\pm 1\}$-vectors. Let T denote the number of coordinates in the vectors $\boldsymbol{V}^{B,x}$. The distribution μ_S is defined as follows,

- Sample a labeling $\ell : \mathcal{S} \to [R]$ from the distribution $\mu_{\mathcal{S}}$, and a coordinate $i \in [T]$ uniformly at random.
- For every vertex $(B, x) \in S$, assign $Y^{B,x}$ to be the ith coordinate of the vector $x_{\ell(B)} b_{\ell(B)}^{\otimes t}$.

We will now argue that the first two moments of the local distributions μ_S defined above, approximately match the corresponding inner products between SDP vectors.

Consider the inner product $\langle \boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y} \rangle$ of the SDP vectors corresponding to some pair of vertices (A, x) and (B, y) in S. The inner product consists of R^2 terms of the form $\langle x_i a_i^{\otimes t}, y_j b_j^{\otimes t} \rangle$. The crucial observation we will utilize is that the inner product $\langle \boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y} \rangle$ is approximately determined by the R terms corresponding to the matching $\pi_{B \leftarrow A}$. In other words, we have

$$\langle \boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y} \rangle \approx \tfrac{1}{n} \sum_{\ell \in [R]} x_\ell y_{\pi_{B \leftarrow A}(\ell)} \langle a_\ell^{\otimes t}, b_{\pi_{B \leftarrow A}(\ell)}^{\otimes t} \rangle \qquad \leqslant \quad \rho(A, B)^t$$

(see Section 12.4.2 for details)

If $\rho(A,B) < 0.9$, then with high probability the clustering would place the clouds A, B in different clusters. Hence the labels assigned to A, B would be completely independent of each other, and so would the assignments to (A,x) and (B,y). Hence, we would have $\mathbb{E}[Y^{A,x}Y^{B,y}] = 0$. On the other hand, by the above inequality the inner product $\langle \boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y}\rangle \leqslant 0.9^t \approx 0$. Therefore, for clouds A, B that are not *highly correlated*, the inner product of vectors $\boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y}$ agree approximately with the distribution over local assignments.

At the other extreme, if $\rho(A,B) \approx 1$, then with high probability the clustering would not separate A from B. If A, B are not separated, then the distribution $\mu_{\mathcal{S}}$ over labelings will respect the matching between A and B. Specifically, whenever A is assigned label ℓ by $\mu_{\mathcal{S}}$, with high probability B is assigned the label $\pi_{B\leftarrow A}(\ell)$. Consequently, in this case we have

$$\mathop{\mathbb{E}}_{\mu_S}[Y^{A,x}Y^{B,y}] = \frac{1}{n}\sum_{\ell\in[R]} \langle x_\ell a_\ell^{\otimes t}, y_{\pi_{B\leftarrow A}(\ell)} b_{\pi_{B\leftarrow A}(\ell)}^{\otimes t}\rangle \approx \langle \boldsymbol{V}^{A,x}, \boldsymbol{V}^{B,y}\rangle$$

Smoothing In Chapter 4, we showed a robustness property for the LH_r and SA_r relaxations by which approximately feasible solutions to these hierarchies can be converted (*smoothed*) into perfectly feasible solutions with a small loss in the objective value.

To illustrate the idea behind the robustness, consider a set of unit vectors $\{v_i\}_{i=1}^R$ that satisfy all triangle inequalities up to an additive error of ε, i.e.,

$$\|v_i - v_j\|^2 + \|v_j - v_k\|^2 - \|v_i - v_k\|^2 \geqslant -\varepsilon$$

For the sake of completeness, we include the statements of the claims about the robustness of solutions to LH_r and SA_r (Theorems 12.8, 12.7) in Section 12.3. We refer the reader to Section 4.8 for the proofs of these claims.

Extending to $\mathsf{E2Lin}_q$ The above argument for Max Cut can be made precise. However, to obtain an SDP gap for larger number of rounds, we use a slightly more involved construction of SDP vectors.

$\{\pm 1\}$-vectors were natural in the above discussion, since Max Cut is a CSP over $\{0,1\}$. For $\mathsf{E2Lin}_q$, it is necessary to work with vectors whose coordinates are from $\mathbb{F}_q$, as opposed to $\{\pm 1\}$. The tensoring operation for $\mathbb{F}_q$-integral vectors is to be appropriately defined to ensure that while the behaviour of the inner products resemble traditional tensoring, the tensored vectors are $\mathbb{F}_q$-integral themselves (see Section 12.6 for details).

For the case of Max Cut, we used a gap instance Φ for Unique Games all of whose SDP vectors where $\{\pm 1\}$-vectors. In case of $\mathsf{E2Lin}_q$, the SDP vectors corresponding to the Unique Games instance Φ would have to be $\mathbb{F}_q$-integral vectors. We outline a generic transformation to convert an arbitrary Unique Games SDP gap into one that satisfies this property (see Observation 12.6.4).

12.3 Preliminaries

Here we recall the definitions of the LH_r and SA_r hierarchies. We refer the reader to Section 4.7 for more details. Let $\Im$ be a GCSP (say Unique Games) instance over a set of variables $\mathcal{V}$, alphabet size q and arity k. A feasible solution to the LH_r relaxation consists of the following:

1. A *collection of (local) distributions* $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$, where $\mu_S\colon [q]^S \to \mathbb{R}_+$ is a distribution over $[q]$-assignments to S, that is, $\mu_S \in \blacktriangle([q]^S)$.

2. A *(global) vector solution* $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in[q]}$, where $\boldsymbol{b}_{i,a}\in\mathbb{R}^d$ for every $i\in\mathcal{V}$ and $a\in[q]$.

LH_r-Relaxation.

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}}\mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \tag{LH_r}$$

$$\text{subject to} \quad \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_S}\Big\{x_i = a, x_j = b\Big\} \qquad S\subseteq\mathcal{V},\ |S|\leqslant r,\ \ i,j\in S,\ \ a,b\in[q]\,, \tag{12.1}$$

$$\mu_S \in \blacktriangle\left([q]^S\right) \tag{12.2}$$

Here, $\blacktriangle\left([q]^S\right)$ denotes probability distributions over $[q]^S$. As usual, we denote by $\mathsf{LH}_r(\Im)$ the value of an optimal solution to this relaxation.

The above relaxation succinctly encodes all possible inequalities on up to r vectors. The next remark makes this observation precise.

Remark 12.3.1. A linear inequality on the inner products of a subset of vectors $\{\boldsymbol{b}_{i,a}\}_{i\in S,a\in[q]}$ for $S\subseteq\mathcal{V}$ is *valid* if it inequality if it holds for all distributions over $[q]$-assignments to the variables S. A feasible solution to the LH_r-relaxation satisfies all valid inequalities on sets of up to r vectors.

12.3.1 SA_r-Relaxation

Enforcing consistency between the marginals of the local distributions yields the SA_r-relaxation.

SA_r-Relaxation:

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}}\mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \tag{SA_r}$$

$$\text{subject to} \quad \langle \boldsymbol{b}_{i,a}, \boldsymbol{b}_{j,b}\rangle = \Pr_{x\sim\mu_S}\Big\{x_i = a, x_j = b\Big\} \qquad S\subseteq\mathcal{V},\ \ |S|\leqslant r,\ \ i,j\in S,\ \ a,b\in[q]\,, \tag{12.3}$$

$$\|\mathsf{margin}_{A\cap B}\,\mu_A - \mathsf{margin}_{A\cap B}\,\mu_B\|_1 = 0 \qquad A,B\subseteq\mathcal{V},\ \ |A|,|B|\leqslant r\,. \tag{12.4}$$

$$\mu_S \in \blacktriangle\left([q]^S\right) \tag{12.5}$$

Remark 12.3.2. The SA_r relaxation is closely related to the r^{th} level of the Sherali–Adams hierarchy. In fact, SA_r is obtained from the basic SDP relaxation by r-rounds Sherali–Adams lift-and-project.

12.3.2 Robustness

In Section 4.9, we showed that the LH_r and SA_r are *robust* in that approximately feasible solutions to these relaxations can be converted into a completely feasible relaxation with a small loss in objective value. Here we restate the claims regarding robustness for the convenience of the reader.

Definition 12.3.1. An SDP solution $\{\boldsymbol{v}_{i,a}\}_{i\in\mathcal{V},a\in\mathbb{F}_q}$, $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ is said to be *ε-infeasible for* LH_r *(or* SA_r*)* if it satisfies all the constraints of the program up to an additive error of ε.

Theorem 12.7. *Given an ε-infeasible solution $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in\mathbb{F}_q}$, $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ to the LH_r relaxation, there exists a feasible solution $\{\boldsymbol{b}'_{i,a}\},\{\mu'_S\}_{S\subset\mathcal{V},|S|\leqslant r}$ for LH_r such that for all subsets $S\subseteq\mathcal{V},|S|\leqslant r$, $\|\mu_S-\mu'_S\|_1\leqslant \mathrm{poly}(q)\cdot r^2\varepsilon$.*

Theorem 12.8. *Given an ε-infeasible solution $\{\boldsymbol{b}_{i,a}\}_{i\in\mathcal{V},a\in\mathbb{F}_q}$, $\{\mu_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ to the SA_r relaxation, there exists a feasible solution $\{\boldsymbol{b}'_{i,a}\},\{\mu'_S\}_{S\subseteq\mathcal{V},|S|\leqslant r}$ for SA_r such that for all subsets $S\subseteq\mathcal{V},|S|\leqslant r$, $\|\mu_S-\mu'_S\|_1\leqslant \mathrm{poly}(q)\cdot\varepsilon\cdot q^r$.*

12.4 Weak Gaps for Unique Games

We refer to an integrality gap instance for a fairly simple SDP relaxation of Unique Games as a *weak gap* instance. Formally, a *weak gap* instance for Unique games is defined as follows.

Definition 12.4.1. *(Weak SDP solutions and weak gap instances)* Let $\Upsilon=(V,E,\{\pi_e\colon[n]\to[n]\}_{e\in E})$. We say a collection $\mathcal{B}=\{B_u\}_{u\in V}$ is a *weak SDP solution of value* $1-\eta$ for Υ if the following conditions hold:

1. (**Orthonormality**) For every vertex $u\in V$, the collection $\mathcal{B}$ contains an ordered set $B_u=\{b_{u,1},\ldots,b_{u,n}\}$ of n orthonormal vectors in $\mathbb{R}^d$.

2. (**ℓ_2^2-triangle inequality**) Any two vectors in $\bigcup\mathcal{B}$ have non-negative inner product and any three vectors in $\bigcup\mathcal{B}$ satisfy the ℓ_2^2-triangle inequality ($\|x-y\|^2\leqslant\|x-z\|^2+\|z-y\|^2$).

3. (**Strong Matching Property**) For every pair of vertices $u,v\in V$, the sets B_u and B_v satisfy the following *strong matching property*: There exists n disjoint matchings between B_u,B_v given by bijections $\pi^{(1)},\ldots,\pi^{(n)}\colon B_u\to B_v$ such that for all $i\in[n],b,b'\in B_u$, we have $\langle b,\pi^{(i)}(b)\rangle=\langle b',\pi^{(i)}(b')\rangle$.

4. (**High SDP value**) For every edge $e=(u,v)\in E$, the vector sets B_u and B_v have significant correlation under the permutation $\pi=\pi_e$. Specifically, $\langle b_{u,\ell},b_{v,\pi(\ell)}\rangle^2\geqslant 0.99$ for all $\ell\in[n]$.

5. The collection $\mathcal{B}$ of orthonormal sets is a good SDP solution for Υ, in the sense that

$$\mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{\substack{w,w'\in N(v)\\ \pi=\pi_{w,v},\ \pi'=\pi_{w',v}}} \frac{1}{n}\sum_{\ell\in[n]} \langle b_{w,\pi(\ell)}, b_{w',\pi'(\ell)}\rangle \geqslant 1-\eta.$$

We say that Υ is a *weak* $(1-\eta,\delta)$*-gap instance* of Unique Games if Υ has a weak SDP solution of value $1-\eta$ and no labeling for Υ satisfies more than a δ fraction of the constraints.

Remark 12.4.1. The weak gap instances defined here are fairly natural objects. In fact, if $\Im$ is an instance of Γ-Max-2Lin(R) with $\mathrm{sdp}(\Im) \geqslant 1-\eta$ and $\mathrm{opt}(\Im) \leqslant \delta$, it is easy to construct a corresponding weak gap instance $\Im'$. The idea is to start with an optimal SDP solution for $\Im$, symmetrize it (with respect to the group Φ), and delete all edges of $\Im$ that contribute less than $\sqrt{3/4}$ to the SDP objective.

We observe the following consequence of Fact 12.4.1 and item 4 of Definition 12.4.1.

Observation 12.4.1. *If $\mathcal{B} = \{B_u\}_{u\in V}$ is a weak SDP solution for $\Phi = (V, E, \{\pi_e\}_{e\in E})$, then for any two edges $(w,v),(w',v) \in E$, the two bijections $\pi = \pi_{(w',v)}^{-1}\circ\pi_{(w,v)}$ and $\pi_{B_{w'}\leftarrow B_w}$ (see Def. 12.4.2) give rise to the same matching between the vector sets B_w and $B_{w'}$,*

$$\pi(i) = j \quad\Longleftrightarrow\quad \pi_{B_{w'}\leftarrow B_w}(b_{w,i}) = b_{w',j}\,.$$

The previous observation implies that in a weak gap instance Φ the collection of permutations $\{\pi_e\}_{e\in E}$ is already determined by the geometry of the vector sets in a weak SDP solution $\mathcal{B}$.

12.4.1 Constructing Weak Gap Instances

There are a few explicit constructions of weak gap instances of Unique Games, most prominently the Khot–Vishnoi instance [104]. In particular, the following observation is a restatement of Theorem 9.2 and Theorem 9.3 in [104].

Observation 12.4.2. *For all $\eta,\delta > 0$, there exists a weak $(1-\eta,\delta)$-gap instance with $2^{2^{O(\log(1/\delta)/\eta)}}$ vertices.*

Here we sketch the construction of an integrality gap for a simple SDP relaxation of Unique Games. The instance we present does not have all the properties required in Definition 12.4.1. However, the construction is extremely simple and intuitive, and with a little more effort modified into a *weak gap* instance.

To begin with, we describe a UG instance with infinitely many vertices. Using standard techniques, it can be discretized to obtain finite instances with the same SDP integrality gap. Recall that $\mathcal{G}^d$ denotes the d-dimensional Gaussian space. It is most natural to describe our UG instance as a 2-player game. The details of the game are described in Figure 12.1. Recall that $\blacktriangle_R$ denote the R-dimensional simplex. Let $\boldsymbol{\mathcal{F}} : (\mathcal{G}^d)^R \to \blacktriangle_R$ corresponding to the strategy $\mathcal{F}_1$ as follows:

$$\boldsymbol{\mathcal{F}}(\boldsymbol{Q}) = \boldsymbol{e}_{\mathsf{A}(\boldsymbol{Q})}$$

A Simple Unique Games Integrality Gap

– The verifier samples $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_R)$ where each $\boldsymbol{g}_i = \frac{1}{\sqrt{d}}\boldsymbol{g}'_i$ for a sample $\boldsymbol{g}'_i$ drawn independently from $\mathcal{G}^d$. Define $\boldsymbol{h} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_R)$ as follows:

$$\boldsymbol{h}_i = (1-\varepsilon)\boldsymbol{g}_i + \sqrt{\varepsilon}\boldsymbol{\zeta}_i$$

where $\boldsymbol{\zeta}_i = \frac{1}{\sqrt{d}}\boldsymbol{\zeta}'_i$ for an i.i.d sample $\boldsymbol{\zeta}'_i$ from $\mathcal{G}^d$.

– Generate two random permutations $\pi_A, \pi_B : [R] \to [R]$. The questions $\boldsymbol{Q}_A, \boldsymbol{Q}_B \in (\mathcal{G}^d)^R$ to the two provers A, B are defined by:

$$\begin{aligned} \boldsymbol{Q}_A &= (\boldsymbol{g}_{\pi_A(1)}, \boldsymbol{g}_{\pi_A(2)}, \ldots, \boldsymbol{g}_{\pi_A(R)}) \\ \boldsymbol{Q}_B &= (\boldsymbol{h}_{\pi_B(1)}, \boldsymbol{h}_{\pi_B(2)}, \ldots, \boldsymbol{h}_{\pi_B(R)}) \end{aligned}$$

– The provers choose one of the R vectors in $\mathcal{G}^d$ presented to them. The provers win, if they choose the corresponding vectors $\boldsymbol{g}_i, \boldsymbol{h}_i$ for some i. Formally, let $\mathsf{A}, \mathsf{B} : (\mathcal{G}^d)^R \to [R]$ denote the strategies of the two provers. Provers win if,

$$\pi_A^{-1}(\mathsf{A}(\boldsymbol{Q}_A)) = \pi_B^{-1}(\mathsf{B}(\boldsymbol{Q}_B))$$

Figure 12.1: A Simple Unique Games Integrality Gap

Similarly, define the function $\boldsymbol{\mathcal{F}}'$ corresponding to the strategy B. Note that each of the functions $\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{F}}'$ consist of R real valued functions. Specifically, let $\boldsymbol{\mathcal{F}} = (\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_R)$ and $\boldsymbol{\mathcal{F}}' = (\mathcal{F}'_1, \mathcal{F}'_2, \ldots, \mathcal{F}'_R)$.

Arithmetizing the probability of success, we get:

$$\mathbb{P}\mathrm{r}[\text{Success}] = \Pr_{\boldsymbol{g},\boldsymbol{h},\pi_A,\pi_B}\left[\pi_A^{-1}(\mathsf{A}(\boldsymbol{Q}_A)) = \pi_B^{-1}(\mathsf{B}(\boldsymbol{Q}_B))\right] = \mathop{\mathbb{E}}_{\substack{\boldsymbol{g},\boldsymbol{h},\\ \pi_A,\pi_B}}\left[\sum_{i=1}^{R} \mathcal{F}_{\pi_A(i)}(\boldsymbol{Q}_A)\mathcal{F}'_{\pi_B(i)}(\boldsymbol{Q}_B)\right] \tag{12.6}$$

For a permutation $\pi : [R] \to [R]$, and a vector $\boldsymbol{g} = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_R)$ in $(\mathcal{G}^d)^R$, define $\pi(\boldsymbol{g}) = (\boldsymbol{g}_{\pi(1)}, \ldots, \boldsymbol{g}_{\pi(R)})$. With this notation, we can rewrite

$$\mathbb{P}\mathrm{r}[\text{Success}] \quad = \quad \sum_{i=1}^{R} \mathop{\mathbb{E}}_{\boldsymbol{g},\boldsymbol{h},\pi_A,\pi_B}\left[\mathcal{F}_{\pi_A(i)}(\pi_A(\boldsymbol{g}))\mathcal{F}'_{\pi_B(i)}(\pi_B(\boldsymbol{h}))\right] \tag{12.7}$$

Define functions $a_i, b_i : (\mathcal{G}^d)^R \to [0,1]$ for $i \in [R]$ as follows:

$$a_i(\boldsymbol{g}) = \mathop{\mathbb{E}}_{\pi}\left[\mathcal{F}_{\pi(i)}(\pi(\boldsymbol{g}))\right] \qquad b_i(\boldsymbol{h}) = \mathop{\mathbb{E}}_{\pi}\left[\mathcal{F}'_{\pi(i)}(\pi(\boldsymbol{h}))\right]$$

where the expectation is over a uniform choice of the permutation π. Further, observe that

$$\mathop{\mathbb{E}}_{\boldsymbol{g}}\left[a_i(\boldsymbol{g})\right] = \mathop{\mathbb{E}}_{\pi}\left[\mathop{\mathbb{E}}_{\boldsymbol{g}}\left[\mathcal{F}_{\pi(i)}(\pi(\boldsymbol{g}))\right]\right] = \mathop{\mathbb{E}}_{\pi}\left[\mathop{\mathbb{E}}_{\boldsymbol{g}}\left[\mathcal{F}_{\pi(i)}(\boldsymbol{g})\right]\right] = \mathop{\mathbb{E}}_{\boldsymbol{g}}\left[\mathop{\mathbb{E}}_{\pi}\left[\mathcal{F}_{\pi(i)}(\boldsymbol{g})\right]\right] = \mathop{\mathbb{E}}_{\boldsymbol{g}}\left[\frac{1}{R}\right] = \frac{1}{R} \tag{12.8}$$

In the above computation, we used the fact that for a fixed permutation π, $\boldsymbol{g}$ and $\pi(\boldsymbol{g})$ are identically distributed. Now we rewrite equation 12.7 in terms of functions a_i, b_i to get:

$$\mathbb{P}\mathrm{r}[\text{Success}] \quad = \quad \sum_{i=1}^{R} \mathop{\mathbb{E}}_{\boldsymbol{g},\boldsymbol{h}}\left[a_i(\boldsymbol{g})b_i(\boldsymbol{h})\right] \tag{12.9}$$

By construction, the vectors $\boldsymbol{g}, \boldsymbol{h}$ are $1-\varepsilon$ correlated random Gaussian vectors of dimension dR. Hence the above expression is equal to $\sum_{i=1}^{R}\langle a_i, U_{1-\varepsilon}b_i\rangle$ where $U_{1-\varepsilon}$ is the Ornstein-Uhlenbeck operator (see Section 3.7). Using the Gaussian isoperimetric theorem of Borell (Theorem 3.4) we obtain an upper bound of $\Gamma_{1-\varepsilon}(1/R)$ on the probability of success for any strategy of the provers. Using estimates for Gaussian noise stability $\Gamma_{1-\varepsilon}$ [125], this quantity is roughly equal to $O\left(1/R^{\frac{\varepsilon}{2-\varepsilon}}\right)$.

Now we shall construct an SDP vector solution with value $1 - O(\varepsilon)$. Firstly, observe that the inner product of two random Gaussian vectors is concentrated at around $O(1/\sqrt{d})$. In particular, for large enough d, with very probability close to 1, the vectors

$\{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_R, \boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_R\}$ satisfy:

$$\begin{aligned} |\langle \boldsymbol{g}_i, \boldsymbol{g}_j \rangle| \leqslant \varepsilon/R^3 \qquad |\langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle| \leqslant \varepsilon/R^3 \qquad \forall i \neq j \in [R] \\ 1 - \varepsilon/R^3 \leqslant \langle \boldsymbol{g}_i, \boldsymbol{g}_i \rangle, \langle \boldsymbol{h}_i, \boldsymbol{h}_i \rangle \leqslant 1 + \varepsilon/R^3 \leqslant \varepsilon/R^3 \qquad \forall i \in [R] \\ \sum_i \langle \boldsymbol{g}_i, \boldsymbol{h}_i \rangle \geqslant R(1-\varepsilon) \end{aligned}$$

Thus for most questions $\boldsymbol{Q}_A, \boldsymbol{Q}_B$ to the two provers, the R vectors part of the question form a good candidate for SDP vectors. The SDP solution we will use is obtained by performing a surgery operation on these candidate solutions. In fact, all we need to do is to perform a Gram-Schmidt orthogonalization and normalization for the set of vectors $(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_R)$ to obtain an orthonormal set of R vectors. Formally, the vectors $\{\boldsymbol{b}_{\boldsymbol{Q}_A,i}\}$ associated with the question $\boldsymbol{Q}_A$ are:

$$\boldsymbol{b}_{\boldsymbol{Q}_A,i} = \frac{1}{\sqrt{R}} \boldsymbol{G}_{\pi_A(i)}$$

where the vectors $(\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_R)$ are obtained by orthonormalizing the set of vectors $(\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_R)$. Similarly, the vectors for $\boldsymbol{Q}_B$ are obtained starting from vectors $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_R)$. As the vectors are nearly orthonormal to begin with, this surgery does not affect the inner products between $\{\boldsymbol{g}_i\}$ and $\{\boldsymbol{h}_j\}$ by more than ε. Thus the objective value of the SDP is at least $1 - O(\varepsilon)$.

We stress here that the SDP gap instance obtained is for the simplest semidefinite program for Unique Games. To make the instance satisfy additional constraints, it could be necessary to perform more complex operations like tensoring of the vectors.

12.4.2 Properties of Weak Gap Instances

Observation 12.6.4 implies that without much loss we can assume that a weak SDP solution is $\mathbb{F}_q$-*integral*, that is, all vectors are $\mathbb{F}_q$-integral. Here we use again $\langle \cdot, \cdot \rangle_\psi := \langle \psi(\cdot), \psi(\cdot) \rangle$ as inner product for $\mathbb{F}_q$-integral vectors.

Lemma 12.8.1. *Let $\Phi = (V, E, \{\pi_e\}_{e \in E})$ be a weak $(1-\eta, \delta)$-gap instance. Then, for every $q \in \mathbb{N}$, we can find a weak $\mathbb{F}_q$-integral SDP solution of value $1 - O(\sqrt{\eta \log q})$ for a* Unique Games *instance Φ' which is obtained from Φ by deleting $O(\sqrt{\eta \log q})$ edges.*

Proof. Let $\mathcal{B}$ be a weak SDP solution for Φ of value $1 - \eta$. By applying the transformation from Observation 12.6.4 to the vectors in $\mathcal{B}$, we obtain a collection $\mathcal{B}' = \{B'_u\}_{u \in V}$ of sets of $\mathbb{F}_q$-integral vectors. For every $u \in V$, the vectors in B'_u are orthonormal. Furthermore, any two sets B'_u, B'_v in $\mathcal{B}'$ satisfy the strong matching property (using the facts that the original sets B_u, B_v satisfy this property and that $\langle b'_{u,i}, b'_{v,j} \rangle_\psi$ is a function of $\langle b_{u,i}, b_{v,j} \rangle$).

Let $\eta_{v,w,w',\ell} = 1 - \langle b_{w,\pi(\ell)}, b_{w',\pi'(\ell)} \rangle$. Using Jensen's inequality, we can verify that the

value of the SDP solution $\mathcal{B}'$ is high,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{\substack{w,w'\in N(v)\\ \pi=\pi_{w,v},\ \pi'=\pi_{w',v}}} \frac{1}{R}\sum_{\ell\in[R]} \langle b'_{w,\pi(\ell)}, b'_{w',\pi'(\ell)}\rangle_\psi \\
&\quad\geqslant \mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{\substack{w,w'\in N(v)\\ \pi=\pi_{w,v},\ \pi'=\pi_{w',v}}} \frac{1}{R}\sum_{\ell\in[R]} 1 - O\left(\sqrt{\eta_{v,w,w',\ell}\log q}\right) \qquad \text{(by Obs. 12.6.4)} \\
&\quad\geqslant 1 - O(\sqrt{\eta\log q}) \qquad \text{(using Jensen's inequality)}\,.
\end{aligned}$$

So far, we verified that $\mathcal{B}'$ satisfies all requirements of a weak SDP solution besides item 4 of Definition 12.4.1. We can ensure that this condition is also satisfied by deleting all edges from E where the condition is violated. Using standard averaging arguments, it is easy to see that the matching property and the high SDP value imply that this condition is satisfied for all but at most an $O(\sqrt{\eta\log q})$ fraction of edges. ■

We will refer to the set of orthonormal vectors associated with a vertex B as a *cloud*. In what follows, we identify the vertices B in a *weak* gap instance with their corresponding clouds, and thus refer to vertices/clouds interchangeably.

Definition 12.4.2. For $A, B \in \mathcal{B}$, we denote

$$\rho(A,B) \stackrel{\text{def}}{=} \max_{a\in A, b\in B} |\langle a,b\rangle|\,.$$

We define $\pi_{B\leftarrow A}\colon A\to B$ to be any[1] bijection from A to B such that $|\langle a,\pi_{B\leftarrow A}(a)\rangle| = \rho(A,B)$ for all $a\in A$.

As a direct consequence of the orthogonality of the clouds in $\mathcal{B}$, we have the following fact about the uniqueness of $\pi_{B\leftarrow A}$ for highly correlated clouds $A, B\in\mathcal{B}$.

Fact 12.4.1. *Let $A,B\in\mathcal{B}$. If $\rho(A,B)^2 > 3/4$, then there exists exactly one bijection $\pi\colon A\to B$ such that $|\langle a,\pi(a)\rangle| = \rho(A,B)$ for all $a\in A$.*

Remark 12.4.2. The collection $\mathcal{B}$ succinctly encodes a Unique Games instance. For a graph $G=(\mathcal{B},E)$ on $\mathcal{B}$, the goal is to find a labeling $\{\ell_A\in A\}_{A\in\mathcal{B}}$ (a labeling can be seen as a system of representatives for the clouds in $\mathcal{B}$) so as to maximize the probability

$$\Pr_{(A,B)\in E}\left\{\ell_A = \pi_{A\leftarrow B}(\ell_B)\right\}\,.$$

Tensoring

Lemma 12.8.2. *For $t\in\mathbb{N}$ and every pair of clouds $A,B\in\mathcal{B}$,*

$$\frac{1}{n}\sum_{\substack{a\in A, b\in B\\ a\neq\pi_{A\leftarrow B}(b)}} |\langle a,b\rangle|^t \leqslant 2\cdot(3/4)^{t/2}\,.$$

[1]The matching property asserts that such a matching exists. If it is not unique, we pick an arbitrary one. We will assume $\pi_{A\to B} = \pi_{B\to A}^{-1}$.

Proof. By near-orthogonality, $\sum_{a\in B}\langle a,b\rangle^2 \leqslant 3/2$ for every $b \in B$. Hence, $\langle a,b\rangle^2 \leqslant 3/4$ for all $a \neq \pi_{A\leftarrow B}(b)$. Thus,

$$\frac{1}{n}\sum_{\substack{a\in A,b\in B\\ a\neq\pi_{A\leftarrow B}(b)}} |\langle a,b\rangle|^t \leqslant (3/4)^{\frac{t-2}{2}}\cdot\frac{1}{n}\sum_{a\in A,b\in B}|\langle a,b\rangle|^2 \leqslant (3/4)^{\frac{t-2}{2}}\cdot 3/2\,.$$

■

The notation $X = Y \pm Z$ means that $|X - Y| \leqslant Z$.

Corollary 12.8.1. *For $t \in \mathbb{N}$ and every pair of clouds $A, B \in \mathcal{B}$,*

$$\frac{1}{n}\sum_{a\in A,b\in B}\langle a,b\rangle^t = \frac{1}{n}\sum_{a\in A}\langle a,\pi_{B\leftarrow A}(a)\rangle^t \quad \pm 2\cdot(3/4)^{t/2}\,.$$

Remark 12.4.3. The left-hand side in the corollary is the inner product of the vectors $1/\sqrt{R}\sum_{u\in A}u^{\otimes t}$ and $1/\sqrt{R}\sum_{v\in B}v^{\otimes t}$. If t is even, then we can replace the right-hand side by $\rho(A,B)^t$. This fact that the functional $\rho(A,B)^t$ is closely approximated by inner products averaged-tensored vectors has implicitly been used in [104] and was explicitly noted in [14, Lemma 2.2].

12.5 Integrality Gap Instance for Unique Games

In this section, we will exhibit the construction strong SDP integrality gap for the $\mathsf{E2Lin}_q$ problem. Recall that the $\mathsf{E2Lin}_q$ problem is a special case of Unique Games. To this end, we follow the approach of Khot–Vishnoi [104] to construct the gap instance.

Khot et al. [99] show a UGC-based hardness result for the $\mathsf{E2Lin}_q$ problem. Specifically, they exhibit a reduction $\Phi_{\gamma,q}$ that maps a Unique Games instance Φ to an $\mathsf{E2Lin}_q$ instance $\Phi_{\gamma,q}(\Phi)$ such that the following holds: For every $\gamma > 0$ and all $q \geqslant q_0(\gamma)$,

- Completeness: If Φ is $1-\eta$-satisfiable then $\Phi_{\gamma,q}(\Phi)$ is $1-\gamma-o_{\eta,\delta}(1)$ satisfiable.
- Soundness: If Φ has no labeling satisfying more than δ-fraction of the constraints, then no assignment satisfies more than $q^{-\eta/2}+o_{\eta,\delta}(1)$-fraction of equations in $\Phi_{\gamma,q}(\Phi)$.

Here the notation $o_{\eta,\delta}(1)$ refers to any function that tends to 0 whenever η and δ go to naught. The details of the $\Phi_{\gamma,q}$ reduction are included in Figure 12.2 for the sake of completeness.

The rest of the chapter is devoted to the proof of the following theorem.

Theorem 12.9. *Let Φ be a weak $(1-\eta,\delta)$-gap instance of* Unique Games. *Then, for every q of order unity, there exists an SDP solution for the $\mathsf{E2Lin}_q$ instance $\Phi_{\gamma,q}(\Phi)$ such that*

- *the SDP solution is feasible for LH_r with $r = 2^{\Omega(1/\eta^{1/4})}$,*

$\mathsf{E2Lin}_q$ Hardness Reduction [99]

Input A Unique Games instance Φ with vertex set V, edge set $E \subseteq V \times V$ (we assume the graph (V, E) to be regular), and permutations $\{\pi_e \colon [R] \to [R]\}_{e \in E}$.

Output An $\mathsf{E2Lin}_q$ instance $\Phi_{\gamma,q}(\Phi)$ with vertex set $\mathcal{V} = V \times \mathbb{F}_q^R$. Let $\{\mathcal{F}_v \colon \mathbb{F}_q^R \to \mathbb{F}_q\}_{v \in V}$ denote an $\mathbb{F}_q$-assignment to $\mathcal{V}$. The constraints of $\Phi_{\gamma,q}(\Phi)$ are given by the tests performed by the following probabilistic verifier:

- Pick a random vertex $v \in V$. Choose two random neighbours $w, w' \in N(v) \subseteq V$. Let π, π' denote the permutations on the edges (w, v) and (w', v).
- Sample $x \in \mathbb{F}_q^R$ uniformly at random. Generate $y \in \mathbb{F}_q^R$ as follows:

$$y_i = \begin{cases} x_i \text{ with probability } 1 - \gamma \\ \text{uniform random element from } \mathbb{F}_q \text{ with probability } \gamma \end{cases}$$

- Generate a uniform random element $c \in \mathbb{F}_q$.
- Test if $\mathcal{F}_w(y \circ \pi + c \cdot \mathbb{1}) = \mathcal{F}_{w'}(x \circ \pi') + c$. (Here, $x \circ \pi$ denotes the vector $(x_{\pi(i)})_{i \in [R]}$.)

Figure 12.2: Reduction from Unique Games to $\mathsf{E2Lin}_q$

– *the SDP solution is feasible for* SA_r *with* $r = \Omega(\eta^{1/4})$,

– *the SDP solution has value* $1 - \gamma - o_{\eta,\delta}(1)$ *for* $\Phi_{\gamma,q}(\Phi)$.

In particular, the $\mathsf{E2Lin}_q$ *instance* $\Phi_{\gamma,q}(\Phi)$ *is a* $(1 - \gamma - o_{\eta,\delta}(1), q^{-\eta/2} + o_{\eta,\delta}(1)))$ *integrality gap instance for the relaxation* LH_r *for* $r = 2^{\Omega(1/\eta^{1/4})}$. *Further,* $\Phi_{\gamma,q}(\Phi)$ *is a* $(1 - \gamma - o_{\eta,\delta}(1), q^{-\eta/2} + o_{\eta,\delta}(1))$ *integrality gap instance for the relaxation* SA_r *for* $r = \Omega(1/\eta^{1/4})$.

12.6 Integral Vectors

In this section, we will develop tools to create and manipulate vectors all of whose coordinates are "integral".

$\{\pm 1\}$-integral vectors We begin by defining our notion of a $\{\pm 1\}$-integral vector.

Definition 12.6.1. Let $\mathcal{R} = (\Omega, \mu)$ be a probability space. A function $u\colon \mathcal{R} \to \{\pm 1\}$ is called an $\{\pm 1\}$-*integral vector*. In other words, u is a $\{\pm 1\}$-valued random variable defined on the probability space $\mathcal{R}$. We define an inner product of functions $u, v\colon \mathcal{R} \to \{\pm 1\}$ by

$$\langle u, v\rangle = \mathop{\mathbb{E}}_{r\sim\mathcal{R}} u(r)v(r)\,.$$

In our construction, we often start with $\{\pm 1\}$-integral vectors given by the hypercube $\{\pm 1\}^R$. In the terminology of $\{\pm 1\}$-integral vectors, we can think of the hypercube $\{\pm 1\}^R$ as the set of $\{\pm 1\}$-integral vectors where $\mathcal{R}$ is the uniform distribution over $\{1, \ldots, R\}$.

The following lemma shows how the Goemans–Williamson [65] rounding scheme can be thought of as a procedure to "round" arbitrary real vectors to $\{\pm 1\}$-integral vectors.

Observation 12.6.1. *Given a family of unit vectors* $\{v_1, \ldots, v_R\} \in \mathbb{R}^d$, *define the set of* $\{\pm 1\}$-*valued functions* $v_1^*, \ldots, v_R^* : \mathcal{R} \to \{\pm 1\}$ *with* $\mathcal{R} = \mathcal{G}^d$ *- the Gaussian space of appropriate dimension as follows:*

$$v_i^*(g) = \mathsf{sign}(\langle v_i, g\rangle)$$

for $g \in \mathcal{G}^d$. *The* $\{\pm 1\}$-*valued functions* $\{v_i^*\}$ *satisfy* $\langle v_1^*, v_2^*\rangle = 2\arccos(\langle v_1, v_2\rangle)/\pi$. *Specifically, this operation obeys the following properties:*

$$\langle u, v\rangle = 0 \Longleftrightarrow \langle u^*, v^*\rangle = 0 \qquad \langle u, v\rangle = 1 - \varepsilon \Longrightarrow \langle u^*, v^*\rangle \geqslant 1 - O(\sqrt{\varepsilon})$$

The tensor product operation on $\{\pm 1\}$-integral vectors, yields a $\{\pm 1\}$-integral vector.

Definition 12.6.2. Given two $\{\pm 1\}$-valued functions $u : \mathcal{R}_1 \to \{\pm 1\}$ and $v : \mathcal{R}_2 \to \{\pm 1\}$, the tensor product $u \otimes v : \mathcal{R}_1 \times \mathcal{R}_2 \to \{\pm 1\}$ is defined as $u \otimes v(r_1, r_2) = u(r_1)v(r_2)$.

Observation 12.6.2. *For* $u, u'\colon \mathcal{R}_1 \to \{\pm 1\}$ *and* $v, v'\colon \mathcal{R}_2 \to \{\pm 1\}$, *we have*

$$\begin{aligned}\langle u \otimes v, u' \otimes v'\rangle &= \mathop{\mathbb{E}}_{r_1,r_2}[u \otimes v(r_1, r_2)u' \otimes v'(r_1, r_2)] \\ &= \mathop{\mathbb{E}}_{r_1}[u(r_1)u'(r_1)] \mathop{\mathbb{E}}_{r_2}[v(r_2)v'(r_2)] = \langle u, u'\rangle\langle v, v'\rangle\end{aligned}$$

$\mathbb{F}_q$-integral vectors Let q be a prime. Now, we will define $\mathbb{F}_q$-integral vectors and their tensor products.

Definition 12.6.3. A $\mathbb{F}_q$-integral vector $v\colon \mathcal{R} \to \mathbb{F}_q$ is a function from a measure space $\mathcal{R}$ to $\mathbb{F}_q$. For a $\mathbb{F}_q$-integral vector $v\colon \mathcal{R} \to \mathbb{F}_q$, its symmetrization $\tilde{v}\colon \mathcal{R} \times \mathbb{F}_q^* \to \mathbb{F}_q$ is defined by $\tilde{v}(r,t) = t \cdot v(r)$.

Given a map $f\colon \mathbb{F}_q \to \mathbb{C}^d$, we denote by $f(v) := f \circ v$ the composition of functions f and v. Here are few examples of functions that will be relevant to us:

1. The function $\chi\colon \mathbb{F}_q \to \mathbb{C}^{q-1}$ given by

$$\chi(i) \stackrel{\text{def}}{=} \tfrac{1}{\sqrt{q-1}}(\omega^{1\cdot i}, \ldots, \omega^{j\cdot i}, \ldots, \omega^{(q-1)i})\,,$$

   where ω is a primitive q^{th} root of unity. The vector $\chi(i) \in \mathbb{C}^{q-1}$ is the restriction of the i^{th} character function of the group $\mathbb{Z}_q$ to the set $\mathbb{F}_q^*$. It is easy to see that

$$\langle \chi(a), \chi(b) \rangle = \mathop{\mathbb{E}}_{t \in \mathbb{F}_q^*} \left[\omega^{ta} \cdot \omega^{-tb}\right] = \begin{cases} 1 & \text{if } a = b\,, \\ -\frac{1}{q-1} & \text{if } a \neq b\,. \end{cases}$$

2. Let $\psi_0, \psi_1, \ldots, \psi_{q-1}$ denote the corners of the q-ary simplex in $\mathbb{R}^{q-1}$, translated so that the origin is its geometric center. Define the function $\psi\colon \mathbb{F}_q \to \mathbb{R}^{q-1}$ as $\psi(i) := \psi_i$. Again, the vectors satisfy

$$\langle \psi(a), \psi(b) \rangle = \begin{cases} 1 & \text{if } a = b\,, \\ -\frac{1}{q-1} & \text{if } a \neq b\,. \end{cases}$$

Remark 12.6.1. A $\mathbb{F}_q$-integral vector $v \in \mathbb{F}_q^N$ can be thought of as a $\mathbb{F}_q$-valued function over the measure space $([N], \mu)$ where μ is the uniform distribution over $[N]$.

Remark 12.6.2. The following notions are equivalent: Collection of $\mathbb{F}_q$-valued functions on some measure space $\mathcal{R} \iff$ Collection of jointly-distributed, $\mathbb{F}_q$-valued random variables $\iff$ Distribution over $\mathbb{F}_q$-assignments.

For the case of $\mathbb{F}_q$-integral vector, the tensor product operation is to be defined carefully, in order to mimic the properties of the traditional tensor product. We will use the following definition for the tensor operation $\otimes_q$.

Definition 12.6.4. Given two $\mathbb{F}_q$-valued functions $u\colon \mathcal{R} \to \mathbb{F}_q$ and $u'\colon \mathcal{R}' \to \mathbb{F}_q$, define the symmetrized tensor product $u \otimes_q u'\colon (\mathcal{R} \times \mathbb{F}_q^*) \times (\mathcal{R}' \times \mathbb{F}_q^*) \to \mathbb{F}_q$ as

$$(u \otimes_q u')\left(r, t, r', t'\right) \stackrel{\text{def}}{=} t \cdot u(r) + t' \cdot u'(r')\,.$$

Lemma 12.9.1. *For any $\mathbb{F}_q$-valued functions $u, v\colon \mathcal{R} \to \mathbb{F}_q$ and $u', v'\colon \mathcal{R}' \to \mathbb{F}_q$,*

$$\langle \psi(u \otimes_q u'), \psi(v \otimes_q v') \rangle = \langle \psi(u), \psi(v) \rangle \langle \psi(u'), \psi(v') \rangle\,.$$

Proof.

$$\begin{aligned}
&\langle \psi(u \otimes_q u'), \psi(v' \otimes_q v') \rangle \\
&= \langle \chi(u \otimes_q u'), \chi(v' \otimes_q v') \rangle \qquad (\text{using } \langle \psi_a, \psi_b \rangle = \langle \chi(a), \chi(b) \rangle) \\
&= \mathop{\mathbb{E}}_{(r,t)} \mathop{\mathbb{E}}_{(r',t')} \mathop{\mathbb{E}}_{\ell \in \mathbb{F}_q^*} \omega^{\ell t u(r) + \ell t' u'(r')} \cdot \omega^{-\ell t v(r) - \ell t' v'(r')} \qquad (\text{by definitions of } \otimes_q \text{ and } \chi) \\
&= \mathop{\mathbb{E}}_{\ell \in \mathbb{F}_q^*} \left(\mathop{\mathbb{E}}_{(r,t)} \omega^{\ell t u(r) - \ell t v(r)} \right) \cdot \left(\mathop{\mathbb{E}}_{(r',t')} \omega^{\ell t' u'(r') - \ell t' v'(r')} \right) \\
&= \mathop{\mathbb{E}}_{\ell \in \mathbb{F}_q^*} \left(\mathop{\mathbb{E}}_{r} \langle \chi(\ell u(r)), \chi(\ell v(r)) \rangle \right) \cdot \left(\mathop{\mathbb{E}}_{r'} \langle \chi(\ell u'(r')), \chi(\ell v'(r')) \rangle \right) \\
&= \mathop{\mathbb{E}}_{\ell \in \mathbb{F}_q^*} \langle \chi(\ell u), \chi(\ell v) \rangle \langle \chi(\ell u'), \chi(\ell v') \rangle \\
&= \langle \chi(u), \chi(v) \rangle \langle \chi(u'), \chi(v') \rangle \qquad (\text{using } \langle \chi(\ell a), \chi(\ell b) \rangle = \langle \chi(a), \chi(b) \rangle \text{ for } \ell \in \mathbb{F}_q^*) \\
&= \langle \psi(u), \psi(v) \rangle \langle \psi(u'), \psi(v') \rangle \qquad (\text{using } \langle \psi_a, \psi_b \rangle = \langle \chi(a), \chi(b) \rangle)
\end{aligned}$$

■

Remark 12.6.3. Unlike the ordinary tensor operation, the q-ary tensor operation we defined is not associative. Formally, we define the tensoring operation to be right-associative

$$u_1 \otimes_q u_2 \otimes_q \ldots \otimes_q u_{k-1} \otimes_q u_k \stackrel{\text{def}}{=} u_1 \otimes_q \left(u_2 \otimes_q \left(\ldots (u_{k-1} \otimes_q u_k) \cdots \right) \right).$$

The lack of associativity will never be an issue in our constructions.

We need the following simple technical observation in one of our proofs.

Observation 12.6.3. *Let $u, v \colon \mathcal{R} \to \mathbb{F}_q$ be two "symmetric" $\mathbb{F}_q$-integral vectors. that is, $\mathbb{P}\mathrm{r}_r\{u(r) - v(r) = a\} = \mathbb{P}\mathrm{r}_r\{u(r) - v(r) = b\}$ for all $a, b \in \mathbb{F}_q^*$. Then, for all $a, b \in \mathbb{F}_q$, we have $\mathbb{E}_r \langle \psi(a + u(r)), \psi(b + v(r)) \rangle = \langle a \otimes u, b \otimes v \rangle$.*

We wish to point out that in our applications, the vectors u and v will be tensor powers. In this case, the symmetry condition is always satisfied.

Proof. Using the symmetry assumption, we see that

$$\begin{aligned}
&\mathop{\mathbb{P}\mathrm{r}}_{r \sim \mathcal{R}, t, t' \in \mathbb{F}_q^*} \left\{ ta + t'u(r) = tb + t'v(r) \right\} \\
&\qquad = \mathop{\mathbb{P}\mathrm{r}}_{r \sim \mathcal{R}, t \in \mathbb{F}_q^*} \left\{ a - b = t \cdot \left(v(r) - u(r) \right) \right\} \\
&\qquad\qquad = \mathop{\mathbb{P}\mathrm{r}}_{r \sim \mathcal{R}} \left\{ a - b = v(r) - u(r) \right\} \qquad (12.10)
\end{aligned}$$

If we let ρ denote this probability, then we have $\langle a \otimes u, b \otimes v \rangle = \rho - (1 - \rho)/(q - 1)$ (using the left-hand side of Eq. (12.10) as well as $\mathbb{E}_r \langle \psi(a + u(r)), \psi(b + v(r)) \rangle = \rho - (1 - \rho)/(q - 1)$ (using the right-hand side of Eq. (12.10)). ■

The following procedure yields a way to generate $\mathbb{F}_q$-integral vectors from arbitrary vectors. The transformation is inspired by the rounding scheme for Unique Games in Charikar et al. [35].

Observation 12.6.4. *Define the function $\zeta\colon \mathcal{G}^q \to \mathbb{F}_q$ on the Gaussian domain as follows:*

$$\zeta(x_1, \ldots, x_q) = \mathrm{argmax}_{i\in[q]} x_i \tag{12.11}$$

Given a family of unit vectors $\{v_1, \ldots, v_R\} \in \mathbb{R}^d$, define the set of $\mathbb{F}_q$-valued functions $v_1^*, \ldots, v_R^*\colon \mathcal{R} \to \mathbb{F}_q$ with $\mathcal{R} = (\mathcal{G}^d)^q$ —the Gaussian space of appropriate dimension— as follows:

$$v_i^*(g_1, \ldots, g_q) = \zeta(\langle v_i, g_1\rangle, \ldots, \langle v_i, g_q\rangle)$$

for $g_1, \ldots, g_q \in (\mathcal{G}^d)^q$. The $\mathbb{F}_q$-valued functions $\{v_i^*\}$ satisfy,

1. $\langle u, v\rangle = 0 \Longrightarrow \langle \psi(u^*), \psi(v^*)\rangle = 0$,
2. $\langle u, v\rangle = 1 - \varepsilon \Longrightarrow \langle \psi(u^*), \psi(v^*)\rangle = 1 - f(\varepsilon, q) = 1 - O(\sqrt{\varepsilon \log q})$.

Proof. To see (1), observe that if $\langle u, v\rangle = 0$, then the sets of random variables $\{\langle u, g_1\rangle, \ldots, \langle u, g_q\rangle\}$ and $\{\langle v, g_1\rangle, \ldots, \langle v, g_q\rangle\}$ are completely independent of each other. Therefore,

$$\langle \psi(u^*), \psi(v^*)\rangle = \mathop{\mathbb{E}}_{r\in\mathcal{G}^{dq}}\Big[\psi(u^*(r))\Big] \cdot \mathop{\mathbb{E}}_{r\in\mathcal{G}^{dq}}\Big[\psi(u^*(r))\Big] = 0\,.$$

Assertion 2 follows from Lemma $C.8$ in [35]. ∎

12.7 *Local Distributions for* Unique Games

In this section, we will construct local distribution over labelings to a Unique Games instance.

The following facts are direct consequences of the (symmetrized) ℓ_2^2-triangle inequality.

Fact 12.7.1. *Let $a, b, c \in \bigcup \mathcal{B}$ with $|\langle a, b\rangle| = 1 - \eta_{ab}$ and $|\langle b, c\rangle| = 1 - \eta_{bc}$. Then, $|\langle a, c\rangle| \geqslant 1 - \eta_{ab} - \eta_{bc}$.*

Fact 12.7.2. *Let $A, B, C \in \mathcal{B}$ with $\rho(A, B) = 1 - \eta_{AB}$ and $\rho(B, C) = 1 - \eta_{BC}$. Then, $\rho(A, C) \geqslant 1 - \eta_{AB} - \eta_{BC}$.*

The construction in the proof of the next lemma is closely related to propagation-style UG algorithms [155, 14].

Definition 12.7.1. A set $\mathcal{S} \subseteq \mathcal{B}$ is *consistent* if

$$\forall A, B \in \mathcal{S}. \qquad \rho(A, B) \geqslant 1 - 1/16\,.$$

Lemma 12.9.2. *If $\mathcal{S} \subseteq \mathcal{B}$ is consistent, there exists bijections $\{\pi_A\colon [R] \to A\}_{A\in\mathcal{S}}$ such that*

$$\forall A, B \in \mathcal{S}. \qquad \pi_B = \pi_{B\leftarrow A} \circ \pi_A\,.$$

Proof. We can construct the bijections in a greedy fashion: Start with an arbitrary cloud $C \in \mathcal{S}$ and choose an arbitrary bijection $\pi_C \colon [R] \to C$. For all other clouds $B \in \mathcal{S}$, choose $\pi_B := \pi_{B \leftarrow C} \circ \pi_C$.

Let A, B be two arbitrary clouds in $\mathcal{S}$. Let $\sigma_{A \leftarrow B} := \pi_A \circ \pi_B^{-1}$. To prove the lemma, we have to verify that $\sigma_{A \leftarrow B} = \pi_{A \leftarrow B}$. By construction, $\sigma_{A \leftarrow B} = \pi_{A \leftarrow C} \circ \pi_{C \leftarrow B}$. Let $\eta = 1/16$. Since $\rho(A, C) \geqslant 1 - \eta$ and $\rho(B, C) \geqslant 1 - \eta$, we have $|\langle b, \sigma_{A \leftarrow B}(b) \rangle| \geqslant 1 - 2\eta$ for all $b \in B$ (using Fact 12.7.1). Since $(1 - 2\eta)^2 > 1 - 4\eta = 3/4$, Fact 12.4.1 (uniqueness of bijection) implies that $\sigma_{A \leftarrow B} = \pi_{A \leftarrow B}$. ■

Hence, for a consistent set of clouds $\mathcal{S}$, the distribution over local Unique Games labelings $\mu_{\mathcal{S}}$ can be defined easily as follows:

> Sample $\ell \in [R]$ uniformly at random, and for every cloud $A \in \mathcal{S}$, assign $\pi_A(\ell)$ as label.

To construct a local distribution for a set $\mathcal{S}$ which is not consistent, we partition the set $\mathcal{S}$ into consistent clusters. To this end, we make the following definition:

Definition 12.7.2. A set $\mathcal{S} \subseteq \mathcal{B}$ is *consistent* with respect to a partition P of $\mathcal{B}$ (denoted $\mathsf{Consistent}(\mathcal{S}, P)$) if

$$\forall \mathcal{C} \in P. \quad \forall A, B \in \mathcal{C} \cap \mathcal{S}. \qquad \rho(A, B) \geqslant 1 - 1/16 \,.$$

We use $\mathsf{Inconsistent}(\mathcal{S}, P)$ to denote the event that $\mathcal{S}$ is not consistent with P. The following is a corollary of Lemma 12.9.2.

Corollary 12.9.1. *Let P be a partition of $\mathcal{B}$ and let $\mathcal{S} \subseteq \mathcal{B}$. If $\mathsf{Consistent}(\mathcal{S}, P)$, then there exists bijections $\{\pi_A \colon [R] \to A \mid A \in \mathcal{S}\}$ such that*

$$\forall \mathcal{C} \in P. \quad \forall A, B \in \mathcal{C} \cap \mathcal{S}. \qquad \pi_B = \pi_{B \leftarrow A} \circ \pi_A \,.$$

The following lemma relies on the fact that the correlations $\rho(A, B)$ behave up to a small errors like inner products of real vectors. In other words, there is a geometric representation of the correlations $\rho(A, B)$ that can be used for the decomposition. This insight has also been used in UG algorithms[14].

Lemma 12.9.3. *For every $t \in \mathbb{N}$, there exists a distribution over partitions P of $\mathcal{B}$ such that*

- *if $\rho(A, B) \geqslant 1 - \varepsilon$, then*

$$\mathbb{P}\mathrm{r}\left\{P(A) = P(B)\right\} \geqslant 1 - O(t\sqrt{\varepsilon}) \,.$$

- *if $\rho(A, B) \leqslant 1 - 1/16$, then*

$$\mathbb{P}\mathrm{r}\left\{P(A) = P(B)\right\} \leqslant (3/4)^t \,.$$

Proof. Let $s \in \mathbb{N}$ be even and large enough (we will determine the value of s later). For every set $B \in \mathcal{B}$, define a vector $\boldsymbol{v}_B \in \mathbb{R}^D$ with $D := d^s$ as

$$\boldsymbol{v}_B := \tfrac{1}{\sqrt{R}} \sum_{v \in B} v^{\otimes s} .$$

We consider the following distribution over partitions P of $\mathcal{B}$: Choose t random hyperplanes $H_1, \dots, H_t$ through the origin in $\mathbb{R}^D$. Consider the partition of $\mathbb{R}^D$ formed by these hyperplanes. Output the induced partition P of $\mathcal{B}$ (two sets $A, B \in \mathcal{B}$ are in the same cluster of P if and only if $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ are not separated by any of the hyperplanes $H_1, \dots, H_t$).

Since s is even, Corollary 12.8.1 shows that for any two sets $A, B \in \mathcal{B}$,

$$\langle \boldsymbol{v}_A, \boldsymbol{v}_B \rangle = \rho(A, B)^s \pm 2 \cdot (3/4)^{-s/2} .$$

Furthermore, if $\rho(A, B) = 1 - \varepsilon$, then

$$\langle \boldsymbol{v}_A, \boldsymbol{v}_B \rangle \geqslant (1 - \varepsilon)^s \geqslant 1 - s\varepsilon .$$

Let $\eta = 1/16$. We choose s minimally such that $(1-\eta)^s + 2 \cdot (3/4)^{-s/2} \leqslant 1/\sqrt{2}$. (So s is an absolute constant.) Then for any two sets $A, B \in \mathcal{B}$ with $\rho(A, B) \leqslant 1 - \eta$, their vectors have inner product $\langle \boldsymbol{v}_A, \boldsymbol{v}_B \rangle \leqslant 1/\sqrt{2}$. Thus, a random hyperplane through the origin separates $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ with probability at least $1/4$. Therefore,

$$\Pr\{P(A) = P(B)\} \leqslant (3/4)^t .$$

On the other hand, if $\rho(A, B) = 1 - \varepsilon$, then the vectors of A and B have inner product $\langle \boldsymbol{v}_A, \boldsymbol{v}_B \rangle \geqslant 1 - s\varepsilon$. Thus, a random hyperplane through the origins separates the vectors with probability at most $O(\sqrt{\varepsilon})$. Hence,

$$\Pr\{P(A) = P(B)\} \geqslant \left(1 - O(\sqrt{\varepsilon})\right)^t \geqslant 1 - O(t\sqrt{\varepsilon}) .$$

■

Remark 12.7.1. Using a more sophisticated construction, we can improve the bound $1 - O(t\sqrt{\varepsilon})$ to $1 - O(\sqrt{t\varepsilon})$.

The previous lemma together with a simple union bound imply the next corollary.

Corollary 12.9.2. *The distribution over partitions from Lemma 12.9.3 satisfies the following property: For every set $\mathcal{S} \subseteq \mathcal{B}$,*

$$\Pr\left\{\mathsf{Inconsistent}(\mathcal{S}, P)\right\} \leqslant |S|^2 \cdot (3/4)^t$$

Remark 12.7.2. Using a slightly more refined argument (triangle inequality), we could improve the bound $r^2 \cdot (3/4)^t$ to $r \cdot (3/4)^t$.

12.8 Construction of SDP Solutions for E2LIN(q)

In this section, we construct SDP vectors and local distributions for $\mathcal{B} \times \mathbb{F}_q^R$ that form the variables in the $\Phi_{\gamma,q}(\Phi)$ instance described in Section 12.5. The set $\mathcal{B} \times \mathbb{F}_q^R$ correspond to the set of vertices in the instance obtained by applying a q-ary long code based reduction on the Unique Games instance encoded by $\mathcal{B}$. For a vertex $(B, x) \in \mathcal{B} \times \mathbb{F}_q^R$, we index the coordinates of x by the elements of B. Specifically, we have $x = (x_b)_{b \in B} \in \mathbb{F}_q^B$.

Geometric Partitioning Apply Lemma 12.9.3 to the collection of sets of vectors $\mathcal{B}$. We obtain a distribution $\mathcal{P}$ over partitions P of $\mathcal{B}$ into T disjoint subsets $\{P_\alpha\}_{\alpha=1}^T$. For a subset $\mathcal{S} \subset \mathcal{B}$, let $\mathcal{S} = \{\mathcal{S}_\alpha\}_{\alpha=1}^T$ denote the partition induced on the set $\mathcal{S}$, that is, $\mathcal{S}_\alpha := P_\alpha \cap \mathcal{S}$. For a family $B \in \mathcal{B}$, let α_B denote the index of the set P_{α_B} in the partition P that contains B.

12.8.1 Vector Solution

For a vertex $(B, x) \in \mathcal{B} \times \mathbb{F}_q^R$, the corresponding SDP vectors are given by functions $\boldsymbol{V}_j^{B,x} \colon \mathcal{P} \times [T] \times \mathcal{R} \to \mathbb{R}^q$ defined as follows:

$$\boldsymbol{W}_j^{B,x}(r) = \tfrac{1}{\sqrt{R}} \sum_{b \in B} \psi\big(x_b - j + b^{\otimes t}(r)\big) \tag{12.12}$$

$$\boldsymbol{U}_j^{B,x}(P, \alpha, r) = P_\alpha(B) \cdot \boldsymbol{W}_j^{B,x}(r) \tag{12.13}$$

$$\boldsymbol{V}_j^{B,x} = \tfrac{1}{q}\boldsymbol{V}_0 + \tfrac{\sqrt{q-1}}{q}\boldsymbol{U}_j^{B,x} \tag{12.14}$$

Here $\mathcal{R}$ is the measure space over which the tensored vectors $b^{\otimes t}$ are defined. The notation $P_\alpha(B)$ denotes the 0/1-indicator for the event $B \in P_\alpha$. Further, $\boldsymbol{V}_0$ is a unit vector orthogonal to all the vectors $\boldsymbol{U}_j^{B,x}$.

Let us evaluate the inner product between two vectors $\boldsymbol{V}_i^{A,x}$ and $\boldsymbol{V}_j^{B,y}$, (in this way, we also clarify the intended measure on the coordinate set)

$$\begin{aligned}
\langle \boldsymbol{V}_i^{A,x}, \boldsymbol{V}_j^{B,y} \rangle &= \tfrac{1}{q^2} + \tfrac{q-1}{q^2} \langle \boldsymbol{U}_i^{A,x}, \boldsymbol{U}_j^{B,y} \rangle \\
&= \tfrac{1}{q^2} + \tfrac{q-1}{q^2} \mathop{\mathbb{E}}_{P \sim \mathcal{P}} \textstyle\sum_{\alpha=1}^T P_\alpha(A) P_\alpha(B) \langle \boldsymbol{W}_i^{A,x}, \boldsymbol{W}_j^{B,y} \rangle \\
&= \tfrac{1}{q^2} + \tfrac{q-1}{q^2} \mathop{\mathbb{P}\mathrm{r}}_{P \sim \mathcal{P}} \{P(A) = P(B)\} \langle \boldsymbol{W}_i^{A,x}, \boldsymbol{W}_j^{B,y} \rangle
\end{aligned} \tag{12.15}$$

Let us also compute the inner product of $\boldsymbol{W}_i^{A,x}$ and $\boldsymbol{W}_j^{B,y}$. Recall the notation $\langle u, v \rangle_\psi :=$

$\langle \psi(u), \psi(v) \rangle$.

$$\begin{aligned}\langle \boldsymbol{W}_i^{A,x}, \boldsymbol{W}_j^{B,y} \rangle &= \tfrac{1}{n} \sum_{a \in A, b \in B} \mathop{\mathbb{E}}_{r \sim \mathcal{R}} \langle x_a - i + a^{\otimes t}(r), y_b - j + b^{\otimes t}(r) \rangle_\psi \\ &= \tfrac{1}{n} \sum_{a \in A, b \in B} \langle (x_a - i) \otimes a^{\otimes t}, (y_b - j) \otimes b^{\otimes t} \rangle_\psi \quad \text{(by Observation 12.6.3)} \\ &= \tfrac{1}{n} \sum_{a \in A, b \in B} \langle \psi(x_a - i), \psi(y_b - j) \rangle \langle a, b \rangle_\psi^t \quad \text{(by Lemma 12.9.1)} \qquad (12.16)\end{aligned}$$

12.8.2 Local Distributions

Fix a subset $\mathcal{S} \subset \mathcal{B}$ of size at most r. In this section, we will construct a local distribution over $\mathbb{F}_q$-assignments for the vertex set $S = \mathcal{S} \times \mathbb{F}_q^R$ (see Figure 12.3). Clearly, the same construction also yields a distribution for a general set of vertices $S' \subset \mathcal{B} \times \mathbb{F}_q^R$ of size at most r.

Remark 12.8.1. In the construction in Figure 12.3, the steps 6–7 are not strictly necessary, but they simplify some of the following calculations. Specifically, we could use the $\mathbb{F}_q$-assignment $\{F^{B,x}\}_{(B,x) \in S}$ to define the local distribution for the vertex set S. The resulting collection of local distributions could be extended to an approximately feasible SDP solution (albeit using a slightly different vector solution).

We need the following two simple observations.

Observation 12.8.1. *For all $a, b \in \mathbb{F}_q$, we have*

$$\Pr_{\kappa \in \mathbb{F}_q} [a + \kappa = i \wedge b + \kappa = j] = \tfrac{1}{q^2} + \tfrac{q-1}{q^2} \langle \psi(a - i), \psi(b - j) \rangle \,.$$

Proof. If $a - i = b - j$ then both LHS and RHS are equal to $1/q$, otherwise both are equal to 0. ■

Observation 12.8.2. *Fix $a, b \in \mathbb{F}_q$, over a random choice of $h_1, h_2 \in \mathbb{F}_q$,*

$$\mathop{\mathbb{E}}_{h_1, h_2 \in \mathbb{F}_q} [\langle \psi(a + h_1), \psi(b + h_2) \rangle] = 0 \,.$$

Proof. Follows easily from the fact that $\langle \psi(i), \psi(j) \rangle = 1$ if $i = j$ and $-1/q-1$ otherwise. ■

The next lemma shows that the second-order correlations of the distribution $\mu_{\mathcal{S}}$ approximately match the inner products of the vector solution $\{\boldsymbol{V}_i^{A,x}\}$.

Lemma 12.9.4. *For any two vertices $(A, x), (B, y) \in S$,*

$$\Pr_{Z \sim \mu_S} \left[Z^{A,x} = i \wedge Z^{B,y} = j \right] = \langle \boldsymbol{V}_i^{A,x}, \boldsymbol{V}_j^{B,y} \rangle \quad \pm 10 |\mathcal{S}|^2 (3/4)^{t/2} \,.$$

For $S = \mathcal{S} \times \mathbb{F}_q^R$, the local distribution μ_S over assignments $\mathbb{F}_q^S$ is defined by the following sampling procedure:

Partitioning:

1. Sample a partition $P = \{P_\alpha\}_{\alpha=1}^T$ of $\mathcal{B}$ from the distribution $\mathcal{P}$ obtained by Lemma 12.9.3. Let α_A, α_B denote the indices of sets in the partition P that contain $A, B \in \mathcal{S}$ respectively.

2. If $\mathsf{Inconsistent}(\mathcal{S}, P)$ then output a uniform random $\mathbb{F}_q$-assignment to $S = \mathcal{S} \times \mathbb{F}_q^R$. Specifically, set

$$Z^{(B,x)} = \text{ uniform random element from } \mathbb{F}_q \qquad \forall B \in \mathcal{S}, x \in \mathbb{F}_q^R \,.$$

Choosing Consistent Representatives:

4. If $\mathsf{Consistent}(\mathcal{S}, P)$ then by Corollary 12.9.1, for every part $\mathcal{S}_\alpha = P_\alpha \cap \mathcal{S}$, there exists bijections $\Pi_{\mathcal{S}_\alpha} = \{\pi_B \colon [R] \to B \mid B \in \mathcal{S}_\alpha\}$ such that for every $A, B \in \mathcal{S}_\alpha$,

$$\pi_A = \pi_{A \leftarrow B} \circ \pi_B \,.$$

5. Sample $L = \{\ell_\alpha\}_{\alpha=1}^T$ by choosing each ℓ_α uniformly at random from $[R]$. For every cloud $B \in \mathcal{S}$, define $\ell_B = \ell_{\alpha_B}$. The choice of L determines a set of representatives for each $B \in \mathcal{S}$. Specifically, the representative of B is fixed to be $\pi_B(\ell_B)$.

Sampling Assignments:

5. Sample $r \in \mathcal{R}$ from the corresponding probability measure and assign

$$F^{B,x}(P, L, r) = x_{\pi_B(\ell_B)} + \pi_B(\ell_B)^{\otimes t}(r) \,.$$

6. Sample $H = \{h_\alpha\}_{\alpha=1}^T$ by choosing each h_α uniformly at random from $[q]$. For every cloud $B \in \mathcal{B}$, define $h_B = h_{\alpha_B}$.

7. Sample κ uniformly at random from $[q]$.

8. For each $B \in \mathcal{S}_\alpha$ and $x \in \mathbb{F}_q^R$, set

$$Z^{B,x}(P, L, r, H, \kappa) = F^{B,x}(P, L, r) + h_B + \kappa \,.$$

9. Output the $\mathbb{F}_q$-assignment $\{Z^{B,x}\}_{(B,x) \in S}$.

Figure 12.3: Local distribution over $\mathbb{F}_q$-assignments

Proof. Firstly, since $\mathbb{P}\mathrm{r}[\mathsf{Consistent}(\mathcal{S},P)] \geqslant 1-|S|^2(3/4)^t$ (by Corollary 12.9.2),

$$\Pr_{\mu_S}\left[Z^{A,x}=i\wedge Z^{B,y}=j\right]=\Pr_{\mu_S}\left[Z^{A,x}=i\wedge Z^{B,y}=j \;\middle|\; \mathsf{Consistent}(\mathcal{S},P)\right] \quad \pm|S|^2(3/4)^t\,. \tag{12.17}$$

Using Observation 12.8.1, and the definition of $Z^{A,x}$ and $Z^{B,y}$ we can write

$$\begin{aligned}&\Pr_{\mu_S}\left[Z^{A,x}=i\wedge Z^{B,y}=j \;\middle|\; \mathsf{Consistent}(\mathcal{S},P)\right]\\ &\quad=\tfrac{1}{q^2}+\tfrac{q-1}{q^2}\mathop{\mathbb{E}}_{P,H,L,r}\left[\langle\psi(F^{A,x}+h_A-i),\psi(F^{B,y}+h_B-j)\rangle \;\middle|\; \mathsf{Consistent}(\mathcal{S},P)\right]\,.\end{aligned} \tag{12.18}$$

If A,B fall in the same set in the partition P (that is $\alpha_A=\alpha_B$), then we have $h_A=h_B$. If A,B fall in different sets (that is $\alpha_A\neq\alpha_B$), then h_A,h_B are independent random variables uniformly distributed over $\mathbb{F}_q$. Using Observation 12.8.2, we can write

$$\begin{aligned}&\mathop{\mathbb{E}}_{P,H,L,r}\left[\langle\psi(F^{A,x}+h_A-i),\psi(F^{B,y}+h_B-j)\rangle \;\middle|\; \mathsf{Consistent}(\mathcal{S},P)\right]\\ &\quad=\mathop{\mathbb{E}}_{P,L,r}\left[\mathbf{1}(\alpha_A=\alpha_B)\langle\psi(F^{A,x}-i),\psi(F^{B,y}-j)\rangle \;\middle|\; \mathsf{Consistent}(\mathcal{S},P)\right]\,.\end{aligned} \tag{12.19}$$

Let P be a partition such that $\mathsf{Consistent}(\mathcal{S},P)$ and $\alpha_A=\alpha_B=\alpha$. The bijections π_A,π_B (see step 4 Figure 12.3) satisfy $\pi_A=\pi_{A\leftarrow B}\circ\pi_B$. Note that therefore $a=\pi_{A\leftarrow B}(b)$ whenever $a=\pi_A(\ell)$ and $b=\pi_B(\ell)$ for some $\ell\in[R]$. Hence,

$$\begin{aligned}&\mathop{\mathbb{E}}_{L}\mathop{\mathbb{E}}_{r}\left[\langle\psi(F^{A,x}(P,L,r)-i),\psi(F^{B,y}(P,L,r)-j)\rangle\right]\\ &=\mathop{\mathbb{E}}_{\ell_\alpha}\mathop{\mathbb{E}}_{r}\left[\langle\psi(x_{\pi_A(\ell_\alpha)}-i+\pi_A(\ell_\alpha)^{\otimes t}(r)),\psi(y_{\pi_B(\ell_\alpha)}-j+\pi_B(\ell_\alpha)^{\otimes t}(r))\rangle\right]\\ &=\tfrac{1}{R}\sum_{\substack{a\in A,b\in B\\ a=\pi_{A\leftarrow B}(b)}}\mathop{\mathbb{E}}_{r}\langle\psi(x_a-i+a^{\otimes t}(r)),\psi(y_b-j+b^{\otimes t}(r))\rangle \qquad (\text{using } \pi_A=\pi_{A\leftarrow B}\circ\pi_B)\\ &=\tfrac{1}{R}\sum_{\substack{a\in A,b\in B\\ a=\pi_{A\leftarrow B}(b)}}\langle\psi(x_a-i),\psi(y_b-j)\rangle\cdot\langle a,b\rangle_\psi^t \quad (\text{using Observation 12.6.3 and Lemma 12.9.1})\\ &=\langle\boldsymbol{W}_i^{A,x},\boldsymbol{W}_j^{B,y}\rangle\pm 2\cdot(3/4)^{t/2} \qquad (\text{using Eq. (12.16) and Lemma 12.8.2})\,.\end{aligned}$$

Combining the last equation with the previous equations (12.17)–(12.19), we can finish the proof

$$\begin{aligned}&\Pr_{\mu_S}\left[Z^{A,x}=i\wedge Z^{B,y}=j\right]\\ &\quad=\tfrac{1}{q^2}+\tfrac{q-1}{q^2}\mathop{\mathbb{E}}_{P}\left[\mathbf{1}(\alpha_A=\alpha_B) \;\middle|\; \mathsf{Consistent}(\mathcal{S},P)\right]\langle\boldsymbol{W}_i^{A,x},\boldsymbol{W}_j^{B,y}\rangle \quad \pm(|S|^2(3/4)^t+2\cdot(3/4)^{t/2})\\ &\quad=\tfrac{1}{q^2}+\tfrac{q-1}{q^2}\Pr_{P}\left[P(A)=P(B)\right]\cdot\langle\boldsymbol{W}_i^{A,x},\boldsymbol{W}_j^{B,y}\rangle \quad \pm 10|\mathcal{S}|^2(3/4)^{t/2}\\ &\qquad\qquad (\text{using } \Pr\{\mathsf{Consistent}(\mathcal{S},P)\}\geqslant 1-|S|^2(3/4)^t \text{ and } |\langle\boldsymbol{W}_i^{A,x},\boldsymbol{W}_j^{B,y}\rangle|\leqslant 1)\\ &=\langle\boldsymbol{V}_i^{A,x},\boldsymbol{V}_j^{B,y}\rangle \quad \pm 10|\mathcal{S}|^2(3/4)^{t/2} \qquad (\text{using Eq. (12.15)})\,.\end{aligned}$$

■

Lemma 12.9.5. *Let $\mathcal{S}' \subset \mathcal{S}$ be two subsets of $\mathcal{B}$ and let $S' = \mathcal{S}' \times \mathbb{F}_q^R$ and $S = \mathcal{S} \times \mathbb{F}_q^R$. Then,*

$$\|\mu_{S'} - \mathsf{margin}_{S'}\, S\|_1 \leqslant 2|\mathcal{S}|^2 (3/4)^t \,.$$

Proof. For a partition $P \in \mathcal{P}$, let $\mu_{S|P}$ denote the distribution μ_S conditioned on the choice of partition P. Firstly, we will show the following claim:

Claim 12.9.1. *If* $\mathsf{Consistent}(\mathcal{S}', P)$ *and* $\mathsf{Consistent}(\mathcal{S}, P)$*, then* $\mu_{S'|P} = \mathsf{margin}_{S'}\, \mu_{S|P}$.

Proof. Let $\{\mathcal{S}_\alpha\}$ and $\{\mathcal{S}'_\alpha\}$ denote the partitions induced by P on the sets $\mathcal{S}$ and $\mathcal{S}'$ respectively. Since $\mathcal{S}' \subseteq \mathcal{S}$, we have $\mathcal{S}'_\alpha \subseteq \mathcal{S}_\alpha$ for all $\alpha \in [T]$. By our assumption, each of the sets $\mathcal{S}'_\alpha$ are *consistent* in that $\rho(A, B) \geqslant 1 - 1/16$ for all $A, B \in \mathcal{S}'_\alpha$. Similarly, the sets $\mathcal{S}_\alpha$ are also *consistent*.

Let us consider the pair of sets $\mathcal{S}'_\alpha \subset \mathcal{S}_\alpha$ for some $\alpha \in [T]$. Intuitively, the vectors within these sets fall into R distinct clusters. Thus the distribution over the choice of consistent representatives are the same in $\mu_{S'|P}$ and $\mathsf{margin}_{S'}\, \mu_{S|P}$. Formally, we have two sets of bijections $\Pi_{\mathcal{S}'_\alpha} = \{\pi'_A \mid A \in \mathcal{S}'_\alpha\}$ and $\Pi_{\mathcal{S}_\alpha} = \{\pi_A \mid A \in \mathcal{S}_\alpha\}$ satisfying the following property:

$$\pi_{A\to B} \circ \pi'_A(\ell) = \pi'_B(\ell) \qquad \pi_{A\to B} \circ \pi_A(\ell) = \pi_B(\ell) \qquad \forall A, B \in \mathcal{S}'_\alpha, \ell \in [R] \,.$$

Fix a collection $A \in \mathcal{S}'_\alpha$. Let $\sim$ denote that two sets of random variables are identically distributed.

$$\begin{aligned}\{\pi'_B(\ell_\alpha) \mid B \in \mathcal{S}'_\alpha\} &\sim \{\pi_{A\to B} \circ \pi'_A(\ell_\alpha) \mid B \in \mathcal{S}'_\alpha\} \\ &\sim \{\pi_{A\to B}(a) \mid B \in \mathcal{S}'_\alpha,\ a \text{ is uniformly random in } A\} \\ &\sim \{\pi_{A\to B} \circ \pi_A(\ell_\alpha) \mid B \in \mathcal{S}'_\alpha\} \sim \{\pi_B(\ell_\alpha) \mid B \in \mathcal{S}'_\alpha\} \,.\end{aligned}$$

The variables $L = \{\ell_\alpha\}$ are independent of each other. Therefore,

$$\{\pi'_B(\ell_B) \mid B \in \mathcal{S}'\} \sim \{\pi_B(\ell_B) \mid B \in \mathcal{S}'\} \,.$$

Notice that the choice of $r \in \mathcal{R}$, H and κ are independent of the set $\mathcal{S}$. Hence, the final assignments $\{Z^{B,x} \mid B \in \mathcal{S}', x \in \mathbb{F}_q^R\}$ are identically distributed in both cases. ■

Returning to the proof of Lemma 12.9.5, we can write

$$\begin{aligned}\|\mu_{S'} - \mathsf{margin}_{S'}\, \mu_S\|_1 &= \left\| \mathop{\mathbb{E}}_P \mu_{S'|P} - \mathop{\mathbb{E}}_P \mathsf{margin}_{S'}\, \mu_{S|P} \right\|_1 \\ &\leqslant \mathop{\mathbb{E}}_P \left[\|\mu_{S'|P} - \mathsf{margin}_{S'}\, \mu_{S|P}\|_1 \right] \qquad \text{(using Jensen's inequality)} \\ &= \Pr[\mathsf{Inconsistent}(\mathcal{S}, P)] \cdot \mathop{\mathbb{E}}_P \left[\|\mu_{S'|P} - \mathsf{margin}_{S'}\, \mu_{S|P}\|_1 \;\middle|\; \mathsf{Inconsistent}(\mathcal{S}, P) \right] .\end{aligned}$$

The first step uses that the operator $\mathsf{margin}_{S'}$ is linear. The final step in the above calculation makes use of Claim 12.9.1. The lemma follows by observing that $\Pr[\mathsf{Inconsistent}(\mathcal{S}, P)] \leqslant |\mathcal{S}|^2 (3/4)^t$ and $\|\mu_{S'|P} - \mathsf{margin}_{S'}\, \mu_{S|P}\|_1 \leqslant 2$. ■

The next corollary follows from the previous lemma (Lemma 12.9.5) and the triangle inequality.

Corollary 12.9.3. *Let $\mathcal{S}, \mathcal{S}'$ be two subsets of $\mathcal{B}$ and let $S' = \mathcal{S}' \times \mathbb{F}_q^R$ and $S = \mathcal{S} \times \mathbb{F}_q^R$. Then,*

$$\|\mathsf{margin}_{S\cap S'}\,\mu_S - \mathsf{margin}_{S\cap S'}\,\mu_{S'}\|_1 \leqslant 4\max\left(|\mathcal{S}|^2, |\mathcal{S}'|^2\right)(3/4)^t\,.$$

Proof. Suppose Φ is given by the vertex set V, the edge set $E \subseteq V \times V$, and the collection of permutations $\{\pi_e\}_{e\in E}$. Using Lemma 12.8.1, we obtain a weak $\mathbb{F}_q$-integral SDP solution $\mathcal{B} = \{B_u\}_{u\in V}$ of value $1 - O(\sqrt{\eta \log q})$ for Φ.

We construct a vector solution $\{\boldsymbol{V}_i^{B,x} \mid i \in \mathbb{F}_q, B \in \mathcal{B}, x \in \mathbb{F}_q^R\}$ and local distributions $\{\mu_S \mid S \subseteq \mathcal{B} \times \mathbb{F}_q^R\}$ as defined in Section (Section 12.8).

Note that since each set $B \in \mathcal{B}$ correspond to a vertices in $u \in V$, we can view these vectors and local distributions as an SDP solution for the $\mathsf{E2Lin}_q$ instance $\Phi_{\gamma,q}(\Phi)$. Specifically, we make the identifications $\boldsymbol{V}_i^{u,x} := \boldsymbol{V}_i^{B_u,x}$ and $\mu_S := \mu_{\{(B_u,x)|(u,x)\in S\}}$ for all $u \in V$, $x \in \mathbb{F}_q^R$, and sets $S \subseteq V \times \mathbb{F}_q^R$.

Lemma 12.9.4 and Corollary 12.9.3 show that this SDP solution is ε-infeasible for SA_r and LH_r, where $\varepsilon = O(r^2 \cdot (3/4)^{t/2})$. The value of the SDP solution for $\Phi_{\gamma,q}(\Phi)$ (see Fig. 12.2) is given by

$$\mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{\substack{w,w'\in N(v)\\ \pi=\pi_{w,v},\ \pi'=\pi_{w',v}}} \mathop{\mathbb{E}}_{\{x,y\}} \mathop{\mathbb{E}}_{c\in\mathbb{F}_q} \sum_{i=1}^{q} \langle \boldsymbol{V}_i^{w,(x\circ\pi+c\cdot\mathbb{1})}, \boldsymbol{V}_{i-c}^{w',\,y\circ\pi'}\rangle\,.$$

Using Eq. (12.15)–(12.16),

$$\begin{aligned}&\langle \boldsymbol{V}_i^{w,(x\circ\pi+c\cdot\mathbb{1})}, \boldsymbol{V}_{i-c}^{w',\,y\circ\pi'}\rangle\\ &= \tfrac{1}{q^2} + \tfrac{q-1}{q^2} \mathop{\mathbb{P}\mathrm{r}}_{P\sim\mathcal{P}}[P(B_w) = P(B_{w'})] \cdot \tfrac{1}{n} \sum_{\ell,\ell'\in[R]} \langle \psi(x_{\pi(\ell)} + c - i), \psi(y_{\pi'(\ell')} - (i-c))\rangle \langle b_{w,\ell}, b_{w',\ell'}\rangle_\psi^t\,.\end{aligned}$$

Note that $\langle \psi(x_{\pi(\ell)}+c-i), \psi(y_{\pi'(\ell')}-(i-c))\rangle = \langle \psi(x_{\pi(\ell)}, \psi(y_{\pi'(\ell')})\rangle$. Using Observation 12.4.1, we have $\pi_{(w,v)}(\ell) = \pi_{(w',v)}(\ell')$ if and only if $\ell = \pi_{B_w \leftarrow B_{w'}}(\ell')$. Hence, by Lemma 12.8.2,

$$\begin{aligned}&\tfrac{1}{n} \sum_{\ell,\ell'\in[R]} \langle \psi(x_{\pi(\ell)}), \psi(y_{\pi'(\ell')})\rangle \langle b_{w,\ell}, b_{w',\ell'}\rangle_\psi^t\\ &\quad= \tfrac{1}{n} \sum_{\ell} \langle \psi(x_{\pi(\ell)}), \psi(y_{\pi(\ell)})\rangle \langle b_{w,\pi(\ell)}, b_{w',\pi(\ell)}\rangle_\psi^t \quad \pm 2\cdot r^2 (3/4)^{t/2}\\ &\quad= \tfrac{1}{n} \sum_{\ell} \langle \psi(x_\ell), \psi(y_\ell)\rangle \rho(B_w, B_{w'})^t \pm O(\varepsilon)\,.\end{aligned}$$

Note that the distribution of $\{x, y\}$ is independent of the vertices v, w, w', and

$$\mathop{\mathbb{E}}_{\{x,y\}} \tfrac{1}{R} \sum_{\ell\in[R]} \langle \psi(x_\ell), \psi(y_\ell)\rangle = 1 - \gamma\,.$$

Therefore, if we let $\eta_{w,w'} = \rho(B_w, B_{w'})$, we can lower bound the value of the SDP solution

as follows

$$\begin{aligned}
&\mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{\substack{w,w'\in N(v)\\ \pi=\pi_{w,v},\ \pi'=\pi_{w',v}}} \mathop{\mathbb{E}}_{\{x,y\}} \mathop{\mathbb{E}}_{c\in\mathbb{F}_q} \sum_{i=1}^{q} \langle \boldsymbol{V}_i^{w,(x\circ\pi+c\cdot\mathbb{1})}, \boldsymbol{V}_{i-c}^{w',\,y\circ\pi'}\rangle \\
&= \mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{w,w'\in N(v)} \left[\tfrac{1}{q^2} + \tfrac{q-1}{q^2} \mathop{\mathbb{P}\mathrm{r}}_{P\sim\mathcal{P}} [P(B_w) = P(B_{w'})] \cdot q \cdot \rho(B_w, B_{w'})^t (1-\gamma)\right] \qquad \pm O(\varepsilon) \\
&\geqslant (1-\gamma) \mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{w,w'\in N(v)} \mathop{\mathbb{P}\mathrm{r}}_{P\sim\mathcal{P}} [P(B_w) = P(B_{w'})]\, \rho(B_w, B_{w'})^t \qquad \pm O(\varepsilon) \\
&\geqslant (1-\gamma) \mathop{\mathbb{E}}_{v\in V} \mathop{\mathbb{E}}_{w,w'\in N(v)} (1 - O(t\sqrt{\eta_{w,w'}})) \qquad \pm O(\varepsilon) \qquad \text{(using Lemma 12.9.3)}
\end{aligned}$$

Using Jensen's inequality and the fact that $\mathbb{E}_{v,w,w'} \eta_{v,w,w'} = O(\sqrt{\eta \log q})$ (Lemma 12.8.1), we see that the the value of our SDP solution is at least $1 - \gamma - O(\varepsilon + t\eta^{1/4})$ (recall that we assume q to be constant).

On smoothing the SDP solution using Theorem 12.7, we lose $O(r^2\varepsilon) = O(r^4(3/4)^t)$ in the SDP value. Thus we can set $t = o(\eta^{-1/4})$ and $r = (3/4)^{t/10}$ in order to get a feasible SDP solution for LH_r with value $1 - \gamma - o_{\eta,\delta}(1)$.

On smoothing the SDP solution using Theorem 12.8, we lose $O(q^r\varepsilon) = O(q^r(3/4)^t)$ in the SDP value. Thus we can set, $t = o(\eta^{-1/4})$ and $r = t/\log^2 q$, we would get a feasible SDP solution for SA_r with value $1 - \gamma - o_{\eta,\delta}(1)$. ■

Proof of Theorems 12.1–12.2. Using Theorem 12.9 with the Khot–Vishnoi integrality gap instance (Lemma 12.4.2), we have $N = 2^{2^{\log(1/\delta)/\eta}}$ and thus $r = 2^{O((\log\log N)^{1/4})}$. Similarly for SA_r, we get $r = O((\log\log N)^{1/4})$.

Chapter 13

3-QUERY PCP OVER INTEGERS

13.1 Introduction

Solving a system of linear equations over the rationals or reals is a fundamental algorithmic task arising in numerous applications. It is possible to tell in polynomial time, by Gaussian elimination, if a given system admits a solution, and if so to find one. However, Gaussian elimination is not robust against noise, and given an over-determined system of equations, of which say only 99% of the equations are simultaneously satisfiable, no efficient algorithm for finding a good solution satisfying a good fraction (say 50%) of equations is known. Indeed, it was recently shown that, for any constant $\varepsilon > 0$, given a $(1-\varepsilon)$-satisfiable linear system over the rationals, it is **NP**-hard to find an assignment to the variables that satisfies even a fraction ε of the equations [74, 61]. A similar hardness result over large finite fields was established in a classic paper by Håstad [86].

A celebrated hardness result due to Håstad [86] shows that for every constant $\varepsilon > 0$, given a $(1-\varepsilon)$-satisfiable system of linear equations over a finite field $\mathbb{F}_q$ where each equation depends on at most 3 variables, it is **NP**-hard to satisfy more than a fraction $\left(\frac{1}{q}+\varepsilon\right)$ of the equations. Underlying this result is a 3-query PCP verifier that queries 3 symbols from purported codewords of the "long code" (a code first defined and considered in [22]) and checks a linear constraint on them, and a tight estimate of the soundness of such a verifier using Fourier analysis. The method of designing long-code based PCP verifiers with tests that closely parallel the underlying constraint in the optimization problem of interest (3-variable linear equations in the above case), and analyzing their performance using Fourier analysis has been highly influential since (for instance, see Khot's survey [98]).

In this chapter, we prove the analog of Håstad's 3-variable linear equations result for equations over the integers (as well as the reals). Formally, we prove that for every $\varepsilon, \delta > 0$, given a system of linear equations with integer coefficients where each equation is on 3 variables, it is **NP**-hard to distinguish between the following two cases: (i) There is an assignment of integer values to the variables that satisfies at least a fraction $(1-\varepsilon)$ of the equations, and (ii) No assignment even of real values to the variables satisfies more than a fraction δ of the equations.

We stress that there seems to be no easy reduction from the problem of solving linear equations over finite fields to solving equations over the real numbers. It is straightforward to obtain a hardness result over integers from the hardness result of Håstad [86] over finite fields. Specifically, for every $\mod p$ equation of the form $x + y - z = c \mod p$, introduce an auxiliary variable w and an equation $x + y - z - pw = c$ over integers. However this reduction yields hardness of linear systems with 4 variables per equation instead of 3. More importantly, this reduction does not yield any hardness for linear systems over real numbers.

Obtaining a hardness of approximation result for linear systems with very few variables per constraint was mentioned as an open question in [61]. The result for general linear equations was obtained via a simple reduction from Label Cover in [74], and via a natural tensoring based approach to amplify the gap in [61].

Moreover, the problem of sparse linear systems is closely related to Unique Games. Γ-Max-2Lin which is a special case of Unique Games is exactly the problem of satisfying the maximum number of linear equations in a system where the sparsity is 2 (exactly two variables per equation) over a finite field. Moreover, by the work of Khot et al. [99],

the unique games conjecture is equivalent to the following hardness for solving sparse linear systems: for every $\varepsilon, \delta > 0$, given a system of sparse linear equations of the form $x_i - x_j = c_{ij} (\mod p)$, modulo a number p such that $1-\varepsilon$ of the equations can be simultaneously satisfied, it is **NP**-hard to find an assignment satisfying more than δ-fraction of the equations. It is natural to ask whether the **UGC** is equivalent to a similar hardness of solving sparse linear systems of integers or real numbers. This question still remains open.

In Section 13.2, we present an overview of our proof technique highlighting some of the key challenges in the integers case, our technical contributions to address them, and connections to derandomized linearity testing.

13.1.1 Previous related results

For sparse linear equations over integers, in fact with at most 2 variables per equation, it is shown in [7] (via a reduction from vertex cover on bounded degree graphs) that for some absolute constants $\rho_2 < \rho_1 < 1$, it is **NP**-hard to tell if such a system is at least ρ_1-satisfiable or at most ρ_2-satisfiable. By boosting this gap using a natural "product" construction, strong hardness results have been shown for the problem (called MAX-SATISFY in the literature) of approximating the number of satisfied equations in an over-determined system of (not necessarily sparse) linear equations over the rationals [7, 60]. In [60], it is shown that unless NP $\subset$ BPP, for every $\varepsilon > 0$, MAX-SATISFY cannot be approximated within a ratio of $n^{1-\varepsilon}$ where n is the number of equations in the system. (On the algorithmic side, the best approximation algorithm for the problem, due to Halldorsson [76], achieves ratio $O(n/\log n)$.)

However, the product construction destroys the sparsity of the original system, and also reduces the completeness to about ρ_1^k for a k-fold product. Consequently, even without the sparsity requirement, these results do not yield any hardness for near-satisfiable instances where an assignment satisfying a $(1-\varepsilon)$ fraction of the equations is promised to exist (for an arbitrarily small parameter $\varepsilon > 0$). For such near-satisfiable instances, a result showing **NP**-hardness of satisfying even an ε fraction of the equations was obtained only recently in [74, 61].

For the complementary objective of minimizing the number of unsatisfied equations, a problem called MIN-UNSATISFY, hardness of approximation within ratio $2^{\log^{0.99} n}$ is shown in [7] (see also [5]).

13.2 Proof Overview

Our proof follows along the lines of Håstad's result for 3-variable linear equations over prime fields $\mathbb{F}_q$. We give a 3-query PCP verifier that reads 3 appropriately chosen locations of the proof (each of whose entry holds an integer in some finite range) and checks a linear equation on them. The starting point is an instance of Label Cover over a fixed alphabet $[R]$ consisting of a bipartite graph and projection constraints $\pi_e : [R] \to [R]$ on the edges e; the projection constraint on edge (u, v) imposes the condition $\pi_{u \leftarrow v}(\ell(v)) = \ell(u)$ where $\ell(w)$ is the label assigned to vertex w. The verifier checks satisfiability of the Label Cover instance by picking a random edge (u, v) of the Label Cover graph and then checking

that the labels assigned to the endpoints of that edge satisfy the projection constraint. To aid the verifier to perform the latter check in a query-efficient way, the prover is expected to write down the *integer long code* encodings (in some large finite range) of all the vertex labels. The verifier picks one location x, with probability $P(x)$ for some distribution P, from the supposed long code $\mathcal{F}$ of u's label, and two locations y, y' with probability $P'(y)$ according to distribution P', from the suppose long code $\mathcal{H}$ of v's label. (Here $y' = y + x$ is determined once x, y are picked — in the actual test, as in Håstad's test [86], a small noise according to some distribution is added to $y + x$ to get y', and this is crucial. However, for the following description let us pretend that y' is determined once x, y are picked.) The verifier then checks that the values $\mathcal{F}(x), \mathcal{H}(y)$ and $\mathcal{H}(y')$ obey a linear constraint.

Let M be a large enough integer such that the total mass of distributions P and P' outside a cube of dimension M is tiny. Now any test of the above form that works for integers must also work modulo all large enough primes (that are much bigger than the range in which we allow the long code values to lie). In particular, picking q large enough compared to M, we will have a 3-query long code test modulo q that only queries a negligible fraction of the domain $\mathbb{F}_q^{[R]}$ of the long code. Therefore, our results imply a highly derandomized version of Håstad's test (though our target soundness ε is necessarily much larger than $1/p$). In particular, we obtain a test whose total randomness used depends only on the soundness and the dimension, and is independent of the domain size.

Technically, the difficulty imposed by this manifests itself in trying to extend the "decoding" procedure where the tables $\mathcal{F}$ and $\mathcal{H}$ are used to produce a small list of candidate labels for u and v. Håstad's decoding procedure uses the large Fourier coefficients of $\mathcal{F}$ to decode a small list of labels for u. The Fourier transform $\hat{\mathcal{F}}_P$ of $\mathcal{F}$ with respect to the distribution P can have many large coefficients since P is very far from uniform. In fact, the sum of squares of the Fourier coefficients grows exponentially in the dimension (size of the alphabet). A key technical lemma we show (Lemma 13.2.2) implies that the Fourier spectrum $\hat{\mathcal{F}}_P$ cannot have many large coefficients that are "far-off" from each other. Here the notion of two Fourier coefficients being "far-off" refers to the natural l_∞ metric between the corresponding linear functions being large. We then show how this can be exploited to decode a small set of labels for u from $\mathcal{F}$ (Claim 13.2.1). A "folding" property of the long code ensures that the set of decoded labels is in fact nonempty (Lemma 13.2.4). The property of the distribution P needed to show that $\mathcal{F}$ has few large pairwise far-off coefficients is an (ε, δ)-*concentration* property, namely $\sum_x P(x)e^{-i\omega\cdot x} \leqslant \varepsilon$ for all $\|\omega\|_\infty \geqslant 2\pi\delta$. Essentially for an (ε, δ)-concentrated distribution P, most of its weight is concentrated around the origin in the Fourier domain.

We are certainly not the first to attempt a derandomization of PCP tests. In particular, we want to point out the work of Ben-Sasson, Sudan, Vadhan, and Wigderson [23] who studied derandomized versions of the BLR linearity test [25] and the low-degree tests underlying PCP constructions. Their derandomized BLR test (for the field $\mathbb{F}_2$) picks a triple $(x, y, y' = y + x)$ of locations to query where y is uniformly distributed on the whole domain $\mathbb{F}_2^R$, but x is distributed uniformly on a much smaller subset S of the domain — the only requirement is that S is ε-biased, which means that for all nonzero $\omega \in \{0, \pi\}^R$, the Fourier coefficient $\frac{1}{|S|} \cdot \sum_{x\in S} e^{-i\omega\cdot x} \leqslant \varepsilon$. In our terminology, this means that the distribution on x is $(\varepsilon, 1/2)$-concentrated. However this derandomization is inadequate for our case, since y

ranges over the entire domain.

It is our hope that ideas from this work might perhaps be useful to reduce the size of long code based PCPs. This could enable giving such PCP constructions for much larger values of parameters, and in turn lead to some improved hardness of approximation results.

13.3 Results

We begin with formal definitions of the problems for which we obtain hardness results. The problem $\mathsf{MAX3LIN}_{\mathbb{Z}}$ consists of finding an assignment that satisfies maximum number of a set of linear equations over integers, each of which has 3 variables. Formally

Definition 13.3.1. For constants c, s satisfying $0 \leqslant s < c \leqslant 1$, define $\mathsf{MAX3LIN}_{\mathbb{Z}}(c, s)$ to be the following Promise problem: The input consists of a multiset of linear equations over variables $\{x_1, x_2 \ldots, x_n\}$ with each equation consisting of at most 3 variables. The problem is to distinguish between the following two cases:

- There is an integer assignment that satisfies at least a fraction c of the equations.
- Every integer assignment satisfies less than a fraction s of the equations.

$\mathsf{MAX3LIN}_{\mathbb{R}}(c, s)$ is the corresponding problem over real numbers instead of integers.

In this chapter, we prove the following hardness result for $\mathsf{MAX3LIN}_{\mathbb{Z}}$,

Theorem 13.1 (Main)**.** *For all constants $\varepsilon, \delta > 0$ the problem $\mathsf{MAX3LIN}_{\mathbb{Z}}(1-\varepsilon, \delta)$ is NP-hard. Further it is NP-hard even when all the equations are of the form $x_i + x_j = x_k + c$ for some integer constants c.*

It is easy to see that the above result implies a similar hardness result for $\mathsf{MAX3LIN}_{\mathbb{R}}$. The details of the reduction from $\mathsf{MAX3LIN}_{\mathbb{Z}}$ are as follows:

Theorem 13.2. *For all constants $\varepsilon, \delta > 0$, the problem $\mathsf{MAX3LIN}_{\mathbb{R}}(1-\varepsilon, \delta)$ is NP-hard.*

Proof. Let $\mathcal{I}$ be an instance of $\mathsf{MAX3LIN}_{Z}(1-\varepsilon, \frac{\delta}{8})$ with the additional restriction that all equations are of the form $x_i + x_j = x_k + c$ for some integer constants c. View this system of equations, as equations over $\mathbb{R}$ to obtain a $\mathsf{MAX3LIN}_{\mathbb{R}}(1-\varepsilon, \delta)$ instance.

In the completeness case, there is an integer assignment that satisfies at least $(1-\varepsilon)$ fraction of the equations. Clearly the same assignment is also a real assignment that satisfies at least $(1-\varepsilon)$ fraction of the equations.

Suppose there is a real assignment $\mathcal{F}_{\mathbb{R}}$ that satisfies more than δ fraction of the equations. Obtain an integer assignment $\mathcal{F}_{\mathbb{Z}}$ as follows: For each variable x_i, $\mathcal{F}_{\mathbb{Z}}(x_i)$ is randomly assigned either $\lceil \mathcal{F}_{\mathbb{R}}(x_i) \rceil$ or $\lfloor \mathcal{F}_{\mathbb{R}}(x_i) \rfloor$. For every equation $x_i + x_j = x_k + c$ that is satisfied by $\mathcal{F}_{\mathbb{R}}$ we have

$$\mathcal{F}_{\mathbb{R}}(x_i) + \mathcal{F}_{\mathbb{R}}(x_j) - \mathcal{F}_{\mathbb{R}}(x_k) = c$$

Since c is an integer, there exists at least one rounding (either ceiling or floor) of $\mathcal{F}_{\mathbb{R}}(x_i), \mathcal{F}_{\mathbb{R}}(x_j), \mathcal{F}_{\mathbb{R}}(x_k)$ such that the above equation continues to hold after rounding. With two choices for each $\mathcal{F}_{\mathbb{R}}(x_i)$, there are 8 possible ways to round the 3 variables. Hence with probability at least $\frac{1}{8}$ the equation still holds after rounding. So the expected number of equations satisfied by the rounded solution $\mathcal{F}_{\mathbb{Z}}$ is at least $\frac{\delta}{8}$. ■

13.4 Analytic Machinery

13.4.1 Fourier Preliminaries

Let $\mathbb{F}_q$ denote the prime field with q elements. Here we recall the definition of Fourier transform and a few useful identities. For a function $\mathcal{F} : \mathbb{F}_q^R \to \mathbb{C}$, define the function $\hat{\mathcal{F}}(\omega)$ as follows:

$$\hat{\mathcal{F}}(\omega) = \frac{1}{q^R} \sum_{x \in \mathbb{F}_q^R} \mathcal{F}(x) e^{-i\omega \cdot x}$$

Hence $\hat{\mathcal{F}}(\omega)$ is a function defined over $[0, 2\pi]^R$. Let $S_p = \{0, \frac{2\pi}{p}, \ldots, \frac{2\pi j}{p}, \ldots, \frac{2\pi(p-1)}{p}\}$. The values of $\hat{\mathcal{F}}(\omega)$ on the finite set S_p^R is the Fourier transform of the function $\mathcal{F}$ on $\mathbb{F}_q^R$. Throughout the analysis, we will only be using these Fourier coefficients, i.e., the values of $\hat{\mathcal{F}}(\omega)$ on S_p^R. The Fourier coefficients satisfy the following identities:
Inverse Transform:

$$\mathcal{F}(x) \quad = \quad \sum_{\omega \in S_p^R} \hat{\mathcal{F}}(\omega) e^{i\omega \cdot x}$$

Parseval's identity:

$$\frac{1}{q^R} \sum_{x \in \mathbb{F}_q^R} |\mathcal{F}(x)|^2 \quad = \quad \sum_{\omega \in S_p^R} |\hat{\mathcal{F}}(\omega)|^2$$

Although we will be applying Fourier Transform over a large prime field $\mathbb{F}_p$, it is instructive to think of the Fourier transform $\hat{\mathcal{F}}(\omega)$ as a function over the continuous domain $[0, 2\pi]^R$. Operations like addition, subtraction, multiplication by scalars, of elements in $[0, 2\pi]^R$ are all done modulo 2π. For instance, if $\omega' = 3\omega$ then the i^{th} coordinate of ω' is given by $\omega'_i = 3\omega_i \mod 2\pi$. For $\theta \in [0, 2\pi]$ we will use $\|\theta\|_{2\pi}$ to denote $\min(\theta, 2\pi - \theta)$. For any $\omega \in [0, 2\pi]^R$ define $\|\omega\|_\infty = \max_{i \in \{1,\ldots,R\}} \|\omega_i\|_{2\pi}$ This defines a metric on $[0, 2\pi]^R$ given by $d(\omega, \omega') = \|\omega - \omega'\|_\infty$ for any two ω, ω'.

We shall denote by $\mathbb{Z}_+$ the set of non negative integers. For a general probability distribution P on $\mathbb{Z}_+^R$, and a function $\mathcal{F} : \mathbb{Z}_+^R \to \mathbb{C}$, we define

$$\hat{\mathcal{F}}_P(\omega) = \mathop{\mathbb{E}}_{x \in P}[\mathcal{F}(x) e^{-i\omega \cdot x}] \tag{13.1}$$

The numbers $\hat{\mathcal{F}}_P(\omega)$ can be thought of as the Fourier coefficients of $\mathcal{F}$ with respect to the distribution P. Notice that, if $\mathcal{F}$ were a function on $\mathbb{F}_q^R$, and P was the uniform distribution over $\mathbb{F}_q^R$, $\mathcal{F}_P(\omega)$ would reduce to the traditional definition of Fourier coefficient of $\mathcal{F}$.

13.4.2 (ε, δ)-concentrated distributions

Let $\hat{\mathbf{1}}$ denote the constant function on $\mathbb{Z}_+^R$ which is always equal to 1. The notion of an (ε, δ)-concentrated distribution is defined as follows:

Definition 13.4.1. For $\varepsilon, \delta > 0$, a probability distribution P on $\mathbb{Z}_+^R$ is said to be (ε, δ)-concentrated if $|\hat{\mathbf{1}}_P(\omega)| \leqslant \varepsilon$ for all $\|\omega\|_\infty \geqslant 2\pi\delta$.

Intuitively a probability distribution is (ε, δ)-concentrated if its Fourier transform is concentrated around the origin. In what follows, we will derive some results on the distribution of large Fourier coefficients $\hat{\mathcal{F}}_P(\omega)$ in $[0, 2\pi]^R$ if an arbitrary function $\mathcal{F}$, and an (ε, δ)-concentrated distributions P. Let $\ell_2(\mathbb{Z}_+^R)$ denote the vector space of all functions from $\mathbb{Z}_+^R$ to $\mathbb{C}$ such that $\sum_{x\in\mathbb{Z}_+^R} |\mathcal{F}(x)|^2 < \infty$. Let $\langle v_1, v_2\rangle = \sum_{x\in\mathbb{Z}_+^R} v_1(x)\overline{v_2(x)}$ denote the natural inner product for two functions v_1, v_2 in $\ell_2(\mathbb{Z}_+^R)$.

Lemma 13.2.1. *Let P be a (ε, δ)-concentrated probability distribution. For any $\omega_1, \omega_2 \in [0, 2\pi]^R$ such that $\|\omega_1 - \omega_2\|_\infty \geqslant 2\pi\delta$ the functions $v_1(x) = \sqrt{P(x)}e^{i\omega_1\cdot x}$ and $v_2(x) = \sqrt{P(x)}e^{i\omega_2\cdot x}$ are nearly orthogonal, i.e., $\langle v_1, v_2\rangle \leqslant \varepsilon$.*

Proof. We have

$$\begin{aligned} v_1 \cdot v_2 = &\sum_{x\in\mathbb{Z}_+^R} \sqrt{P(x)}e^{i\omega_1\cdot x}\overline{\sqrt{P(x)}e^{i\omega_2\cdot x}} \\ = &\quad \hat{\mathbf{1}}_P(\omega_2 - \omega_1) \leqslant \varepsilon \end{aligned}$$

where the last inequality follows from $\|\omega_1 - \omega_2\|_\infty \geqslant 2\pi\delta$ and the fact that P is (ε, δ)-concentrated. ∎

Let $\mathcal{F} : \mathbb{F}_q^R \to \mathbb{C}$ be a function that is bounded, say $|\mathcal{F}(x)| = 1$ for all x. By Parseval's identity, the sum of squares of Fourier coefficients $\hat{\mathcal{F}}(\omega)$ is 1. In particular, this implies that not more than $\frac{1}{\varepsilon^2}$ of the Fourier coefficients can be more than ε. Now, consider a function $\mathcal{F} : \mathbb{Z}_+^R \to \mathbb{C}$ satisfying $|\mathcal{F}(x)| = 1$ for all x. The Fourier coefficients $\hat{\mathcal{F}}_P(\omega)$ do not satisfy the Parseval's identity. In fact, the sum of the squares of Fourier coefficients could be exponentially large in R, thus giving us no bound on the number of large Fourier coefficients.

However, the following lemma asserts that there cannot be many large Fourier coefficients that are all far from each other. Specifically, although there could be exponentially many ω for which $\hat{\mathcal{F}}_P(\omega)$ is large, they are all clustered together into very few clusters.

Lemma 13.2.2 (Few far-off Fourier coefficients)**.** *For $0 \leqslant \varepsilon < \frac{1}{9}, \delta > 0$, let P be a (ε^5, δ)-concentrated probability distribution. Let $\mathcal{F} : \mathbb{Z}_+^R \to \mathbb{C}$ be a function such that $|\mathcal{F}(x)| = 1$ for all $x \in \mathbb{Z}_+^R$. Let $\Omega = \{\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(k)}\} \subset [0, 2\pi]^R$ be a set such that $\|\omega^{(j)} - \omega^{(j')}\|_\infty \geqslant 2\pi\delta$ and $|\hat{\mathcal{F}}_P(\omega^{(j)})| \geqslant \varepsilon$ for all j, j'. Then $|\Omega| < \frac{3}{\varepsilon^2}$.*

Proof. On the contrary, let us say there exists a set Ω such that $|\Omega| \geqslant \frac{3}{\varepsilon^2}$. By deleting some elements from the set, we can assume $k = |\Omega| = \frac{3}{\varepsilon^2}$. Consider functions $v(x) = \sqrt{P(x)}\mathcal{F}(x)$, $v_j(x) = \sqrt{P(x)}e^{i\omega^{(j)}\cdot x}$ for all $1 \leqslant j \leqslant |\Omega|$. Observe that all of them are unit vectors in $\ell_2(\mathbb{Z}_+^R)$. Since $v \cdot v_j = \hat{\mathcal{F}}_P(\omega^{(j)})$ we have $|v \cdot v_j| \geqslant \varepsilon$. Further using lemma 13.2.1, we know

$|v_j \cdot v_{j'}| \leqslant \varepsilon^5$. Now consider

$$\begin{aligned} |v - \sum_{i=j}^{k}(v \cdot v_j)v_j|^2 &= |v|^2 + \sum_{j=1}^{k}(v \cdot v_j)^2|v_j|^2 - 2\sum_{j=1}^{k}(v \cdot v_j)^2 \\ &\quad +2 \sum_{1 \leqslant j' < j \leqslant k} (v \cdot v_{j'})(v \cdot v_j)(v_{j'} \cdot v_j) \\ &\leqslant 1 - k\varepsilon^2 + 2\binom{k}{2}\varepsilon^5 \end{aligned}$$

Substituting $k = \frac{3}{\varepsilon^2}$, $|v - \sum_{j=1}^{k}(v \cdot v_j)v_j|^2 < 1 - 3 + 18\varepsilon < 0$, a contradiction. Hence we must have $|\Omega| < \frac{3}{\varepsilon^2}$. ■

13.4.3 An explicit (ε, δ)-concentrated distribution

It can be shown that the uniform distribution over the cube $[M]^R$ is (ε, δ)-concentrated for a sufficiently large integer M. However we will use the exponential probability distribution to simplify some of the calculations. Formally, define a probability distribution P on $\mathbb{Z}_+^R$ as :

$$P(x) = \gamma e^{-c\sum_{i=1}^{R} x_i} \text{ for some } c > 0 \text{ and } \gamma = (1 - e^{-c})^R \tag{13.2}$$

The constant γ in the above definition is the correct normalization constant to ensure that P is a distribution. In showing that P has the desired properties, we will use the following fact:

Fact 13.4.1. *For $c > 0$ and $\omega \in [0, 2\pi]$ the following inequality holds $|1 - e^{-c-i\omega}| \geqslant \frac{2e^{-c}}{\pi}\|\omega\|_{2\pi}$.*

Proof. We have $|1 - e^{-c-i\omega}| \geqslant |e^{-c} - e^{-c-i\omega}| \geqslant e^{-c}|1 - e^{-i\omega}| \geqslant e^{-c}|2\sin\frac{\omega}{2}|$. Using the fact that $|\sin\theta| \geqslant \frac{2\theta}{\pi}$ for $\theta \in [0, \frac{\pi}{2}]$, we conclude

$$|1 - e^{-c-i\omega}| \geqslant \frac{2e^{-c}}{\pi}|\min(\omega, 2\pi - \omega)| = \frac{2e^{-c}}{\pi}\|\omega\|_{2\pi}$$

■

Lemma 13.2.3. *For all constants $\varepsilon, \delta > 0$ and $0 < c < \ln(1 + 4\delta\varepsilon)$, the distribution P defined in Equation (13.2) is (ε, δ)-concentrated.*

Proof. Let $\omega \in [0, 2\pi]^R$ be such that $\|\omega\|_\infty \geqslant 2\pi\delta$. In particular, let j_0 be an index such

that $\min(\omega_{j_0}, 2\pi - \omega_{j_0}) > 2\pi\delta$.

$$\begin{aligned}\hat{\mathbf{1}}_P(\omega) &= \sum_{x\in\mathbb{Z}_+^R} P(x)e^{-i\omega\cdot x} \\ &= (1-e^{-c})^R \prod_{j=1}^{R}\sum_{x_j=0}^{\infty} e^{-cx_j}e^{-i\omega_j x_j} \\ &= \prod_{j=1}^{R}\frac{(1-e^{-c})}{(1-e^{-c-i\omega_j})}\end{aligned}$$

However from Fact 13.4.1 we know

$$\begin{aligned}|1-e^{-c-i\omega_{j_0}}| &\geqslant \frac{2e^{-c}}{\pi}\|\omega_{j_0}\|_{2\pi} \\ &\geqslant 4e^{-c}\delta\end{aligned}$$

Substituting in the expression for $\hat{\mathbf{1}}_P(\omega)$ we get

$$|\hat{\mathbf{1}}_P(\omega)| = \left(\prod_{j=1,j\neq j_0}^{R}\frac{|(1-e^{-c})|}{|(1-e^{-c-i\omega_j})|}\right)\frac{|(1-e^{-c})|}{|(1-e^{-c-i\omega_{j_0}})|} \leqslant \frac{|(1-e^{-c})|}{4e^{-c}\delta} = \frac{e^c-1}{4\delta}$$

which is less than ε for $c < \ln(1+4\delta\varepsilon)$. ■

13.5 Label Cover *Test*

The reduction from Label Cover proceeds along the lines of [86]. We will present the reduction as a PCP system for Label Cover which makes linear tests on three proof locations. The connection to $\mathsf{MAX3LIN}_{\mathbb{Z}}$ will be immediate. Towards this, we define a long code over integers as follows:

Definition 13.5.1. The long code for label $i \in \{1,\ldots,R\}$ consists of the function $\mathbf{F}_i : \mathbb{Z}_+^R \to \mathbb{Z}$ defined by $\mathbf{F}_i(x) = x_i$ for all $x \in \mathbb{Z}_+^R$.

13.5.1 Folding

Denote by $\mathbb{Z}_0^R \subset \mathbb{Z}_+^R$ the set of all points in $\mathbb{Z}_+^R$ with the one of its coordinates equal to zero.

Definition 13.5.2. A $\mathbf{1}$-folded long code is a function $\mathbf{F'}_i : \mathbb{Z}_0^R \to \mathbb{Z}$ defined by $\mathbf{F}'_i(x) = x_i$. More generally, a function $a : \mathbb{Z}_+^R \to \mathbb{Z}$ is a $\mathbf{1}$-folded function if $a(x+\mathbf{1}) = a(x)+1$.

Given a $\mathbf{1}$-folded long code $\mathbf{F'}_i$, it is possible to retrieve the value of the full long code at any location x. This is achieved by expressing $x \in \mathbb{Z}_+^R$ as $x = x_0 + t\mathbf{1}$ where $x_0 \in \mathbb{Z}_0^R$, and then using $\mathbf{F}(x) = \mathbf{F'}(x_0) + t$. By using $\mathbf{1}$-folded long codes, the reduction ensures that all functions under consideration are $\mathbf{1}$-folded functions.

If a **1**-folded function a is linear, then clearly it must be of the form $a(x) = \sum_{i=1}^{R} a_i x_i$ where $\sum_{i=1}^{R} a_i = 1$. The following lemma asserts that the significant Fourier coefficients ω corresponding to an arbitrary **1**-folded function a also approximately satisfy $\sum_{i=1}^{R} \omega_i = 1$.

Lemma 13.2.4 (Folding lemma)**.** *Let $a : \mathbb{Z}_+^R \to \mathbb{Z}$ be a function such that $a(x+\mathbf{1}) = a(x)+1$ for all $x \in \mathbb{Z}_+^R$. Let $\mathcal{F}(x) = e^{i\frac{2\pi k a(x)}{p}}$. For all $\delta > 0$ and $c < \frac{1}{R}\ln\left(1+4\delta^2\right)$ the following holds: for all $\omega \in [0, 2\pi]^R$ with $\|\omega \cdot \mathbf{1} - \frac{2\pi k}{p}\|_{2\pi} \geqslant 2\pi\delta$:*

$$|\hat{\mathcal{F}}_P(\omega)| \leqslant \delta$$

Proof. Recall that $\mathbb{Z}_0^R \subset \mathbb{Z}_+^R$ denotes the set of all points in $\mathbb{Z}_+^R$ with the one of its coordinates equal to zero. For every $x \in \mathbb{Z}_+^R$, there exists unique $x_0 \in \mathbb{Z}_0^R, t \in \mathbb{Z}$ such that $x = x_0 + t\mathbf{1}$. By definition of P we have $P(x) = P(x_0)e^{-cRt}$. Hence picking x with probability $P(x)$ is the same as:

- Pick $x_0 \in \mathbb{Z}_0^R$ with probability $\tilde{P}(x_0) = \sum_{t=0}^{\infty} P(x_0 + t\mathbf{1})$
- Pick t with probability $p(t) = (1 - e^{-cR})e^{-cRt}$

Decompose the expression for $\hat{\mathcal{F}}_P(\omega)$ as follows:

$$\begin{aligned}\hat{\mathcal{F}}_P(\omega) &= \mathop{\mathbb{E}}_{x\in P}[\mathcal{F}(x)e^{-i\omega\cdot x}] \\ &= \mathop{\mathbb{E}}_{x_0\in\tilde{P}}\mathop{\mathbb{E}}_{t\in p}[\mathcal{F}(x_0+t\mathbf{1})e^{-i\omega\cdot(x_0+t\mathbf{1})}]\end{aligned}$$

However since $a(x_0+\mathbf{1}) = a(x)+1$, we know $\mathcal{F}(x_0+t\mathbf{1}) = \mathcal{F}(x_0)e^{\frac{2\pi kt}{p}}$. Substituting we get

$$\hat{\mathcal{F}}_P(\omega) = \mathop{\mathbb{E}}_{x_0\in\tilde{P}}[\mathcal{F}(x_0)e^{-i\omega\cdot x_0}]\mathop{\mathbb{E}}_{t\in p}[e^{\frac{2\pi kt}{p}}e^{-i\omega\cdot t\mathbf{1}}]$$

Now to compute

$$\begin{aligned}\left|\mathop{\mathbb{E}}_{t\in p}[e^{\frac{2\pi kt}{p}}e^{-i\omega\cdot t\mathbf{1}}]\right| &= \left|(1-e^{-cR})\sum_{t=0}^{\infty}e^{-cRt}e^{it(\frac{2\pi k}{p}-\omega\cdot\mathbf{1})}\right| \\ &= \frac{|(1-e^{-cR})|}{|1-e^{-cR+i\Delta}|}\end{aligned}$$

where $\Delta = \frac{2\pi k}{p} - \omega\cdot\mathbf{1}$. By our assumption $\|\Delta\|_{2\pi} \geqslant 2\pi\delta$, hence using fact 13.4.1, we get

$$|\mathop{\mathbb{E}}_{t\in p}[e^{\frac{2\pi kt}{p}}e^{-i\omega\cdot t\mathbf{1}}]| \leqslant \frac{|(1-e^{-cR})|}{4e^{-cR}|\delta|} \leqslant \delta$$

for all $c < \frac{1}{R}\ln\left(1+4\delta^2\right)$. Since $|\mathcal{F}(x)| = 1$ for all x, we know $|\mathbb{E}_{x_0\in\tilde{P}}[\mathcal{F}(x_0)e^{-i\omega\cdot x_0}]| \leqslant 1$. Together with the bound on $|\mathbb{E}_{t\in p}[e^{\frac{2\pi kt}{p}}e^{-i\omega\cdot t\mathbf{1}}]|$, this implies the required result. ∎

13.5.2 Verifier

As defined above, long codes are infinite objects that cannot be written down. Throughout this article, we will be dealing with long codes that are truncated by restricting the domain from $\mathbb{Z}_+^R$ to $[M]^R$ for some large M. However for the purposes of analysis, it is convenient to ignore the truncation and assume that the entire long code is available. As we shall see later, this truncation can be carried out since the verifier queries the values outside a sufficiently large box $[M]^R$ with very low probability.

Let $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, [R], \Pi)$ be an instance of Label Cover. Given an assignment $\mathcal{F}$ to the instance Φ, the corresponding PCP proof consists of the **1**-folded long codes of the labels assigned to each of the vertices in $\mathcal{W}_\Phi \cup \mathcal{V}_\Phi$. For instance if $\mathcal{F}$ is an assignment then for every vertex $w \in \mathcal{W}_\Phi \cup \mathcal{V}_\Phi$ the proof contains the **1**-folded long code $\mathbf{F}'_{\mathcal{F}(w)}$.

Recall that given a **1**-folded long code $\mathbf{F}'_i$, it is possible to retrieve the value of the full long code at any location x. Henceforth, we shall describe the verifier as having access to the full long code of the labels. Clearly the linear tests of the verifier on the full long code can be converted to linear tests on the **1**-folded long code.

Given a function $\pi : [R] \to [R]$ and a vector $x \in \mathbb{Z}_+^R$ define $x \circ \pi \in \mathbb{Z}_+^R$ as $(x \circ \pi)_i = x_{\pi(i)}$. Let P and P' be exponential decay probability distributions over $\mathbb{Z}_+^R$ whose parameters will be chosen later. Intuitively, the distribution P will be chosen to be a sufficiently slowly decaying exponential distribution, while the distribution P' decays at a much slower rate than P. The verifier is described below:

3-Query PCP Verifier

1. Pick a random edge $e = (w, v) \in E$. Let $a : \mathbb{Z}_+^R \to Z$,$b : \mathbb{Z}_+^R \to \mathbb{Z}$ be the long codes corresponding to vertices w, v respectively.

2. Pick a random $x \in \mathbb{Z}_+^R$ with the distribution P, a random $y \in \mathbb{Z}_+^R$ with the distribution P'

3. Generate a *noise* vector $\mu \in \mathbb{Z}_+^R$ from the following distribution : Each coordinate μ_i is chosen
   - 0 with probability $(1 - \varepsilon)$.
   - Chosen uniformly at random from $\{1, \ldots, m\}$ with probability ε.

4. Accept if the following equation holds

$$a(x) = b(x \circ \pi + y + \mu) - b(y)$$

For technical reasons, we will need the following simple lemmas in the soundness analysis.

Lemma 13.2.5. *The total weight of the distribution P outside the set $[N]^R$ (on the set $\mathbb{Z}_+^R - [N]^R$) is less than δ for $N \geqslant \frac{1}{c} \ln \frac{R}{\delta(1-e^{-c})^R}$*

Proof. We have

$$\begin{aligned}\sum_{x\in\mathbb{Z}_+^R-[N]^R} P(x) &\leqslant \sum_{i=1}^{R}\sum_{x_i\geqslant N} P(x)\\ &\leqslant \frac{Re^{-cN}}{(1-e^{-c})^R}\end{aligned}$$

which is less than δ for $N \geqslant \frac{1}{c}\ln\frac{R}{\delta(1-e^{-c})^R}$. ■

Lemma 13.2.6. *For all $M > 0$, $c \leqslant \frac{\ln 4}{RM}$, for all $x \in [M]^R, y \in \mathbb{Z}_+^R$ the following is true:* $P(x+y) \geqslant P(y)/4$

Proof. Clearly we have

$$\frac{P(x+y)}{P(y)} = e^{-c\sum_i x_i} \geqslant e^{-cRM} \geqslant \frac{1}{4}$$

■

13.5.3 Noise Stability

Notice that in Step (3), the 3-query PCP verifier generates a *noise* vector μ. Finally, instead of querying the location $b(x \circ \pi + y)$, the verifier queries the value of a nearby location $b(x \circ \pi + y + \mu)$.

Introducing *noise* into the locations queried by the verifier is a powerful recurring theme in dictatorship (long code) tests and PCP constructions ever since its use in Håstad [86]. Roughly speaking, using this technique, the verifier can ensure that the function being queried does not depend on too many coordinates. Specifically, if the function b was a long code then $b(x \circ \pi + y + \mu) = b(x \circ \pi + y)$ with high probability over the choice of the noise vector μ. On the other hand, if b is a linear function depending on too many coordinates, then the noise μ would affect the value, thus reducing the probability of success.

Denote by Q the distribution on $\mathbb{Z}_+^R$ of the *noise* vector μ. That is each coordinate of μ is chosen independently to be 0 with probability $(1-\varepsilon)$ and a uniformly random element in $\{1,\ldots,m\}$ with probability ε. Along the lines of Håstad [86], we need to bound the contribution of the Fourier coefficients of b corresponding to linear forms depending on many coordinates. However, in our setting, the coefficients of the linear forms are not discrete. Thus, we say a linear function depends on many coordinates if it has more than C (defined below in Lemma 13.2.7) *large enough* coefficients. The following lemma will be used in the soundness analysis to bound the contribution of the Fourier coefficients corresponding to these linear functions:

Lemma 13.2.7. *For all $\varepsilon_1 > 0, 0 < \delta_1 \leqslant \frac{1}{4}$ and constants $m = \lceil\frac{1}{\delta_1}\rceil$, $C = \lceil\log_{1-\frac{\varepsilon}{2}}\varepsilon_1\rceil$ the following is true: For all $\omega \in [0,2\pi]^R$ with more than C coordinates ω_i satisfying $\|\omega_i\|_{2\pi} \geqslant 2\pi\delta_1$,*

$$|\hat{\mathbf{1}}_Q(\omega)| \leqslant \varepsilon_1$$

Proof. Let S denote the set of indices $j \in \{1, \ldots, R\}$ such that $\|\omega_j\|_{2\pi} \geqslant 2\pi\delta_1$. Then by definition $|S| \geqslant C$

$$\begin{aligned}
|\hat{\mathbf{1}}_Q(\omega)| &= \left| \sum_{x \in \mathbb{Z}_+^R} Q(x) e^{-i\omega \cdot x} \right| \\
&= \left| \prod_{j=1}^{R} \left[(1-\varepsilon) e^{i\omega_j \cdot 0} + \frac{\varepsilon}{m} \sum_{t=1}^{m} e^{i\omega_j t} \right] \right| \\
&\leqslant \prod_{j=1}^{R} \left[1 - \varepsilon + \frac{\varepsilon}{m} \left| \frac{e^{i\omega_j (m+1)} - e^{i\omega_j}}{e^{i\omega_j} - 1} \right| \right] \\
&\leqslant \prod_{j \in S} \left[1 - \varepsilon + \frac{\varepsilon}{m} \frac{2}{|e^{i\omega_j} - 1|} \right]
\end{aligned}$$

By definition of S, $\|\omega_j\|_{2\pi} > 2\pi\delta_1$ for $j \in S$. Hence using Fact 13.4.1 with $c = 0$ we get

$$|\hat{\mathbf{1}}_Q(\omega)| \leqslant \prod_{j \in S} \left(1 - \varepsilon + \frac{\varepsilon}{2m\delta_1} \right)$$

which for $m \geqslant 1/\delta_1$ and $C \geqslant \log_{1-\varepsilon/2} \varepsilon_1$ is at most

$$\prod_{j \in S} \left(1 - \varepsilon + \frac{\varepsilon}{2} \right) = \left(1 - \frac{\varepsilon}{2} \right)^C \leqslant \varepsilon_1 \ .$$

■

13.6 Proof of Main Theorem

In this section, we will present the proof of Theorem 13.1. Towards this, we first describe the parameters for the verifier in section 13.5.

Choose an integer $m > \frac{R^2}{\delta}$. Choose a c less than both $\frac{1}{R}\ln(1+4\delta^2)$ and $\ln(1+4(\frac{\delta}{4})^5 \frac{\delta}{2R})$. Denote by P the exponential decay probability distribution with parameter c. In particular, P is $((\frac{\delta}{4})^5, \frac{\delta}{2R})$-concentrated. Let N be the integer obtained from Lemma 13.2.5, such that weight of P outside $[N]^R$ is less than δ. Let c' be a real number less than $\frac{\ln 4}{R(N+m)}$. Let P' denote the exponential probability distribution with parameter c'.

Completeness: Suppose $\mathcal{F}$ is an assignment that satisfies all the edge constraints Π. The corresponding long code assignment is accepted by the verifier with probability at least $1-\varepsilon$. For an edge $e = (w, v) \in E$, the verifier rejects the long code assignment only if $\mu_{\mathcal{F}(v)} \neq 0$. It is clear from the choice of μ that this happens with probability exactly $1 - \varepsilon$.

Soundness: Suppose the verifier accepts with probability greater than 19δ. Let $\chi^{uv}(x, y, \mu)$ be the indicator variable that is 1 if the test on edge $e = (w, v)$ succeeds with random choices

x, y, μ. Then we can write the probability of acceptance of the test as follows:

$$\mathbb{P}\mathrm{r}[\text{test accepts}] = \mathop{\mathbb{E}}_{w,v}\left[\sum_{\substack{x,y,\\ \mu\in\mathbb{Z}_+^R}} P(x)P'(y)Q(\mu)\chi^{uv}(x,y,\mu)\right] \geqslant 19\delta$$

Notice that the support of the distribution μ is $\{0,1,\ldots m\}^R$. Further from Lemma 13.2.5 the total weight of the distribution P outside $[N]^R$ is less than δ. Hence we can truncate the summation over x and conclude

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\substack{x\in[N]^R,\ \mu\in[m]^R\\ y\in\mathbb{Z}_+^R}} P(x)P'(y)Q(\mu)\chi^{uv}(x,y,\mu)\right] \geqslant 18\delta$$

where $[N]^R \subset \mathbb{Z}_+^R$ defined as $[N]^R = \{0,1,\ldots,N\}^R$. Clearly for $x \in [N]^R, \mu \in [m]^R$ the vector $x \circ \pi + \mu \in [N+m]^R$. Recall that the distribution P' is chosen to be sufficiently slowly decaying in comparison to $P(x)$ and $Q(\mu)$. That is by Lemma 13.2.6 for all $y \in \mathbb{Z}_+^R, z \in [N+m]^R$ we have $P'(y+z) \geqslant \frac{P'(y)}{4}$. In particular, $P'(y + x \circ \pi_{v\leftarrow w} + \mu) \geqslant \frac{P'(y)}{4}$, or equivalently $2\sqrt{P'(y + x \circ \pi_{v\leftarrow w} + \mu)P'(y)} \geqslant P'(y)$. Henceforth we will use y' to denote $y + x \circ \pi_{v\leftarrow w} + \mu$.

Using this inequality in the expression for probability of acceptance we get:

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\substack{x\in[N]^R,\mu\in[m]^R\\ y\in\mathbb{Z}_+^R}} P(x)\sqrt{P'(y)P'(y')}Q(\mu)\chi^{uv}(x,y,\mu)\right] \geqslant 9\delta$$

For a prime q define $\chi_p^{uv}(x,y,\mu)$ to be 1 if $a(x) + b(y) - b(x \circ \pi + y + \mu) = 0 \mod p$ and zero otherwise. Clearly $\chi_p^{uv}(x,y,\mu) \geqslant \chi^{uv}(x,y,\mu)$ for all integers x, y, μ. Replacing χ^{uv} by χ_p^{uv} we get:

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\substack{x\in[N]^R,\mu\in[m]^R\\ y\in\mathbb{Z}_+^R}} P(x)\sqrt{P'(y)P'(y')}Q(\mu)\chi_p^{uv}(x,y,\mu)\right] \geqslant 9\delta$$

The prime q can be chosen to be sufficiently large so that truncating the summation over y to $[q]^R$ does not alter the probability value significantly. Further, by picking q sufficiently large, it is possible to ensure that the total weight of the distributions P, P', Q outside $[\frac{p}{3}]^R$ is less than δ. Hence computing $y' = y + x \circ \pi_{v\leftarrow w} + \mu$ modulo q is same as computing y' over integers for all but a δ fraction of (x,y,μ). In particular, we can conclude

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\substack{x,\mu,\\ y\in[q]^R}} P(x)\sqrt{P'(y)P'(y')}Q(\mu)\chi_p^{uv}(x,y,\mu)\right] \geqslant 8\delta \tag{13.3}$$

where $y' = y + x \circ \pi_{uv} + \mu$ is computed modulo q. Notice that the parameter q is an artifact in the analysis, and is chosen to be sufficiently large compared to all other parameters. It is instructive to think of q as tending to infinity while all other parameters are fixed.

Now we fix an edge $e = (w, v)$ and analyze the probability that the test succeeds. Let π denote the projection constraint on the edge e. The following is an arithmetization for χ_p^{uv}:

$$\chi_p^{uv}(x, y, \mu) = \frac{1}{p} \sum_{k=0}^{p-1} \beta^{k[a(x)+b(y)-b(x\circ\pi+y+\mu)]}$$

where $\beta = e^{\frac{2\pi i}{p}}$. Now we define the following notation:

$$\begin{aligned} \mathcal{F}(x) &= \beta^{a(x)} & \mathcal{H}(x) &= \beta^{b(x)} \\ \mathcal{F}^k(x) &= P(x)\beta^{ka(x)} & \mathcal{H}(x) &= \sqrt{P'(x)}\beta^{kb(x)} \end{aligned}$$

Substituting the above expressions in (13.3) we get:

$$\mathop{\mathbb{E}}_{w,v}\left[\frac{1}{p}\sum_{k=0}^{p-1}\sum_{\substack{x,\mu,\\ y\in[q]^R}} Q(\mu)\mathcal{F}^k(x)\mathcal{H}^k(y)\overline{\mathcal{H}^k(y')}\right] \geqslant 8\delta\ . \tag{13.4}$$

Given an $\omega \in [0, 2\pi]^R$ and a function $\pi : [R] \to [R]$, the vector $\pi(\omega) \in [0, 2\pi]^R$ is defined by $(\pi(\omega))_i = \sum_{j\in\pi^{-1}(i)} \omega_j$. The expression inside the expectation in (13.4) is similar to the one obtained in [86], and using a standard computation over $\mathbb{F}_q$ it can be written in terms of the Fourier coefficients. For the sake of completeness, we include the details below. The expression within the expectation in (13.4) is equal to

$$\frac{1}{p}\sum_{k=0}^{p-1}\sum_{\substack{x,\mu,\\ y\in[q]^R}} Q(\mu) \sum_{\omega_1\in S_p^R} \hat{\mathcal{F}}^k(\omega_1)e^{i\omega_1\cdot x} \sum_{\omega_2\in S_p^R} \hat{\mathcal{H}}^k(\omega_2)e^{i\omega_2\cdot y} \overline{\sum_{\omega_3\in S_p^R} \hat{\mathcal{H}}^k(\omega_3)e^{i\omega_3\cdot(x\circ\pi+y+\mu)}}$$

$$= \frac{1}{p}\sum_{k=0}^{p-1} \sum_{\omega_1,\omega_2,\omega_3\in S_p^R} \hat{\mathcal{F}}^k(\omega_1)\hat{\mathcal{H}}^k(\omega_2)\overline{\hat{\mathcal{H}}^k}(\omega_3) \sum_{\mu\in[q]^R} Q(\mu)e^{-i\omega_3\cdot\mu} \sum_{x\in[q]^R} e^{i(\omega_1-\pi(\omega_3))\cdot x} \sum_{y\in[q]^R} e^{i(\omega_2-\omega_3)\cdot y}$$

Since $\omega_1, \omega_2, \omega_3 \in S_p^R$, we have

$$\begin{aligned} \sum_{x\in[q]^R} e^{i(\omega_1-\pi(\omega_3))\cdot x} &= 0 \text{ unless } \omega_1 = \pi(\omega_3) \\ \sum_{y\in[q]^R} e^{i(\omega_2-\omega_3)\cdot y} &= 0 \text{ unless } \omega_2 = \omega_3 \end{aligned}$$

Using these relations in the expression, and renaming ω_3 to be ω we get

$$\frac{1}{p}\sum_{k=0}^{p-1}\sum_{\omega\in S_p^R}\left(q^R\hat{\mathcal{F}}^k(\pi(\omega))\right)\left(q^R\left|\hat{\mathcal{H}}^k(\omega)\right|^2\right)\left(\sum_{\mu\in[q]^R}Q(\mu)e^{-i\omega\cdot\mu}\right)$$

Recall that for $Q(\mu)=0$ for all $\mu\notin[m]^R$, hence for $p>m$ we have $\sum_{\mu\in[q]^R}Q(\mu)e^{-i\omega\cdot\mu}=\hat{\mathbf{1}}_Q(\omega)$. Therefore we have

$$\frac{1}{p}\sum_{k=0}^{p-1}\mathop{\mathbb{E}}_{w,v}\left[\sum_{\omega\in S_p^R}\left(q^R|\hat{\mathcal{H}}^k(\omega)|^2\right)\left|q^R\hat{\mathcal{F}}^k(\pi(\omega))\hat{\mathbf{1}}_Q(\omega)\right|\right]\geqslant 8\delta \tag{13.5}$$

From Parseval's identity we have,

$$q^R\sum_{\omega\in S_p^R}|\hat{\mathcal{H}}^k(\omega)|^2=\sum_{x\in[q]^R}|\sqrt{P'(x)}\beta^{kb(y)}|^2\leqslant 1 \tag{13.6}$$

Further we have $|q^R\hat{\mathcal{F}}^k(\omega)|,|\hat{\mathbf{1}}_Q(\omega)|\leqslant 1$ for all ω. Hence for all k

$$\sum_{\omega\in S_p^R}\left(q^R|\hat{\mathcal{H}}^k(\omega)|^2\right)\left|q^R\hat{\mathcal{F}}^k(\pi(\omega))\hat{\mathbf{1}}_Q(\omega)\right|\leqslant 1$$

The inequality (13.5) asserts that the average of q such terms is larger than 8δ. By an averaging argument, there exists $2\delta p\leqslant k\leqslant p(1-2\delta)$ such that

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\omega\in S_p^R}\left(q^R|\hat{\mathcal{H}}^k(\omega)|^2\right)\left|q^R\hat{\mathcal{F}}^k(\pi(\omega))\hat{\mathbf{1}}_Q(\omega)\right|\right]\geqslant 4\delta$$

Fix some such k for the rest of the argument. Observe that

$$q^R\hat{\mathcal{F}}^k(\pi(\omega))=\sum_{x\in[q]^R}P(x)\mathcal{F}^k(x)e^{-i\pi(\omega)\cdot x}$$

By Definition 13.1, the Fourier coefficient $\hat{\mathcal{F}}_P^k(\pi(\omega))$ with respect to distribution P is given by

$$\hat{\mathcal{F}}_P^k(\pi(\omega))=\sum_{x\in\mathbb{Z}_+^R}P(x)\mathcal{F}^k(x)e^{-i\pi(\omega)\cdot x}\ .$$

For sufficiently large choice of the prime q, we have

$$|q^R\hat{\mathcal{F}}^k(\pi(\omega))-\hat{\mathcal{F}}_P^k(\pi(\omega))|\leqslant\delta\ .$$

Substituting $q^R\hat{\mathcal{F}}^k(\pi(\omega))$ by $\hat{\mathcal{F}}^k_P(\pi(\omega))$ and using equation 13.6 we get

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\omega\in S_p^R}\left(q^R|\hat{\mathcal{H}}^k(\omega)|^2\right)\left|\hat{\mathcal{F}}^k_P(\pi(\omega))\hat{\mathbf{1}}_Q(\omega)\right|\right] \geqslant 3\delta \tag{13.7}$$

13.6.1 Restricting to "sparse" Fourier coefficients

The expectation (13.7) above looks similar to the expression that is used to derive labels in Håstad's work on 3-variable linear equations modulo 2. This latter expression is of the form $\sum_\beta |\hat{\mathcal{H}}(\beta)|^2|\hat{\mathcal{F}}_{\pi_2(\beta)}|$ summed over all β of small size — see [86] for details. Along the lines of [86], we will use the Fourier coefficients in the above expression to obtain a decoding of labels to the vertices w, v. Roughly speaking, Håstad's decoding proceeds as follows:

> For each vertex $v \in \mathcal{W}_\Phi \cup \mathcal{V}_\Phi$, sample a sparse Fourier coefficient ω from an appropriate distribution, and sample uniformly random non-zero coordinate of ω. Assign to vertex v the label corresponding to the coordinate.

The Fourier coefficients ω in our case do not take discrete values. Although for the purposes of analysis we have used ω in a discrete set S_q^R, recall that q is chosen to be sufficiently large compared to every other parameter including R. In fact, it is instructive to think of $p \to \infty$ while all other parameters stay fixed.

In the continuous setting, the notion of a *sparse* Fourier coefficient ω needs to be redefined. Specifically, a sparse Fourier coefficient ω would have a few large coordinates ω_i, while the remaining coordinates are small in absolute value. To this end, we define two subsets $\Omega_1, \Omega_2 \subset S_p^R$ as follows:

- Ω_1: set of ω such that $\|\omega \cdot \mathbf{1}\|_{2\pi} \geqslant 2\pi\delta$. In other words, for every $\omega \in \Omega_1$ there is at least one *large* coordinate, i.e, a coordinate ω_i with $\|\omega_i\|_{2\pi} \geqslant \frac{2\pi\delta}{R}$.

- Ω_2: subset of ω which have very few *large* coordinates. In particular, for all $\omega \in \Omega_2$ at most C of its coordinates satisfy $\|\omega_i\|_{2\pi} \geqslant \frac{2\pi\delta}{R^2}$. (Here C is the constant from Lemma 13.2.7.)

Here $\Omega_1 \cap \Omega_2$ would be the set of *sparse* Fourier coefficients for our purpose.

Firstly, we will bound the contribution of Fourier coefficients with no large coordinate using Lemma 13.2.4. This corresponds to bounding the contribution of trivial Fourier coefficient in [86]. Notice that $\omega \cdot \mathbf{1} = \pi(\omega) \cdot \mathbf{1}$. Hence for $\omega \notin \Omega_1$, $\|\pi(\omega) \cdot \mathbf{1} - \frac{2\pi k}{p}\|_{2\pi} \geqslant \|2\pi\delta - \frac{2\pi(2\delta p)}{p}\|_{2\pi} \geqslant 2\pi\delta$. From Lemma 13.2.4 and choice of distribution P, $|\hat{\mathcal{F}}^k_P(\pi(\omega))| < \delta$ when $\|\pi(\omega) \cdot \mathbf{1} - \frac{2\pi k}{p}\|_{2\pi} \geqslant 2\pi\delta$. This implies that $|\hat{\mathcal{F}}^k_P(\pi(\omega))| < \delta$ for all $\omega \notin \Omega_1$.

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\omega\in \Omega_1}\left(q^R|\hat{\mathcal{H}}^k(\omega)|^2\right)\left|\hat{\mathcal{F}}^k_P(\pi(\omega))\hat{\mathbf{1}}_Q(\omega)\right|\right] \geqslant 2\delta$$

To bound the contribution of Fourier coefficients with too many large coordinates, we will use the noise μ introduced by the verifier. More precisely, we have $|\hat{\mathbf{1}}_Q(\omega)| \leqslant \delta$ for all $\omega \notin \Omega_2$ from Lemma 13.2.7. Therefore,

$$\mathop{\mathbb{E}}_{w,v}\left[\sum_{\omega\in\Omega_1\cap\Omega_2}\left(q^R|\hat{\mathcal{H}}^k(\omega)|^2\right)\left|\hat{\mathcal{F}}^k_P(\pi(\omega))\right|\right] \geqslant \delta \tag{13.8}$$

We will next see how one can decode labels satisfying many Label Cover constraints based on (13.8).

13.6.2 Decoding Label Sets

For $\omega \in [0, 2\pi]^R$ and $\delta > 0$, let $\mathsf{L}_\delta(\omega) \subseteq [R]$ denote the subset of indices ω_i such that $\|\omega_i\|_{2\pi} \geqslant 2\pi\delta$.

For every vertex $v \in \mathcal{V}_\Phi$ with the corresponding Fourier transform $\hat{\mathcal{H}}^k$, define P_v to be the distribution obtained by normalizing $q^R|\hat{\mathcal{H}}^k(\omega)|^2$. Since $\sum_{\omega\in S_p^R} q^R|\hat{\mathcal{H}}^k(\omega)|^2 \leqslant 1$, $P_v = \gamma q^R|\hat{\mathcal{H}}^k(\omega)|^2$ for some $\gamma \geqslant 1$. For a vertex $w \in \mathcal{W}_\Phi$ with the corresponding Fourier transform $\hat{\mathcal{F}}^k_P$, define the set $\Omega_{\mathcal{F}}$ of significant frequencies as follows:

$$\Omega_{\mathcal{F}} = \left\{\omega \in \Omega_1 \cap \Omega_2 \ : \ |\hat{\mathcal{F}}^k_P(\omega)| \geqslant \frac{\delta}{4}\right\} . \tag{13.9}$$

Define the set $\mathsf{L}(w)$ as follows:

$$\mathsf{L}(w) = \bigcup_{\omega\in\Omega_{\mathcal{F}}} \mathsf{L}_{\frac{\delta}{R}}(\omega) . \tag{13.10}$$

Intuitively $\mathsf{L}(w)$ is the set of all *large* coordinates of those ω for which the Fourier coefficient $|\hat{\mathcal{F}}^k_P(\omega)|$ is *large*. The decoding algorithm proceeds as follows:

- For $v \in \mathcal{V}_\Phi$, pick a $\omega \in S_p^R$ with probability P_v. Assign a label uniformly at random from $\mathsf{L}_{\frac{\delta}{R^2}}(\omega)$ if it is nonempty, else assign a random label.

- For every vertex $w \in \mathcal{W}_\Phi$, assign a label uniformly at random from $\mathsf{L}(w)$ if it is nonempty, else assign a random label.

Every Fourier coefficient $\omega \in \Omega_{\mathcal{F}}$ is *sparse* in that it at most C large coordinates. A trivial bound on the size of $\mathsf{L}(w)$ is given by $C \cdot |\Omega_{\mathcal{F}}|$. In Håstad's work [86], this bound suffices since the size of $\Omega_{\mathcal{F}}$ is bounded using Parseval's identity. The main technical challenge in our setting is that $\sum_\alpha |\hat{\mathcal{F}}^k_P(\alpha)|^2$, the sum of squared Fourier coefficients with respect to the distribution P, could be very large. In particular, bounding this sum by Parseval's we get

$$\sum_\alpha \left(q^R|\hat{\mathcal{F}}^k(\alpha))|\right)^2 \leqslant q^R \sum_x P(x)^2 .$$

When $P(x)$ is uniform, i.e., $P(x) = 1/q^R$ for every x, this bound equals 1, but the bound could be exponentially larger in R for distributions P that are very non-uniform (as in our case). Thus the obvious extension of Håstad's argument will lead to a list size bound that is too large to be useful as a decoding strategy.

Although the size of $\Omega_{\mathcal{F}}$ could be exponentially large, Lemma 13.2.2 shows that the large Fourier coefficients are all clustered into a few clusters. Using this property, we obtain the following bound on the size of $\mathsf{L}(w)$.

Claim 13.2.1. *For every vertex $w \in \mathcal{W}_\Phi$, the cardinality of the set $\mathsf{L}(w)$ is at most $\frac{48C}{\delta^2}$.*

Proof. Recall that by definition, every $\omega \in \Omega_2$ has at most C coordinates ω_i satisfying $\|\omega_i\|_{2\pi} \geqslant \frac{2\pi\delta}{R^2}$. Hence for all $\omega \in \Omega_1 \cap \Omega_2$ each of the sets $\mathsf{L}_{\frac{\delta}{R}}(\omega)$ and $\mathsf{L}_{\frac{\delta}{2R}}(\omega)$ have a cardinality of at most C.

Suppose the assertion of the claim is false. We will inductively produce a *large* set of distant ω, for all of which $\hat{\mathcal{F}}_P^k(\omega)$ is *large*. This will contradict the Lemma 13.2.2 since the distribution P is concentrated.

Construct the set $\Omega' \subset \Omega_{\mathcal{F}}$ iteratively as follows: To start with pick an $\omega^{(1)} \in \Omega_{\mathcal{F}}$. After $t \geqslant 1$ steps, let $\mathsf{L}_t = \cup_{i=1}^t \mathsf{L}_{\frac{\delta}{2R}}(\omega^{(i)})$. Since each $\mathsf{L}_{\frac{\delta}{2R}}$ has at most C elements, the cardinality of L_t is at most $C \cdot t$. Since $\mathsf{L}(w) > \frac{48C}{\delta^2}$, when $t \leqslant \frac{48}{\delta^2}$ we have $|\mathsf{L}(w)| > |\mathsf{L}_t|$. In particular, there exists some $\omega^{(t+1)} \in \Omega_{\mathcal{F}}$ such that the set $\mathsf{L}_{\frac{\delta}{R}}(\omega^{(t+1)}) - \mathsf{L}_t$ is nonempty. Let us assume $j \in \mathsf{L}_{\frac{\delta}{R}}(\omega^{(t+1)}) - \mathsf{L}_t$. For any $1 \leqslant i \leqslant t$, the distance $\|\omega^{(i)} - \omega^{(t+1)}\|_\infty \geqslant \|\omega_j^{(t+1)} - \omega_j^{(i)}\|_{2\pi}$. Since $j \in \mathsf{L}_{\frac{\delta}{R}}(\omega^{(t+1)}) - \mathsf{L}_t$, we have $\|\omega_j^{(t+1)}\|_{2\pi} \geqslant \frac{2\pi\delta}{R}$ and $\|\omega_j^{(i)}\|_{2\pi} \leqslant \frac{2\pi\delta}{2R}$. Hence the distance $\|\omega^{(i)} - \omega^{(t+1)}\|_\infty$ is at least $\frac{2\pi\delta}{2R}$.

By iterating the above process, it is possible to construct a set $\Omega' \subseteq \Omega_{\mathcal{F}}$ with cardinality at least $\frac{48}{\delta^2}$ such that for all $\omega^{(i)}, \omega^{(j)} \in \Omega'$, $\|\omega^{(i)} - \omega^{(j)}\|_\infty \geqslant \frac{2\pi\delta}{2R}$. This will contradict Lemma 13.2.2, since P is a $(\left(\frac{\delta}{4}\right)^5, \frac{\delta}{2R})$-concentrated. ∎

13.6.3 Soundness analysis wrap-up using the label sets

By an averaging argument applied to (13.8), at least for a fraction $\frac{\delta}{2}$ of the edges the following inequality holds:

$$\sum_{\omega \in \Omega_1 \cap \Omega_2} \left(q^R |\hat{\mathcal{H}}^k(\omega)|^2\right) \left|\hat{\mathcal{F}}_P^k(\pi(\omega))\right| \geqslant \frac{\delta}{2}$$

We refer to these edges (w, v) as *good* edges. Consider a good edge $e = (w, v)$. On choosing ω over the probability distribution $P_v(\omega)$ with probability at least $\frac{\delta}{4}$ we have $|\hat{\mathcal{F}}_P^k(\pi_{v \leftarrow w}(\omega))| \geqslant \frac{\delta}{4}$ and $\omega \in \Omega_1 \cap \Omega_2$. Since $\omega \in \Omega_1$ we have $\|\pi(\omega) \cdot 1\|_{2\pi} \geqslant 2\pi\delta$. Consequently, there have to be large coordinates of $\pi(\omega)$, i.e., there must exist $i \in [R]$ such that $\|[\pi(\omega)]_i\|_{2\pi} \geqslant \frac{2\pi\delta}{R}$. Suppose $i \in \mathsf{L}_{\frac{\delta}{R}}(\pi(\omega))$ is a large coordinate of $\pi(\omega)$ then there must be a large coordinate of ω in $\pi^{-1}(i)$, i.e., a $j \in \pi^{-1}(i)$ such that $\|\omega_j\|_{2\pi} \geqslant \frac{2\pi\delta}{R^2}$. Recall that $\omega \in \Omega_2$ has at most C large coordinates. Therefore with probability at least $\frac{1}{C}$, the vertex v is assigned label j. Further using Claim 13.2.1, we conclude that vertex w is assigned label i with probability

at least $\frac{\delta^2}{48C}$. The edge (w, v) is satisfied when w is assigned i and v is assigned j. Hence the edge e is satisfied with probability at least $\frac{\delta}{4} \cdot \frac{1}{C} \cdot \frac{\delta^2}{48C} = \frac{\delta^3}{192C^2}$. As there are at least a fraction $\frac{\delta}{2}$ of *good* edges, the expected fraction of edges satisfied is at least $\frac{\delta^4}{384C^2}$ which is greater than $\frac{1}{R^\gamma}$ for large enough R.

We have thus shown that the 3-query PCP has completeness $(1 - \varepsilon)$ and soundness at most 19δ. The tests it makes are linear equations. Therefore, we immediately get that the promise problem $\mathsf{MAX3LIN}_{1-\varepsilon,19\delta}$ is **NP**-hard. Since $\varepsilon, \delta > 0$ are arbitrary, the proof of Theorem 13.1 is complete.

Chapter 14

CONCLUSIONS AND FUTURE WORK

In this thesis, we exhibited a black-box reduction from SDP integrality gaps to UG-hardness results. Not only has this work formalized the widely suspected connection, it has led to new insights in terms of optimal rounding schemes, and algorithms for computing integrality gaps [136]. At the outset, this dissertation following fact:

> *"For large classes of combinatorial optimization problems such as constraint satisfaction problems, metric Labeling problems and ordering constraint satisfaction problems, a fairly minimal semidefinite programming relaxation (LC) give the best approximation computable in polynomial time assuming the Unique Games Conjecture."*

Going further, the thesis demonstrates a single generic algorithm for every constraint satisfaction problems (CSP) which yields the optimal approximation under the **UGC**. Independent of the truth of the **UGC**, this algorithm is guaranteed to be at least as good all known approximation algorithms for specific CSPs. Finally, independent of the truth of the **UGC**, including all valid constraints on at most $2^{(\log \log N)^{1/4}}$ vectors does not improve upon the approximation given by the LC SDP relaxation. Among other corollaries that hold irrespective of the truth of **UGC**, is an algorithm to compute the Grothendieck constant - a fundamental mathematical constant whose value is unknown.

Overall, the thesis demonstrates that Unique Games is a common barrier that all existing techniques seem to have reached, for a surprisingly wide variety of combinatorial optimization problems which were seemingly unconnected earlier. Thus, the study of approximability is at an exciting juncture. An affirmation of the unique games conjecture would resolve long standing open questions and demonstrate an underlying unity in combinatorial optimization problems, while a disproof would hopefully lead to new algorithmic techniques.

14.1 Directions for Future Work

14.1.1 Understanding Unique Games

The Unique Games Conjecture (**UGC**) remains a notorious open problem today. Not only is the conjecture unresolved, but there is no consensus among theorists about its truth. In fact, attempts at disproving the conjecture [155, 72, 41, 35] have failed by a close margin.

There are three different scenarios that seem equally likely as of now:

- **Scenario I** Unique Games *Conjecture is True, and so are the hardness results shown assuming the conjecture.*
- **Scenario II** Unique Games *Conjecture is False, but showing this needs new algorithmic techniques other than Linear/Semidefinite Programming*
- **Scenario III** Unique Games *Conjecture is False, and can be efficiently solved using existing algorithmic techniques*

There are many questions of independent interest, answers to which should shed more light on the conjecture. By their very nature, answers to these questions would yield insights into limitations of techniques like Linear and Semidefinite programming (SDP). This section outlines the specific research directions that could be pursued.

Perhaps the most pressing open problem to pursue is to show strong SDP integrality gaps for Unique Games. Specifically, this would show that SDPs cannot be used to disprove the **UGC**, thus lending some credibility to the conjecture. While preliminary results in this direction were presented in Chapter 12, the question remains essentially open. Except for k-CSP [157], for any problem whose hardness is shown under **UGC**, no integrality gap is known for $\Omega(\log n)$ rounds of a SDP hierarchy. In particular, it is entirely possible that $\log n$ rounds of a SDP hierarchy not only disproves the **UGC** but also yields better approximations for Max Cut and Minimum Vertex Cover.

Question. Obtain strong SDP gaps for Unique Games, specifically against Lovasz-Schriver, Sherali-Adams or Lasserre hierarchies.

The interplay between Unique Games and Graph Expansion needs to be understood with greater clarity. One of the biggest applications of UG-hardness reductions was the SDP integrality gap for graph expansion obtained by Khot Vishnoi [104]. Recently, Arora .et .al [14] showed that Unique Games problem is easy when the constraint graph is an expander. On a different note, all the hard instances of Unique Games that are constructed so far ([104, 99]) rely on expansion of small sets in graphs.

More specifically, let us define a computational problem related to expansion of small sets in graphs. For a subset of vertices S in a graph $G = (V, E)$ let $Vol(S)$ denote the number of edges with at least one endpoint in S.

Definition 14.1.1. An instance of the problem $\mathsf{GEX}(\beta, 1-\varepsilon, \delta)$ consists of a graph $G = (V, E)$ and the objective is to distinguish between the following two cases:

- There exists a subset of vertices $S \subset V$ satisfying $|S| \leqslant \beta n$ and

$$\frac{E(S, \bar{S})}{Vol(S)} \leqslant \varepsilon$$

  In other words, $1-\varepsilon$ fraction of edges incident on S stay inside the set.

- For every subset S of size at most βn, we have

$$\frac{E(S, \bar{S})}{Vol(S)} \geqslant 1-\delta$$

  That is, at most δ fraction of the edges incident on S stay inside S for every small set S.

There is evidence suggesting that the GEX problem is closely related to Unique Games. Specifically, we wish to make two observations here:

– Given an instance of Unique Games, construct the label extended graph by replacing each vertex v by R different vertices $(v,1),(v,2),\ldots,(v,R)$ for the R different labels. Then connect (v,i) to $(u,\pi_{u\leftarrow v}(i))$ for an edge $e=(u,v)$ in the Unique Games instance. A UG solution satisfying $(1-\varepsilon)$ fraction of the constraints corresponds to a subset of $1/R$ fraction of vertices in the label-extended graph, with $(1-\varepsilon)$ fraction of edges staying inside.

– All hard instances of Unique Games stem from expansion of small sets. For instance, in integrality gap from Khot and Vishnoi [104], there is no good integral solution because every small set of the hypercube (with ε-perturbation edges) expands. The hardest instances of UG are those produced by Khot .et .al [99], in the sense that for a given alphabet size R, these instances have the best possible soundness $1/R^{\varepsilon/2-\varepsilon}$. Even this number $1/R^{\varepsilon/2-\varepsilon}$ is arrived at from the expansion of small sets in Gaussian space.

More formally, let us define the GEX conjecture as follows:

Conjecture 14.1.1. *For every $1>\varepsilon,\delta>0$, there exists β such that the problem $\mathsf{GEX}(\beta,1-\varepsilon,\delta)$ is NP-hard.*

Question. Explore the relation between the GEX conjecture and the Unique Games conjecture. Specifically, show a reduction from either problem to the other.

Like the Unique Games, a $2-to-1$ game is a special case of Label Cover wherein the projections $\pi_{v\to w}$ involved map two labels to w to a single label for v. The 2-to-1 conjecture of Khot [97] asserts that for every δ, for a large enough choice of alphabet size R, it is **NP**-hard to determine whether the value of a 2-to-1 game is equal to 1 or at most δ. Although the Unique Games conjecture is seemingly closely related to the 2-to-1 conjecture, no formal reduction is known in either direction.

Question. Find a reduction from 2-to-1 conjecture to Unique Games conjecture.

Recent work of O'Donnell and Wu [133] on showing hardness using 2-to-1 conjecture could shed some light in this direction.

14.1.2 Algorithmic Issues

In this work, we obtained rounding schemes for the optimal SDP for CSPs. Unfortunately, the running time of the algorithm is doubly exponential in the accuracy required. Specifically, to obtain an approximation that is within η of the optimal ratio, the running time required is $2^{2^{1/poly(\eta)}}\times poly(n)$. For the most part, we are interested in the regime where η is a constant. Hence the doubly exponential running time in η does not matter for theoretical purposes. Yet, to make the rounding schemes applicable practically, it would be great to improve the running time. In fact, a drastic reduction in the running time would have very interesting consequences to determining the value of the Grothendieck constant [137].

Question. Can the running time of the rounding scheme for CSPs be made efficient?

Unlike the case of CSPs, for minimization problems like Multiway Cut and Metric Labeling we did not obtain a rounding scheme for the SDP. For any CSP, the number of constraints satisfied by the optimal assignment is always at least a constant fraction of the set of all constraints. To see this, observe that a random assignment satisfies a constant fraction of the constraints of the CSP. Thus, for a CSP, a constant additive error of η in the reductions does not alter the approximation ratio. However, for minimization problems or GCSP s with negative payoffs, there is no lower bound on the value of the optimum solution. For instance, the value of the optimal Multiway Cut could be $\frac{1}{\log n}$. Clearly, a constant additive error of η would completely alter the approximation ratio.

Question. Obtain optimal rounding schemes for minimization problems like Multiway Cut and Metric Labeling .

14.1.3 More Hardness Results?

It would be very interesting to obtain UG-hardness results for problems like Metric TSP whose approximability has been a long standing open question. Most of the problems considered in this work are either constraint satisfaction problems or are similar to them. Problems like Metric TSP and Steiner Tree are of a completely different flavor, making it a challenge to obtain such reductions from gaps to UG-hardness results.

Question. Can linear programming gaps for Metric Traveling Salesman Problem, Steiner tree, Asymmetric Traveling Salesman Problem, Steiner network design be converted in to UG-hardness results?

In this work, we considered CSPs in which no given constraint is obligatory, but we wish to maximize the number of constraint satisfied. It is interesting to consider mixed CSPs with both *hard* and *soft* constraints. Here the objective would be to maximize the number of *soft* constraints satisfied, while satisfying all the *hard* constraints. Perhaps, the most important example of such a mixed CSP is the Minimum Vertex Cover problem.

Question. Extend the gap to hardness paradigm to CSPs with a hard constraints all of which are required to be satisfied.

Yet another interesting direction to extend the CSP result is to consider the hardness of CSPs under perfect completeness. Specifically, given a CSP instance all of whose constraints can be satisfied simultaneously, how good a solution can a polynomial time algorithm find? For instance, by the classic result of Hastad [86], given a satisfiable 3-CNF formula, it is **NP**-hard to find an assignment satisfying more than 7/8-fraction of the clauses. However, UG-hardness reductions have are inherently limited to not have perfect completeness.

Question. Extend the gap to hardness paradigm to CSPs with perfect completeness, assuming the 2-to-1 conjecture.

The 2-to-1 conjecture by Khot [97] is a related conjecture to **UGC**. Recently, O'Donnell and Wu used the 2-to-1 conjecture to obtain optimal hardness for boolean 3-CSPs with perfect completeness. Their techniques could shed some light on how a general reduction must proceed.

In Chapter 13, we presented **NP**-hardness of satisfying maximum number of equations in a sparse system over reals. Over finite fields, **UGC** yields hardness of maximally sparse (2 variables in an equation) linear equations [99]. The problem remains open over integers.

Question. Are two variable linear equations over integers easy to solve even with noise? Is it possible to design a 2-query dictatorship over integers that yields a hardness for two variable linear equations?

14.1.4 Reductions

UGC implies tight hardness results for classic problems like Minimum Vertex Cover and Max Cut. In other words, **UGC** being true is a *sufficient* condition to show these hardness results. The natural question is whether it also a necessary condition? Hence a very interesting research direction would be the following:

Question. Show **UGC** assuming a hardness result for a well-known problem like Minimum Vertex Cover, Max Cut, Sparsest Cut or Multicut.

The most natural candidate to reduce to **UGC** is Max Cut. Max Cut is already a Unique Games problem over a much smaller alphabet (2). Thus a natural approach would be to use Parallel Repetition Theorem [140] on Max Cut to obtain a Unique Games. In recent years, this problem has sparked a lot of interest, leading to interesting work on the Parallel Repetition Theorem itself [87, 139, 58, 20, 106]. Unfortunately, the Parallel Repetition Theorem necessary for such a reduction was shown to not hold [141, 20].

BIBLIOGRAPHY

[1] Michael Alekhnovich, Sanjeev Arora, and Iannis Tourlakis. Towards strong nonapproximability results in the Lovász–Schrijver hierarchy. In *STOC*, pages 294–303, 2005. 225

[2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995. 6

[3] Noga Alon, Konstantin Makarychev, Yury Makarychev, and Assaf Naor. Quadratic forms on graphs. In *STOC*, pages 486–493, 2005. 194, 195

[4] Noga Alon and Assaf Naor. Approximating the cut-norm via grothendieck's inequality. *SIAM J. Comput*, 35(4):787–803, 2006. 7, 193, 194

[5] E. Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 109:237–260, 1998. 255

[6] Aaron Archer, Jittat Fakcharoenphol, Chris Harrelson, Robert Krauthgamer, Kunal Talwar, and Éva Tardos. Approximate classification via earthmover metrics. In *SODA '04: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1079–1087, 2004. 27, 147, 149

[7] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer System Sciences*, 54(2):317–331, 1997. 255

[8] Sanjeev Arora. Polynomial time approximation schemes for euclidean TSP and other geometric problems. In *FOCS*, pages 2–11, 1996. 8

[9] Sanjeev Arora, Eli Berger, Elad Hazan, Guy Kindler, and Muli Safra. On non-approximability for quadratic programs. In *FOCS*, pages 206–215, 2005. 195

[10] Sanjeev Arora, Béla Bollobás, and László Lovász. Proving integrality gaps without knowing the linear program. In *FOCS*, pages 313–322, 2002. 225

[11] Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(1):19–51, 2006. 225

[12] Sanjeev Arora and Eden Chlamtac. New approximation guarantee for chromatic number. In Jon M. Kleinberg, editor, *STOC*, pages 215–224. ACM, 2006. 6, 7, 225

[13] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 227–236. ACM, 2007. 25, 58

[14] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 21–28. ACM, 2008. 9, 238, 243, 244, 275

[15] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *JACM: Journal of the ACM*, 45, 1998. 8, 24

[16] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC-04)*, pages 222–231, New York, June 13–15 2004. ACM Press. 7, 9, 60, 81, 225

[17] Per Austrin. Balanced max 2-sat might not be the hardest. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 189–197. ACM, 2007. 9, 10, 93, 224

[18] Per Austrin. Towards sharp inapproximability for any 2-csp. In *FOCS*. IEEE, 2007. 11, 93

[19] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. In *IEEE Conference on Computational Complexity*, pages 249–258. IEEE Computer Society, 2008. 9, 208, 209

[20] Boaz Barak, Moritz Hardt, Ishay Haviv, Anup Rao, Oded Regev, and David Steurer. Rounding parallel repetitions of unique games. In *FOCS*, pages 374–383. IEEE Computer Society, 2008. 278

[21] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998. 149

[22] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, June 1998. 92, 254

[23] Eli Ben-Sasson, Madhu Sudan, Salil P. Vadhan, and Avi Wigderson. Randomness-efficient low degree tests and short PCPs via epsilon-biased sets. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 612–621, 2003. 256

[24] Bonnie Berger and Peter W. Shor. Tight bounds for the Maximum Acyclic Subgraph problem. *J. Algorithms*, 25(1):1–18, 1997. 165

[25] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. 256

[26] Boppana and Magnús M. Halldorsson. Approximating maximum independent sets by excluding subgraphs. *BIT: BIT*, 32, 1992. 9

[27] Christer Borell. Geometric bounds on the Ornstein-Uhlenbeck velocity process. *Z. Wahrsch. Verw. Gebiete*, 70(1):1–13, 1985. 46

[28] Joshua Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. Rank bounds and integrality gaps for cutting planes procedures. *Theory of Computing*, 2(1):65–90, 2006. 225

[29] B.W.Johnson and Johnson Lindenstrauss. Basic concepts in the geometry of Banach spaces. *Handbook of the geometry of Banach spaces, North Holland, Amsterdam*, 1:1–84, 2001. 194

[30] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *SODA '01: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 8–16, 2001. 27, 147, 149

[31] Moses Charikar, Venkatesan Guruswami, and Rajsekar Manokaran. Every permutation csp of arity 3 is approximation resistant. *To appear in IEEE Conference on Computational Complexity*, 2009. 166, 169

[32] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 62–68. SIAM, 2007. 6, 9, 26, 28, 78, 208, 219

[33] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. On the advantage over random for maximum acyclic subgraph. In *FOCS*, pages 625–633. IEEE Computer Society, 2007. 7, 165, 166, 168, 173

[34] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for Sherali–Adams relaxations. In *STOC*, 2009. To appear. 224, 225, 226

[35] Moses Charikar, Yury Makarychev, and Konstantin Makarychev. Near-optimal algorithms for unique games. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2006. 6, 9, 26, 60, 78, 243, 274

[36] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending grothendieck's inequality. In *FOCS*, pages 54–60. IEEE Computer Society, 2004. 6, 7, 26, 78, 194

[37] Sourav Chatterjee. A simple invariance theorem. *arxiv:math/0508213v1.*, 2005. 43

[38] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006. 9, 224

[39] Chandra Chekuri, Sanjeev Khanna, Joseph (Seffi) Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. *SIAM J. Disc. Math.*, 18:608–625, 2005. 27, 55, 56, 147, 149, 153, 154, 155

[40] Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2007. 6, 7, 225

[41] Eden Chlamtac, Yury Makarychev, and Konstantin Makarychev. How to play unique games using embeddings. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006. 6, 9, 26, 59, 274

[42] Eden Chlamtac and Gyanit Singh. Improved approximation guarantees through higher levels of SDP hierarchies. In Ashish Goel, Klaus Jansen, José D. P. Rolim, and Ronitt Rubinfeld, editors, *APPROX-RANDOM*, volume 5171 of *Lecture Notes in Computer Science*, pages 49–62. Springer, 2008. 6, 7, 225

[43] Benny Chor and Madhu Sudan. A geometric approach to betweenness. *SIAM Journal on Discrete Mathematics*, 11(4):511–523, November 1998. 7

[44] Benny Chor and Madhu Sudan. A geometric approach to betweenness. *SIAM Journal on Discrete Mathematics*, 11(4):511–523, November 1998. 166

[45] Julia Chuzhoy and Joseph (Seffi) Naor. The hardness of metric labeling. *SIAM J. Comput.*, 36(5):1376–1386, 2006. 147, 150

[46] Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 48–52, 1998. 21, 27, 54, 147, 149

[47] Elias Dahlhaus, David Johnson, Christos Papadimitriou, Paul Seymour, and Mihalis Yannakakis. The complexity of multiway cuts (extended abstract). In *STOC '92: Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 241–251, 1992. 27, 147, 149

[48] Nikhil R. Devanur, Subhash Khot, Rishi Saket, and Nisheeth K. Vishnoi. Integrality gaps for sparsest cut and minimum linear arrangement problems. In *STOC*, pages 537–546, 2006. 225

[49] I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM J. Computing*, 34(5):1129–1146, 2005. 209

[50] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *annmath*, 162:439–485, 2005. 8, 9, 93

[51] Lars Engebretsen and Jonas Holmerin. More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP. In Volker Diekert and Bruno Durand, editors, *STACS*, volume 3404 of *Lecture Notes in Computer Science*, pages 194–205. Springer, 2005. 208, 209, 211

[52] Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998. 165

[53] Jittat Fakcharoenphol, Chris Harrelson, Satish Rao, and Kunal Talwar. An improved approximation algorithm for the 0-extension problem. In *SODA '03: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 257–265, 2003. 27, 147, 148, 149

[54] Feige and Goemans. Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. In *ISTCS: 3rd Israeli Symposium on the Theory of Computing and Systems*, 1995. 6, 26, 78

[55] U. Feige and G. Schechtman. On the optimality of the random hyperplane rounding technique for MAX CUT. *Random Structures and Algorithms*, 20(3):403–440, 2002.

[56] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. 9, 10

[57] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *JCSS: Journal of Computer and System Sciences*, 57, 1998. 9

[58] Uriel Feige, Guy Kindler, and Ryan O'Donnell. Understanding parallel repetition requires understanding foams. In *IEEE Conference on Computational Complexity*, pages 179–192. IEEE Computer Society, 2007. 278

[59] Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput*, 31(4):1090–1118, 2002.

[60] Uriel Feige and Daniel Reichman. On the hardness of approximating Max-Satisfy. *Information Processing Letters*, 97(1):31–35, 2006. 9, 255

[61] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572, 2006. 254, 255

[62] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18:67–81, 1997. 6, 7, 26, 78

[63] Alan Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999. 81, 193

[64] Konstantinos Georgiou, Avner Magen, Toniann Pitassi, and Iannis Tourlakis. Integrality gaps of 2 - o(1) for vertex cover SDPs in the lovész-schrijver hierarchy. In *FOCS*, pages 702–712. IEEE Computer Society, 2007. 223, 225

[65] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. 6, 9, 16, 17, 26, 78, 111, 240

[66] Michel X. Goemans and David P. Williamson. Approximation algorithms for M_{AX}-3-C_{UT} and other problems via complex semidefinite programming. *J. Comput. Syst. Sci*, 68(2):442–470, 2004. 6, 26, 78

[67] W. T. Gowers and J. Wolf. The true complexity of a system of linear equations. *arXiv:math.NT/0711.0185*, 2007. 217

[68] B. Green and T. Tao. Linear equations in primes. *arXiv:math.NT/0606088v1*, 2006. 216, 217

[69] Alexander Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques. *Bol. Soc. Mat*, 8:1–79, 1953. 193

[70] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, Berlin-Heidelberg, 1988. 6

[71] Anupam Gupta and Éva Tardos. A constant factor approximation algorithm for a class of classification problems. In *STOC '00: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 652–658, 2000. 27, 147, 149

[72] Anupam Gupta and Kunal Talwar. Approximating unique games. In *SODA*, pages 99–106. ACM Press, 2006. 9, 274

[73] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard : Inapproximability of maximum acyclic subgraph. *In FOCS '08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. 93, 224, 226

[74] Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. In *FOCS*, pages 543–552. IEEE Computer Society, 2006. 254, 255

[75] Venkatesan Guruswami and Prasad Raghavendra. Constraint satisfaction over a non-boolean domain: Approximation algorithms and unique-games hardness. In Ashish Goel, Klaus Jansen, José D. P. Rolim, and Ronitt Rubinfeld, editors, *APPROX-RANDOM*, volume 5171 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2008. 9

[76] Magnus Halldorsson. Approximations of weighted indpendent set and hereditary subset problems. *J. Gaph Algorithms Appl.*, 4(1), 2000. 255

[77] Magnús M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45(1):19–23, 25 January 1993. 9

[78] Eran Halperin and Uri Zwick. Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. *J. Algorithms*, 40(2):184–211, 2001. 6, 26, 78

[79] Gustav Hast. Beating a random assignment. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *APPROX-RANDOM*, volume 3624 of *Lecture Notes in Computer Science*, pages 134–145. Springer, 2005. 6, 26, 78, 208

[80] Gustav Hast. Beating a random assignment - approximating constraint satisfaction problems. 2005. 209

[81] Johan Håstad. Every 2-csp allows nontrivial approximation. *Computational Complexity*. 167

[82] Johan HÅstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, FOCS'96 (Burlington, VT, October 14-16, 1996)*, pages 627–636, Los Alamitos-Washington-Brussels-Tokyo, 1996. IEEE Computer Society, IEEE Computer Society Press. 92

[83] Johan Håstad. On the approximation resistance of a random predicate. In *APPROX-RANDOM*, pages 149–163, 2007. 209

[84] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and PCP. *Random Struct. Algorithms*, 22(2):139–160, 2003. 209

[85] Johann Hastad. Clique is hard to approximate within $n^{1-epsilon}$. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 1996. 9

[86] Johann Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 9, 12, 21, 30, 92, 208, 209, 254, 256, 261, 264, 267, 269, 270, 277

[87] Holenstein. Parallel repetition: Simplifications and the no-signaling case. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2007. 278

[88] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001. 78

[89] A. Karazanov. On the minimal number of arcs of a digraph meeting all its directed cutsets. *Graph Theory Newsletters*, 8, 1979. 165

[90] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998. 6

[91] David R. Karger, Philip Klein, Cliff Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *STOC '99: Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 668–678, 1999. 148, 149

[92] Howard Karloff, Subhash Khot, Aranyak Mehta, and Yuval Rabani. On earthmover distance, metric labeling, and 0-extension. In *STOC '06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 547–556, 2006. 27, 147, 148, 149, 150

[93] Howard Karloff and Uri Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, FOCS'97 (Miami Beach, Florida, October 20-22, 1997)*, pages 406–415, Los Alamitos-Washington-Brussels-Tokyo, 1997. IEEE Computer Society, IEEE Computer Society Press. 6, 26, 78

[94] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations. New York: Plenum*, pages 85–103, 1972. 21, 165

[95] Alexander Karzanov. Minimum 0-extension of graph metrics. *Europ. J. Combinat.*, 19:71–101, 1998. 147

[96] Alexander Karzanov. A combinatorial algorithm for the minimum $(2, r)$-metric problem and some generalizations. *Combinatorica*, 18:549–569, 1999. 147

[97] Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775, 2002. 8, 92, 276, 277

[98] Subhash Khot. Inapproximability results via long code based PCPs. *SIGACT News*, 36(2), June 2005. 92, 254

[99] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007. 8, 9, 10, 26, 29, 89, 92, 93, 122, 138, 160, 223, 224, 226, 227, 238, 239, 254, 275, 276, 278

[100] Subhash Khot and Assaf Naor. Approximate kernel clustering. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008. 7

[101] Subhash Khot and Ryan O'Donnell. SDP gaps and UGC-hardness for MAXCUT-GAIN. In *FOCS*, pages 217–226. IEEE Computer Society, 2006. 11, 195

[102] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within /spl epsi/. In *Annual IEEE Conference on Computational Complexity (formerly Annual Conference on Structure in Complexity Theory)*, volume 18, 2003. 150

[103] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-ε. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. 9, 24, 25

[104] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l_1. In *FOCS*, pages 53–62. IEEE Computer Society, 2005. 9, 10, 26, 89, 93, 113, 122, 150, 164, 167, 223, 224, 225, 226, 227, 228, 229, 233, 238, 275, 276

[105] Guy Kindler, Assaf Naor, and Gideon Schechtman. The UGC hardness threshold of the ℓ_p Grothendieck problem. In *SODA*, pages 64–73, 2008. 194

[106] Guy Kindler, Ryan O'Donnell, Anup Rao, and Avi Wigderson. Spherical cubes and rounding in high dimensions. In *FOCS*, pages 189–198. IEEE Computer Society, 2008. 278

[107] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002. 21, 27, 147, 148, 149

[108] Robert Krauthgamer, James R. Lee, Manor Mendel, and Assaf Naor. Measured descent: A new embedding method for finite metrics. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 434–443, 2004. 21, 149

[109] J. L. Krivine. Constantes de Grothendieck et fonctions de type positif sur les sphères. *Adv. Math.*, 31:16–30, 1977-1978. 193

[110] J. B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In K. Aardel and A. M. H. Gerards, editors, *IPCO 2001*, volume 2081 of *Lecture Notes in Computer Science*, pages 293–303. Springer, Berlin, 2001. 7, 63, 225

[111] Lap Chi Lau and Mohit Singh. Approximating minimum bounded degree spanning trees to within one of optimal. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 661–670. ACM, 2007. 78

[112] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for $0-1$ programming. *Mathematics of Operations Research*, 28(3):470–496, 2003. 7, 225

[113] James R. Lee and Assaf Naor. Extending lipschitz functions via random metric partitions. *Inventiones Mathematicae*, 160(1):59–95, 2005. 21, 149

[114] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *IPCO: 9th Integer Programming and Combinatorial Optimization Conference*, 2002. 6, 9, 10, 26, 62, 78

[115] Johnson Lindenstrauss and A. Pelczyński. Absolutely summing operators in L_p spaces and their applications. *Studia Math*, 29:275–326, 1968. 193

[116] Nati Linial and Adi Shraibman. Lower bounds in communication complexity based on factorization norms. In *STOC*, pages 699–708, 2007. 193

[117] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979. 6

[118] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991. 7, 63, 225

[119] Cláudio Lucchesi and Daniel H. Younger. A minimax theorem for directed graphs. *Journal London Math. Society*, 17:369–374, 1978. 165

[120] Konstantin Makarychev and Yury Makarychev. Personal communication. 2008. 216

[121] Rajsekar Manokaran, Seffi Naor, Prasad Raghavendra, and Roy Schwartz. Sdp gaps and ugc hardness for multiway cut, 0-extension and metric labelling. *In STOC '08: Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008. 93, 224, 226

[122] Matuura and Matsui. 0.863-approximation algorithm for MAX DICUT. In *APPROX: International Workshop on Approximation Algorithms for Combinatorial Optimization*, 2001. 6, 26, 78

[123] A. Megretski. Relaxation of quadratic programming in operator theory and systems analysis. *Systems, approximation, singular integral operators and related topics*, 2001. 194

[124] E. Mossel. Gaussian bounds for noise correlation of functions. *In FOCS '08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. 37, 44, 45, 124, 133, 168, 194

[125] Elchannan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005. 43, 45, 47, 107, 171, 197, 198, 235

[126] A. Nemirovski, C. Roos, and T. Terlaky. On maximization of quadratic form over intersection of ellipsoids with common center. *Mathematical Programming*, 86(3):463–473, 1999. 194

[127] A. S. Nemirovsky and Y. Nesterov. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994. 6

[128] Y. E. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Methods and Software*, 9:141–160, 1998. 7

[129] Alantha Newman. Approximating the maximum acyclic subgraph. Master's thesis, MIT, June 2000. 164, 165

[130] Alantha Newman. Cuts and orderings: On semidefinite relaxations for the linear ordering problem. In *APPROX-RANDOM*, pages 195–206, 2004. 164

[131] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer-Verlag, New York, 1999.

[132] Ryan O'Donnell and Yi Wu. An optimal sdp algorithm for max-cut, and equally optimal long code tests. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 335–344. ACM, 2008. 117, 223

[133] Ryan O'Donnell and Yi Wu. 3-bit dictator testing : 1 vs 5/8. *In SODA '09: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009. 276

[134] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991. 165

[135] S. Poljak and Z. Tuza. The expected relative error of the polyhedral approximation of the max-cut problem. *Operations Research Letters*, 16:191–198, 1994. 6

[136] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? *In STOC '08: Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 245–254, 2008. 81, 93, 150, 166, 167, 168, 169, 175, 209, 224, 226, 274

[137] Prasad Raghavendra and David Steurer. Towards computing the grothendieck constant. *In SODA '09: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009. 224, 276

[138] V. Ramachandran. Finding a minimum feedback arc set in reducible flow graphs. *J. Algorithms*, 9:299–313, 1988. 165

[139] Anup Rao. Parallel repetition in projection games and a concentration bound. In Richard E. Ladner and Cynthia Dwork, editors, *STOC*, pages 1–10. ACM, 2008. 278

[140] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998. 24, 278

[141] Ran Raz. A counterexample to strong parallel repetition. In *FOCS*, pages 369–373. IEEE Computer Society, 2008. 278

[142] J. Reeds. A new lower bound on the real Grothendieck constant. *Unpublished Manuscript, available at http://www.dtc.umn.edu/ reedsj*, 1993. 193, 195

[143] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *STOC*, pages 191–199, 2000. 9, 208, 211

[144] Alex Samorodnitsky and Luca Trevisan. Gowers uniformity, influence of variables, and PCPs. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2006. 9, 28, 208, 209, 210, 211, 212, 226

[145] Grant Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *IEEE Foundations of Computer Science*. IEEE Computer Society, 2008. 223, 225

[146] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. A linear round lower bound for Lovász-Schrijver SDP relaxations of vertex cover. In *IEEE Conference on Computational Complexity*, pages 205–216, 2007. 224, 225

[147] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. A linear round lower bound for lovasz-schrijver SDP relaxations of vertex cover. In *IEEE Conference on Computational Complexity*, pages 205–216. IEEE Computer Society, 2007. 225, 226

[148] Paul Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995. 165

[149] H. D. Sherali and W. P. Adams. A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Appl. Math.*, 52(1):83–106, 1994. 7

[150] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990. 7, 63, 223, 225

[151] David Steurer. Fast sdp algorithms for constraint satisfaction problems. 2009. 12, 51, 58, 65

[152] Iannis Tourlakis. Towards optimal integrality gaps for hypergraph vertex cover in the lovász–Schrijver hierarchy. In *APPROX-RANDOM*, pages 233–244, 2005. 225

[153] Iannis Tourlakis. New lower bounds for vertex cover in the Lovász-Schrijver hierarchy. In *IEEE Conference on Computational Complexity*, pages 170–182, 2006. 225

[154] Luca Trevisan. Parallel approximation algorithms by positive linear programming. *Algorithmica*, 21(1):72–88, 1998. 208

[155] Luca Trevisan. Approximation algorithms for unique games. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 197–205. IEEE Computer Society, 2005. 9, 243, 274

[156] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM J. Comput*, 29(6):2074–2097, 2000. 26, 78

[157] Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *STOC*, 2009. To apperar. 223, 224, 226, 275

[158] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin-Heidelberg-New York-Barcelona-Hong Kong-London-Milan-Paris-Singapur-Tokyo, 2001. 6, 78

[159] V.I.Rotar'. Limit theorems for polylinear forms. 43

[160] S. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, 1996.

[161] U. ZWICK. Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *stoc99*, pages 679–687, 1999. 6, 26, 78

[162] Uri Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'98 (San Francisco, California, January 25-27, 1998)*, pages 201–210, Philadelphia, PA, 1998. ACM SIGACT, SIAM, Society for Industrial and Applied Mathematics. 6, 26, 78

[163] Uri Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210, 25–27 January 1998. 209

[164] Uri Zwick. Finding almost-satisfying assignments. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC-98)*, pages 551–560, New York, May 23–26 1998. ACM Press. 6, 26, 78

Appendix A

PROBLEM DEFINITIONS

Problem 12 (Label Cover)**.** An instance of Label Cover (c, s) is given by $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, [R], \Pi)$ consists of a bipartite graph over node sets $\mathcal{V}_\Phi$,$\mathcal{W}_\Phi$ with the edges E between them, such that all nodes in $\mathcal{V}_\Phi$ are of the same degree. Also part of the instance is a set of labels $[R]$, and a set of mappings $\Pi = \{\pi_{v \leftarrow w} : [R] \to [R]\}$ for each edge $e = (w, v) \in E$. An assignment A of labels to vertices is said to satisfy an edge $e = (w, v)$, if $\pi_{v \leftarrow w}(A(w)) = A(v)$. The problem is to distinguish between the following two cases:

- There exists an assignment A that satisfies at least a fraction c of the edge constraints Π.
- Every assignment satisfies less than a fraction s of the constraints in Π.

Problem 13 ($\mathsf{E2Lin}_q$)**.** Given a variable set $\mathcal{V}$ and a system of linear equations over the finite field $\mathbb{F}_q$, with equation of the form $x_i - x_j = c_{ij}$ for variables $i, j \in \mathcal{V}$, the goal is to find an $\mathbb{F}_q$-assignment x to $\mathcal{V}$ that satisfies the maximum number of equations.

Note that Max Cut is a slight generalization of $\mathsf{E2Lin}_2$.

Problem 14 (Γ-Max-2Lin)**.** Given a variable set $\mathcal{V}$ and a list of constraints of the form $x_i x_j^{-1} = c_{ij}$ over the group Φ for variables $i, j \in \mathcal{V}$, i, j, the goal is to find a Φ-assignment to $\mathcal{V}$ so as to maximize the number of satisfied constraints.

Note that $\mathsf{E2Lin}_q$ is the same problem as Γ-Max-2Lin for $\Phi = \mathbb{F}_q$. We sometimes use the notation Γ-Max-2Lin(k) to refer to the more general problem where a group Φ of order k is given as part of the input.

Problem 15 (Unique Games (R))**.** Given a variable set $\mathcal{V}$ and a list of constraints of the form $x_u = \pi_{u \leftarrow v}(x_v)$ where $u, v \in \mathcal{V}$ are two variables and $\pi_{u \leftarrow v}$ is a permutation of $[R]$, the goal is to find a $[R]$-assignment to $\mathcal{V}$ so as to maximize the number of satisfied constraints.

Problem 16 (Unique Games $(R, 1 - \gamma, \delta)$)**.** Given a bipartite unique games instance $\Phi = (\mathcal{W}_\Phi \cup \mathcal{V}_\Phi, E, \Pi = \{\pi_{v \leftarrow w} : [R] \to [R] \mid e = (w, v) \in E\}, [R])$ with number of labels R, distinguish between the following two cases:

- $(1 - \gamma)$-strongly satisfiable instances: There exists an assignment A of labels such that for $1 - \delta$ fraction of vertices $w \in \mathcal{W}_\Phi$ are strongly satisfied, i.e., *all* the edges (w, v) are satisfied.
- Instances that are not δ-satisfiable: No assignment satisfies more than a δ-fraction of the edges E.

Problem 17. A METRIC LABELING problem is specified as $\Lambda = (\mathsf{L}, d)$ where d is a metric over the set of labels L.

We will use q to denote the number of labels $|\mathsf{L}|$

Problem 18 (Λ-METRIC LABELING)**.** An instance $\Im = (\mathcal{V}, \mathcal{E}, \{\mathsf{L}(v)\}_{v\in\mathcal{V}})$ of the Λ-METRIC LABELING problem consists of a set of vertices $\mathcal{V}$, a probability distribution $\mathcal{E}$ over pairs from $\mathcal{V} \times \mathcal{V}$ (equivalent to edges with weights) and a family of subsets $\{\mathsf{L}(v)\}_{v\in\mathcal{V}}$ of L. A valid labeling is a mapping $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each vertex, $v \in \mathcal{V}$, $\mathcal{L}(v)$ belongs to $\mathsf{L}(v)$. The cost of a labeling $\mathcal{L}$, $\mathrm{val}_{\Im}(\mathcal{L})$, is

$$\mathop{\mathbb{E}}_{(u,v)=e\in\mathcal{E}} d(\mathcal{L}(u), \mathcal{L}(v)).$$

The optimum value of the instance, $\mathrm{opt}(\Im)$, is the minimum cost labeling for the instance.

An important special case of the Λ-METRIC LABELING problem is the Λ-ZERO-EXTENSION problem defined below.

Problem 19 (Λ-ZERO-EXTENSION)**.** An instance $\Im = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of Λ-ZERO-EXTENSION problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of terminals $\mathsf{L} \subset \mathcal{V}$ with a metric d on them. The objective is to assign each vertex v a terminal $\mathcal{L}(v) \in \mathsf{L}$ such that the following cost is minimized:

$$\mathop{\mathbb{E}}_{(u,v)=e\in\mathcal{E}} d(\mathcal{L}(u), \mathcal{L}(v)).$$

The value of the instance, $\mathrm{opt}(\Im)$ is the minimum cost labeling for the instance.

Observe that a valid solution to the above problem consists of a labeling $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each terminal $t \in \mathsf{L}$, $\mathcal{L}(t) = t$. This corresponds to Λ-METRIC LABELING over the graph $(\mathcal{V}, \mathcal{E})$ with the family of sets $\{\mathsf{L}(v)\}_{v\in\mathcal{V}}$ defined as,

$$\mathsf{L}(v) = \begin{cases} \{v\} & \text{if } v \in \mathsf{L} \\ \mathsf{L} & \text{otherwise} \end{cases}$$

Problem 20 (MULTIWAY CUT)**.** An instance $\Im = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of MULTIWAY CUT problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of terminals $\mathsf{L} \subset \mathcal{V}$. The objective is to delete a set of edges of minimum weight so as to separate every pair of terminals.

The MULTIWAY CUT problem can be formulated as a labeling problem (with a uniform metric) as follows: A valid multiway cut corresponds to a labeling $\mathcal{L} : \mathcal{V} \to \mathsf{L}$ such that for each terminal $t \in \mathsf{L}$, $\mathcal{L}(t) = t$. The cost of such a labeling $\mathcal{L}$, $\mathrm{val}_{\Im}(\mathcal{L})$ is given by $\mathbb{E}_{(u,v)\in\mathcal{E}}\left[\mathbf{1}[\mathcal{L}(u) \neq \mathcal{L}(v)]\right]$. The optimum value of the instance $\mathrm{opt}(\Im)$ is the minimum cost labeling for the instance.

A special case of MULTIWAY CUT problem is the q-WAYCUT for a positive integer q.

Problem 21 (q-WAYCUT)**.** An instance $\Im = (\mathcal{V}, \mathcal{E}, \mathsf{L})$ of q-WAYCUT problem consists of a weighted graph $(\mathcal{V}, \mathcal{E})$, along with a set of q terminals $\mathsf{L} \subset \mathcal{V}$. The objective is to delete a set of edges of minimum weight so as to separate every pair of terminals.

Problem 22 ($K_{N,N}$-QuadraticProgramming)**.** Given an $m \times n$ matrix $A = (a_{ij})$, compute the optimal value of the following optimization problem,

$$\text{opt}(A) \stackrel{\text{def}}{=} \max \sum_{ij} a_{ij} x_i y_j \,,$$

where the maximum is over all $x_1, \ldots, x_m \in [-1, 1]$ and $y_1, \ldots, y_n \in [-1, 1]$. Note that the optimum value opt(A) is always attained for numbers with $|x_i| = |y_j| = 1$.

Problem 23 (Max Cut)**.** Given a graph $G = (V, E)$ with vertices $V = \{v_1, \cdots, v_n\}$ and edges E, find a partition $S \cup \bar{S} = V$ of the set of vertices that maximizes the number of edges *cut* by the partition. An edge $e = (v_i, v_j)$ is cut, if $v_i \in S$ and $v_j \in \bar{S}$ or vice versa.

Problem 24 (Minimum Vertex Cover)**.** An instance of Minimum Vertex Cover problem consists of a graph $G = (V, E)$ over a set of vertices V and edges E. A vertex cover is a set of vertices S, such that every edge in the graph, has one of its endpoints in the set S. The goal is to find a vertex cover S with the minimum number of vertices.

Problem 25 (b-Balanced Separator)**.** Given a graph G on vertex set V, the goal is to find a set $S \subseteq V$ with $b \leqslant {}^{|S|}/_{|V|} \leqslant {}^{1}/_{2}$ so as to minimize the fraction of edges cut by S.

Problem 26 (Sparsest Cut)**.** Given a weighted graph $G = (V, E)$ with weight $\{w_{ij}\}_{i,j \in V}$ and a set of demands $\{d_{ij}\}_{i,j \in V}$, compute the cut $(S, \bar{S})$ of the vertices, that minimizes the ratio of the total weight of edges cut to total demands that are separated. Formally, the goal is to

$$\text{Minimize} \quad \frac{\sum_{i \in S, j \in \bar{S}} w_{ij}}{\sum_{i \in S, j \in \bar{S}} d_{ij}}$$

Problem 27 (Max 2-Sat)**.** Given a set of 2-CNF clauses of the form $\ell_i \vee \ell_j$ where ℓ_i, ℓ_j are literals (variables or their negations) over a set of variables V, find an assignment to the variables that satisfies the maximum number of clauses.

Max 3-Sat, and $Max - k - SAT$ are the corresponding problems on 3-CNF and k-CNF clauses respectively.

Problem 28 (Maximum Acyclic Subgraph)**.** Given a weighted directed graph $G = (V, E)$ with weights $\{w_{ij}\}_{i,j \in V}$, find an ordering of the vertices such that the maximum weight of edges are in the *forward direction*. Formally, find an ordering $\mathcal{O} : V \to [n]$ of the n vertices V that maximizes the following:

$$\sum_{\mathcal{O}(i) < \mathcal{O}(j)} w_{ij}$$

VITA

Prasad Raghavendra obtained his bachelors degree at Indian Institute of Technology, Madras, and is currently a Phd candidate at University of Washington. His research spans several areas of theoretical computer science including approximation algorithms and inapproximability results for NP-hard optimization problems, error correcting codes, metric embeddings, computational learning and complexity. He welcomes your comments to `prasad@cs.washington.edu`.