

# On the Power of Lasserre SDP Hierarchy

*Ning Tan*



Electrical Engineering and Computer Sciences  
University of California at Berkeley

Technical Report No. UCB/EECS-2015-236

<http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-236.html>

December 15, 2015

Copyright © 2015, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**On the Power of Lasserre SDP Hierarchy**

by

Ning Tan

A dissertation submitted in partial satisfaction of the  
requirements for the degree of  
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Prasad Raghavendra, Chair

Professor Satish Rao

Professor Nikhil Srivastava

Fall 2015

# **On the Power of Lasserre SDP Hierarchy**

Copyright 2015  
by  
Ning Tan

## Abstract

On the Power of Lasserre SDP Hierarchy

by

Ning Tan

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Prasad Raghavendra, Chair

Constraint Satisfaction Problems (CSPs) are a class of fundamental combinatorial optimization problems that have been extensively studied in the field of approximation algorithms and hardness of approximation. In a ground breaking result, Raghavendra showed that assuming Unique Games Conjecture, the best polynomial time approximation algorithm for all Max CSPs are given by a family of basic standard SDP relaxations. With Unique Games Conjecture remains as one of the most important open question in the field of Theoretical Computer Science, it is natural to ask whether hypothetically stronger SDP relaxations would be able to achieve better approximation ratio for Max CSPs and their variants.

In this work, we study the power of Lasserre/Sum-of-Squares SDP Hierarchy. First part of this work focuses on using Lasserre/Sum-of-Squares SDP Hierarchy to achieve better approximation ratio for certain CSPs with global cardinality constraints. We present a general framework to obtain Sum-of-Squares SDP relaxation, round SDP solution and analyze the rounding algorithm for CSPs with global cardinality constraints. To demonstrate the approach, we show that one could use Sum-of-Squares SDP to achieve a 0.85-approximation algorithm for Max Bisection problem, improving on the previously best known 0.70 ratio.

In the second part of this work, we study the computational power of general symmetric relaxations. Specifically, we show that Lasserre/Sum-of-Squares SDP solution achieves the best possible approximation ratio for all Max CSPs among all symmetric SDP relaxations of similar size. This result gives the first lower bounds for symmetric SDP relaxations of Max CSPs, and indicates that the Sum-of-Squares SDP is indeed the "right" SDP relaxation for this class of problems.

To my parents.

# Contents

<b>Contents</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The Relaxation and Rounding Paradigm for Designing Approx- imation Algorithms . . . . .	2
1.2 Relaxation Techniques and Hierarchies . . . . .	4
1.3 Contribution of the Thesis . . . . .	7
<b>2 Preliminary and Organization of Thesis</b>	<b>8</b>
2.1 Definitions and Terminologies . . . . .	8
2.2 Relaxation and Rounding of Combinatorial Optimization Problems	10
2.3 Constraint Satisfaction Problems . . . . .	16
2.4 Results and Organization . . . . .	18
<b>3 Mathematical Notations and Tools</b>	<b>20</b>
3.1 Sets and Families . . . . .	20
3.2 Linear Algebra . . . . .	20
3.3 Convex Optimization and Semidefinite Programming . . . . .	21
3.4 Information Theory, Entropy and Mutual Information . . . . .	22
<b>4 LP and SDP Relaxations</b>	<b>26</b>
4.1 Introduction . . . . .	26
4.2 Generic LP and SDP Relaxation For Max-CSPs . . . . .	27
4.3 Sum-of-Squares SDP Hierarchy . . . . .	30
<b>5 An Improved Approximation Algorithm for MAX BISECTION</b>	<b>34</b>
5.1 Introduction . . . . .	34
5.2 Statement of Results . . . . .	36
5.3 Overview of Techniques . . . . .	36
5.4 Preliminaries . . . . .	39
5.5 Globally Uncorrelated SDP Solutions . . . . .	43
5.6 Rounding Scheme for MAX BISECTION . . . . .	46

5.7	Analysis of Cut Value . . . . .	50
5.8	Dictatorship Tests from Globally Uncorrelated SDP Solutions . .	54
<b>6</b>	<b>Optimal Symmetric SDP Relaxation for Max-CSPs</b>	<b>62</b>
6.1	Introduction . . . . .	62
6.2	Statement of Results . . . . .	63
6.3	Preliminaries . . . . .	65
6.4	Symmetric SDPs . . . . .	67
6.5	Instance Optimal Symmetric LP for Traveling Salesman Problem	74
<b>7</b>	<b>Future Directions</b>	<b>78</b>
	<b>Bibliography</b>	<b>80</b>



## Acknowledgments

First and foremost, I am greatly indebted to my advisor Prasad Raghavendra. Prasad is truly the nicest and most brilliant person that I've ever met in my life and I was extremely lucky to have become his first student. I was always amazed by Prasad's brilliance, technical mastery, wide knowledge of almost everything and his ability to look at things from an angle that I could never have imagined. Prasad spent countless hours of his time to go through every ridiculous ideas of mine and was always be able to point out a new direction after we hit the dead end. Over the years, I immensely enjoyed the conversations that I had with Prasad, both technical and non-technical, through which I learnt a great deal. For all these and the endless support and encouragement he gave me during the years, as well as a lot more left out – Thank you Prasad, I could never asked for a better advisor!

I am very grateful to Satish Rao, Luca Trevisan and Nikhil Srivastava for serving on my qualifying exam committee as well as my thesis committee. Thank you for taking time from your busy schedules to make my thesis defense.

I would also like to thank Robin Thomas for admitting me into the prestigious ACO program at Georgia Tech as well as his support throughout the years that I have spent at Georgia Tech. He provided tremendous help and support to some of my decisions and this thesis couldn't have been done without him.

Throughout the years I luckily had the opportunity to interact with several faculty members either through courses or research projects. They have taught me the beauty of math and computer science and also motivated me to pursue my study in theoretical computer science. I would like to thank them all.

Special thanks to my girlfriend Yijie Wang for her support and encouragement, she has been a great inspiration to me. Also I'm greatly thankful for a few of my middle school friends including Fei Meng, Yue Li, Nan Wang, Mengke Xing and Tie Zan. They have been beside me through all my highs and lows for more than 15 years and I am extremely grateful for knowing you guys.

Also thanks to the friends that I have met during my stay at Georgia Tech as well as UC Berkeley for making my life colorful: Albert Bush, Jonah Brown-Cohen, Aviad Rubinstein, Tselil Schramm, Jarett Schwartz, Ruidong Wang, Ruodu Wang, Qianyi Wang, Xiaolin Wang, Benjamin Weitz and Yi Xiao. Sincere apology to those that I missed.

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

# Chapter 1

## Introduction

Many important computational tasks can be modeled as combinatorial optimization problems, where the goal is to find a solution that maximizes or minimizes a certain objective function (value) on a certain discrete set of feasible solutions. Combinatorial optimization problems have been extensively studied with an tremendous progress in the past few decades. To give the readers a flavor of the problems studied in this dissertation, we present a few examples below.

**Problem 1.0.1 (MAX CUT).** Given an unweighted graph  $G = (V, E)$ , find a partition of the vertices  $V = (S, \bar{S})$  such that the number of edges between  $S$  and  $\bar{S}$  is maximized.

**Problem 1.0.2 (MAX 3-SAT).** Given a set of 3-CNF clauses of the form  $\ell_i \wedge \ell_j \wedge \ell_k$  where  $\ell_i, \ell_j$  and  $\ell_k$  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.

These two problems belong to the class of Constraint Satisfaction Problems (CSPs) that have numerous applications, from artificial intelligence and planning to VLSI chip design.

**Problem 1.0.3 (MAX BISECTION).** Given an unweighted graph  $G = (V, E)$ , where  $|V|$  is even, find a partition of the vertices  $V = (S, \bar{S})$  such that  $|S| = |\bar{S}|$  and the number of edges between  $S$  and  $\bar{S}$  is maximized.

This problem is a close variant of the MAX CUT problem and belongs to the class of Constraint Satisfaction Problems with Global Cardinality Constraints.

**Problem 1.0.4 (TRAVELING SALESMAN PROBLEM).** Given a list of cities and the distances between each pair of cities, find the shortest possible route i.e., the one with minimum total distance that visits each city exactly once and return to the original city.

**Problem 1.0.5** (VERTEX COVER). Given an undirected graph, find the smallest set of vertices such that each edge of the graph is incident to at least one vertex in the set.

The reader may refer to Section 2.3 where more combinatorial problems (specifically CSPs) are defined.

Unfortunately, overwhelming majority of the combinatorial optimization problem are computationally intractable (**NP**-hard). Therefore, unless  $\mathbf{P} = \mathbf{NP}$ , one would not be able to solve these problems optimally efficiently.

One popular and extensively studied way to cope with this intractability is to compute solutions that are (provably) *approximately* optimal. For example, one might settle for an algorithm that always returns a solution that is guaranteed to be at least half as good as the optimal solution. We will formally introduce the notion of approximation ratio in Chapter 2.

## 1.1 The Relaxation and Rounding Paradigm for Designing Approximation Algorithms

Convex relaxations and rounding schemes play an extremely important role in designing approximation algorithms. In fact, a vast majority of approximation algorithms follow a two step approach consisting of relaxation and rounding. A significant portion of this thesis is devoted to exploring the effectiveness and limitations of convex relaxations. In this section, we will provide a rudimentary introduction to this powerful framework. The readers may refer to the book of Vazirani [71] for more detailed explanations and proofs.

### Relaxation

By definition, the feasible space of combinatorial problem is discrete and finite. Therefore, one can always reformulate a combinatorial problem into an optimization problem over a finite set of binary variables that are required to satisfy some constraints.

Of course, this reformulation does not make the problem any easier, as integer program is **NP**-hard to solve. However, in terms of designing approximation algorithms, this approach enables us to look at the problem from a different angle. One could hope that looking from an integer program perspective, one would be able to gain more insight into the problem and thus be able to find better approximation algorithms. In particular, as the intractability of the integer program stems from the *non-convexity* of the space of solutions. Therefore, one could relax the constraints of the integer programs in order to make it tractable.

Specifically we can 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. For example, a simple relaxation would be to allow variables to take any real numbers in  $[0, 1]$  instead of  $\{0, 1\}$ . By doing this, if the objective function and other constraints in the integer program were also linear, we would get a linear program which is efficiently solvable in polynomial time [34].

However this does not solve the problem entirely. By relaxing the integral constraints, we're effectively permitting more solutions than the original integer program does. Therefore it immediately follows that the optimum of the relaxation is at least as good as the optimum of the integer program.

Formally, let  $\mathfrak{I}$  be an instance of a minimization problem. Let  $\text{opt}(\mathfrak{I})$  denotes the value of the optimum solution to the instance  $\mathfrak{I}$ , and  $\text{Conv}(\mathfrak{I})$  denotes the optimal value of the corresponding relaxation, then

$$\text{Conv}(\mathfrak{I}) \leq \text{opt}(\mathfrak{I}) \quad (1.1.1)$$

## Rounding

Not every solution for the convex relaxation has a corresponding solution in the original problem. Therefore, when using convex relaxations to design approximation algorithms, there is usually a *rounding* step to convert the relaxation solution to a feasible solution of the original problem. This procedure is called "rounding" because in the linear programming relaxation setting, one usually gets a non-integer solution as the optimal solution to the linear program therefore the goal is often to "round" the fractional assignments to the variables to integral values. However, when using other relaxation techniques such as semidefinite programs, one may have to convert vector-valued variables to integral values.

Formally, a rounding scheme is an algorithm that takes the problem instance  $\mathfrak{I}$  and the optimal solution  $x^*$  to the convex relaxation as input, and outputs a feasible solution  $x$  to the original combinatorial optimization problem. Let  $\text{val}(x^*)$  denote the value of  $x^*$  of the convex relaxation and  $\text{val}(x)$  denote the value of the objective function on the rounded solution  $x$ . Let  $\text{opt}(\mathfrak{I})$  denote the value of the optimal solution to the instance. If one could show that the following holds (for a minimization problem)

$$\text{val}(x) \leq \alpha \text{opt}(\mathfrak{I}) \quad (1.1.2)$$

for every instance  $\mathfrak{I}$ , then we effectively obtained an  $\alpha$ -approximation algorithm by first solving the convex relaxation and then performing the rounding scheme. However, directly proving the 1.1.2 is usually quite difficult as computing  $\text{opt}(\mathfrak{I})$  itself is already **NP**-hard. Therefore, alternative one usually turns to prove

$$\text{val}(x) \geq \alpha \text{val}(x^*) \quad (1.1.3)$$

instead. Observe that 1.1.3 together with 1.1.1 directly implies 1.1.2.

## Integrality Gap of Convex Relaxations

As we mentioned earlier, one usually compares the performance between the rounded solution and the optimum of the relaxation when proving the approximation guarantee of an algorithm. However, this approach inherently introduced some inaccuracy when calculating the approximation ratio. This inaccuracy stems from the relaxation itself. For example, given some combinatorial optimization problem (let's say it's a minimization problem) and an instance  $\mathfrak{I}$ , it's entirely possible that the optimal solution of  $\mathfrak{I}$  has value 1 while the relaxation has optimum 0.5. In this case, irrespective of which rounding algorithm we use, we cannot hope to achieve an approximation guarantee better than 2, as no integral solution achieves value better than 1.

Formally, we define *integrality gap* of a relaxation  $R$  to be the worst case ratio between  $\text{opt}(\mathfrak{I})$  and  $R(\mathfrak{I})$ , taking over all possible instances  $\mathfrak{I}$ , denoted by  $\text{gap}(R)$ .

$$\text{gap}(R) \stackrel{\text{def}}{=} \sup_{\mathfrak{I}} \frac{\text{opt}(\mathfrak{I})}{R(\mathfrak{I})} \geq 1$$

Similarly for maximization problem we can define also define the integrality gap of a relaxation  $R$ .

$$\text{gap}(R) \stackrel{\text{def}}{=} \inf_{\mathfrak{I}} \frac{\text{opt}(\mathfrak{I})}{R(\mathfrak{I})} \leq 1$$

Integrality gap serves as a measure of the quality of the relaxation  $R$ . In most cases it also serves as a limit of approximation ratio of the approximation algorithms obtained via this relaxation. Therefore a great effort has been going into designing relaxations with small integrality gap in the past few decades.

On the other hand, rounding algorithms serve as a concrete proof for integrality gap as integrality gap serves as a lower bound for approximation ratio (for minimization problems), therefore an approximation ratio of 2 would imply that the integrality gap of the relaxation is at most 2.

## 1.2 Relaxation Techniques and Hierarchies

A large number of approximation algorithms use a specific type of convex relaxation - linear programming (LP). A linear program consists of an objective function that is either maximizing or minimizing a linear function over real-valued variables while satisfying certain linear constraints among them. While linear programs can be solved in polynomial time using interior point methods [1, 57], the simplex method is used extensively in practice. We refer the reader to the

book by Vazirani [71] for more details and examples on using linear program to obtain approximation algorithms.

Another relaxation technique that has been extensively used in the design of approximation algorithm is semidefinite programming (SDP). A semidefinite program consists of vector-valued variables, with linear constraints on their inner products. The objective function is a linear function of the inner products of the variables. Semidefinite programs can be solved in polynomial time using the ellipsoid method [34] or interior point methods [1, 57]. More precisely, these algorithms output a solution with the value which differs from the optimum by at most an additive error  $\varepsilon$  in time that is polynomial in the program description size and  $\log \frac{1}{\varepsilon}$ .

SDP was first introduced to the field of combinatorial optimization by the classic work of Lovász [53]. The Lovász Theta function, as it is referred to today, is a semidefinite programming relaxation for the MAXIMUM INDEPENDENT SET problem.

Semidefinite program was popularized in the field of approximation algorithms thanks to the seminal work of Goemans and Williamson in 1994[31]. In their work they used a simple semidefinite programming relaxation and an elegant rounding scheme called halfspace rounding to obtain a 0.878-approximation algorithm for MAX CUT problem. Together with the work of Poljak [60], their algorithm implied that SDP is strictly stronger when designing approximation algorithms.

Ever since then SDP has been the main driving force in advancing of approximation algorithms. It has found application in problems ranging from Constraint Satisfaction Problems [14, 15, 17, 19, 24, 27, 29, 37, 38, 45, 52, 55, 74, 76, 75] to Vertex Cover [3, 18, 20, 43], Vertex Ordering [16, 22] to Graph decomposition and Discrete optimization[2, 48, 56].

**Raghavendra's Result.** With every approximation devised, the question arises as to whether one could find an even better approximation algorithm. Similarly, even though SDP has shown that its extremely powerful when designing approximation algorithms, one could still ask if there exists some better methodologies that could achieve better approximation ratio than SDPs. A groundbreaking result of Raghavendra [61] indicates that the answer might be NO, at least for a large portion of combinatorial problems.

In particular, Raghavendra showed that assuming Unique Games Conjecture, the optimal approximation algorithms for *every* CSP can be obtained by a relatively simple SDP relaxation. He also gave an algorithm that optimally rounds every SDP of this form. Also, he showed that the best approximation ratio for every CSP is given by the integrality gap of this SDP.

Recall that the integrality gap is the worst possible ratio between the opti-

mum of the relaxation and the optimal solution to the instance. Therefore its a measurement of the quality of a *specific* convex relaxation. Hence, Raghavendra's result seems to indicate that SDPs are indeed the "correct" approach in obtaining approximation algorithms.

## LP/SDP Hierarchies

As we mentioned earlier, integrality gap of the relaxation poses as a natural barrier toward obtaining better approximation algorithms. Therefore, in order to achieve improved approximation algorithms, one of the most important step is to find a relaxation with integrality gap as close to 1 as possible. While in some cases the most natural and simple LP/SDP relaxation of the integer program already yields the best possible approximation algorithm [31, 39, 49], there are also cases where a cleverly formulated relaxation could lead to improved approximation algorithms[4, 76].

One natural way to strengthen the algorithmic power of a relaxation is by adding additional constraints, so that the result relaxation is tighter and gives better approximation guarantee. One notable example is the work by Arora, Rao and Vazirani [4] in which they used the so-called " $\ell_2^2$ -triangle inequalities" in addition to the basic SDP relaxation and improved the approximation ration from  $\Theta(\log n)$  to  $O(\sqrt{\log n})$  for SPARSEST CUT problem.

While analysis of the convex relaxations with such extra constraints are usually very problem specific, there are several systematic ways to add additional constraints without even looking at the problem – so called "relaxation hierarchies". These hierarchies provide systematic procedures which work round-by-round: At each round, they produce a stronger convex relaxation at the cost of larger problem size. First such hierarchy was given by Sherali and Adams [68], then followed by Lovász and Schrijver [54], both based on linear programming. The strongest hierarchy among known hierarchies is based on semidefinite relaxation given by Lasserre [51], which will be the focus of study in this thesis.

These hierarchies are known to converge to an integral 0/1 solution, i.e., have an integrality gap of 1 as the number of rounds gets closer to  $n$ . However, at  $k$  rounds, these hierarchies usually takes time  $O(n^{O(k)})$  to solve. Therefore such convergence result does not provide too much value when we restrict on polynomial time regime. The interesting question is to characterize the problems for which a small number of rounds of these hierarchies yields a better approximation algorithm for. On the other hand, lower bound results showing that the integrality gap of the program obtained after many levels of a hierarchy remains large would be a strong indication that the problem might be hard to tackle.

For Sum-of-Squares SDP hierarchy, it is known that even few rounds of Sum-of-Squares SDP is already as strong as many state of the art approximation algorithms for a large set of combinatorial optimization problems. For example, 3

rounds of Sum-of-Squares SDP is enough to capture the ARV SDP relaxation for SPARSEST CUT [4]. Arguably, Sum-of-Squares SDP poses the currently strongest known threat to the famous Unique Games Conjecture. In fact, to the best of our knowledge, it is entirely possible that 4th level of Sum-of-Squares SDP could improve upon the Goemans-Williamson algorithm for MAX CUT and therefore refute the Unique Games Conjecture.

While for weaker hierarchies, many strong algorithmic results as well as integrality gaps were known[67, 66, 70], our understanding of Lasserre SDP hierarchy seems to be lacking behind. On the algorithmic front, to the best of our knowledge, only two results existed prior to our work: Chalmatc and Singh [21] used  $O(1/\gamma^2)$  rounds of Sum-of-Squares SDP hierarchy to find an independent set of size  $\Omega(n^{\gamma^{2/8}})$  in 3-uniform hypergraphs with an independent set of size  $\gamma n$ . Also Karlin et al. [44] showed that  $1/\varepsilon$  rounds of Sum-of-Squares SDP gives a  $(1 + \varepsilon)$  approximation to the Knapsack problem.

Part of the reason in our lack of understanding of Sum-of-Squares SDP is due to its complexity. While additional the variables and constraints provides better approximation guarantee over the basic SDP, they also make analyzing the solution much harder. Therefore it would be beneficial to have some generic framework to round and analyze the Sum-of-Squares SDP solutions.

### 1.3 Contribution of the Thesis

In the first part of our thesis we will try to address this problem. We present a general framework to obtain Sum-of-Squares SDP relaxation, round SDP solution and analyze the rounding algorithm for CSPs with global cardinality constraints. To demonstrate the approach, we show that one could use Sum-of-Squares SDP to achieve a 0.85-approximation algorithm for Max Bisection problem, improving on the previously best known 0.70 ratio.

While Sum-of-Squares SDP hierarchy is extremely powerful, one could still ask the question: is there an SDP relaxation (hierarchy) that is even stronger than Sum-of-Squares? If so, what is the strongest SDP possible? We will try to address this question in the second part of the thesis. Specifically, we show that Sum-of-Squares SDP solution achieves best possible approximation ratio for all Max CSPs among all symmetric SDP relaxations of similar size. This result gives the first lower bounds for symmetric SDP relaxations of Max CSPs, and indicates that the sum-of-squares method provides the "right" SDP relaxation for this class of problems.

Sum-of-Squares SDP hierarchy is sometimes referred to as Parrilo-Lasserre SDP Hierarchy or Lasserre SDP Hierarchy. In this thesis we will use these names interchangeably.



## Chapter 2

# Preliminary and Organization of Thesis

In this chapter, we will first introduce the basic concepts that are involved in this dissertation.

### 2.1 Definitions and Terminologies

To start off this section, we first formally define approximation algorithms and approximation ratio.

**Definition 2.1.1.** An algorithm  $\mathcal{A}$  is said to be an  $\alpha$ -approximation algorithm for a maximization problem  $\Lambda$ , if for every instance  $\mathfrak{I}$  of  $\Lambda$ , we have

$$\frac{\mathcal{A}(\mathfrak{I})}{\text{opt}(\mathfrak{I})} \geq \alpha$$

Similarly we can define approximation ratios for minimization problem as well.

**Definition 2.1.2.** An algorithm  $\mathcal{A}$  is said to be an  $\alpha$ -approximation algorithm for a minimization problem  $\Lambda$ , if for every instance  $\mathfrak{I}$  of  $\Lambda$ , we have

$$\frac{\mathcal{A}(\mathfrak{I})}{\text{opt}(\mathfrak{I})} \leq \alpha$$

We note that for minimization problems, the approximation ratio  $\alpha$  is usually greater than 1.

The notion of approximation ratio can be generalized to randomized algorithms as well.

**Definition 2.1.3.** A *randomized* algorithm  $\mathcal{A}$  is said to be an  $\alpha$ -approximation algorithm for a maximization problem  $\Lambda$ , if for every instance  $\mathfrak{I}$  of  $\Lambda$ , we have

$$\mathbb{E}\left(\frac{\mathcal{A}(\mathfrak{I})}{\text{opt}(\mathfrak{I})}\right) \geq \alpha$$

To illustrate the notion of approximation algorithms and approximation ratio, we give two simple  $1/2$ -approximation algorithms for MAX CUT problem, one deterministic, one randomized.

**Two simple  $1/2$ -approximation algorithms for MAX CUT.** Recall that in MAX CUT problem we're given an unweighted graph  $G = (V, E)$  and want to find a cut  $(S, \bar{S})$  such that the number of edges crossing the cut is maximized.

Firstly we present a simple greedy algorithm.

**Algorithm 2.1.4.** Greedy algorithm for MAX CUT.

Input: An unweighted graph  $G = (V, E)$

Output: A cut  $(S, \bar{S})$  of size at least  $|E|/2$ .

- Step 1. Start with an empty set  $S$ .
- Step 2. For every vertex  $v \in V$ , check if switching side (from  $S$  to  $\bar{S}$  or  $\bar{S}$  to  $S$ ) will increase the cut size.
- Step 3. Repeat step 2 until no such vertex found, then output  $S$ .

**Proposition 2.1.5.** The Algorithm 2.1.4 is a  $1/2$ -approximation algorithm for MAX CUT.

*Proof.* First we have to show that the algorithm terminates. To see that, observe that the size of the cut *strictly* increases after every switch, therefore the algorithm will terminate within  $|E|$  switches.

Now we want to show the approximation ratio. In order to show this statement, we only have to show that the output cut contains at least half of the edges in the graph, as the optimal solution is trivially upper bounded by the total number of edges in the graph.

To see this, observe that for every vertex at least half of its neighbor is in the cut, otherwise switching this vertex to the other side of the cut would strictly increase the size of the cut. Hence the statement follows.  $\square$

Secondly we present a simple randomized algorithm that also achieves  $1/2$  approximation ratio *in expectation*.

**Algorithm 2.1.6.** Randomized algorithm for MAX CUT.

For each vertex  $v \in V$ , randomly put it in  $S$  or  $\bar{S}$  with  $1/2$  probability each.

**Proposition 2.1.7.** *The Algorithm 2.1.6 is a  $1/2$ -approximation algorithm for MAX CUT.*

*Proof.* It's easy to see that for every edge  $e \in E$ ,  $\mathbb{P}(e \in (S, \bar{S})) = 1/2$ . By linearity of expectation,  $\mathbb{E}(|(S, \bar{S})|) = |E|/2$ .  $\square$

## 2.2 Relaxation and Rounding of Combinatorial Optimization Problems

A vast majority of approximation algorithms follow a two step approach consisting of relaxation and rounding. In this section we will give an overview of this approach and use VERTEX COVER and MAX CUT to illustrate the usage of linear programming (LP) and semidefinite programming (SDP) in approximation algorithms.

### Representing Combinatorial Optimization Problems as Integer Program

Recall that for a graph  $G = (V, E)$  a vertex cover  $S$  of  $G$  is a subset of vertices such that for every edge  $e = (u, v) \in E$ , at least one of  $u \in S$  or  $v \in S$  is true. In VERTEX COVER problem, we're given a graph  $G$  and want to find the vertex cover of minimum cardinality.

In order to formulate it as an integer program, we can introduce an integer variable  $X_v$  for every vertex  $v$  in the graph  $G$ . This variable  $X_v$  indicates whether the vertex  $v$  belongs to the vertex cover. Specifically,  $X_v$  is a  $\{0, 1\}$ -variable defined as follows:

$$X_v = \begin{cases} 0, & \text{if } v \text{ is not in } S \\ 1, & \text{if } v \text{ is in } S \end{cases}$$

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  $X_u + X_v \geq 1$ . Also notice that this is a sufficient condition, since  $X_u + X_v \geq 1$  guarantees that at least one of  $X_u$  or  $X_v$  is 1. Also, the size of the vertex cover can be represented as  $\sum_{v \in V} X_v$ . With these observations one can write the integer program for VERTEX COVER as:

Integer program for VERTEX COVER

$$\begin{aligned} & \text{minimize} && \sum_{v \in V} X_v \\ & \text{subject to} && X_u + X_v \geq 1 \quad \text{for every edge } e = (u, v) \\ & && X_v \in \{0, 1\} \quad \text{for every vertex } v \in V \end{aligned}$$

For MAX CUT problem, we're given a graph  $G = (V, E)$  and the goal is to find a cut  $V = (S, \bar{S})$  such that the total number of edges crossing the cut is maximized. Similar to the VERTEX COVER problem, we can also define a random variable  $X_v$  for each  $v \in V$  as follows:

$$X_v = \begin{cases} 0, & \text{if } v \text{ is in } \bar{S} \\ 1, & \text{if } v \text{ is in } S \end{cases}$$

An edge  $e = (u, v)$  is cut if and only if  $|X_u - X_v| = 1$ . Hence the integer program for MAX CUT can be written as the following:

Integer program for MAX CUT

$$\begin{aligned} & \text{maximize} && \sum_{e=(u,v) \in E} |X_u - X_v| \\ & \text{subject to} && X_v \in \{0, 1\} \quad \text{for every vertex } v \in V \end{aligned}$$

## Approximation Algorithm for VERTEX COVER via Linear Program

While the optimum of the integer program gives exact solution to VERTEX COVER problem, it doesn't reduce the complexity. In order to make the problem tractable, we will reduce the constraints in the integer program. Specifically, we will relax the condition that every variable needs to be assigned as either 0 or 1 – instead we will allow 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 program for VERTEX COVER

$$\begin{aligned}
 & \text{minimize} && \sum_{v \in V} X_v \\
 & \text{subject to} && X_u + X_v \geq 1 \quad \text{for every edge } e = (u, v) \\
 & && 0 \leq X_v \leq 1 \quad \text{for every vertex } v \in V
 \end{aligned}$$

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, let  $\text{opt}(G)$  denote the size of the optimal vertex cover of graph  $G$ , let  $\text{IP}(G)$  denote the optimum of the integer program of  $G$ , let  $\text{LP}(G)$  denote the optimum of the linear program of  $G$ , we have

$$\text{LP}(G) \leq \text{IP}(G) = \text{opt}(G)$$

Since linear program can be solved in polynomial time, one could use  $\text{LP}(G)$  as an (efficiently computable) approximation to the optimum of VERTEX COVER problem, although it is unclear how good of an approximation it is.

For example, on a complete graph  $K_n$ , one of the possible solution to the linear program could be  $X_v = 1/2$  for every vertex  $v$  in the graph. This solution clearly satisfies all the linear constraints in the LP. Hence we have that  $\text{LP}(K_n) \leq n/2$ . However on the other hand, it's easy to see that in order for a subset of vertices to cover the complete graph, at least  $n - 1$  vertices needs to be chosen, therefore  $\text{opt}(K_n) \geq n - 1$ . As  $n$  goes to infinity, the ratio between  $\text{LP}(K_n)$  and  $\text{opt}(K_n)$  goes to 2.

The worst case ratio among all possible instances is known as the *integrality gap* of an linear program. It can be thought of a rough measure of the quality of approximation. The instances where the gap is achieved are known as *gap instances*.

**Rounding algorithm for VERTEX COVER.** Recall that our original goal is to find a minimum vertex cover given a graph  $G$ . Instead, we have now relaxed the integer program into a linear program and obtained its solution. Therefore we will need some procedure to transform an LP solution back to a feasible solution of the original VERTEX COVER problem. This process is referred as a *rounding algorithm*. Formally, given an LP solution, a rounding algorithm takes it as the input and produces an integral solution to the integer program that satisfies all the constraints(hence also a feasible solution to the original combinatorial problem).

After a brief thinking one would realize that the rounded solution will most likely not be as good as the LP solution itself. In fact, take the complete graph

for example, the LP solution only uses  $n/2$  vertices while the best vertex cover requires at least  $n - 1$  vertices. Therefore no matter how good the rounding algorithm is, the produced solution will be almost twice as worse as the LP solution.

Below we will give a rounding algorithm and analyze its performance.

**Algorithm 2.2.1.** Rounding algorithm for VERTEX COVER.

Input: The optimum solution of linear program 2.2.

Output: A feasible vertex cover to the graph  $G$ .

For every vertex  $v \in V$ , pick  $v$  to be in the vertex cover if and only if  $X_v \geq 1/2$ .

**Proposition 2.2.2.** *This rounding algorithm outputs a feasible solution  $S$  to the VERTEX COVER problem with  $|S| \leq LP(G) * 2$ .*

*Proof.* Firstly we prove that the solution  $S$  is a feasible solution. In order to show this, we only need to show that for every edge  $e = (u, v) \in E$ , either  $X_u \geq 1/2$  or  $X_v \geq 1/2$ . Since the LP solution is a feasible solution, it satisfies  $X_u + X_v \geq 1$  and  $0 \leq X_u, X_v \leq 1$ , the statement follows.

Secondly we want to show that the size of  $S$  is at most twice the LP value. This is also fairly obvious as we only took the vertices with  $X_v \geq 1/2$ .  $\square$

There are two important corollaries that immediately follows from this proposition.

**Corollary 2.2.3.** *The linear program 2.2 together with rounding algorithm 2.2.1 gives a 2-approximation algorithm for VERTEX COVER problem.*

*Proof.* This follows directly from Proposition 2.2.2 and the fact that the linear program is a relaxation (hence  $LP(G) \leq \text{opt}(G)$ ).  $\square$

**Corollary 2.2.4.** *The integrality gap of linear program 2.2 is 2.*

*Proof.* Using the rounding algorithm above, we showed that  $LP(G) \geq \text{opt}(G)/2$ , thus implying that the integrality gap is at most 2. Using the complete graph we mentioned before, we know that the integrality gap of this LP is at least 2. The statement follows.  $\square$

We remark that we actually used the rounding algorithm as a proof for the integrality gap of the LP.

## Approximation Algorithm for MAX CUT via Semidefinite Program

In this section we will give a high level overview of the Goemans-Williamson algorithm for MAX CUT [31].

Recall that we constructed an integer program 2.2 for MAX CUT problem in the previous section. In the first step, we would take this IP and reformulate it into a new integer quadratic program (IQP). Instead of using  $\{0, 1\}$  variables to denote the vertices, we will use  $\{-1, 1\}$  variables instead.

Integer quadratic program for MAX CUT

$$\begin{aligned} & \text{maximize} && \sum_{e=(u,v) \in E} (X_u - X_v)^2 / 4 \\ & \text{subject to} && X_v^2 = 1 \quad \text{for every vertex } v \in V \end{aligned}$$

It's easy to verify that the IQP above is equivalent to the integer program we have for MAX CUT.

**Goemans-Williamson SDP relaxation for MAX CUT.** Just as what we did for VERTEX COVER in the previous section, we will relax the intractable problem into a polynomial time solvable problem.

Recall that the variables  $X_i$  are equal to  $\pm 1$ , 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 an  $n$ -dimensional unit vector  $\mathbf{X}_v$  to each vertex  $v \in V$ . This yields the following semidefinite program (SDP) relaxation:

Goemans-Williamson SDP relaxation for MAX CUT.

$$\begin{aligned} & \text{maximize} && \sum_{e=(u,v) \in E} (X_u - X_v)^2 / 4 \\ & \text{subject to} && X_v^2 = 1 \quad \text{for every vertex } v \in V \end{aligned}$$

It's easy to see that every integer solution to the original IQP can be easily translate to a corresponding SDP with the same value – for every  $x_i = \pm 1$ , we will construct an  $n$ -dimensional vector with the first dimension equals to  $x_i$  and the other dimensions being 0. Therefore the Goemans-Williamson SDP is an relaxation of the original IQP, henceforth also a relaxation of the MAX CUT problem.

## Rounding Goemans-Williamson SDP

On solving the Goemans-Williamson SDP relaxation, we obtain a set of unit vectors  $\{\mathbf{v}_i\}$  on the  $n$ -dimensional space  $\mathbb{R}^n$ . Recall that the vector  $\mathbf{v}_i$  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.

We will present a randomized rounding algorithm that takes  $n$  unit vectors as input, output a cut in the graph  $G$ .

**Algorithm 2.2.5.** Rounding algorithm for Goemans-Williamson SDP.

Input: A feasible solution to the Goemans-Williamson SDP, which consists of a set of unit vectors  $\{\mathbf{v}_i\}$

Output: A cut of the original graph  $G$ .

- Sample a random hyperplane  $\mathcal{H}$  that passes through the origin. This plane naturally induces a partition of the  $n$ -dimensional space as well as the  $n$ -dimensional unit sphere, say  $S^+$  and  $S^-$ .
- Take the embedding of the graph on  $S_n$ , output the cut induced by the partition.

This simple rounding scheme turns out to be extremely powerful. Below we will prove that it achieves 0.878-approximation for MAX CUT.

In order to prove the effectiveness of the rounding scheme, first we need a fact.

**Fact 2.2.6.** For any  $x \in [-1, 1]$ , the following inequality holds:

$$\frac{\arccos(x)}{2\pi} \geq \alpha_{GW} \times \frac{1-x}{2}$$

where  $\alpha_{GW} \geq 0.878$  is an absolute constant.

Now we will calculate the expected cut value of the rounding algorithm. Consider an edge  $e = (u, v)$  in the graph  $G$ . Let  $\theta$  be the angle between the vectors  $\mathbf{u}$  and  $\mathbf{v}$  hence we have  $\theta = \arccos(\mathbf{u} \cdot \mathbf{v})$ . Also observe that a random hyperplane projects as a random line passing through the origin in the 2 dimensional space spanned by  $\mathbf{u}$  and  $\mathbf{v}$ , therefore the chance of  $\mathbf{u}$  and  $\mathbf{v}$  ended up on different side of the cut is

$$\mathbb{P}(e = (u, v) \text{ is in the cut}) = \frac{\theta}{\pi} = \frac{\arccos(\mathbf{u} \cdot \mathbf{v})}{\pi}$$

Also notice that by fact 2.2.6, we have

$$\mathbb{P}(e = (u, v) \text{ is in the cut}) \geq \alpha_{GW} \times \frac{(\mathbf{u} - \mathbf{v})^2}{4}$$



Note that here we used the fact that  $\mathbf{u}$  and  $\mathbf{v}$  are unit vectors.

Observe that on the LHS we have the probability of an edge being cut, i.e., the *expected* contribution of this edge to the final cut. On the RHS we have the contribution of the same edge to the SDP value. Summing this over all the edges in the graph, we have

$$\mathbb{E}(\text{number of edges cut}) \geq \alpha_{\text{GW}} \text{SDP}(G)$$

Therefore the SDP together with the rounding algorithm achieves 0.878-approximation of the MAX CUT problem.

The two examples showed previously highlighted a generic way of obtaining approximation algorithm for combinatorial optimization problem. In particular, given an instance  $\mathfrak{I}$  of a combinatorial optimization problem, it is first reformulated as an integer program, then relaxed to a convex program that affords efficient solver (usually linear program or semidefinite program).

The optimum of the LP or SDP consists of a set of variables, either real numbers or vectors. In the *rounding* step, an algorithm takes the LP/SDP solution as input, and "round" them into an integral solution to the original combinatorial optimization problem, usually losing in objective value. The analysis of the algorithm usually compares the performance of the rounded solution against the LP/SDP value and use that as an upper bound of the approximation ratio of the algorithm.

By definition, performance of this approach is naturally limited by the integrality gap of the relaxation, as on the gap instances there is simply no integral solution performs better than the integrality gap. However, as it turns out, the integrality gaps of some natural SDP/LP relaxations actually perfectly captures the approximability of a large set of combinatorial problems (namely general CSPs) under Unique Games Conjecture[61].

## 2.3 Constraint Satisfaction Problems

This thesis will heavily focus on the approximability of Constraint Satisfaction Problems (CSPs) as well as CSPs with global cardinality constraints. In this section, we will define these problems and also give a few examples.

**Definition 2.3.1** (Constraint Satisfaction Problems). A constraint satisfaction problem is specified by  $\Lambda = ([q], \mathbb{P}, k)$  where  $[q] = \{0, \dots, q-1\}$  is a finite domain,  $\mathbb{P} = \{P : [q]^t \mapsto [0, 1] \mid t \leq k\}$  is a set of payoff functions. The maximum number of inputs to a payoff function is denoted by  $k$ .

Every instance of the CSP  $\Lambda$  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  $\Lambda$  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  $\Lambda$  is the maximum number of inputs to a predicate in the family  $\Lambda$ .

Below we give a few examples of Constraint Satisfaction Problems.

**Example 2.3.2 (MAX CUT).** Given a graph  $G = (V, E)$  with vertices  $V = \{v_1, \dots, v_n\}$  and edges  $E$ , find a partition  $S \cup S' = V$  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  $v_j \in S'$  or vice versa.

In the MAX CUT example, the domain is the binary domain  $[2] = \{0, 1\}$ , with a single payoff function  $P$  given by:

$$P(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases}$$

**Example 2.3.3 (MAX 2-SAT).** Given a set of 2-CNF clauses of the form  $\ell_i \wedge \ell_j$  where  $\ell_i$  and  $\ell_j$  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.

Similarly one can define MAX 3-SAT and MAX  $k$ -SAT where the number of variables in a clause have 3 and  $k$  literals.

**Example 2.3.4 (LABEL COVER).** An instance of Label Cover is given by  $(\mathcal{W} \cup \mathcal{V}, E, [R], \Pi)$  consists of a bipartite graph over vertex sets  $\mathcal{W}$  and  $\mathcal{V}$  with edges  $E$  between them where all the vertices in  $V$  are of the same degree. Also part of the instance is a set of labels  $[R]$  and a set of mappings  $\Pi = \{\pi_{w \rightarrow v} : [R] \mapsto [R]\}$  for each edge  $e \in E$ . An assignment  $A$  of labels to vertices is said to satisfy an edge  $e = (w, v)$  if  $\pi_{w \rightarrow v}(A(w)) = A(v)$ . The objective is find an assignment that satisfies as many edges as possible.

**Example 2.3.5 (UNIQUE GAMES).** Given a variable set  $V$  and a list of constraints of the form  $x_u = \pi_{v \rightarrow u} x(v)$  where  $u, v \in V$  are two variables and  $\pi_{v \rightarrow u}$  is a permutation of  $[R]$ , the goal is to find an  $[R]$ -assignment to  $V$  so as to maximize the number of satisfied constraints.

Below we define CSPs with global cardinality constraints.

**Definition 2.3.6 (Constraint Satisfaction Problems with Global Cardinality Constraints).** A constraint satisfaction problem with global cardinality constraints is specified by  $\Lambda = ([q], \mathbb{P}, k, c)$  where  $[q] = \{0, \dots, q-1\}$  is a finite domain,  $\mathbb{P} = \{P : [q]^t \mapsto [0, 1] \mid t \leq k\}$  is a set of payoff functions. The maximum number of inputs to a payoff function is denoted by  $k$ . The map  $c : [q] \mapsto [0, 1]$  is the cardinality function which satisfies  $\sum_i c_i = 1$ . For any  $0 \leq i \leq q-1$ , the solution should contain  $c_i$  fraction of the variables with value  $i$ .

**Definition 2.3.7.** An instance  $\Phi$  of constraint satisfaction problems with global cardinality constraints  $\Lambda = ([q], \mathbb{P}, k, c)$  is given by  $\Phi = (\mathcal{V}, \mathbb{P}_{\mathcal{V}}, W)$  where

- $\mathcal{V} = \{x_1, \dots, x_n\}$ : variables taking values over  $[q]$
- $\mathbb{P}_{\mathcal{V}}$  consists of the payoffs applied to subsets  $S$  of size at most  $k$
- Nonnegative weights  $W = \{w_S\}$  satisfying  $\sum_{|S| \leq k} w_S = 1$ . Thus we may interpret  $W$  as a probability distribution on the subsets. By  $S \sim W$ , we denote a set  $S$  chosen according to the probability distribution  $W$
- An assignment should satisfy that the number of variables with value  $i$  is  $c_i n$  (we may assume this is an integer).

Here we also give a few examples of CSPs with global cardinality constraints.

**Example 2.3.8** (Max(Min) Bisection). Given a (weighted) graph  $G = (V, E)$  with  $|V|$  even, the goal is to partition the vertices into two equal pieces such that the number (total weights) of edges that cross the cut is maximized (minimized).

More generally, one could define  $\alpha$ -MAX CUT problem, where the goal is to find a partition having  $\alpha|V|$  vertices on one side, while cutting the maximum number of edges. Furthermore, one could allow weights on the vertices of the graph, and look for cuts with exactly  $\alpha$ -fraction of the weight on one side.

**Definition 2.3.9** (Edge Expansion). Given a graph (w.l.o.g. we may assume it is a unweighted regular graph)  $G = (V, E)$ , and  $\delta \in (0, 1/2)$ , the goal is to find a set  $S \subseteq V$  such that  $|S| = \delta|V|$  and the edge expansion of  $S$ :  $\Phi(S) = \frac{E(S, \bar{S})}{d|S|}$  is minimized.

**Remark 2.3.10.** Although some problems (e.g., BALANCED SEPARATOR) do not fix the cardinality to be some specific quantities, they can easily be reduced to the case above.

## 2.4 Results and Organization

In Chapter 2 and Chapter 3, we present some basic definitions, set up notations and recall some mathematical preliminaries. In Chapter 4, we give an overview of SDP/LP relaxations for constraint satisfaction problems, define several LP/SDP hierarchies and also define general SDP/LP relaxations for combinatorial problem. The rest of the thesis is divided into two parts: in Chapter 5, we present a general framework to obtain Sum-of-Squares SDP relaxation, round

SDP solution and analyze the rounding algorithm for CSPs with global cardinality constraints. To demonstrate the approach, we show that one could use Sum-of-Squares SDP to achieve a 0.85-approximation algorithm for Max Bisection problem, improving on the previously best known 0.70 ratio. In Chapter 6, we study the computational power of general symmetric relaxations. Specifically, we show that Sum-of-Squares SDP solution achieves the best possible approximation ratio for all Max CSPs among all symmetric SDP relaxations of similar size. This result gives the first lower bounds for symmetric SDP relaxations of Max CSPs, and indicates that the sum-of-squares method provides the "right" SDP relaxation for this class of problems. We will wrap up the thesis with future directions and some open questions in Chapter 7.

## Chapter 3

# Mathematical Notations and Tools

In this chapter we set up basic notations and tools that will be used throughout this thesis.

### 3.1 Sets and Families

For any positive integer  $n$ , we will use  $[n]$  to denote the set  $\{1, 2, \dots, n\}$ . We will use  $\emptyset$  to denote empty set. Given set  $S$ , we use  $2^S$  to denote its *power set*, i.e., set of all subsets of  $S$ , also we say  $\mathcal{F}$  is a family over  $S$  if  $\mathcal{F} \subseteq 2^S$ .

For two sets  $A$  and  $B$ , we use  $A^B$  denote the set of mappings from  $A$  to  $B$ . For notational convenience, if  $B = [n]$  then we will write  $A^n$  instead of  $A^{[n]}$ . An element  $\mathbf{x} \in A^n$  is an vector  $\mathbf{x} = (x_1, \dots, x_n)$  where  $x_i \in A$ . We will always use boldface to denote multidimensional objects.

### 3.2 Linear Algebra

We will use  $\mathbb{R}$ ,  $\mathbb{Q}$ ,  $\mathbb{Z}$  and  $\mathbb{N}$  to denote the set of real number, rational number, integers and natural numbers respectively.

Given finite sets  $A$ ,  $B$  and a subset of reals  $R \subseteq \mathbb{R}$ , we will use  $R^A$  and  $R^{A,B}$  to denote the set of vectors and matrices over  $R$  whose rows and columns are identified with elements of  $A$  and  $B$  respectively.

Given two vectors  $\mathbf{x}$  and  $\mathbf{y}$ , we will use  $\langle \mathbf{x} \cdot \mathbf{y} \rangle$  to denote their inner product.

We will use  $\mathcal{S}$  denote the set of real symmetric matrices and  $\mathcal{S}_n$  to denote the set of real  $n$  by  $n$  symmetric matrices.

**Definition 3.2.1** (Positive Semidefiniteness of Matrices). A matrix  $M \in \mathcal{S}_n$  is said to be positive semidefinite if  $\mathbf{x}^T M \mathbf{x} \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^n$ , denoted by  $M \geq 0$ . The set of positive semidefinite matrices is denoted as  $\mathcal{S}_+^n$ .

We state some well known properties regarding positive semidefinite matrices.

**Theorem 3.2.2.** *The following statements are equivalent:*

- *The symmetric matrix  $A$  is positive semidefinite.*
- *All eigenvalues of  $A$  are nonnegative.*
- *All the principal minors of  $A$  are nonnegative.*
- *$\text{Tr}(A \cdot B) \geq 0$  for all  $B \geq 0$ .*

There is another well-known characterization of a matrix being PSD, as it is very important for us, we single it out as a theorem.

**Theorem 3.2.3** (Gram Decomposition of PSD Matrices). *Given  $Y \in \mathcal{S}_+^n$ ,  $y \geq 0$  if and only if there exists a set of vectors  $\{vec x_i\}_{i \in [n]} \in \mathbb{R}^m$  for some  $m \leq n$  such that  $Y = XX^T$  where  $X$  is an  $n \times m$  matrix composite of  $\mathbf{x}_i$  as rows. We refer to  $Y$  as the Gram matrix of  $\{\mathbf{x}_i\}$  and  $\{\mathbf{x}_i\}$  as the Gram decomposition of  $Y$ .*

If  $X - Y \geq 0$  then we write  $X \geq Y$ .

**Definition 3.2.4** (Convex Set). Given  $X \subseteq \mathbb{R}^n$ , we say  $X$  is convex if for any  $y, z \in X$  and  $\theta \in [0, 1]$ ,  $\theta y + (1 - \theta)x \in X$ .

**Definition 3.2.5** (Convex Hull). Given  $X \subseteq \mathbb{R}^n$ , the convex hull of  $X$  is defined as:

$$\text{convex}(X) \stackrel{\text{def}}{=} \bigcap_{C \text{ convex and } X \subseteq C} C$$

### 3.3 Convex Optimization and Semidefinite Programming

**Definition 3.3.1.** Given a convex set  $K \subseteq \mathbb{R}^n$  and a convex function  $f : \mathbb{R}^n \mapsto \mathbb{R}$ , consider the following optimization problem

$$\inf f(x) \quad \text{subject to } x \in K$$

We call problems of this natural convex optimization problems. Here  $f$  is the objective function and  $K$  is called feasible region. For any  $x \in \mathbb{R}^n$  and  $x \in K$  we call  $x$  a feasible solution. If  $K = \emptyset$  we say the convex optimization problem is infeasible. We call the infimum of the problem the optimum. If there exists a point  $x \in K$  achieving this value, we call  $x$  the optimal solution.

**Definition 3.3.2** (Semidefinite Programming). Convex optimization of the following form is called a semidefinite program(SDP):

$$\begin{aligned} & \inf \langle C, M \rangle \\ & \text{subject to } BM = D \\ & M \succeq 0 \end{aligned}$$

**Remark 3.3.3.** While convex optimization problems always concentrate on computing the infimum of the objective function, SDP supports supremum as well. This is because the objective in SDP is an linear function.

**Definition 3.3.4** (Vector Form SDPs). Most of the time we will use an alternate form of SDP where the variables in SDP are vectors with linear constraints and objective function on *inner products* between these vectors. Vector formed SDP usually looks like the following:

$$\begin{aligned} & \inf \sum_{i,j \in [n] \times [m]} C_{i,j} \mathbf{v}_i \cdot \mathbf{v}_j \\ & \text{subject to } \sum_{j \in [m]} B_{1,j} \mathbf{v}_1 \cdot \mathbf{v}_j = d_1 \\ & \quad \vdots \\ & \sum_{j \in [m]} B_{n,j} \mathbf{v}_n \cdot \mathbf{v}_j = d_n \end{aligned}$$

One could easily show that the matrix form and vector form are equivalent.

**Proposition 3.3.5** (Equivalence between matrix form and vector form). *The SDP 3.3.2 and 3.3.4 are equivalent, i.e., given a feasible solution to one SDP one could construct a solution to the other SDP with the same objective function.*

*Proof.* Given a feasible solution to the matrix form SDP, we can take its Gram decomposition and obtain a set of vectors as the solution to the vector form SDP. On the other hand, given a feasible solution to the vector form SDP we can obtain a matrix as the solution to the matrix form SDP by taking its Gram decomposition.  $\square$

## 3.4 Information Theory, Entropy and Mutual Information

In this section we give some basic definitions from information theory.

**Definition 3.4.1** (Entropy). Let  $X$  be a random variable taking values over  $[q]$ . The *entropy* of  $X$  is defined as

$$H(X) \stackrel{\text{def}}{=} - \sum_{i \in [q]} \mathbb{P}(X = i) \log \mathbb{P}(X = i)$$

**Definition 3.4.2** (Mutual Information). Let  $X$  and  $Y$  be two jointly distributed variables taking values over  $[q]$ . The *mutual information* of  $X$  and  $Y$  is defined as

$$I(X; Y) \stackrel{\text{def}}{=} \sum_{i, j \in [q]} \mathbb{P}(X = i, Y = j) \log \frac{\mathbb{P}(X = i, Y = j)}{\mathbb{P}(X = i) \mathbb{P}(Y = j)}$$

**Definition 3.4.3** (Conditional Entropy). Let  $X$  and  $Y$  be two jointly distributed variables taking values over  $[q]$ . The *conditional entropy* of  $X$  conditioned on  $Y$  is defined as

$$H(X|Y) = \mathbb{E}_{i \in [q]} [H(X|Y = i)]$$

We also give two well-known theorems in information theory below.

**Theorem 3.4.4.** *Let  $X$  and  $Y$  be two jointly distributed variables taking value on  $[q]$ , then*

$$I(X; Y) = H(X) - H(X|Y)$$

**Theorem 3.4.5.** (*Data Processing Inequality*) *Let  $X, Y, Z, W$  be random variables such that  $H(X|W) = 0$  and  $H(Y|Z) = 0$ , i.e.,  $X$  is fully determined by  $W$  and  $Y$  is fully determined by  $Z$ , then*

$$I(X; Y) \leq I(W; Z)$$

## Mutual Information, Statistical Distance and Independence

Intuitively, when two random variables have low mutual information, they should be close to being independent. In this section we formalize this intuition by giving an explicit bound on the statistical distance between the joint distribution and the independent distribution. We stress that all the results here are sufficient for our use in this work, but we believe the parameters could be further optimized.

We start by defining a few notions that measures the correlation of two random variables.

**Definition 3.4.6.** Let  $\Omega$  be a finite sample space,  $P$  and  $Q$  be two probability distributions on  $\Omega$ . The *square Hellinger distance* of  $P$  and  $Q$  is defined as

$$H^2(P, Q) = \frac{1}{2} \sum_{x \in \Omega} (\sqrt{P(x)} - \sqrt{Q(x)})^2$$



**Definition 3.4.7.** Let  $\Omega$  be a finite sample space,  $P$  and  $Q$  be two probability distributions on  $\Omega$ . The *Kullback-Leibler divergence* of  $P$  and  $Q$  is defined as

$$D_{KL}(P||Q) = \sum_{x \in \Omega} P(x) \log \frac{P(x)}{Q(x)}$$

Now we give a few facts regarding mutual information, Hellinger distance and Kullback-Leibler divergence without proving them.

**Fact 3.4.8.** Let  $X$  and  $Y$  be two jointly distributed random variables taking value in  $[q]$ , then

$$I(X; Y) = D_{KL}(p(x, y) || p(x) \times p(y)).$$

where  $p(x, y)$  is the joint distribution of  $X$  and  $Y$  on  $[q]^2$  and  $p(x) \times p(y)$  is the product distribution of the marginal distributions of  $X$  and  $Y$ .

**Fact 3.4.9.** Let  $\Omega$  be a finite sample space,  $P$  and  $Q$  be two probability distribution on  $\Omega$ , then

$$D_{KL}(Q||P) \geq \frac{2}{\ln 2} H^2(P, Q)$$

Combining the facts mentioned above, we get the following relation between mutual information and statistical distance.

**Fact 3.4.10.** Let  $X$  and  $Y$  be two jointly distributed random variables on  $[q]$  then,

$$I(X; Y) \geq \frac{1}{2 \ln 2} \sum_{i, j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2,$$

in particular for all  $i, j \in [q]$

$$|\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j)| \leq \sqrt{2I(X; Y)}$$

As a consequence, if  $X$  and  $Y$  are two random variables defined on  $\{-1, 1\}$ ,  $\text{Cov}(X, Y) \leq O(\sqrt{I(X; Y)})$

**Lemma 3.4.11.** Let  $X$  and  $Y$  be two jointly distributed random variables on  $[q]$ , we have

$$I(X; Y) \geq \frac{1}{2 \ln 2} \sum_{i, j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2$$

*Proof.*

$$\begin{aligned} I(X; Y) &= D_{KL}(p(x, y) || p(x) \times p(y)) \\ &\geq \frac{2}{\ln 2} H^2(p(x, y), p(x) \times p(y)) \end{aligned}$$

$$\begin{aligned}
&= \frac{2}{\ln 2} \sum_{i,j \in [q]} \left( \sqrt{\mathbb{P}(X = i, Y = j)} - \sqrt{\mathbb{P}(X = i)\mathbb{P}(Y = j)} \right)^2 \\
&= \frac{2}{\ln 2} \sum_{i,j \in [q]} \left( \frac{\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j)}{\sqrt{\mathbb{P}(X = i, Y = j)} + \sqrt{\mathbb{P}(X = i)\mathbb{P}(Y = j)}} \right)^2 \\
&\geq \frac{1}{2 \ln 2} \sum_{i,j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2
\end{aligned}$$

Upper bounding  $\ln 2$  by 1 finishes the proof. □

# Chapter 4

## LP and SDP Relaxations

### 4.1 Introduction

Given a combinatorial optimization problem  $\Lambda$ , there are numerous ways of writing a relaxation in order to obtain approximation algorithms for  $\Lambda$ . Due to the sheer diversity of combinatorial optimization problems, it's practically impossible to find a canonical way to write the "correct" relaxation for every problem as different problems warrant very different constraints in the relaxation. Coming up with the best relaxation has been a major part of the effort in designing approximation algorithms in the past few decades.

Of course the ultimate question to ask would be: given a combinatorial optimization problem  $\Lambda$ , what is the "best" relaxation one could write for  $\Lambda$ ? However, before even attempting to answer this question, we need to first clarify two questions:

- Given a problem  $\Lambda$ , what is considered as a valid relaxation? For example, given an instance  $\mathfrak{I}$  of  $\Lambda$ , one could use brute-force algorithm to compute the optimum of  $\mathfrak{I}$  then construct an LP/SDP such that the optimum is exactly this value. Of course this kind of LPs/SDPs should not be considered valid. Therefore we need a clear definition of what kind of LPs/SDPs are considered as valid relaxation.
- As one can always add valid constraints to an (non-tight) LP/SDP to make it tighter, it's hard to define "best" relaxation under the polynomial solvable regime. Henceforth it would be preferable to have a clear measurement of the size of an LP/SDP and the question would become: given a combinatorial optimization problem  $\Lambda$  and size  $k$ , what is the "best" relaxation one could write for  $\Lambda$  of size at most  $k$ ?

We will formally define generic LP/SDP relaxation of a combinatorial optimization problem and the size of an LP/SDP in the next section.

Just like the duality between approximation algorithms and hardness of approximation results, there are also two lines of work trying to answer this question. On one hand, people have been trying to develop different relaxations in order to better approximate a combinatorial problem, this is usually done by adding constraints which are satisfied by an integer solution. In particular, one could add constraints that involves at most  $k$  variables and let  $k$  increase. This process of generating stronger relaxations by adding larger but *local* constraints is captured by various hierarchies of LP/SDP relaxations such as the one defined by Lovász and Schrijver [54], Sherali and Adams [68] and Lasserre [51]. Start from a basic relaxations, these hierarchies define various levels of convex relaxations for a problem, with the relaxation at a high level being more powerful than the relaxations at lower levels. These hierarchies are known to capture the LP/SDP used in the best known algorithms for many problems, such as the SDP relaxation for SPARSEST CUT by Arora, Rao and Vazirani [4] and the  $\theta$ -function of Lovász for MAXIMUM INDEPENDENT SET [53], within a constant number of levels. It is also known that for an integer program with  $n$  variables taking values in  $\{0, 1\}$ , the  $n$ -levels of the hierarchies mentioned above all have integrality gap 1, i.e., it produces the exact solution. However, writing/solving  $t$ -levels of the hierarchies takes time  $O(n^{O(t)})$  time which is exponential when  $t = \Sigma(n)$ . We will briefly define these hierarchies and give some examples in this chapter. In Chapter 5, we will give a generic framework to round Sum-of-Squares Hierarchy for CSPs as well as CSPs with global cardinality constraints.

On the other hand, people have also been trying to prove *lower bound* for LP/SDP, i.e., showing that there doesn't exist any LP/SDP within certain size that achieves some approximation guarantee. This long line of work all started with the groundbreaking work of Yannakakis [72]. He proved that the TSP and matching polytopes do not admit symmetric linear programming formulations of size  $2^{o(n)}$ , where  $n$  is the number of vertices in the underlying graph. In the process, he laid the structural framework (in terms of non-negative factorizations) that would underlie all future work in the subject. In Chapter 6, we will survey recent breakthroughs for symmetric and general LP/SDP relaxations for various combinatorial optimization problems, also show that for constraint satisfaction problems (CSPs), Sum-of-Squares SDP gives the best possible approximation ratio among symmetric SDPs of similar size.

## 4.2 Generic LP and SDP Relaxation For Max-CSPs

In this section we formally define the computation model of linear/semidefinite relaxations for combinatorial optimization problems (more specifically Max

CSPs) that we consider in this thesis. In Chapter 6, we will prove symmetric SDP relaxation lower bound using this model.

Let us take MAX CUT as an example. Given a graph  $G = (V, E)$  with  $|V| = n$ . For any  $S \subset V$ , let us define the cut value of  $(S, \bar{S})$  as

$$G(S) = \frac{|E(S, \bar{S})|}{|E|}$$

which represents the fraction of edges crossing the cut  $(S, \bar{S})$ . Therefore the max cut of the graph is  $\text{opt}(G) = \max_{S \subset V} G(S)$ .

**An attempt to write an LP relaxation for MAX CUT.** Recall that the integer program for MAX CUT is as follows:

Integer program for MAX CUT

$$\begin{aligned} & \text{maximize} && \sum_{e=(u,v) \in E} |X_u - X_v| \\ & \text{subject to} && X_v \in \{0, 1\} \quad \text{for every vertex } v \in V \end{aligned}$$

In order to relax the integer program above into an linear program, we need to resolve two issues: First of all, the constraint  $X_v \in \{0, 1\}$  is not a linear constraint, in fact it's not even a convex constraint as the underlying feasible space is discrete. Secondly the objective function  $\sum_{e=(u,v) \in E} |X_u - X_v|$  is also non-linear.

As for the first issue, one natural solution would be to relax the condition  $X_v \in \{0, 1\}$  into a linear constraints  $0 \leq X_v \leq 1$ .

The second issue is slightly trickier, as the function  $|X_u - X_v|$  cannot be naturally expressed as a linear function of  $X_u$  and  $X_v$ . A common trick to deal with situation like this is to introduce auxiliary variables. For example, in order to express a nonlinear constraint that looks like  $|x| \leq c$ , one can introduce a new variable  $y$  that represents  $|x|$ . In order to enforce that, one need to add two new constraints  $y \geq x$  as well as  $y \geq -x$ . These two constraints together with constraint  $y \leq c$  will create a polytope on variables  $x$  and  $y$  such that its *projection* onto variable  $x$  gives us exactly the polytope  $|x| \leq c$  that we wanted. The new polytope of  $x$  and  $y$  we constructed is called an *extended formulation* of the original polytope.

Back to MAX CUT problem, using the similar idea, we could introduce new variables  $Y_{u,v}$  that is intended to represent the value of  $|X_u - X_v|$ . However, simply adding constraints  $Y_{u,v} \geq X_u - X_v$  and  $Y_{u,v} \geq X_v - X_u$  no longer works, as in this case  $Y_{u,v}$  would simply be unbounded since we're dealing with a maximization problem.

Therefore, in order to enforce  $Y_{u,v}$  behaves exactly like  $|X_u - X_v|$ , we need to add more constraints. One possible way to do so is the following: we want to make sure that there exists some *local distribution*  $\mu_{u,v}$  on the variables  $X_u$  and  $X_v$  such that  $Y_{u,v} = \mathbb{E}_{\mu_{u,v}} |X_u - X_v|$ . To do so, we can introduce 4 new variables  $P_{u=0,v=0}$ ,  $P_{u=0,v=1}$ ,  $P_{u=1,v=0}$  and  $P_{u=1,v=1}$  that represents the probability of each event in the distribution, and add constraint  $Y_{u,v} = \mathbb{E}_{\mu_{u,v}} |X_u - X_v| = P_{u=0,v=1} + P_{u=1,v=0}$ . Note that we can easily add linear constraints to ensure that they indeed form a probability distribution. Now we have a reasonable despite cumbersome LP relaxation for MAX CUT problem as follows:

An linear program for MAX CUT

$$\begin{aligned} & \text{maximize} && \sum_{e=(u,v) \in E} Y_{u,v} \\ & \text{subject to} && 0 \leq X_v \leq 1 \quad \text{for every vertex } v \in V \\ & && Y_{u,v} = P_{u=0,v=1} + P_{u=1,v=0} \quad \text{for every pair } (u, v) \\ & && P_{u,v} \in \Delta_{\{0,1\}^2} \quad \text{for every } u, v \in V \end{aligned}$$

Here  $\Delta_{\{0,1\}^2}$  denotes the convex hull of all the distribution on  $\{0, 1\}^2$ .

In fact, the linear program above is exactly the second level of Sherali-Adams relaxation for MAX CUT problem.

**Generic LP/SDP Relaxation.** Of course, the LP relaxation we showed above is only one possible way of writing LP relaxation for MAX CUT problem. We need a way to characterize *all* possible relaxation for this problem.

First off, a seemingly trivial property about our LP relaxation is that, given any cut  $S \subset V$ , one can construct a feasible solution to the LP  $\{X, Y, P\}_S$  (in the most obvious way) such that the objective function is exactly the cut value.

Second thing to observe is that the linear constraints in the LP above does not depend on the actual graph, except the size of the graph  $|V|$ . In fact, all the information of the graph is encoded in the objective function – this is called *linearization* of the graph.

In fact, the properties above are all we need from a relaxation. Formally, we require the a relaxation to satisfy the following condition: For every  $n \in \mathbb{N}$ , there exists a number  $D(n) \in \mathbb{N}$  and two mappings:

- First mapping is from an arbitrary graph  $G$  of size  $n$  to a *feasible* vector  $\mathbf{v}_G \in \mathbb{R}^{D(n)}$  for the LP
- Second mapping is from an arbitrary cut  $S \subseteq [n]$  (note that  $S$  does not depend on  $G$ ) to a vector  $\mathbf{v}_S \in \mathbb{R}^{D(n)}$

such that for all graphs  $G$  and cut  $S$ , we have

$$G(S) = \langle \mathbf{v}_G, \mathbf{y}_S \rangle$$

Moreover, the objective function of the relaxation should be the following:

$$\max \langle \mathbf{v}_G, \mathbf{y}_S \rangle$$

Given a feasible relaxation of this form, we say that  $D(n)$  is the *size* of the relaxation.

Notice that the definition above didn't touch upon the fact that the relaxation is a linear program. The only requirement is that the cuts of the graph has to be able to be embedded into the *feasible region* of the relaxation in a way such that the inner products between these vectors and the graph specific vectors ( $\mathbf{v}_G$  gives exactly the cut value). Hence, this definition handily carries over to SDP as well. We will give a more detailed definition for SDP case in Chapter 6.

### 4.3 Sum-of-Squares SDP Hierarchy

We devote this section to Sum-of-Squares SDP hierarchy, where we will give a relatively detailed description of Sum-of-Squares SDP hierarchy for Max-CSPs, as well as some properties that we will be using in this thesis. Sum-of-Squares SDP hierarchy, also known as Lasserre or Lasserre-Parrilo SDP hierarchy, is an extremely powerful tool for obtaining approximation algorithms. In fact, a few rounds of Sum-of-Squares SDP hierarchy already capture the best known algorithms for many problems including SPARSEST CUT, VERTEX COVER, and all Max-CSPs.

To motivate Sum-of-Squares SDP hierarchy for Max-CSPs, we will again start with MAX CUT problem.

Recall that in the previous section we derived a linear program for MAX CUT as follows:

A linear program for MAX CUT

$$\begin{aligned} & \text{maximize} && \sum_{e=(u,v) \in E} Y_{u,v} \\ & \text{subject to} && 0 \leq X_v \leq 1 \quad \text{for every vertex } v \in V \\ & && Y_{u,v} = P_{u=0,v=1} + P_{u=1,v=0} \quad \text{for every pair } (u,v) \\ & && P_{u,v} \in \Delta_{\{0,1\}^2} \quad \text{for every } u,v \in V \end{aligned}$$

We will start by trying to write an analogue SDP version of this LP. To do so, we will introduce two new variables  $\mathbf{w}_{v,0}$  and  $\mathbf{w}_{v,1}$ . In the intended solution,

one of them is equal to some constant unit vector  $\mathbf{I}$  and the other one is equal to 0, depending on whether  $X_u$  is 0 or 1. Hence we can add the following (valid) constraint:

$$\begin{aligned}\mathbf{w}_{v,0} \cdot \mathbf{w}_{v,1} &= 0 \\ (\mathbf{w}_{v,0} + \mathbf{w}_{v,1} - \mathbf{I})^2 &= 0\end{aligned}$$

Note that the first constraint implies the orthogonality between  $\mathbf{w}_{v,0}$  and  $\mathbf{w}_{v,1}$ .

Also observe that in the intended integral solution, the inner product  $\langle \mathbf{w}_{v,\alpha}, \mathbf{w}_{u,\beta} \rangle$  exactly captures the probability of the event  $X_v = \alpha \wedge X_u = \beta$  happening, this constraint alone already captures all the constraints we wrote above.

Combining the observations above, we obtain the following SDP relaxation:

A simple SDP for MAX CUT

$$\begin{aligned}\text{maximize} \quad & \sum_{e=(u,v) \in E} P_{u=1,v=0} + P_{u=0,v=1} \\ \text{subject to} \quad & \mathbf{w}_{u,\alpha} \cdot \mathbf{w}_{v,\alpha} = P_{u=\alpha,v=\beta} \quad \text{for every } u, v \in V \text{ and possibly } u = v \\ & P_{u,v} \in \Delta_{\{0,1\}^2} \text{ for every } u, v \in V\end{aligned}$$

In fact this SDP looks quite similar to the LP we had before – the only difference is that in the local distributions the probabilities are no longer some arbitrary distribution on  $\{0, 1\}^2$ , but instead has to come from inner product of some vectors, hence more restricted.

On first glance this small restriction does not seem too significant. However, as it turns out, this additional constraint significantly improves the approximation power of the relaxation. While it is well known the the simple LP (and its improved versions) does not achieve anything beyond the somewhat trivial 1/2-approximation for MAX CUT problem[59], this simple SDP actually achieves 0.878-approximation for MAX CUT algorithm, which is optimal under Unique Games Conjecture. Below we prove it by comparing this SDP to the Goemans-Williamson SDP relaxation 2.2. Recall that the Goemans-Williamson SDP is as follows:

Goemans-Williamson SDP relaxation for MAX CUT.

$$\begin{aligned}\text{maximize} \quad & \sum_{e=(u,v) \in E} (X_u - X_v)^2 / 4 \\ \text{subject to} \quad & X_v^2 = 1 \quad \text{for every vertex } v \in V\end{aligned}$$



**Lemma 4.3.1.** *The SDP 4.3 is at least as good as the Goemans-Williamson SDP.*

*Proof.* In order to show the lemma, all we need to show is that given an SDP solution to the simple SDP 4.3, we can convert it into a Goemans-Williamson SDP solution on the same graph with the same objective value.

In fact, the conversion is quite simple: for each vertex  $v \in V$ , define  $X_v = \mathbf{w}_{v,0} - \mathbf{w}_{v,1}$ .

First we show that  $X_v$  is an unit vector.

To see this, observe that

$$\begin{aligned} X_v^2 &= \mathbf{w}_{v,0} \cdot \mathbf{w}_{v,0} + \mathbf{w}_{v,0} \cdot \mathbf{w}_{v,1} + \mathbf{w}_{v,1} \cdot \mathbf{w}_{v,0} + \mathbf{w}_{v,1} \cdot \mathbf{w}_{v,1} \\ &= P_{v=0} + 0 + 0 + P_{v=1} \\ &= 1 \end{aligned}$$

We also have to show that the objective function does not suffer from this conversion. To this end observe that for each edge  $e = (u, v)$ , the following holds:

$$\begin{aligned} \frac{(X_u - X_v)^2}{4} &= \frac{(\mathbf{w}_{u,0} - \mathbf{w}_{u,1} - \mathbf{w}_{v,0} + \mathbf{w}_{v,1})^2}{4} \\ &= \frac{P_{u=0} + P_{u=1} + P_{v=0} + P_{v=1} - P_{u=0,v=0} - P_{u=1,v=1} + P_{u=0,v=1} + P_{u=1,v=0}}{4} \\ &= P_{u=0,v=1} + P_{u=1,v=0} \end{aligned}$$

Hence the SDP value does not change, and the lemma follows.  $\square$

Now we have a simple SDP to start with, we can try to strengthen it by adding more constraints. Recall that we achieved better approximation ratio by enforcing the underlying local distribution to be consistent with inner product of some vectors. There are several natural ways that one could add more constraints:

- Current SDP only enforces local distribution on every subset of 2 variables, we could extend this to local distribution on subset of  $k$  variables for some constant  $k$  while maintaining polynomial size.
- Right now the vectors are indexed by a variable and an assignment of the variable, we can extend this to a subset of  $k$  variables and an assignment of the subset, for some constant  $k$ .
- We can also add constraint to enforce consistency across local distributions, i.e., for two subset  $S$  and  $T$  such that  $S \cap T \neq \emptyset$ , the marginal distribution of  $P_S$  and  $P_T$  restricted onto  $S \cap T$  should be consistent.

In fact, this is exactly what  $k$ -rounds of Sum-of-Squares SDP hierarchy does! For concreteness, we write down the Lasserre SDP relaxation for MAX CUT problem below.

**Sum-of-Squares SDP hierarchy for MAX CUT.** In  $k$ -rounds of Sum-of-Squares SDP consists of the following variables:

- An unit constant vector  $\mathbf{I}$
- For every subset  $S \subseteq V$ ,  $|S| \leq k$  and an assignment of the variables in  $S$   $\alpha$ , a vector  $\mathbf{v}_{S,\alpha}$
- For every subset  $S \subseteq V$ ,  $|S| \leq k$ , a local distribution  $\mu_S \in \Delta(\{0, 1\}^S)$

The SDP is as follows:

$k$ -rounds of Sum-of-Squares SDP relaxation for MAX CUT.

$$\begin{aligned}
 &\text{maximize} && \sum_{e=(u,v) \in E} \mathbb{P}_{\mu_{\{u,v\}}}(X_u \neq X_v) \\
 &\text{subject to} && \sum_{\alpha \in \{0,1\}^S} \mathbf{v}_{S,\alpha} = \mathbf{I} \quad \text{for every subset } |S| \leq k \\
 &&& \langle \mathbf{v}_{S,\alpha}, \mathbf{v}_{T,\beta} \rangle = \mathbb{P}_{\mu_{S \cup T}}(X_S = \alpha, X_T = \beta) \quad \text{for every subset } S, T \text{ such that } |S \cup T| \leq k \\
 &&& \mu_S \in \Delta(\{0, 1\}^S) \quad \text{for subset } |S| \leq k
 \end{aligned}$$

**Remark 4.3.2.** This SDP has  $O(n^k \times 2^k)$  variables therefore can be solved in polynomial time for any constant  $k$ .

**Remark 4.3.3.** The variables  $\mu_S$  are not necessary in the SDP as they can be replaced by the inner product of the vectors. We only write them down to increase readability of the relaxation.

## Chapter 5

# An Improved Approximation Algorithm for MAX BISECTION

In this chapter we will give an example on how to utilize the power of Sum-of-Squares SDP hierarchy in the context of Max-CSPs with global cardinality constraints. In particular, we will give a simple yet powerful framework to *round* Sum-of-Squares SDP solutions. To illustrate this framework, we will present an improved algorithm for the MAX BISECTION problem.

### 5.1 Introduction

As we mentioned in the previous chapters, Constraint Satisfaction Problems (CSP) are a class of fundamental optimization problems that have been extensively studied in approximation algorithms and hardness of approximation. Recall that in a constraint satisfaction problem, the input consists of a set of variables taking values over a fixed finite domain (say  $\{0, 1\}$ ) and a set of *local* constraints on them. The constraints are *local* in that each of them depends on at most  $k$  variables for some fixed constant  $k$ . The goal is to find an assignment to the variables that satisfies the maximum number of constraints.

Over the last two decades, there has been much progress in understanding the approximability of CSPs. On the algorithmic front, semidefinite programming (SDP) has been used with great success in approximating several well-known CSPs such as MAX CUT [31], MAX 2-SAT [13] and MAX 3-SAT [46]. More recently, these algorithmic results have been unified and generalized to the entire class of constraint satisfaction problems [64]. With the development of PCPs and long code based reductions, tight hardness results matching the SDP based algorithms have been shown for some CSPs such as Max-3-SAT [40]. In a surprising development under the Unique Games Conjecture, semidefinite programming based algorithms have been shown to be optimal for MAX CUT [49], MAX 2-SAT

[5] and more generally every constraint satisfaction problem [61].

Unfortunately, neither SDP based algorithms nor the hardness results extend satisfactorily to optimization problems with *non-local* constraints. Part of the reason is that the nice framework of SDP based approximation algorithms and matching hardness results crucially rely on the *locality* of the constraints involved. Perhaps the simplest non-local constraint would be to restrict the cardinality of the assignment, i.e., the number of ones in the assignment. Variants of CSPs with even a single cardinality constraint are not well-understood. Optimization problems of this nature, namely constraint satisfaction problems with global cardinality constraints are the primary focus of this work. Several important problems such as MAX BISECTION, MIN BISECTION, SMALL-SET EXPANSION can be formulated as CSPs with a single global cardinality constraint.

As an illustrative example, let us consider the MAX BISECTION problem which is also part of the focus of this chapter. The MAX BISECTION problem is a variant of the much well-studied MAX CUT problem [31, 49]. In the MAX CUT problem the goal is to partition the vertices of the input graph in to two sets while maximizing the number of crossing edges. The MAX BISECTION problem includes an additional cardinality constraint that both sides of the partition have exactly half the vertices of the graph. The seemingly mild cardinality constraint appears to change the nature of the problem. While MAX CUT admits a factor 0.878 approximation algorithm [31], the best known approximation factor for MAX BISECTION equals 0.7027 [25], improving on previous bounds of 0.6514 [28], 0.699 [73], and 0.7016 [36]. These algorithms proceed by rounding the natural semidefinite programming relaxation analogous to the Goemans-Williamson SDP for MAX CUT. In a recent work, Guruswami et al. [35] showed that this natural SDP relaxation has a large integrality gap: the SDP optimum could be 1 whereas every bisection might only cut less than 0.95 fraction of the edges! In particular, this implies that none of these algorithms guarantee a solution with value close to 1 even if there exists a perfect bisection in the graph. More recently, using a combination of graph-decomposition, brute-force enumeration and SDP rounding, Guruswami et al. [35] obtained an algorithm that outputs a  $1 - O(\epsilon^{1/3} \log(1/\epsilon))$  bisection on a graph that has a bisection of value  $1 - \epsilon$ .

A simple approximation preserving reduction from MAX CUT shows that MAX BISECTION is no easier to approximate than MAX CUT (the reduction is simply to take two disjoint copies of the MAX CUT instance). Therefore, the factor 16/17 NP-hardness [40, 69] and the factor 0.878 Unique-Games hardness for MAX CUT [49] also applies to the MAX BISECTION problem. In fact, a stronger hardness result of factor 15/16 was shown in [41] assuming  $\text{NP} \not\subseteq \bigcap_{\gamma > 0} \text{TIME}(2^{n^\gamma})$ . Yet, these hardness results for MAX BISECTION are far from matching the best known approximation algorithm that only achieves a 0.702 factor.

## 5.2 Statement of Results

In this chapter, we develop a general approach to approximate CSPs with global cardinality constraints using the Sum-of-Squares SDP hierarchy.

We illustrate the approach with an improved approximation algorithm for the MAX BISECTION and balanced MAX 2-SAT problems. For the MAX BISECTION problem, we show the following result.

**Theorem 5.2.1.** *For every  $\delta > 0$ , there exists an algorithm for MAX BISECTION that runs in time  $O(n^{\text{poly}(1/\delta)})$  and obtains the following approximation guarantees,*

- *The output bisection has value at least  $0.85 - \delta$  times the optimal max bisection.*
- *For every  $\varepsilon > 0$ , given an instance  $G$  with a bisection of value  $1 - \varepsilon$ , the algorithm outputs a bisection of value at least  $1 - O(\sqrt{\varepsilon}) - \delta$ .*

Note that the approximation guarantee of  $1 - O(\sqrt{\varepsilon})$  on instances with  $1 - \varepsilon$  is nearly optimal (up to constant factors in the  $O()$ ) under the Unique Games Conjecture. This follows from the corresponding hardness of MAX CUT and the reduction from MAX CUT to MAX BISECTION.

Our approach is robust in that it also yields similar approximation guarantees to the more general  $\alpha$ -MAX CUT problem where the goal is to find a cut with exactly  $\alpha$ -fraction of vertices on one side of the cut. More generally, the algorithm also generalizes to a weighted version of MAX BISECTION, where the vertices have weights and the cut has approximately half the weight on each side.<sup>1</sup>

The same algorithm also yields an approximation to the complementary problem of MIN BISECTION. Formally, we obtain the following approximation algorithm for MIN BISECTION and  $\alpha$ -BALANCED SEPARATOR.

**Theorem 5.2.2.** *For every  $\delta > 0$ , there exists an algorithm running in time  $O(n^{O(\text{poly}(1/\delta))})$ , which given a graph with a bisection ( $\alpha$ -balanced separator) cutting  $\varepsilon$ -fraction of the edges, finds a bisection ( $\alpha$ -balanced separator) cutting at most  $O(\sqrt{\varepsilon}) + \delta$ -fraction of edges.*

## 5.3 Overview of Techniques

In this section, we outline our approach of approximating the MAX BISECTION problem. The techniques are fairly general and can be applied to other CSPs with global cardinality constraints as well.

---

<sup>1</sup>Note that in the weighted case, finding any exact bisection is at least as hard as subset-sum problem.

**Global Correlation.** For the sake of exposition, let us recall the Goemans and Williamson algorithm for MAX CUT. For more detailed description and analysis, please refer to Chapter 2. Given a graph  $G = (V, E)$ , the Goemans-Williamson SDP relaxation for MAX CUT assigns a unit vector  $\mathbf{v}_i$  for every vertex  $i \in V$ , so as to maximize the average squared length  $E_{i,j \in E} \|v_i - v_j\|^2$  of the edges. Formally, the SDP relaxation is given by,

$$\text{maximize } \mathbb{E}_{i,j \in E} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \quad \text{subject to } \|\mathbf{v}_i\|_i^2 = 1 \quad \forall i \in V$$

The rounding scheme picks a random halfspace passing through the origin and outputs the partition of the vertices induced by the halfspace. The value of the cut returned is guaranteed to be within a 0.878-factor of the SDP value.

The same algorithm would be an approximation for MAX BISECTION if the cut returned by the algorithm was near-balanced, i.e.,  $|S| \approx |V|/2$ . Indeed, the expected number of vertices on either side of the partition is  $|V|/2$ , since each vertex  $i \in V$  falls on a given side of a random halfspace with probability  $\frac{1}{2}$ .

If the balance of the partition returned is concentrated around its expectation then the Goemans and Williamson algorithm would yield a 0.878-approximation for MAX BISECTION. However, the balance of the partition need not be concentrated, simply because the values taken by vertices could be highly correlated with each other!

**SDP Relaxation.** As we mentioned earlier, the reason that the Goemans-Williamson algorithm does not work well for MAX BISECTION is because the rounded solution might be highly correlated, therefore the balance of the cut might not be concentrated around the expected value. Due to the special formulation of Goemans-Williamson SDP, it is quite difficult to bound the correlation between the vertices, as every unit vector is a feasible solution to Goemans-Williamson SDP. The situation turns out to be quite different for Sum-of-Squares SDP hierarchy - as the Sum-of-Squares SDP solution gives us local distribution among small subsets of vertices, therefore gives us more room to exploit the correlations between the vertices.

For more detailed explanation of Sum-of-Squares SDP hierarchy we refer the reader to Chapter 4. Here we will recall some basic properties about Sum-of-Squares SDP. On a high level, the solutions to a Sum-of-Squares SDP hierarchy are vectors that *locally behave* like a distribution over integral solutions. The  $k$ -round Sum-of-Squares SDP has the following properties similar to a true distribution over integral solutions.

- *Marginal Distributions* For any subset  $S$  of vertices with  $|S| \leq k$ , the SDP will yield a distribution  $\mu_S$  on partial assignments to the vertices  $(\{-1, 1\}^S)$ . The marginals of  $\mu_S, \mu_T$  for a pair of subsets  $S$  and  $T$  are consistent on their intersection  $S \cap T$ .

- *Conditioning* Analogous to a true distribution over integral solutions, for any subset  $S \subseteq V$  with  $|S| \leq k$  and a partial assignment  $\alpha \in \{-1, 1\}^S$ , the SDP solution can be conditioned on the event that  $S$  is assigned  $\alpha$ .

A detailed description of the Sum-of-Squares SDP hierarchy applied specifically to MAX BISECTION will be given in section 5.4.

**Measuring Correlations.** Throughout this chapter we will use mutual information as a measure of correlation between two random variables. We refer the reader to Chapter 2 for the definitions of Shannon entropy and mutual information. Recall that the correlation(mutual information) between two random variables  $X$  and  $Y$  is given by

$$I(X; Y) = H(X) - H(X|Y),$$

The crucial observation is that the definition only depends on *local* distribution between  $X$  and  $Y$ , therefore it is also well defined in a Sum-of-Squares SDP solution. Specifically, given two vertices  $i$  and  $j$ , the mutual information between them is defined as  $i$  and  $j$  is given by

$$I_{\mu_{i,j}}(X_i; X_j) = H(X_i) - H(X_i|X_j),$$

where the random variables  $X_i, X_j$  are sampled using the local distribution  $\mu_{i,j}$  associated with the Sum-of-Squares SDP solution. An SDP solution will be termed  $\alpha$ -independent if the average mutual information between random pairs of vertices is at most  $\alpha$ , i.e.,  $\mathbb{E}_{i,j \in V}[I(X_i; X_j)] \leq \alpha$ .

For most natural rounding schemes such as the halfspace-rounding, the variance of the balance of the cut returned is directly related to the average correlation between random pairs of vertices in the graph. In other words, if the rounding scheme is applied to an  $\alpha$ -independent SDP solution then the variance of the balance of the cut is at most  $\text{poly}(\alpha)$ .

**Obtaining Uncorrelated SDP Solutions.** Intuitively, if it is the case that globally all the vertices are highly correlated, then conditioning on the value of a vertex should reveal information about the remaining vertices, therefore reducing the total entropy of all the vertices.

Formally, let us suppose the  $k$ -rounds Sum-of-Squares SDP solution is not  $\alpha$ -independent, i.e.,  $\mathbb{E}_{i,j \in V}[I(X_i; X_j)] > \alpha$ . In this case, if we randomly pick a vertex  $i \in V$ , sample its value  $b \in \{-1, 1\}$  and condition the SDP solution to the event  $X_i = b$ , this would result in the average entropy vertices ( $\mathbb{E}_{j \in V}[H(X_j)]$ ) by at least  $\alpha$  in expectation. We can keep repeating this process until  $\alpha$ -independence is achieved.

To see that this process will eventually terminate, observe that the initial average entropy  $\mathbb{E}_{j \in V}[H(X_j)]$  is at most 1 (since these are binary variables), and the quantity always remains non-negative. Therefore, within  $\frac{1}{\alpha}$  conditionings, the SDP solution will be  $\alpha$ -independent with high probability. Starting with a  $k$ -round Sum-of-Squares SDP solution, this process produces a  $k - t$  round  $\alpha$ -independent Sum-of-Squares SDP solution for some  $t > \frac{1}{\alpha}$ .

**Rounding Uncorrelated SDP Solutions.** Given an  $\alpha$ -independent SDP solution, for many natural rounding schemes the balance of the output cut is concentrated around its expectation. That is quite convenient since when designing rounding algorithms we do not have to worry about the concentration anymore. Hence we need to ensure that: 1) the rounding scheme should output a balanced cut in expectation and 2) The expected cut value should be good when compared to the SDP value. We exhibit a simple rounding scheme that preserves the bias of each vertex individually, thereby preserving the global balance property. The details of the rounding algorithm will be described in section 5.6.

## 5.4 Preliminaries

**Constraint Satisfaction Problem with Global Cardinality Constraints.** In this section we recall some definitions of CSPs with global constraints from Chapter 2 as well as some information theory notions from Chapter 3.

**Definition 5.4.1** (Constraint Satisfaction Problems with Global Cardinality Constraints). A constraint satisfaction problem with global cardinality constraints is specified by  $\Lambda = ([q], \mathbb{P}, k, c)$  where  $[q] = \{0, \dots, q - 1\}$  is a finite domain,  $\mathbb{P} = \{P : [q]^t \mapsto [0, 1] \mid t \leq k\}$  is a set of payoff functions. The maximum number of inputs to a payoff function is denoted by  $k$ . The map  $c : [q] \mapsto [0, 1]$  is the cardinality function which satisfies  $\sum_i c_i = 1$ . For any  $0 \leq i \leq q - 1$ , the solution should contain  $c_i$  fraction of the variables with value  $i$ .

**Remark 5.4.2.** Although some problems (e.g., BALANCED SEPARATOR) do not fix the cardinalities to be some specific quantities, they can be easily reduced to the above case.

**Definition 5.4.3.** An instance  $\Phi$  of constraint satisfaction problems with global cardinality constraints  $\Lambda = ([q], \mathbb{P}, k, c)$  is given by  $\Phi = (\mathcal{V}, \mathbb{P}_{\mathcal{V}}, W)$  where

- $\mathcal{V} = \{x_1, \dots, x_n\}$ : variables taking values over  $[q]$
- $\mathbb{P}_{\mathcal{V}}$  consists of the payoffs applied to subsets  $S$  of size at most  $k$



- Nonnegative weights  $W = \{w_S\}$  satisfying  $\sum_{|S| \leq k} w_S = 1$ . Thus we may interpret  $W$  as a probability distribution on the subsets. By  $S \sim W$ , we denote a set  $S$  chosen according to the probability distribution  $W$
- An assignment should satisfy that the number of variables with value  $i$  is  $c_i n$  (we may assume this is an integer).

Here we give a few examples of CSPs with global cardinality constraints.

**Definition 5.4.4** (Max(Min) Bisection). Given a (weighted) graph  $G = (V, E)$  with  $|V|$  even, the goal is to partition the vertices into two equal pieces such that the number (total weights) of edges that cross the cut is maximized (minimized).

More generally, in an  $\alpha$ -MAX CUT problem, the goal is to find a partition having  $\alpha n$  vertices on one side, while cutting the maximum number of edges. Furthermore, one could allow weights on the vertices of the graph, and look for cuts with exactly  $\alpha$ -fraction of the weight on one side. Most of our techniques generalize to this setting.

Throughout this work, we will have a weighted graph  $G$  with weights  $W$  on the vertices. The weights on the vertices are assumed to form a probability distribution. Hence the notation  $i \sim W$  refers to a random vertex sampled from the distribution  $W$ .

**Definition 5.4.5** (Edge Expansion). Given a graph (w.l.o.g. we may assume it is a unweighted regular graph)  $G = (V, E)$ , and  $\delta \in (0, 1/2)$ , the goal is to find a set  $S \subseteq V$  such that  $|S| = \delta|V|$  and the edge expansion of  $S$ :  $\Phi(S) = \frac{E(S, \bar{S})}{d|S|}$  is minimized.

First we recall some information theoretical notions we mentioned in Chapter 3.

## Information Theoretic Notions

**Definition 5.4.6.** Let  $X$  be a random variable taking values over  $[q]$ . The *entropy* of  $X$  is defined as

$$H(X) \stackrel{\text{def}}{=} - \sum_{i \in [q]} \mathbb{P}(X = i) \log \mathbb{P}(X = i)$$

**Definition 5.4.7.** Let  $X$  and  $Y$  be two jointly distributed variables taking values over  $[q]$ . The *mutual information* of  $X$  and  $Y$  is defined as

$$I(X; Y) \stackrel{\text{def}}{=} \sum_{i, j \in [q]} \mathbb{P}(X = i, Y = j) \log \frac{\mathbb{P}(X = i, Y = j)}{\mathbb{P}(X = i) \mathbb{P}(Y = j)}$$

**Definition 5.4.8.** Let  $X$  and  $Y$  be two jointly distributed variables taking values over  $[q]$ . The *conditional entropy* of  $X$  conditioned on  $Y$  is defined as

$$H(X|Y) = \mathbb{E}_{i \in [q]} [H(X|Y = i)]$$

We also give two well-known theorems in information theory below.

**Theorem 5.4.9.** Let  $X$  and  $Y$  be two jointly distributed variables taking value on  $[q]$ , then

$$I(X; Y) = H(X) - H(X|Y)$$

**Theorem 5.4.10.** (*Data Processing Inequality*) Let  $X, Y, Z, W$  be random variables such that  $H(X|W) = 0$  and  $H(Y|Z) = 0$ , i.e.,  $X$  is fully determined by  $W$  and  $Y$  is fully determined by  $Z$ , then

$$I(X; Y) \leq I(W; Z)$$

## Mutual Information, Statistical Distance and Independence

Intuitively, when two random variables have low mutual information, they should be close to being independent. In this section we formalize this intuition by giving an explicit bound on the statistical distance between the joint distribution and the independent distribution. We stress that all the results here are sufficient for our use in this work, but we believe the parameters could be further optimized.

We start by defining a few notions that measures the correlation of two random variables.

**Definition 5.4.11.** Let  $\Omega$  be a finite sample space,  $P$  and  $Q$  be two probability distributions on  $\Omega$ . The *square Hellinger distance* of  $P$  and  $Q$  is defined as

$$H^2(P, Q) = \frac{1}{2} \sum_{x \in \Omega} (\sqrt{P(x)} - \sqrt{Q(x)})^2$$

**Definition 5.4.12.** Let  $\Omega$  be a finite sample space,  $P$  and  $Q$  be two probability distributions on  $\Omega$ . The *Kullback-Leibler divergence* of  $P$  and  $Q$  is defined as

$$D_{KL}(P||Q) = \sum_{x \in \Omega} P(x) \log \frac{P(x)}{Q(x)}$$

Now we give a few facts regarding mutual information, Hellinger distance and Kullback-Leibler divergence without proving them.

**Fact 5.4.13.** *Let  $X$  and  $Y$  be two jointly distributed random variables taking value in  $[q]$ , then*

$$I(X; Y) = D_{KL}(p(x, y) \| p(x) \times p(y)).$$

where  $p(x, y)$  is the joint distribution of  $X$  and  $Y$  on  $[q]^2$  and  $p(x) \times p(y)$  is the product distribution of the marginal distributions of  $X$  and  $Y$ .

**Fact 5.4.14.** *Let  $\Omega$  be a finite sample space,  $P$  and  $Q$  be two probability distribution on  $\Omega$ , then*

$$D_{KL}(Q \| P) \geq \frac{2}{\ln 2} H^2(P, Q)$$

Combining the facts mentioned above, we get the following relation between mutual information and statistical distance.

**Fact 5.4.15.** *(Restatement of Fact 5.5.3) Let  $X$  and  $Y$  be two jointly distributed random variables on  $[q]$  then,*

$$I(X; Y) \geq \frac{1}{2 \ln 2} \sum_{i, j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2,$$

in particular for all  $i, j \in [q]$

$$|\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j)| \leq \sqrt{2I(X; Y)}$$

As a consequence, if  $X$  and  $Y$  are two random variables defined on  $\{-1, 1\}$ ,  $\text{Cov}(X, Y) \leq O(\sqrt{I(X; Y)})$

**Lemma 5.4.16.** *Let  $X$  and  $Y$  be two jointly distributed random variables on  $[q]$ , we have*

$$I(X; Y) \geq \frac{1}{2 \ln 2} \sum_{i, j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2$$

*Proof.*

$$\begin{aligned} I(X; Y) &= D_{KL}(p(x, y) \| p(x) \times p(y)) \\ &\geq \frac{2}{\ln 2} H^2(p(x, y), p(x) \times p(y)) \\ &= \frac{2}{\ln 2} \sum_{i, j \in [q]} \left( \sqrt{\mathbb{P}(X = i, Y = j)} - \sqrt{\mathbb{P}(X = i)\mathbb{P}(Y = j)} \right)^2 \\ &= \frac{2}{\ln 2} \sum_{i, j \in [q]} \left( \frac{\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j)}{\sqrt{\mathbb{P}(X = i, Y = j)} + \sqrt{\mathbb{P}(X = i)\mathbb{P}(Y = j)}} \right)^2 \\ &\geq \frac{1}{2 \ln 2} \sum_{i, j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2 \end{aligned}$$

Upper bounding  $\ln 2$  by 1 finishes the proof.  $\square$

**Sum-of-Squares SDP hierarchy for Globally Constrained CSPs.** Below we will give a detailed description of Sum-of-Squares SDP relaxation for CSPs with global cardinality constraints. The relaxation is quite similar to the Sum-of-Squares SDP relaxation we presented in Chapter 4 for MAX CUT problem. The only difference is that we add extra constraints to enforce the global cardinality property.

Formally, let  $\Lambda = ([q], \mathbb{P}, k, c)$  be a CSP with global constraints and  $\Phi = (\mathcal{V}, \mathbb{P}_{\mathcal{V}}, W)$  be an instance of  $\Lambda$  on variables  $X = \{x_1, \dots, x_n\}$ . A solution to the  $k$ -round Sum-of-Squares SDP consists of vectors  $v_{S,\alpha}$  for all vertex sets  $S \subseteq V$  with  $|S| \leq k$  and local assignments  $\alpha \in [q]^S$ . Also for each subset  $S \subseteq V$  with  $|S| \leq k$ , there is a distribution  $\mu_S$  on  $[q]^S$ . For two subsets  $S, T$  such that  $|S|, |T| \leq k$ , we require that the corresponding distributions  $\mu_S$  and  $\mu_T$  are consistent when restricted to  $S \cap T$ . A Sum-of-Squares solution is feasible if for any  $|S \cup T| \leq k$ ,  $\alpha \in [q]^S, \beta \in [q]^T$ , we have

$$\langle v_{S,\alpha}, v_{T,\beta} \rangle = \mathbb{P}_{\mu_{S \cup T}} \{X_S = \alpha, X_T = \beta\}$$

The SDP also has a vector  $I$  that denotes the constant 1. The global cardinality constraints can be written in terms of the marginals of each variable. Specifically, for every  $S$  with  $|S| \leq k - 1$  and  $\alpha \in [q]^S$ , we have

$$\mathbb{E}_j \mathbb{P}_{\mu_{S \cup \{x_j\}}} (x_j = i | X_S = \alpha) = c_i$$

The objective of the SDP is to maximize

$$\mathbb{E}_{S \in W} \left( \sum_{\beta \in [q]^S} P_S(\beta(S)) \mathbb{P}_{\mu_S}(S, \beta) \right)$$

While the complete description of the Sum-of-Squares SDP hierarchy is somewhat complicated, there are few properties of the hierarchy that we need. The most important property is the existence of consistent local marginal distributions  $\{\mu_S\}_{S \subseteq V, |S| \leq k}$  whose first two moments match the inner products of the vectors. We stress that even though the local distributions are consistent, there might not exist a global distribution that agrees with all of them. The second property of the  $k$ -round Sum-of-Squares SDP solution is that although the variables are not jointly distributed, one can still *condition* on the assignment to any given variable to obtain a solution to the  $k - 1$  round Sum-of-Squares SDP that corresponds to the *conditioned distribution*.

## 5.5 Globally Uncorrelated SDP Solutions

As remarked earlier, it is easy to round SDP solutions to a CSP with cardinality constraint if the variables behave like independent random variables. In this sec-

tion, we show a very simple procedure that starts with a solution to the  $(k + l)$ -rounds Sum-of-Squares SDP and produces a solution to the  $l$ -rounds Sum-of-Squares SDP with the additional property that globally the variables are somewhat "uncorrelated". To this end, we define the notion of  $\alpha$ -independence for SDP solutions below. Roughly speaking, if a distribution is  $\alpha$ -independent, then the "average correlation" among the variables is low. We remark that all the definitions and results in this section can be applied to all CSPs.

**Definition 5.5.1.** Given a solution to the  $k$ -round Sum-of-Squares SDP relaxation, it is said to be  $\alpha$ -independent if  $\mathbb{E}_{i,j \sim W}[I_{\mu_{\{i,j\}}}(X_i; X_j)] \leq \alpha$  where  $\mu_{\{i,j\}}$  is the local distribution associated with the pair of vertices  $\{i, j\}$ .

**Remark 5.5.2.** We stress again that the variables in the SDP solution are not jointly distributed. However, the notion is still well-defined here because of the locality of mutual information: it only depends on the joint distribution of two variables, which is guaranteed to exist by the SDP. Also,  $\mu_{\{i,j\}}$  in the expression can be replaced with  $\mu_S$  for arbitrary  $S$  with  $i, j \in S$  and  $|S| \leq k$  because of the consistency of local distributions.

The notion of  $\alpha$ -independence of random variables using mutual information easily translates into more familiar notion of statistical distance. Specifically, we have the following relation as we showed in For the sake of completeness, we include the proof of this observation in section 5.4.

**Fact 5.5.3.** Let  $X$  and  $Y$  be two jointly distributed random variables on  $[q]$  then,

$$I(X; Y) \geq \frac{1}{2 \ln 2} \sum_{i,j \in [q]} (\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j))^2,$$

in particular for all  $i, j \in [q]$

$$|\mathbb{P}(X = i, Y = j) - \mathbb{P}(X = i)\mathbb{P}(Y = j)| \leq \sqrt{2I(X; Y)}$$

As a consequence, if  $X$  and  $Y$  are two random variables defined on  $\{-1, 1\}$ ,  $\text{Cov}(X, Y) \leq O(\sqrt{I(X; Y)})$

Now we describe the procedure of getting an  $\alpha$ -independent  $l$ -rounds Sum-of-Squares solution. A similar argument was concurrently discovered in [7]. Here we reproduce the argument in information theoretic terms, while [7] present the argument in terms of covariance. The information theoretic argument is somewhat robust and cleaner in that it is independent of the sample space involved.

**Algorithm 5.5.4. Input:** A feasible solution to the  $(k + \ell)$  round Sum-of-Squares SDP relaxation as described in section 5.4 for  $k = 1/\sqrt{\alpha}$ .

**Output:** An  $\alpha$ -independent solution to the  $\ell$  round Sum-of-Squares SDP relaxation.

Sample indices  $i_1, \dots, i_k \subseteq V$  independently according to  $W$ . Set  $t = 1$ .

Until the SDP solution is  $\alpha$ -independent repeat

- Sample the variable  $X_{i_t}$  from its marginal distribution after the first  $t - 1$  fixings, and condition the SDP solution on the outcome.
- $t = t + 1$ .

The following lemma shows that there exists  $t$  such that the resulting solution is  $\alpha$ -independent after  $t$ -conditionings with high probability.

**Lemma 5.5.5.** *There exists  $t \leq k$  such that*

$$\mathbb{E}_{i_1, \dots, i_t \sim W} \mathbb{E}_{i, j \sim W} [I(X_i, X_j | X_{i_1}, \dots, X_{i_t})] \leq \frac{\log q}{k - 1}$$

*Proof.* By linearity of expectation, we have that for any  $t \leq k - 2$

$$\begin{aligned} & \mathbb{E}_{i, i_1, \dots, i_t \sim W} [H(X_i | X_{i_1}, \dots, X_{i_t})] \\ &= \mathbb{E}_{i, i_1, \dots, i_t \sim W} [H(X_i | X_{i_1}, \dots, X_{i_{t-1}})] - \mathbb{E}_{i_1, \dots, i_{t-1} \sim W} \mathbb{E}_{i, i_t \sim W} [I(X_i, X_{i_t} | X_{i_1}, \dots, X_{i_{t-1}})] \end{aligned}$$

adding the equalities from  $t = 1$  to  $t = k - 2$ , we get

$$\mathbb{E}_{i \sim W} [H(X_i)] - \mathbb{E}_{i_1, \dots, i_{k-2} \sim W} [H(X_i | X_{i_1}, \dots, X_{i_{k-2}})] = \sum_{1 \leq t \leq k-1} \mathbb{E}_{i, j, i_1, \dots, i_{t-1} \sim W} [I(X_i, X_j | X_{i_1}, \dots, X_{i_{t-1}})]$$

The lemma follows from the fact that for each  $i$ ,  $H(X_i) \leq \log q$ .  $\square$

**Theorem 5.5.6.** *For every  $\alpha > 0$  and positive integer  $\ell$ , there exists an algorithm running in time  $O(n^{\text{poly}(1/\alpha) + \ell})$  that finds an  $\alpha$ -independent solution to the  $\ell$ -round Sum-of-Squares SDP, with an SDP objective value of at least  $\text{OPT} - \alpha$ , where  $\text{OPT}$  denotes the optimum value of the  $\ell$ -round Sum-of-Squares SDP relaxation.*

*Proof.* Pick  $k = \frac{4 \log q}{\alpha^2}$ . Solve the  $k + \ell$  round Sum-of-Squares SDP solution, and use it as input to the conditioning algorithm described earlier. Notice that the algorithm respects the marginal distributions provided by the SDP while sampling the values to variables. Therefore, the expected objective value of the SDP solution after conditioning is exactly equal to the SDP objective value before conditioning. Also notice that the SDP value is at most 1. Therefore, the probability of the SDP value dropping by at least  $\alpha$  due to conditioning is at most  $1/(1 + \alpha)$ .

Also, by Lemma 5.5.5 and Markov Inequality, the probability of the algorithm failing to find a  $\sqrt{\frac{\log q}{k}}$ -independent solution is at most  $\sqrt{\frac{\log q}{k}}$ . Therefore, by union bound, there exists a fixing such that the SDP value is maintained up to  $\alpha$ , and the solution after conditioning is  $\alpha$ -independent. Moreover, this particular fixing can be found using brute-force search.  $\square$

## 5.6 Rounding Scheme for MAX BISECTION

In this section, we present and analyze a natural rounding scheme for MAX BISECTION. Given an globally uncorrelated SDP solution to a 2-round Sum-of-Squares SDP relaxation of MAX BISECTION, the rounding scheme will output a cut with the approximation guarantees outlined in Theorem 5.2.1. The same rounding scheme also yields a 0.92-approximation algorithm for arbitrary globally constrained MAX 2-SAT problem.

**Constructing Goemans-Williamson type SDP solution.** In the 2-round Sum-of-Squares SDP for MAX BISECTION, there are two orthogonal vectors  $v_{i0}$  and  $v_{i1}$  for each variable  $x_i$ . This can be used to obtain a solution to the Goemans-Williamson SDP solution by simply defining  $v_i \stackrel{\text{def}}{=} v_{i0} - v_{i1}$ . The following proposition is an easy consequence,

**Proposition 5.6.1.** *Let  $v_i = v_{i0} - v_{i1} = (2p_i - 1)I + w_i$  where  $p_i = \mathbb{P}(x_i = 0)$ . Then, for each edge  $e = (i, j) \in E$ ,  $\mathbb{P}_{\mu_e}(x_i \neq x_j) = \|v_i - v_j\|^2/4$ .*

*Proof.*

$$\|v_i - v_j\|^2 = 2 - 2\langle v_{i0} - v_{i1}, v_{j0} - v_{j1} \rangle = 2 - 2(\mathbb{P}_{\mu_e}(x_i = x_j) - \mathbb{P}_{\mu_e}(x_i \neq x_j)) = 4\mathbb{P}_{\mu_e}(x_i \neq x_j)$$

$\square$

Let  $w_i$  be the component of  $v_i$  orthogonal to the  $I$  vector, i.e.,  $w_i \stackrel{\text{def}}{=} (v_i - \langle v_i, I \rangle I)$ . Using  $v_{i0} + v_{i1} = I$  and  $\langle v_{i0}, v_{i1} \rangle = 0$ , we get  $v_{i0} = \langle v_{i0}, I \rangle I + w_i/2$  and  $v_{i1} = \langle v_{i1}, I \rangle I - w_i/2$ .

We remark that  $w_i$  is the crucial component that captures the *correlation* between  $x_i$  and other variables. To formalize this, we show the following lemma.

**Lemma 5.6.2.** *Let  $v_i$  and  $v_j$  be the unit vectors constructed above,  $w_i$  and  $w_j$  be the components of  $v_i$  and  $v_j$  that orthogonal to  $I$ . Then  $|\langle w_i, w_j \rangle| \leq 4\sqrt{2I(x_i, x_j)}$*

*Proof.* Let  $p_i \stackrel{\text{def}}{=} \mathbb{P}(x_i = 0) = \langle v_{i0}, I \rangle$  and  $p_j \stackrel{\text{def}}{=} \mathbb{P}(x_j = 0) = \langle v_{j0}, I \rangle$ . Notice that

$$|\mathbb{P}(x_i = 0, x_j = 0) - \mathbb{P}(x_i = 0)\mathbb{P}(x_j = 0)| = |\langle p_i I + w_i/2, p_j I + w_j/2 \rangle - p_i p_j| = |\langle w_i, w_j \rangle|/4$$

By applying fact 5.5.3, we get  $|\langle w_i, w_j \rangle| \leq 4\sqrt{2I(x_i, x_j)}$   $\square$

Henceforth we will switch from the alphabet  $\{0, 1\}$  to  $\{-1, 1\}$ <sup>2</sup>. After this transformation, we can interpret the inner product  $\mu_i = \langle v_i, I \rangle = p_i - (1 - p_i)$  as the *bias* of vertex  $i$ .

## Rounding Scheme

Roughly speaking, the algorithm applies a hyperplane rounding on the vectors  $w_i = v_i - \langle v_i, I \rangle I$  associated with the vertices  $i \in V$ . However, for each vertex  $i \in V$ , the algorithm shifts the hyperplane according to the bias of that vertex.

**Algorithm 5.6.3.** Given: A set of unit vectors  $\{v_1, \dots, v_n\}$  where  $v_i = \mu_i I + w_i$ , where  $w_i$  is the component of  $v_i$  orthogonal to  $I$ .  
Pick a random Gaussian vector  $g$  orthogonal to  $I$  with coordinates distributed as  $\mathcal{N}(0, 1)$ . For every  $i$ ,

1. Project  $g$  on the direction of  $w_i$ , i.e.,  $\xi_i = \langle g, \bar{w}_i \rangle$ , where  $\bar{w}_i = \frac{w_i}{\sqrt{1-\mu_i^2}}$  is the normalized vector of  $w_i$ . Note that  $\xi_i$  is also a standard Gaussian variable.
2. Pick threshold  $t_i$  as follows:

$$t_i = \Phi^{-1}(\mu_i/2 + 1/2)$$

3. If  $\xi_i \leq t_i$ , set  $x_i = 1$ , otherwise set  $x_i = -1$ .

Notice that, the threshold  $t_i$  is chosen so that individually the bias of  $x_i$  is exactly  $\mu_i$ . Therefore, the expected balance of the rounded solution matches the intended value. The analysis of the rounding algorithm consists of two parts: first we show that the cut returned by the rounding algorithm has high expected value, then we show that the balance of the cut is concentrated around its expectation.

## Analysis of the Cut Value

Analyzing the cut value of the rounding scheme is fairly standard albeit a bit technical. The analysis is *local* as in the case of other algorithms for CSPs, and reduces to bounding the probability that a given edge is cut. The probability that a given edge  $u, v$  is cut corresponds to a probability of an event related to two correlated Gaussians.

By using numerical techniques, we were able to show that the cut value is at least 0.85 times the SDP optimum. Analytically, we show the following asymptotic relation.

---

<sup>2</sup>The mapping is given by  $0 \rightarrow 1$  and  $1 \rightarrow -1$



**Lemma 5.6.4.** *Let  $u = \mu_1 I + w_1, v = \mu_2 I + w_2$  be two unit vectors satisfying  $\|u - v\|^2/4 \leq \varepsilon$ , then the probability of them being separated by Algorithm 5.6.3 is at most  $O(\sqrt{\varepsilon})$ .*

The proof of this lemma is fairly technical and is deferred to section 5.7.

## Analysis of the Balance

In this section we show that the balance of the rounded solution will be highly concentrated. We prove this fact by bounding the variance of the balance. Specifically, we show that if the SDP solution is  $\alpha$ -independent, then the variance of the balance can be bounded above by a function of  $\alpha$ .

The proof in this section is information theoretical – although this approach gives sub-optimal bound, but the proof itself is very simple and clean.

**Lemma 5.6.5.** *Let  $v_i = \mu_i I + w_i$  and  $v_j = \mu_j I + w_j$  be two vectors in the SDP solution that satisfy  $|\langle w_i, w_j \rangle| \leq \zeta$ . Let  $y_i$  and  $y_j$  be the rounded solution of  $v_i$  and  $v_j$ , then  $I(y_i, y_j) \leq O(\zeta^{1/3})$*

*Proof.* Since

$$|\langle w_i, w_j \rangle| = \sqrt{1 - \mu_i^2} \sqrt{1 - \mu_j^2} |\langle \bar{w}_i, \bar{w}_j \rangle| \leq \zeta$$

It implies that one of the three quantities in the equation above is at most  $\zeta^{1/3}$ . If it is the case that  $\sqrt{1 - \mu_i^2} \leq \zeta^{1/3}$  or  $\sqrt{1 - \mu_j^2} \leq \zeta^{1/3}$  (w.l.o.g we can assume it's the first case), then we have

$$\min(|1 - \mu_i|, |1 + \mu_i|) \leq O(\zeta^{2/3})$$

We may assume  $\mu_i > 0$ , therefore  $1 - \mu_i < O(\zeta^{2/3})$ . Notice that our rounding scheme preserves the bias individually, which implies  $y_i$  is a highly biased binary variable, hence

$$I(y_i, y_j) \leq H(y_i) = O(-(1 - \mu_i) \log(1 - \mu_i)) \leq O(\zeta^{1/3})$$

Now let's assume it's the case that  $|\langle \bar{w}_i, \bar{w}_j \rangle| \leq \zeta^{1/3}$ . Let  $g_1 = g \cdot \bar{w}_1$  and  $g_2 = g \cdot \bar{w}_2$  as described in the rounding scheme, and  $\rho = \langle \bar{w}_i, \bar{w}_j \rangle$ . Hence  $g_1$  and  $g_2$  are two jointly distributed standard Gaussian variables with covariance matrix  $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ .

The mutual information of  $g_1$  and  $g_2$  is

$$I(g_1, g_2) = -\frac{1}{2} \log(\det \Sigma) \leq O(-\log(1 - \zeta^{2/3})) \leq O(\zeta^{1/3})$$

Notice that  $y_i$  is fully dependent on  $g_i$ , therefore by the data processing inequality (5.4.10), we have  $I(y_1, y_2) \leq I(g_1, g_2) \leq O(\zeta^{1/3})$   $\square$

**Theorem 5.6.6.** *Given an  $\alpha$ -independent solution to 2-rounds Sum-of-Squares SDP hierarchy. Let  $\{y_i\}$  be the rounded solution after applying Algorithm 5.6.3. Define  $S = \mathbb{E}_{i \sim W} y_i$ , then*

$$\text{Var}(S) \leq O(\alpha^{1/12})$$

*Proof.*

$$\begin{aligned} \text{Var}(S) &= \mathbb{E}_{i,j \sim W} [\text{Cov}(y_i, y_j)] \\ &\leq \mathbb{E}_{i,j \sim W} [O(\sqrt{I(y_i; y_j)})] \quad (\text{by Fact 5.5.3}) \\ &\leq \mathbb{E}_{i,j \sim W} [O(\sqrt{|w_i, w_j|^{1/3}})] \quad (\text{by Lemma 5.6.5}) \\ &\leq \mathbb{E}_{i,j \sim W} [O(\sqrt{I(x_i; x_j)^{1/6}})] \quad (\text{by Lemma 5.6.2}) \\ &\leq O((\mathbb{E}_{i,j \sim W} [I(x_i; x_j)])^{1/12}) \quad (\text{by concavity of the function } x^{1/12}) \\ &\leq O(\alpha^{1/12}) \end{aligned}$$

□

**Corollary 5.6.7.** *Given an  $\alpha$ -independent solution to 2-rounds Sum-of-Squares SDP hierarchy  $v_i = \mu_i + w_i$ . The rounding algorithm will find an  $O(\alpha^{1/24})$ -balanced (that is, the balance of the cut differs from the expected value by at most  $O(\alpha^{1/24})$  fraction of the total weights) with probability at least  $1 - O(\alpha^{1/24})$ .*

## Wrapping Up

Here we present the proofs of the main theorems of this chapter.

Suppose we're given a MIN BISECTION instance  $G = (V, E)$  with value at most  $\varepsilon$  and constant  $\delta > 0$ . By setting  $\alpha = \delta^{24}$  and applying Theorem 5.5.6, we will get an  $\alpha$ -independent solution with value at most  $\varepsilon + \alpha$ . By Lemma 5.6.4 and the concavity of the function  $\sqrt{x}$ , the expected size of the cut returned by Algorithm 5.6.3 is at most  $O(\sqrt{\varepsilon + \alpha}) = O(\sqrt{\varepsilon} + \sqrt{\alpha})$ . Therefore, with constant probability (say  $1/2$ ), the cut returned by the rounding algorithm has size at most  $O(\sqrt{\varepsilon} + \sqrt{\alpha})$ . Also, by Corollary 5.6.7, the cut will be  $O(\delta)$ -balanced with probability at least  $1 - O(\delta)$ . Therefore, by union bound, the algorithm will return an  $O(\delta)$ -balanced cut with value at most  $O(\sqrt{\varepsilon} + \sqrt{\alpha})$  with constant probability. Notice that this probability can be amplified to  $1 - \varepsilon$  by running the algorithm  $O(\log(1/\varepsilon))$  times. Given such a cut, we can simply move  $O(\delta)$  fraction of the vertices with least degree from the larger side to the smaller side to get an exact bisection – this process will increase the value of the cut by at most  $O(\delta)$ . Therefore, in this case, we get a bisection of value at most  $O(\sqrt{\varepsilon} + \sqrt{\alpha} + \delta) = O(\sqrt{\varepsilon} + \delta)$ .

Hence, the expected value of the bisection returned by the rounding algorithm is at most  $(1 - \varepsilon)O(\sqrt{\varepsilon} + \delta) + \varepsilon = O(\sqrt{\varepsilon} + \delta)$ .

**Proof of Theorem 5.2.1.** The proof is similar in the case of MAX BISECTION. The only difference is that we have to use the fact that the rounding scheme is balanced, i.e.,  $\mathbb{P}(F(v) \neq F(-v)) = 1$ . Hence, by Lemma 5.6.4, for any edge  $(u, v)$  with value  $1 - \varepsilon$  in the SDP solution, the algorithm separates them with probability at least  $1 - O(\sqrt{\varepsilon})$ . The rest of the proof is identical.

Using a computer-assisted proof, we can show that the approximation ratio of this algorithm for MAX BISECTION is between 0.85 and 0.86. Thus further narrowing down the gap between approximation and inapproximability of MAX BISECTION. Using the same algorithm, we obtain a 0.92-approximation for globally constrained MAX 2-SAT. It is known that under the Unique Games Conjecture, MAX 2-SAT is NP-Hard to approximate within 0.9401.

## 5.7 Analysis of Cut Value

We analyze the rounding algorithm in an indirect way – first we show that under certain conditions, Algorithm 5.6.3 returns a better cut compared to Goemans-Williamson algorithm (in expectation). Then we use an union-bound type argument to give the proof for general cases.

First, we present a bound on the tail of the standard gaussian distribution.

**Lemma 5.7.1.** For  $t \geq 0$ ,

$$\Phi^c(t) = 1 - \Phi(t) \leq \frac{\sqrt{2/\pi} e^{-t^2/2}}{t + \sqrt{t^2 + 8/\pi}}$$

*Proof.* We apply the following bound on the error function given in [50]

$$e^{x^2} \int_x^\infty e^{-y^2} dy \leq \frac{1}{x + \sqrt{x^2 + 4/\pi}}$$

by replacing  $x$  with  $\frac{\sqrt{2}t}{2}$ , we get the desired bound. □

From now on, let  $\mu_0 = \sqrt{1 - 4/\pi^2} \approx 0.7712$  and  $t_0 = \Phi^{-1}(\mu_0/2 + 1/2) \approx 1.2034$ .

**Lemma 5.7.2.** Let  $g(t) = e^{t^2/2}(1 - \mu^2(t))$ , where  $\mu(t) = 2\Phi(t) - 1$ .  $g(t)$  is decreasing when  $t \geq t_0$ .

*Proof.* By simple calculation, we get

$$g'(t) = 4 \left( te^{t^2/2}(1 - \Phi(t))\Phi(t) + \frac{1}{\sqrt{2\pi}}(1 - 2\Phi(t)) \right)$$

we want to show

$$te^{t^2/2}(1 - \Phi(t))\Phi(t) + \frac{1}{\sqrt{2\pi}}(1 - 2\Phi(t)) < 0$$

by applying Lemma 5.7.1, we only need to show

$$te^{t^2/2} \frac{\sqrt{\frac{2}{\pi}}e^{-t^2/2}}{t + \sqrt{t^2 + 8/\pi}} \Phi(t) + \frac{1}{\sqrt{2\pi}}(1 - 2\Phi(t)) < 0$$

by simplification, we get

$$2\Phi(t) - 1 > \frac{t}{\sqrt{t^2 + 8/\pi}}$$

By applying the lemma again and further simplification, we get

$$e^{t^2} - t^2 > \frac{8}{\pi}$$

This can easily be verified for  $t = t_0$ . Also LHS is increasing when  $t \geq t_0$ , therefore the lemma follows.  $\square$

**Lemma 5.7.3.** Let  $f_1(x)$  and  $f_2(x)$  be twice differentiable decreasing functions defined on  $[0, \infty)$  satisfying the following conditions

1.  $f_1(0) = f_2(0)$
2.  $\lim_{x \rightarrow \infty} f_1(x) = \lim_{x \rightarrow \infty} f_2(x)$
3.  $\lim_{x \rightarrow 0} \frac{f_1'(x)}{f_2'(x)} > 1$
4.  $\frac{f_1'(x)}{f_2'(x)} = 1$  has only one solution

then

$$f_1(x) \leq f_2(x), \quad \forall x \geq 0$$

*Proof.* For the sake of contradiction we assume there exists  $x_0$  such that  $f_1(x_0) > f_2(x_0)$ . By the mean value theorem, there exists  $x_1 < x_0$  such that  $f_1'(x_1) > f_2'(x_1)$ , which means  $\frac{f_1'(x_1)}{f_2'(x_1)} < 1$  (since both  $f_1'$  and  $f_2'$  are negative). By the fourth assumption, for any  $x > x_0 > x_1$ ,  $f_1'(x) > f_2'(x)$ , therefore  $f_1(x) - f_2(x) \geq f_1(x_0) - f_2(x_0) > 0$ , contradicting the second assumption.  $\square$

Now we show the key lemma in this section.

**Lemma 5.7.4.** *Let  $u = \mu I + w_1$  and  $v = \mu I + w_2$  be two unit vectors with the same projection on the direction of  $I$ . Also we assume that  $\langle \bar{w}_1, \bar{w}_2 \rangle = 1 - \rho \geq 0$ , where  $\bar{w}_1$  and  $\bar{w}_2$  are the normalized vectors of  $w_1$  and  $w_2$ . Then the probability that these two vectors are separated by a random hyperplane is at least the probability that these two vectors are cut by Algorithm 5.6.3.*

*Proof.* First notice that since  $u$  and  $v$  have the same bias  $\mu$ , they will be assigned the same threshold  $t = \Phi^{-1}(2\mu - 1)$  in Algorithm 5.6.3.

Henceforth, we fix  $\langle \bar{w}_1, \bar{w}_2 \rangle = 1 - \rho \geq 0$ , and express the probabilities as a function of  $\mu$  and  $t$ . We stress that  $\mu$  and  $t$  are fully dependent on each other, therefore the functions are only single variable functions. We use both  $\mu$  and  $t$  (and other notations that are about to be introduced) in the expression only for simplicity.

Let  $\varepsilon = (1 - \mu^2)\rho$ , which characterizes  $\langle u, v \rangle$  as a function of  $\mu$ , i.e.,

$$\langle u, v \rangle = \langle \mu I + \sqrt{1 - \mu^2} \bar{w}_1, (\mu I + \sqrt{1 - \mu^2} \bar{w}_2) \rangle = 1 - \varepsilon$$

Let  $H(t)$  be the probability of the two vectors being separated by a random hyperplane. It is well-known that [31]

$$H(t) = \arccos(u \cdot v) / \pi = \arccos(1 - \varepsilon) / \pi$$

For Algorithm 5.6.3, notice that  $\bar{w}_1 \cdot g$  and  $\bar{w}_2 \cdot g$  are two jointly distributed standard Gaussian variables with covariance matrix  $\Sigma = \begin{pmatrix} 1 & 1 - \rho \\ 1 - \rho & 1 \end{pmatrix}$ . Thus the probability of  $u$  and  $v$  being separated by Algorithm 5.6.3 is

$$B(t) = 2 \int_{-\infty}^t \int_t^{\infty} \frac{1}{2\pi|\Sigma|^{1/2}} e^{-(x_1 \ x_2)\Sigma^{-1}(x_1 \ x_2)^T} dx_1 dx_2$$

It's easy to see that when  $\mu = t = 0$ , these two rounding schemes are equivalent, thus  $B(0) = H(0)$ . Also  $\lim_{t \rightarrow \infty} B(t) = \lim_{t \rightarrow \infty} H(t) = 0$ . The derivatives of  $H(t)$  and  $B(t)$  are as follows:

$$H'(t) = -\frac{2\sqrt{2}\rho}{\pi^{3/2}\sqrt{2\varepsilon - \varepsilon^2}} \tilde{\Phi}(t) e^{-t^2/2}$$

and

$$B'(t) = -\sqrt{\frac{2}{\pi}} \tilde{\Phi}(at) e^{-t^2/2}$$

where  $a = \frac{\rho}{\sqrt{2\rho - \rho^2}} \leq 1$  when  $\rho \leq 1$ , and  $\tilde{\Phi}(t)$  is defined as

$$\tilde{\Phi}(t) = \Phi(t) - \Phi(-t)$$

Let  $f(t) = \frac{B'(t)}{H'(t)}$ . Notice that  $f(0) = \pi/2 \geq 1$ , thus by Lemma 5.7.3, we only have to show that  $f(t) = 1$  has only one solution. Moreover, it suffices to show that  $f'(t) < 0$  when  $f(t) \leq 1$ .

Notice that when  $f(t) \leq 1$ , we have

$$\begin{aligned} \frac{\sqrt{2\varepsilon - \varepsilon^2} \tilde{\Phi}(at)}{\sqrt{2\rho - \rho^2} a \tilde{\Phi}(t)} &\leq \frac{2}{\pi} \\ \Rightarrow \frac{2\varepsilon - \varepsilon^2}{2\rho - \rho^2} &\leq \frac{4}{\pi^2} \quad (\text{By convexity of } \tilde{\Phi}, \frac{\tilde{\Phi}(at)}{a \tilde{\Phi}(t)} \geq 1 \text{ when } a \leq 1) \\ \Rightarrow \frac{\varepsilon(2 - \varepsilon)}{\rho(2 - \rho)} &\leq \frac{4}{\pi^2} \\ \Rightarrow (1 - \mu^2) \frac{2 - \rho}{2 - \varepsilon} &\leq \frac{4}{\pi^2} \quad \left( \frac{\varepsilon}{\rho} = 1 - \mu^2 \right) \\ \Rightarrow \mu &\geq \sqrt{1 - 4/\pi^2} = \mu_0 \quad \left( \frac{2 - \rho}{2 - \varepsilon} \leq 1 \right) \\ &\Rightarrow t \geq t_0 \end{aligned}$$

By calculation, one can show that

$$f'(t) = \frac{\sqrt{2/\pi} e^{-t^2/2} \sqrt{2\varepsilon - \varepsilon^2}}{\tilde{\Phi}(t)} \left( \frac{1 - \varepsilon}{2\varepsilon - \varepsilon^2} (-2\mu\rho) \tilde{\Phi}(at) + e^{(1-a^2)t^2/2} a - \frac{\tilde{\Phi}(at)}{\tilde{\Phi}(t)} \right)$$

Now we show  $f'(t) < 0$  when  $t \geq t_0$ . In order to show this, one only needs to show that

$$\frac{1 - \varepsilon}{2\varepsilon - \varepsilon^2} (2\mu\rho) \tilde{\Phi}(at) + \frac{\tilde{\Phi}(at)}{\tilde{\Phi}(t)} > e^{(1-a^2)t^2/2} a$$

By substituting  $\varepsilon = (1 - \mu^2)\rho$  and simplification, we get

$$\frac{\tilde{\Phi}(at)}{a \tilde{\Phi}(t)} \frac{1}{1 - \mu^2} \left( \frac{1 - \varepsilon}{2 - \varepsilon} 2\mu^2 + 1 - \mu^2 \right) \geq e^{(1-a^2)t^2/2}$$

Since  $\frac{\tilde{\Phi}(at)}{a \tilde{\Phi}(t)} \geq 1$  when  $a \leq 1$  and  $e^{(1-a^2)t^2/2} \leq e^{t^2/2}$ , it suffices to show

$$\left( 2\mu^2 \frac{1 - \varepsilon}{2 - \varepsilon} + 1 - \mu^2 \right) \geq e^{t^2/2} (1 - \mu^2)$$

holds when  $t \geq t_0$ .

By Lemma 5.7.2, we know that RHS is decreasing when  $t \geq t_0$ . Now we show LHS is increasing when  $\mu \geq \mu_0$ . It can be shown that the derivative of LHS is

$$2\mu\rho(1 - \mu^2)\mu^2 - (2\mu - 4\mu^3)(2 - \varepsilon) \geq -\mu(2 - 4\mu^2)(2 - \varepsilon) \geq 0$$

when  $\mu \geq \mu_0$ .

Now we only have to verify the inequality when  $t = t_0$ , and that can be done numerically. The calculation shows that  $\text{LHS}(t_0) \approx 0.8489$  while  $\text{RHS}(t_0) \approx 0.836$ .

□

Finally, we show Lemma 5.6.4.

**Lemma 5.7.5.** (Restatement of Lemma 5.6.4) *Let  $u = \mu_1 I + w_1, v = \mu_2 I + w_2$  be two unit vectors satisfying  $\|u - v\|^2/4 \leq \varepsilon$ , then the probability of them being separated by Algorithm 5.6.3 is at most  $O(\sqrt{\varepsilon})$ .*

*Proof.* (Proof of Lemma 5.6.4)

First we prove the case when  $\mu_1 = \mu_2 = \mu$ . Notice that when  $\langle w_1, w_2 \rangle > 0$ , the lemma follows from Lemma 5.7.4 and the fact that Goemans-Williamson algorithm will separate  $u$  and  $v$  with probability  $O(\sqrt{\varepsilon})$  [31].

If  $\langle w_1, w_2 \rangle < 0$ , then  $\|u - v\|^2/4 = \|w_1 - w_2\|^2/4 \geq (\|w_1\|^2 + \|w_2\|^2)/4 = (1 - \mu^2)/2$ . Hence  $|\mu| \geq 1 - O(\sqrt{\varepsilon})$ . By union bound, the probability of the algorithm separating  $u$  and  $v$  is at most  $O(\sqrt{\varepsilon})$ .

Now we consider the case when  $\mu_1 \neq \mu_2$ , w.l.o.g. we may assume  $|\mu_1| > |\mu_2|$ . We construct an auxiliary vector  $v'$  as follow:  $v' = \mu_1 I + \sqrt{1 - \mu_1^2} \bar{w}_2$ . It's easy to see that  $\|u - v'\| \leq \|u - v\|$ . Let  $F$  denote the rounding function, we analyze the probability of  $u$  and  $v$  being separated as follows:

$$\begin{aligned} & \mathbb{P}(F(u) \neq F(v)) \\ &= \mathbb{P}(F(u) \neq F(v'), F(v') = F(v)) + \mathbb{P}(F(u) = F(v'), F(v') \neq F(v)) \\ &\leq \mathbb{P}(F(u) \neq F(v')) + \mathbb{P}(F(v') \neq F(v)) \end{aligned}$$

Since  $\|u - v'\| \leq \|u - v\|$  and  $\langle u, I \rangle = \langle v', I \rangle = \mu_1$ , by the first part of the proof  $\mathbb{P}(F(u) \neq F(v')) \leq O(\sqrt{\varepsilon})$ . Also,

$$\mathbb{P}(F(v') \neq F(v)) \leq |\mu_1 - \mu_2|/2 \leq \|u - v\|/2 \leq O(\sqrt{\varepsilon}).$$

Therefore the lemma follows. □

## 5.8 Dictatorship Tests from Globally Uncorrelated SDP Solutions

A dictatorship test DICT for the MAX BISECTION 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  $(z, z') \in \text{DICT}$  to denote an edge sampled from the graph  $\text{DICT}$  (here  $z, z' \in \{\pm 1\}^R$ ).

A cut of the  $\text{DICT}$  graph can be thought of as a boolean function  $\mathcal{F} : \{\pm 1\}^R \rightarrow \{\pm 1\}$ . The value of a cut  $\mathcal{F}$  given by

$$\text{DICT}(\mathcal{F}) = \frac{1}{2} \mathbb{E}_{(z, z') \in \text{DICT}} [1 - \mathcal{F}(z)\mathcal{F}(z')],$$

is the probability that  $z, z'$  are on different sides of the cut. It is also useful to define  $\text{DICT}(\mathcal{F})$  for non-boolean functions  $\mathcal{F} : \{\pm 1\}^R \rightarrow [-1, 1]$  that take values in the interval  $[-1, 1]$ . To this end, we will interpret a value  $\mathcal{F}(z) \in [-1, 1]$  as a random variable that takes  $\{\pm 1\}$  values. Specifically, we think of a number  $a \in [-1, 1]$  as the following random variable

$$a = \begin{cases} -1 & \text{with probability } \frac{1-a}{2} \\ 1 & \text{with probability } \frac{1+a}{2} \end{cases} \quad (5.8.1)$$

With this interpretation, the natural definition of  $\text{DICT}(\mathcal{F})$  for such a function is as follows:

$$\text{DICT}(\mathcal{F}) = \frac{1}{2} \mathbb{E}_{(z, z') \in \text{DICT}} [1 - \mathcal{F}(z)\mathcal{F}(z')].$$

Indeed, the above expression is equal to the expected value of the cut obtained by randomly rounding the values of the function  $\mathcal{F} : \{\pm 1\}^R \rightarrow [-1, 1]$  to  $\{\pm 1\}$  as described in Equation (5.8.1).

We will construct a dictatorship test for the weighted version of  $\text{MAX BISECTION}$ . In particular, each vertex  $x \in \{\pm 1\}^R$  of  $\text{DICT}$  is associated a weight  $W(x)$ , and the weights  $W$  form a probability distribution over  $\{\pm 1\}^R$  (sum up to 1). The balance condition on the cut can now be expressed as  $\mathbb{E}_{z \sim W}[\mathcal{F}(z)] = 0$ .

The dictatorship test  $\text{DICT}$  can be easily transformed in to a dictatorship test  $\text{DICT}'$  for unweighted  $\text{MAX BISECTION}$ . The idea is to replace each vertex  $x \in \{\pm 1\}^R$  with a cluster  $V_x$  of  $\lfloor W(x) \cdot M \rfloor$  vertices for some large integer  $M$ . For every edge  $(x, y)$  in  $\text{DICT}$ , connect every pair of vertices in the corresponding clusters  $V_x, V_y$  with edge of the same weight. Given any bisection  $\mathcal{F}' : \text{DICT}' \rightarrow \{\pm 1\}$  of the graph  $\text{DICT}'$  with value  $c$ , define  $\mathcal{F}(z) = \mathbb{E}_{v \in V_z} \mathcal{F}'(v)$ . By slightly correcting the balance of  $\mathcal{F}$ , it is easy to obtain a bisection  $\mathcal{F} : \{\pm 1\}^R \rightarrow [-1, 1]$  satisfying

$$\text{DICT}(\mathcal{F}) \geq c - o_M(1) \quad \mathbb{E}_z \mathcal{F}(z) = 0.$$

Conversely, given a bisection  $\mathcal{F} : \{\pm 1\}^R \rightarrow [-1, 1]$  of  $\text{DICT}$ , assign  $(1 + \mathcal{F}(z))/2$  fraction of vertices of  $V_z$  to be 1 and the rest to  $-1$ . The resulting partition of  $\text{DICT}'$  is very close to balanced (up to rounding errors), and can be modified in to a bisection with value  $\text{DICT}(\mathcal{F}) - o_M(1)$ .

The dictator cuts are given by the functions  $\mathcal{F}(z) = z^{(\ell)}$  for some  $\ell \in [R]$ . The dictatorship test graph is so constructed that each dictator cut will yield a



bisection and the Completeness of the test DICT is the minimum value of a dictator cut, i.e.,

$$\text{Completeness}(\text{DICT}) = \min_{\ell \in [R]} \text{DICT}(z^{(\ell)})$$

The soundness of the dictatorship test is the value of bisections of DICT that are far from every dictator. We will formalize the notion of being far from every dictator using the notion of influences.

**Influences and Noise Operators.** To this end, we recall the definitions of influences and noise operators. Let  $\Omega = (\{\pm 1\}, \mu)$  denote the probability space with atoms  $\{\pm 1\}$  and a distribution  $\mu$  on them. Then, the influences and noise operators for functions over the product space  $\Omega^R$  are defined as follows.

**Definition 5.8.1** (Influences). The influence of the  $\ell^{\text{th}}$  coordinate on a function  $\mathcal{F} : \{\pm 1\}^R \rightarrow \mathbb{R}$  under a distribution  $\mu$  over  $\{\pm 1\}$  is given by  $\text{Inf}_\ell^\mu(\mathcal{F}) = \mathbb{E}_{x^{(-\ell)}} [\mathbb{V}_{x^{(\ell)}}[\mathcal{F}(x)]] = \sum_{S \ni \ell} \hat{\mathcal{F}}_S^2$ .

**Definition 5.8.2.** For  $0 \leq \varepsilon \leq 1$ , define the operator  $T_{1-\varepsilon}$  on  $L_2(\Omega^R)$  as,

$$T_{1-\varepsilon}\mathcal{F}(z) = \mathbb{E}[\mathcal{F}(\tilde{z}) \mid z]$$

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

**Invariance Principle.** The following invariance principle is an immediate consequence of Theorem 3.6 in the work of Isaksson and Mossel [42].

**Theorem 5.8.3.** (Invariance Principle [42]) Let  $\Omega$  be a finite probability space with the least non-zero probability of an atom at least  $\alpha \leq 1/2$ . Let  $\mathcal{L} = \{\ell_1, \ell_2\}$  be an ensemble of random variables over  $\Omega$ . Let  $\mathcal{G} = \{g_1, g_2\}$  be an ensemble of Gaussian random variables satisfying the following conditions:

$$\mathbb{E}[\ell_i] = \mathbb{E}[g_i] \quad \mathbb{E}[\ell_i^2] = \mathbb{E}[g_i^2] \quad \mathbb{E}[\ell_i \ell_j] = \mathbb{E}[g_i g_j] \quad \forall i, j \in \{1, 2\}$$

Let  $K = \log(1/\alpha)$ . Let  $F$  denote a multilinear polynomial and let  $H = (T_{1-\varepsilon}F)$ . Let the variance of  $H$ ,  $\mathbb{V}[H]$  be bounded by 1 and all the influences are smaller than  $\tau$ , i.e.,  $\text{Inf}_i(H) \leq \tau$  for all  $i$ .

If  $\Psi : \mathbb{R}^2 \rightarrow \mathbb{R}$  is a Lipschitz-continuous function with Lipschitz constant  $C_0$  (with respect to the  $L_2$  norm) then

$$\left| \mathbb{E}[\Psi(H(\mathcal{L}^R))] - \mathbb{E}[\Psi(H(\mathcal{G}^R))] \right| \leq C \cdot C_0 \cdot \tau^{\varepsilon/18K} = o_\tau(1)$$

for some constant  $C$ .

**Construction.** Let  $G = (V, E)$  be an arbitrary instance of MAX BISECTION. Let  $\mathbf{V} = \{v_{i,0}, v_{i,1}\}_{i \in V}$  denote a globally uncorrelated feasible SDP solution for two rounds of the Sum-of-Squares hierarchy. Specifically, for every pair of vertices  $i, j \in V$ , there exists a distribution  $\mu_{ij}$  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

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \mathbb{E}[z_i \cdot z_j].$$

Furthermore, the correlation between random pair of vertices is at most  $\delta$ , i.e.,

$$\mathbb{E}_{i,j \in V} [I(z_i, z_j)] \leq \delta.$$

Starting from  $G = (V, E)$  along with the SDP solution  $\mathbf{V}$  and a parameter  $\varepsilon$  we construct a dictatorship test  $\text{DICT}_V^\varepsilon$ . The dictatorship test gadget is exactly the same as the construction by Raghavendra [61] for the MAX CUT problem. For the sake of completeness, we include the details below.

**$\text{DICT}_V^\varepsilon$  (MAX BISECTION)** The set of vertices of  $\text{DICT}_V^\varepsilon$  consists of the  $R$ -dimensional hypercube  $\{\pm 1\}^R$ . The distribution of edges in  $\text{DICT}_V^\varepsilon$  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  $\mu_e$  to obtain  $\mathbf{z}_i^R = (z_i^{(1)}, \dots, z_i^{(R)})$  and  $\mathbf{z}_j^R = (z_j^{(1)}, \dots, z_j^{(R)})$ , both in  $\{\pm 1\}^R$ .
- Perturb each coordinate of  $\mathbf{z}_i^R$  and  $\mathbf{z}_j^R$  independently with probability  $\varepsilon$  to obtain  $\tilde{\mathbf{z}}_i^R, \tilde{\mathbf{z}}_j^R$  respectively. Formally, for each  $\ell \in [R]$ ,

$$\tilde{z}_i^{(\ell)} = \begin{cases} z_i^{(\ell)} & \text{with probability } 1 - \varepsilon \\ \text{random sample from distribution } \mu_i & \text{with probability } \varepsilon \end{cases}$$

- Output the edge  $(\tilde{\mathbf{z}}_i^R, \tilde{\mathbf{z}}_j^R)$ .

The weights on the vertices of  $\text{DICT}_V^\varepsilon$  is given by

$$W(x) = \mathbb{E}_{i \in V} \left[ \mathbb{P}_{\mathbf{z} \in \mu_i^R} [\mathbf{z} = x] \right].$$

We will show the following theorem about the completeness and soundness of the dictatorship test.

**Theorem 5.8.4.** *There exist absolute constants  $C, K$  such that for all  $\varepsilon, \tau \in [0, 1]$  there exists  $\delta$  such that following holds. Given a graph  $G$  and a  $\delta$ -independent*

SDP solution  $\mathbf{V} = \{\mathbf{v}_{i,0}, \mathbf{v}_{i,1} | i \in V\}$  for the two round Sum-of-Squares SDP for MAX BISECTION, the dictatorship test  $DICT_{\mathbf{V}}^{\varepsilon}$  is such that

- The dictator cuts are bisections with value within  $2\varepsilon$  of the SDP value, i.e.,  $\text{Completeness}(DICT_{\mathbf{V}}^{\varepsilon}) \geq \text{val}(\mathbf{V}) - 2\varepsilon$
- If  $\mathcal{F} : \{\pm 1\}^R \rightarrow [-1, 1]$  is a bisection of  $DICT_{\mathbf{V}}^{\varepsilon}$  ( $\mathbb{E}_{x \sim W}[\mathcal{F}(x)] = 0$ ) and all its influences are at most  $\tau$ , i.e.,

$$\text{Inf}_{\ell}^{\mu_i}(\mathcal{F}) \leq \tau \quad \forall i \in V, \ell \in [R],$$

then,

$$DICT_{\mathbf{V}}^{\varepsilon}(\mathcal{F}) \leq \text{opt}(G) + C\tau^{K\varepsilon}.$$

*Proof.* The analysis of the dictatorship test is along the lines of the corresponding proof for MAX CUT in [61].

**Completeness.** First, the dictatorship test gadget is exactly the same as that constructed for MAX CUT in [61]. Therefore from [61], the fraction of edges cut by the dictators is at least  $\text{val}(\mathbf{V}) - 2\varepsilon$ . To finish the proof of completeness, we need to show that the dictator cuts are indeed *balanced*. However, this is an easy calculation since the balance of the  $j^{\text{th}}$  dictator cut is given by,

$$\mathbb{E}_{x \in W} [x^{(j)}] = \mathbb{E}_{i \in V} \mathbb{E}_{x \in \mu_i^R} [x^{(j)}] = \mathbb{E}_{i \in V} \mathbb{E}_{a \in \mu_i} [a] = 0,$$

where the last equality uses the fact that the SDP solution satisfies the balance condition.

**Soundness.** Let  $\mathcal{F} : \{\pm 1\}^R \rightarrow [-1, 1]$  be a balanced cut all of whose influences are at most  $\tau$ . As in [61], we will use the function  $\mathcal{F}$  to round the SDP solution  $\mathbf{V}$ . The rounding algorithm is exactly the same as the one in [61]. For the sake of completeness, we reproduce the rounding scheme below.

Round $\mathcal{F}$  Scheme

**Truncation Function.** Let  $f_{[-1,1]} : \mathbb{R} \rightarrow [-1, 1]$  be a Lipschitz-continuous function such that for all  $x \in [-1, 1]$ ,  $f_{[-1,1]}(x) = x$ . Let  $C_0$  denote the Lipschitz constant of the function  $f_{[-1,1]}$ .

**Bias.** For each vertex  $i \in V$ , let the bias of vertex  $i$  be  $\theta_i = \langle \mathbf{v}_{i,0}, \mathbf{I} \rangle$  and let  $\mathbf{w}_i = \mathbf{v}_{i,0} - \langle \mathbf{v}_{i,0}, \mathbf{I} \rangle \mathbf{v}_{i,0}$  be the component of  $\mathbf{v}_{i,0}$  orthogonal to the vector  $\mathbf{I}$ .

**Scheme.** Sample  $R$  vectors  $\zeta^{(1)}, \dots, \zeta^{(R)}$  with each coordinate being i.i.d normal random variable.

For each  $i \in V$  do

- For all  $1 \leq j \leq R$ , compute the projection  $g_i^{(j)}$  of the vector  $\mathbf{w}_i$  as follows:

$$g_i^{(j)} = \theta_i + [\langle \mathbf{w}_i, \zeta^{(j)} \rangle]$$

and let  $\mathbf{g}_i = (g_i^{(1)}, \dots, g_i^{(R)})$

- Let  $F_i$  denote the multilinear polynomial corresponding to the function  $\mathcal{F}$  under the distribution  $\mu_i^R$  and let  $H_i = T_{1-\varepsilon} F_i$ . Evaluate  $H_i$  with  $g_i^{(j)}$  as inputs to obtain  $p_i$ , i.e.,  $p_i = H_i(g_i^{(1)}, \dots, g_i^{(R)})$ .
- Round  $p_i$  to  $p_i^* \in [-1, 1]$  by using the Lipschitz-continuous truncation function  $f_{[-1,1]} : \mathbb{R} \rightarrow [-1, 1]$ .

$$p_i^* = f_{[-1,1]}(p_i).$$

- Assign the vertex  $i$  to be 1 with probability  $(1 + p_i^*)/2$  and  $-1$  with the remaining probability.

Let Round $\mathcal{F}(\mathbf{V})$  denote the expected value of the cut returned by the rounding scheme Round $\mathcal{F}$  on the SDP solution  $\mathbf{V}$  for the MAX BISECTION instance  $G$ .

Again, by appealing to the soundness analysis in [61], we conclude that the fraction of edges cut by the resulting partition is lower bounded by

$$\text{Round}_{\mathcal{F}}(\mathbf{V}) \geq \text{DICT}_{\mathbf{V}}^{\varepsilon}(\mathcal{F}) - C' \tau^{K\varepsilon}.$$

for an absolute constant  $C'$ . To finish the proof, we need to argue that if the SDP solution  $\mathbf{V}$  is  $\delta$ -independent, then the resulting partition is close to balanced with high probability.

First, note that the expected balance of the cut is given by,

$$\mathbb{E}_{\zeta} \left[ \mathbb{E}_i [p_i^*] \right] = \mathbb{E}_{\zeta} \left[ \mathbb{E}_i [f_{[-1,1]}(H(g_i))] \right].$$

Fix a vertex  $i \in V$ . By construction, the random variables  $z_i^{(\ell)} \sim \mu_i$  and  $g_i^{(\ell)}$  have matching moments up to order two for each  $\ell \in [R]$ . Therefore, by applying the invariance principle of Isaksson and Mossel [42] with the smooth function  $f_{[-1,1]}$  and the multilinear polynomial  $F_i$  yields the following inequality,

$$\mathbb{E}_{\zeta} [f_{[-1,1]}(H_i(g_i))] \leq \mathbb{E}_{z_i^R \in \mu_i^R} [f_{[-1,1]}(H_i(z_i^R))] + C\tau^{K\varepsilon}.$$

Since the cut  $\mathcal{F}$  is balanced we can write,

$$\mathbb{E}_i \mathbb{E}_{z_i^R \in \mu_i^R} [f_{[-1,1]}(H_i(z_i^R))] = \mathbb{E}_i \mathbb{E}_{z_i^R \in \mu_i^R} [H_i(z_i^R)] = \mathbb{E}_i \mathbb{E}_{z_i^R \in \mu_i^R} [F_i(z_i^R)] = \mathbb{E}_i \mathbb{E}_{z_i^R \in \mu_i^R} [\mathcal{F}(z_i^R)] = 0.$$

In the previous calculation, the first equality uses the fact that  $f_{[-1,1]}(x) = x$  for  $x \in [-1, 1]$  while the second equality uses the fact that  $\mathbb{E}_z[T_{1-\varepsilon}H_i(z)] = \mathbb{E}_z[F_i(z)]$ . Therefore, we get the following bound on the expected value of the balance of the cut,  $\mathbb{E}_{\zeta} [f_{[-1,1]}(H_i(g_i))] \leq C\tau^{K\varepsilon}$ .

Finally, we will show that the balance of the cut is concentrated around its expectation. To this end, we first show the following continuity of the rounding algorithm.

**Lemma 5.8.5.** *For each  $i \in V$  and any vector  $\mathbf{w}'_i$  satisfying  $\|\mathbf{w}'_i\|_2 = \|\mathbf{w}_i\|_2$ , if  $p'_i$  denotes the output of the rounding scheme  $\text{Round}_{\mathcal{F}}$  with  $\mathbf{w}'_i$  instead of  $\mathbf{w}_i$  then,*

$$\|\mathbb{E}_{\zeta} [p'_i - p_i^*]^2\| \leq C(R)\|\mathbf{w}_i - \mathbf{w}'_i\|_2^2,$$

for some function of  $R$  ( $C(R) = 2^{2R}$  suffices).

*Proof.* Let  $\mathbf{g}'_i = (g_i^{(1)}, \dots, g_i^{(R)})$  denote the projections of the vector  $\mathbf{w}'_i$  along the directions  $\zeta^{(1)}, \zeta^{(2)}, \dots, \zeta^{(R)}$ . The output of the rounding scheme on  $\mathbf{w}'_i$  is given by  $p'_i = f_{[-1,1]}(H_i(\mathbf{g}'_i))$ . Recall that the output of the rounding scheme is given by  $p_i^* = f_{[-1,1]}(H_i(\mathbf{g}_i))$ .

The result is a consequence of the fact that the function  $f_{[-1,1]} \circ H_i$  is Lipschitz continuous. Since the variance of  $\mathcal{F}(z_i^R)$  is at most 1, the sum of squares of coefficients of  $H_i$  is at most 1. Therefore, all the  $2^R$  coefficients of  $H_i$  are bounded by 1 in absolute value.

The proof is a simple hybrid argument, where we replace  $g_i^{(\ell)}$  by  $g_i'^{(\ell)}$  one by one. The details of the proof are deferred to the full version.  $\square$

**Lemma 5.8.6.** *For every  $i, j$ ,*

$$|\mathbb{E}_{\zeta} [p_i^* p_j^*] - \mathbb{E}_{\zeta} [p_i^*] \mathbb{E}_{\zeta} [p_j^*]| \leq C(R)|\langle w_i, w_j \rangle|$$

for some function  $C(R)$  of  $R$  ( $C(R) = 1002^{2R}$  suffices).

*Proof.* Set  $\mathbf{w}'_j = \mathbf{w}_j - \langle \mathbf{w}_i, \mathbf{w}_j \rangle \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} + \langle \mathbf{w}_i, \mathbf{w}_j \rangle \bar{u}$  for a unit vector  $\bar{u}$  orthogonal to  $\mathbf{w}_i$  and  $\mathbf{w}_j$ . Note that  $\mathbf{w}'_j$  is orthogonal to  $\mathbf{w}_i$  and satisfies  $\|\mathbf{w}_j - \mathbf{w}'_j\| \leq 4|\langle \mathbf{w}_i, \mathbf{w}_j \rangle|$ . Let  $p'_j$  denote the output of the rounding with  $\mathbf{w}'_j$  instead of  $\mathbf{w}_j$ . Since  $\mathbf{w}'_j$  is orthogonal to  $\mathbf{w}_i$  all their projections are independent random variables, which implies that,

$$\mathbb{E}_{\zeta}[p'_j p_i^*] = \mathbb{E}_{\zeta}[p'_j] \mathbb{E}_{\zeta}[p_i^*].$$

. Moreover, by Lemma 5.8.5 we have,

$$\mathbb{E}_{\zeta}[(p'_j - p_j^*)^2] \leq C(R) \|\mathbf{w}_j - \mathbf{w}'_j\|_2^2 \leq C(R) \cdot 16 |\langle \mathbf{w}_i, \mathbf{w}_j \rangle|^2.$$

. Combining these inequalities and using Cauchy-Schwartz, we finish the proof as follows,

$$\begin{aligned} |\mathbb{E}_{\zeta}[p_i^* p_j^*] - \mathbb{E}_{\zeta}[p_i^*] \mathbb{E}_{\zeta}[p_j^*]| &\leq |\mathbb{E}_{\zeta}[p_i^* (p_j^* - p'_j)]| + |\mathbb{E}_{\zeta}[p_i^*] \mathbb{E}_{\zeta}[p'_j - p_j^*]| \\ &\leq 2 \left( \mathbb{E}_{\zeta}[(p'_j - p_j^*)^2] \right)^{\frac{1}{2}} \left( \mathbb{E}_{\zeta}[(p_i^*)^2] \right)^{\frac{1}{2}} \\ &\leq 8C(R) |\langle w_i, w_j \rangle| \end{aligned}$$

□

To finish the proof, now we bound the variance of the balance of the cut returned using Lemma 5.8.6. The variance of the balance of the cut returned is given by,

$$\mathbb{E}_{\zeta}(\mathbb{E}_i[p_i^*])^2 - (\mathbb{E}_{\zeta} \mathbb{E}_i[p_i^*])^2 = \mathbb{E}_{i,j} \left[ \mathbb{E}_{\zeta}[p_i^* p_j^*] - \mathbb{E}_{\zeta}[p_i^*] \mathbb{E}_{\zeta}[p_j^*] \right] \leq C(R) \mathbb{E}_{i,j} [|\langle w_i, w_j \rangle|]$$

For a  $\delta$ -independent SDP solution, the above quantity is at most  $C(R) \text{poly}(\delta)$ . This gives the desired result. □

# Chapter 6

## Optimal Symmetric SDP Relaxation for Max-CSPs

### 6.1 Introduction

As mentioned earlier, the best known (approximation) algorithms for a vast range of combinatorial optimization problems are based on (polynomial-size) symmetric LP or SDP relaxations. In this chapter we will study the computational power of such relaxations and compares it to the power of explicit relaxation, e.g., obtained from hierarchies [51, 54, 68, 58]. The motivation for this comparison is two fold: On the one hand, we can deduce new lower bounds for general symmetric relaxations (from known lower bounds for hierarchies). On the other hand, our comparison identifies the best symmetric relaxations of a certain size. These relaxations are therefore a promising basis for new approximation results.

A groundbreaking work of Yannakakis [72] initiated the study of general LP formulations and showed exponential lower bounds on the size of symmetric LP formulations for TRAVELING SALESMAN and MAXIMUM MATCHING. This work also provided a framework for proving lower bounds on general LP formulations (based on the notion of *nonnegative rank* of matrices). Recent breakthroughs [26, 65] extended Yannakakis’s lower bounds to the non-symmetric case using techniques from communication complexity.

There has been some progress to extend these lower bounds on LP formulations to the approximation setting [9, 10, 12], but so far only for CLIQUE<sup>1</sup> and Max CSPs. In the SDP setting, no lower bounds are known for explicit problems (neither exact nor approximate). This work gives the first lower bounds for general symmetric SDP relaxations.

---

<sup>1</sup>In the case of CLIQUE, the LP relaxations considered in the lower bounds do not subsume all LP relaxations for CLIQUE that appear in the literature.

The ultimate goal of this line of research is to identify the “right” LP and SDP relaxations (not necessarily symmetric) for classes of optimization problems. We conjecture that Sherali–Adams and Sum-of-Squares(Lasserre) relaxations of polynomial size indeed achieve the best possible approximation guarantees among all polynomial-size LP and SDP relaxations for many problems. Some of our proof techniques are tailored toward the symmetric case (especially the group-theoretic arguments). However, our basic framework also works in the non-symmetric case and could therefore form the basis of a proof for the non-symmetric case, in the same way that Yannakakis’s framework was instrumental in the lower bound results for general LP formulations.

**Symmetric SDP Relaxations for Max-CSPs.** Semidefinite programming marries linear programming and spectral methods. Prominent examples like MAX CUT and SPARSEST CUT show that semidefinite relaxations can achieve approximation guarantees that are not (known to be) achievable by linear relaxations or spectral methods on their own [30, 4].

The Unique Games Conjecture [47] predicts that a particularly simple SDP relaxation achieves best-possible approximation guarantees for every Max CSP [62]. It’s an outstanding open question whether more complicated SDP relaxation can refute this conjecture (by providing better approximations than the basic SDP relaxation). Indeed, recent works show that polynomial-size SDP relaxations based on the Sum-of-Squares method / Lasserre hierarchy provide better approximations on families of instances for which many other methods fail [8].

Analogous to Yannakakis’s characterization of general LP relaxations, there exists a characterization of general SDP relaxations (in terms of the notion of *positive-semidefinite rank* of matrices) [26, 32], but no explicit lower bounds are known. We provide an alternative characterization in terms of sums-of-squares of linear subspaces, inspired by the viewpoint developed in previous work [12]. This characterization allows us to compare the power of general symmetric SDP relaxation and the power of low-degree Sum-of-Squares relaxations [58, 51] for Max-CSPs.

## 6.2 Statement of Results

In this section we present the main results of this chapter.

**Theorem 6.2.1.** *For every Max-CSP  $\text{MAX-}\Pi$  and  $k < n/4$ , degree- $k$  Sum-of-Squares relaxations achieve the best-possible approximation guarantee among all symmetric SDP relaxations of size at most  $\binom{n}{k}$ .*



(This result also holds if  $k$  is a function of  $n$ , up to exponential-size relaxations.)

Moreover, we exhibit an augmented degree  $k$  Sum-of-Squares relaxation that achieves the best approximation guarantees among all symmetric SDP relaxations on an instance-by-instance basis. Specifically, we show the following:

**Theorem 6.2.2.** *For every Max-CSP  $\text{MAX-}\Pi$  and  $k < n/4$ , there exists an augmented degree- $k$  Sum-of-Squares relaxation of size  $n^{k+10}$  that on every instance  $\mathfrak{I}$  of the  $\text{MAX-}\Pi$ , achieves the best-possible approximation guarantees among all symmetric SDP relaxations of size at most  $\binom{n}{k}$ .*

It is interesting that the guarantee of optimality holds on every instance, and therefore would apply even when one is interested in special classes of instances such as planar instances.

Combined with known lower bounds for Sum-of-Squares relaxations [33, 66, 70], this result implies the first explicit lower bounds for general symmetric SDP relaxations of natural optimization problems. (A recent work shows that random 0/1 polytopes require exponential-size SDP relaxations [11], but these polytopes do not correspond to natural combinatorial optimization problems.) A concrete implication is that for every positive constant  $\varepsilon > 0$ , symmetric SDP relaxation require exponential size to achieve approximation ratio  $7/8 + \varepsilon$  for MAX 3-SAT.

**Symmetric LP Relaxations for Traveling Salesman Problem.** Recent years have seen a lot of progress on the approximability of constraint satisfaction problems (e.g., in the context of the Unique Games Conjecture). It is a very interesting question whether these results could lead to new insights about other notorious combinatorial optimization problems, e.g., TRAVELING SALESMAN.

Previous work showed that symmetric LP relaxations for Max-CSPs are exactly as powerful as Sherali–Adams relaxations [12]. Here, we show an analogous result for TRAVELING SALESMAN.

**Theorem 6.2.3.** *For every  $k \in \mathbb{N}$ ,  $k < n/4$ , there exists an  $O(n^{2k})$ -size LP relaxation that can be generated in time  $O(n^{2k})$  for TRAVELING SALESMAN on  $n$  sites such that the following holds: the relaxation achieves the best-possible approximation guarantees among all symmetric LP relaxations of size at most  $\binom{n}{k}$ , even on a per-instance basis.*

**Related Work.** In an independent effort, Fawzi et al. [23] show similar lower bounds for symmetric semidefinite programs. The results of [23] are incomparable to those presented in this work (See Section 6.4 for more details).

## 6.3 Preliminaries

**Constraint Satisfaction Problems.** Constraint Satisfaction Problems (CSPs) are a broad class of discrete optimization problems that include MAX CUT and MAX 3-SAT. The main focus of this work is CSPs over a boolean domain; the same ideas can be generalized to CSPs over general finite domains.

Fix some  $k \in \mathbb{N}$ . A  $k$ -ary predicate is a mapping  $P : \{-1, 1\}^k \rightarrow \{0, 1\}$ . For a given  $n \in \mathbb{N}$  and a subset  $S \subseteq [n]$  with  $|S| = k$ , we use the notation  $P^S : \{-1, 1\}^n \rightarrow \{0, 1\}$  to denote the mapping

$$P^S(x_1, x_2, \dots, x_n) = P(x_S),$$

where  $x_S \in \{-1, 1\}^k$  denotes the projection of  $x \in \{-1, 1\}^n$  to the coordinates in  $S$ .

Let  $\Pi$  be a collection of  $k$ -ary predicates. We will often refer to such a collection as a  $k$ -ary CSP. An instance of  $\mathfrak{I}$  of MAX- $\Pi$  consists of  $n$  boolean variables  $x_1, x_2, \dots, x_n$ ,  $m$  predicates  $P_1, P_2, \dots, P_m \in \Pi$ , and  $m$  subsets  $S_1, S_2, \dots, S_m \subseteq [n]$ . The constraints of the CSP are naturally of the form  $P_i^{S_i}(x) = 1$ . The associated optimization problem is to find an assignment  $x \in \{-1, 1\}^n$  that satisfies as many constraints as possible, i.e. that maximizes

$$\text{val}_{\mathfrak{I}}(x) = \frac{1}{m} \sum_{i=1}^m P_i^{S_i}(x).$$

Given a CSP instance  $\mathfrak{I}$ , we denote its optimal value by  $\text{opt}_{\mathfrak{I}} = \max_{x \in \{-1, 1\}^n} \text{val}_{\mathfrak{I}}(x)$ . Finally, we will use MAX- $\Pi_n$  to denote the set of MAX- $\Pi$  instances on  $n$  variables.

**Positive Semidefinite Matrices.** We will use the notation  $\mathcal{S}_k^+$  to denote the cone of  $k \times k$  symmetric, positive semidefinite (PSD) matrices with real entries. We equip  $\mathcal{S}_k^+$  with the Frobenius inner product  $\langle U, V \rangle = \text{Tr}(U^T V) = \sum_{i=1}^k \sum_{j=1}^k U_{ij} V_{ij}$ .

One may naturally identify  $\mathcal{S}_k^+$  with a subset of  $\mathbb{R}^{k(k+1)/2}$  so that the inner product of two PSD matrices is equal to the inner product of the corresponding vectors. We will use these two representations interchangeably when the context is clear.

**SDP Relaxations for CSPs.** In Chapter 4 we defined generic LP relaxation for Max-CSPs. Here we will similarly define generic SDP relaxation for Max-CSPs.

Let  $\Pi$  be a  $k$ -ary CSP and let  $n \in \mathbb{N}$ . An SDP relaxation for MAX- $\Pi_n$  consists of two objects: A linearization and a spectrahedron. Fix a number  $R \in \mathbb{N}$  called the *size of the relaxation*.

**Linearization:** A linearization associates to each assignment  $x \in \{-1, 1\}^n$  an element  $\tilde{x} \in \mathcal{S}_R^+$  and to each instance  $\mathfrak{I}$  a vector  $\tilde{\mathfrak{I}} \in \mathbb{R}^{R(R+1)/2}$  satisfying the property that  $\text{val}_{\mathfrak{I}}(x) = \langle \tilde{\mathfrak{I}}, \tilde{x} \rangle$ .

**Spectrahedron:** A spectrahedron  $\mathcal{S}$  is the intersection of the PSD cone with an affine linear subspace, i.e.

$$\mathcal{S} = \{y \in \mathbb{R}^{R(R+1)/2} \mid Ay = b, y \in \mathcal{S}_R^+\},$$

where  $A$  is an  $\frac{R(R+1)}{2} \times \frac{R(R+1)}{2}$  matrix and  $b \in \mathbb{R}^{R(R+1)/2}$ . To be a valid relaxation,  $\mathcal{S}$  must contain all the integral points in its linearization, i.e.  $\{\tilde{x} : x \in \{-1, 1\}^n\} \subseteq \mathcal{S}$ .

Thus the SDP associated with a MAX- $\Pi_n$  instance  $\mathfrak{I}$  is given by

$$\begin{aligned} & \text{maximize} \quad \langle \tilde{\mathfrak{I}}, y \rangle \\ & \text{subject to} \quad Ay = b \\ & \quad \quad y \in \mathcal{S}_R^+. \end{aligned}$$

It is worth noting that the spectrahedron is independent of the instance  $\mathfrak{I}$  (note also that one has a possibly different spectrahedron for every input size  $n$ ). The instance itself is entirely encoded by the objective function. This is the same as the linear program case defined in Chapter 4/

We refer to  $R$  as the size of the SDP relaxation even though it has  $\frac{R(R+1)}{2}$  variables and equality constraints. Finally, we say that an SDP relaxation is a  $(c, s)$ -approximation for MAX- $\Pi_n$  if, for every instance  $\mathfrak{I}$ , the following implication holds true:

$$\text{opt}(\mathfrak{I}) \leq s \implies \max_{y \in \mathcal{S}} \langle \tilde{\mathfrak{I}}, y \rangle \leq c.$$

**Sum-of-Squares Hierarchy.** We will briefly recall the Sum-of-Squares SDP for Max-CSPs, in addition to a brief review of what we introduced in Chapter 4, we will also give an alternative point of view in terms of *pseudo expectation functionals*.

A solution to the the  $d$ -round SoS hierarchy consists of vectors  $v_{S,\alpha}$  for all sets of variables  $S \subseteq [n]$  with  $|S| \leq d$  and assignments  $\alpha \in \{-1, 1\}^S$ . The constraints are described as follows: for every subset  $S$  such that  $|S| \leq d$ , there should exist a probability distribution  $\mu_S$  on  $\{-1, 1\}^S$ . Furthermore, these distributions should be consistent in the sense that for any two subsets  $S$  and  $T$  with  $|S|, |T| \leq d$ , the marginal distributions of  $\mu_S$  and  $\mu_T$  on  $S \cap T$  should be identical. One then requires that for any subsets  $S, T \subseteq [n]$  with  $|S \cup T| \leq d$  and any assignments  $\alpha \in \{-1, 1\}^S, \beta \in \{-1, 1\}^T$ , we have

$$\langle v_{S,\alpha}, v_{T,\beta} \rangle = \mathbb{P}_{\mu_{S \cup T}} \{X_S = \alpha, X_T = \beta\},$$

where  $X$  denotes a random variable distributed according to  $\mu_{S \cup T}$  and  $X_S$  and  $X_T$  denote the projections of  $X$  to the coordinates in  $S$  and  $T$ , respectively.

Alternatively, one can think of the SoS SDP as optimizing an objective function over “local expectation” functionals. Consider a map  $\tilde{\mathbb{E}}$  that sends  $n$ -variate polynomials of degree at most  $d$  (over  $\mathbb{R}$ ) to real numbers. We say that  $\tilde{\mathbb{E}}$  is a *level- $d$  pseudo expectation functional* if it satisfies the following properties:

- **Linearity.** For every pair of  $n$ -variate real polynomials  $P$  and  $Q$  with  $\deg(P), \deg(Q) \leq d$ , and every pair of numbers  $a, b \in \mathbb{R}$ , we have

$$\tilde{\mathbb{E}}(aP + bQ) = a\tilde{\mathbb{E}}(P) + b\tilde{\mathbb{E}}(Q).$$

- **Positivity.** For every polynomial  $P$  with  $\deg(P) \leq d/2$ , we have  $\tilde{\mathbb{E}}(P^2) \geq 0$
- **Normalization.**  $\tilde{\mathbb{E}}(1) = 1$ .

For CSPs over the Boolean cube  $\{-1, 1\}^n$ , we may assume the following additional constraints on the functionals; this is because  $x_i^2 = 1$  for  $x_i \in \{-1, 1\}$ .

- **Folding.** For every monomial  $x_\alpha = \prod_{i=1}^n (x_i)^{\alpha_i}$  of degree at most  $d$ , we have  $\tilde{\mathbb{E}}[x_\alpha] = \tilde{\mathbb{E}}[x_{\alpha \bmod 2}]$ , where  $x_{\alpha \bmod 2} = \prod_i (x_i)^{\alpha_i \bmod 2}$ .

Consider now a  $k$ -ary CSP instance  $\mathfrak{S}$ . We may naturally consider the functional  $\text{val}_{\mathfrak{S}} : \{-1, 1\}^n \rightarrow [0, 1]$  as a multilinear polynomial of degree at most  $k$  by expressing it in the Fourier basis:  $\text{val}_{\mathfrak{S}} = \sum_{S \subseteq [n]: |S| \leq k} a_S \chi_S$ , where  $\chi_S(x) = \prod_{i=1}^n x_i$ . By abuse of notation, we can consider  $\text{val}_{\mathfrak{S}}$  also as a multilinear polynomial over  $\mathbb{R}^n$ .

We can now express the *degree- $d$  SoS value of the instance  $\mathfrak{S}$*  by

$$\text{SoS}_d(\mathfrak{S}) = \max \left\{ \tilde{\mathbb{E}}[\text{val}_{\mathfrak{S}}] : \tilde{\mathbb{E}} \text{ is a level-}d \text{ pseudo expectation functional} \right\}.$$

## 6.4 Symmetric SDPs

### Symmetry

Let  $\mathfrak{S}_n$  denote the set of permutations on  $n$  objects. Clearly  $\mathfrak{S}_n$  acts on  $\mathbb{R}^n$  by permutation of the coordinates. We call a subset  $S \subseteq \mathbb{R}^n$  *symmetric* if it is invariant under the action of  $\mathfrak{S}_n$ . In [72], an extended formulation of an  $n$ -dimensional convex polytope  $\mathcal{P} \subseteq \mathbb{R}^n$  is a convex polytope  $Q \subseteq \mathbb{R}^{n+n'}$  such that  $\mathcal{P}$  is the projection of  $Q$  to the first  $n$  coordinates. Suppose  $\mathcal{P}$  is symmetric. One says that the *extended formulation is symmetric* if, for every  $\sigma \in \mathfrak{S}_n$ , there is a  $\sigma' \in \mathfrak{S}_{n'}$  such that the permutation  $(\sigma, \sigma') \in \mathfrak{S}_{n+n'}$  preserves  $Q$ , i.e.  $Q = (\sigma, \sigma')Q$ .

A direct analog of this definition is unsuitable for SDPs. Consider again the natural identification of  $\mathcal{S}_R^+$  with a subset of  $\mathbb{R}^{R(R+1)/2}$ . If  $\sigma \in \mathfrak{S}_{R(R+1)/2}$  and  $Y \in \mathbb{R}^{R(R+1)/2}$  is PSD, it is not necessarily the case that  $\sigma Y$  is PSD. It is more

natural to define the action of  $\mathfrak{S}_R$  on  $\mathbb{R}^{R(R+1)/2}$  as that which permutes the rows and columns simultaneously. Thus if  $\sigma \in \mathfrak{S}_R$  and  $Y = (Y_{ij}) \in \mathbb{R}^{R(R+1)/2}$ , we define  $\sigma \cdot Y = (Y_{\sigma(i)\sigma(j)})_{ij} \in \mathbb{R}^{R(R+1)/2}$ . It is manifestly clear that  $\mathcal{S}_R^+ \subseteq \mathbb{R}^{R(R+1)/2}$  is invariant under this action. If one thinks about an SDP as a vector program, this corresponds naturally to permuting the underlying variables. It leads to the following notion of symmetry.

**Definition 6.4.1.** An SDP relaxation of size  $R$  for  $\text{MAX-}\Pi_n$  is *symmetric* if, for any  $\sigma \in \mathfrak{S}_n$ , there is a  $\tilde{\sigma} \in \mathfrak{S}_R$ , such that for every  $x \in \{-1, 1\}^n$ ,

$$\widetilde{\sigma(x)} = \tilde{\sigma} \cdot \tilde{x},$$

where  $\tilde{x}$  is the linearization of  $x$  and  $\widetilde{\sigma(x)}$  is the linearization of  $\sigma(x)$ .

**Remark 6.4.2.** Fawzi et al. [23] use a more general notion of symmetry wherein for any  $\sigma \in \mathfrak{S}_n$  there exists an invertible matrix  $\rho(\sigma)$  such that  $\widetilde{\sigma(x)} = \rho(\sigma)\tilde{x}\rho(\sigma)^T$ . In our setup, the matrices  $\rho(\sigma)$  is restricted to being permutation matrices.

## Function families

We now present a necessary condition for there to exist a good SDP relaxation for  $\text{MAX-}\Pi_n$  in terms of families of functions on the discrete cube. This is analogous to the characterization for LPs given in [12], and follows closely the semidefinite generalization of Yannakakis' factorization theorem presented in [26]. In what follows,  $\|\cdot\|$  denotes the Euclidean norm.

**Theorem 6.4.3.** Consider some boolean CSP  $\Pi_n$ . Suppose that for some  $c \geq s \geq 0$ , there exists an SDP relaxation of size  $R$  that  $(c, s)$ -approximates  $\text{MAX-}\Pi_n$ . Then there exists a family of functions  $f_1, f_2, \dots, f_R : \{-1, 1\}^n \rightarrow \mathbb{R}^R$ , such that for each instance  $\mathfrak{S}$  with  $\text{opt}(\mathfrak{S}) \leq s$ , there are numbers  $\{\lambda_{i,j} : 1 \leq i, j \leq R\} \subseteq \mathbb{R}$  and  $\eta \geq 0$  satisfying: For all  $x \in \{-1, 1\}^n$ ,

$$c - \text{val}_{\mathfrak{S}}(x) = \sum_{i=1}^R \left\| \sum_{j=1}^R \lambda_{i,j} f_j(x) \right\|^2 + \eta.$$

Furthermore, if the SDP relaxation is symmetric, then the family  $\{f_i : 1 \leq i \leq R\}$  is invariant under permutation of inputs, i.e. for all  $\sigma \in \mathfrak{S}_R$ ,

$$\{\sigma f_i : 1 \leq i \leq R\} = \{f_i : 1 \leq i \leq R\}.$$

*Proof.* Let  $\mathcal{S}$  be the spectrahedron associated with an SDP relaxation of size  $R$  that  $(c, s)$ -approximates  $\text{MAX-}\Pi_n$  and write

$$\mathcal{S} = \{y \mid Ay = b, y \in \mathcal{S}_R^+\}.$$

Suppose that  $\text{opt}(\mathfrak{S}) \leq s$ . Since the SDP relaxation  $(c, s)$ -approximates  $\text{MAX-}\Pi_n$ , we have  $\langle \tilde{\mathfrak{S}}, y \rangle \leq c$  for all  $y \in \mathcal{S}$ .

In particular, this implies that  $c - \langle \tilde{\mathfrak{S}}, y \rangle \geq 0$  is valid for all  $y \in \mathcal{S}$ . Therefore, by the strong separation theorem (and the fact that the SDP cone is self-dual), there exists a PSD matrix  $\Lambda \in \mathcal{S}_R^+$ , a vector  $\beta \in \mathbb{R}^{R(R+1)/2}$  and a number  $\eta \geq 0$  such that for all  $y \in \mathcal{S}$ ,

$$c - \langle \tilde{\mathfrak{S}}, y \rangle = \langle \Lambda, y \rangle + \langle \beta, Ay - b \rangle + \eta.$$

Specializing to  $y = \tilde{x}$  for  $x \in \{-1, 1\}^n$ , we have

$$c - \text{val}_{\mathfrak{S}}(x) = c - \langle \tilde{\mathfrak{S}}, \tilde{x} \rangle = \langle \Lambda, \tilde{x} \rangle + \langle \beta, A\tilde{x} - b \rangle + \eta$$

As  $\tilde{x} \in \mathcal{S}$  for  $x \in \{-1, 1\}^n$  (by the definition of a valid relaxation), we will have  $A\tilde{x} - b = 0$ , which implies that

$$c - \text{val}_{\mathfrak{S}}(x) = \langle \Lambda, \tilde{x} \rangle + \eta \quad \forall x \in \{\pm 1\}^n.$$

Write  $\Lambda = \sum_{i=1}^R \lambda_i \lambda_i^T$  for a set of vectors  $\{\lambda_i\} \subseteq \mathbb{R}^R$ . For each  $x \in \{-1, 1\}^n$ , let  $\tilde{x} = L_x L_x^T$  be a Cholesky decomposition of  $\tilde{x}$ , and define the functions  $\{f_i\}$  so that  $f_1(x), f_2(x), \dots, f_R(x)$  are the rows of  $L_x$ . In this case, we have

$$c - \text{val}_{\mathfrak{S}}(x) = \left\langle \sum_i \lambda_i \lambda_i^T, \tilde{x} \right\rangle + \eta = \left\langle \sum_i \lambda_i \lambda_i^T, \sum_i f_i(x) f_i(x)^T \right\rangle + \eta = \sum_{i=1}^R \left\| \sum_{j=1}^R \lambda_{i,j} f_j(x) \right\|^2 + \eta.$$

Suppose now that the SDP relaxation is symmetric. By definition, for each permutation  $\sigma \in \mathfrak{S}_n$ , there exists a permutation  $\tilde{\sigma} \in \mathfrak{S}_R$  such that

$$\widetilde{\sigma(x)} = \tilde{\sigma} \cdot \tilde{x}$$

for all  $x \in \{\pm 1\}^n$ .

Note that  $f_i(x)$  is the  $i^{\text{th}}$  row of the Cholesky decomposition of  $\tilde{x}$ . From the above condition, it is clear that the  $i^{\text{th}}$  row of the Cholesky decomposition of  $\widetilde{\sigma(x)}$  is the  $\tilde{\sigma}(i)^{\text{th}}$  row of  $\tilde{x}$ . Hence we will have

$$f_i(\sigma(x)) = f_{\tilde{\sigma}(i)}(x)$$

for all  $x \in \{\pm 1\}^n$  and therefore the function family  $\{f_i : 1 \leq i \leq R\}$  is invariant under the action of  $\mathfrak{S}_R$ , as desired.  $\square$

## Instance optimal symmetric SDPs

We now present an augmented version of the SoS hierarchy and show that the approximation it achieves on *every*  $\text{MAX-}\Pi_n$  instance is at least as good as *any* symmetric SDP of roughly the same size. Our starting point is a structural lemma on symmetric families of functions.

**Definition 6.4.4.** A function  $f : \{-1, 1\}^n \rightarrow \mathbb{R}$  is a  $k$ -near-junta if  $f(x_1, x_2, \dots, x_n)$  depends on at most  $k$  variables and the value  $\sum_{i=1}^n x_i$ . In other words, there is a subset  $S \subseteq [n]$  with  $|S| = k$  such that if  $x$  and  $x'$  have  $\sum_{i=1}^n x_i = \sum_{i=1}^n x'_i$  and differ only on coordinates outside  $S$ , then  $f(x) = f(x')$ .

The proof of the following lemma is very similar to an analogous claim in the work of Yannakakis [72].

**Lemma 6.4.5** ([12], Lemma 4.3). *Let  $\mathcal{F}$  be a finite family of functions of the form  $f : \{-1, 1\}^n \rightarrow \mathbb{R}$  such that  $|\mathcal{F}| \leq \binom{n}{k}$  for some  $k < n/4$ . If  $\mathcal{F}$  is invariant under the action of  $\mathfrak{S}_n$ , then each  $f \in \mathcal{F}$  is a  $k$ -near-junta.*

Recall that a  $d$ -rounds SoS hierarchy is corresponding to a normalized psuedo expectation functional over low degree polynomials. Specifically, the psuedo expectation functional  $\tilde{E}$  is a linear functional that maps polynomials of degree at most  $d$  to  $\mathbb{R}$  and satisfies linearity and positivity. Note that this functional can be represented by a table containing the psuedo expectations of every monomial of degree at most  $d$ , the positivity constraint is equivalent to the quadratic form  $P \mapsto \tilde{E}P^2$  being positive semidefinite.

In the modified SoS hierarchy, we require a psuedo expectation functional on a slightly larger class of polynomials than low-degree polynomials. In particular, fix a positive integer  $d$  and consider the vector space of polynomials of the form

$$P = \sum_{0 \leq i \leq 2n} P_i(x) \left( \sum_j x_j \right)^i, \quad (6.4.1)$$

where each  $P_i(x)$  is a polynomial of degree at most  $d$ . Note that the dimension of this vector space is at most  $2n$  times the dimension of the vector space of degree  $d$  polynomials.

In the modified SoS SDP we will maximize the objective function over psuedo expectation functionals on this vector space of polynomials. Similar to SoS hierarchy, we require the psuedo expectation functional  $\tilde{E}$  to satisfy the following properties:

– **Linearity**

$\tilde{E}(P + Q) = \tilde{E}P + \tilde{E}Q$  for every polynomial  $P$  and  $Q$  of the form 6.4.1. This is slightly more subtle compared to the usual SoS, since assigning an arbitrary table of values of  $\tilde{E}m(x)(\sum_i x_i)^k$  for every monomial  $m(x)$  of degree at most  $d$  and  $k \leq n$  no longer guarantees linearity, as they're not linearly independent. However we can specify a basis of the space spanned by these polynomials and let SDP output the pseudo-moments of the basis.

Compared to SoS, the size of this SDP is at most  $2n$  times bigger, as the number of polynomials in the basis is at most  $2n$  times bigger.

– **Positivity**

We want that for  $P = \sum_{0 \leq i \leq n} P_i(x)(\sum_j x_j)^i$ ,  $\deg(P_i) \leq d/2$ ,  $\tilde{\mathbb{E}}P^2 \geq 0$ . Once we specify the basis, this is equivalent to the quadratic form being semidefinite.

– **Normalization**

$$\tilde{\mathbb{E}}1 = 1$$

Finally as the CSP is over the boolean cube  $\{\pm 1\}^n$ , the following additional constraints on the functional arise from the fact that  $x_i^2 = 1$ .

- **Folding** For every monomial  $x_\alpha = \prod_i (x_i)^{\alpha_i}$  of degree  $\leq d$ , we will have  $\tilde{\mathbb{E}}[x_\alpha (\sum_i x_i)^j] = \tilde{\mathbb{E}}[x_{\alpha \bmod 2} (\sum_i x_i)^j]$  for all  $j \in \{0, \dots, 2n\}$  wherein  $x_{\alpha \bmod 2} = \prod_i (x_i)^{\alpha_i \bmod 2}$ .

Now we prove that this modified SoS is instance-wise optimal.

**Theorem 6.4.6.** *Given an instance  $\mathfrak{S}$  of Max-CSP  $\Pi_n$ , suppose  $2d$ -rounds of the modified SoS hierarchy does not achieve  $(c, s)$ -approximation, then no symmetric SDP of size  $\binom{n}{d}$  can achieve  $(c, s)$ -approximation on  $\mathfrak{S}$ .*

*Proof.* We prove the result by contradiction. Suppose there exists a symmetric SDP that achieves  $(c, s)$ -approximation on  $\mathfrak{S}$ . By Theorem 6.4.3, there exists a family of vector valued functions  $\{f_i\}$  such that

$$c - \text{val}_{\mathfrak{S}}(x) = \sum_i \left\| \sum_j \lambda_{i,j} f_j(x) \right\|^2 + \eta, \quad (6.4.2)$$

for some  $\eta > 0$  and real numbers  $\lambda_{i,j}$ . Note that by Lemma 6.4.5, each  $f_i$  is  $d$  near-junta.

Let  $f_{j,k}$  be the  $k$ -th coordinate of  $f_j$ , it is easy to see that  $f_{j,k}$  is also  $d$  near-junta. Therefore,

$$f_{j,k} = \sum_{l=0}^{n-1} \left( \sum_t x_t \right)^l P_{j,k,l}, \quad (6.4.3)$$

for some polynomials  $P_{j,k,l}$  with degree at most  $d$ . Here we are also using the fact that  $\sum_t x_t$  takes at most  $n+1$  different values.

Let  $\tilde{E}$  denote the psuedo expectation functional obtained by solving the  $2d$ -rounds modified SoS hierarchy on the instance  $\mathfrak{S}$ . Clearly,  $\tilde{E}$  can be evaluated on the LHS of (6.4.2) since  $\text{val}_{\mathfrak{S}}$  is a low-degree polynomial. By (6.4.3), the psuedo expectation can also be evaluated on the RHS of (6.4.2).



On evaluating  $\tilde{E}$  on the RHS of (6.4.2),

$$\tilde{\mathbb{E}} \sum_i \left\| \sum_j \lambda_{i,j} f_j \right\|^2 + \eta = \sum_{i,k} \tilde{\mathbb{E}} \left( \sum_j \lambda_{i,j} f_{j,k} \right)^2 + \eta = \sum_{i,k} \tilde{\mathbb{E}} \left( \sum_j \lambda_{i,j,l} P_{j,k,l} \left( \sum_t x_t \right)^l \right)^2 + \eta \geq 0$$

However, on the LHS we will have

$$\tilde{\mathbb{E}}(c - \text{val}_{\mathfrak{S}}) = c - \text{SoS}(\mathfrak{S}) < 0,$$

a contradiction.  $\square$

Note that one can also modify the Sherali-Adam Hierarchy in the same manner to obtain instance optimal LP for CSPs.

We remark that this modified SoS SDP is not stronger than the usual SoS SDP in terms of general approximation guarantee (that is, the worst case approximation ratio over all possible instances), as we will show in the next section. However, it is possible that this SDP performs better than SoS on some specific instances.

## Sum-of-Squares SDPs

In this section we prove that the Sum-of-Squares SDP achieves the best possible approximation amongst symmetric SDPs of similar size (not per instance-wise). Specifically we show that the approximation guarantee of SoS SDP on instances with  $n$  variables is at least as good as the approximation guarantee of *any* symmetric SDP of similar size on  $2n$  variables.

**Lemma 6.4.7.** *Suppose that the conditions of Theorem 6.4.3 hold for  $N = 2n$ , then there exists a family of  $k$ -juntas  $\{g_i\}$  on  $n$  variables of size at most  $2^k n^k$ , such that for every instance  $\mathfrak{S}$  on  $n$  variables, there exists  $\lambda_{i,j}$ , such that,*

$$c' - \text{val}_{\mathfrak{S}} = \sum_i \left\| \sum_j \lambda_{i,j} g_j \right\|^2$$

*Proof.* Given a Max-CSP instance  $\mathfrak{S}$ , we construct another instance  $\mathfrak{S}'$  of size  $2n$  by adding  $n$  extra dummy variables, while keeping the constraints the same on first  $n$  variables. There are no constraints amongst the dummy variables. Since the conditions of Theorem 6.4.3 hold, we have

$$c' - \text{val}_{\mathfrak{S}'}(y) = \sum_i \left\| \sum_j \lambda_{i,j} f_j(y) \right\|^2$$

for every  $y \in \{-1, 1\}^{2n}$ .

In particular, we have

$$c' - \text{val}_{\mathfrak{Z}'}(x, -x) = \sum_i \left\| \left( \sum_j \lambda_{i,j} f_j(x, -x) \right) \right\|^2$$

Define  $g_j(x) = f_j(x, -x)$ , since  $f_j$  is  $k$  near-junta,  $g_j$  is  $k$ -junta.

It's easy to see that  $\text{val}_{\mathfrak{Z}'}(x, -x) = \text{val}_{\mathfrak{Z}}(x)$ , hence we have

$$c' - \text{val}_{\mathfrak{Z}}(x) = c' - \text{val}_{\mathfrak{Z}'}(x, -x) = \sum_i \left\| \left( \sum_j \lambda_{i,j} f_j(x, -x) \right) \right\|^2 = \sum_i \left\| \left( \sum_j \lambda_{i,j} g_j \right) \right\|^2$$

□

Now we prove the main theorem of this section.

**Theorem 6.4.8.** *Given Max-CSP  $\Pi$ , suppose that  $2k$ -rounds SoS relaxation cannot achieve  $(c, s)$ -approximation on instances with  $n$  variables, then no symmetric SDP of size  $\binom{N}{k}$  achieves  $(c, s)$ -approximation on instances with  $N$  variables, with  $N = 2n$ .*

*Proof.* We prove it by contradiction. Suppose there exists an SDP relaxation that achieves  $(c, s)$ -approximation on instances with  $N$  variables, by Lemma 6.4.7, there exists a family of  $k$ -juntas  $g_i$  such that for every  $\mathfrak{Z}$  on  $n$  variables,

$$c' - \text{val}_{\mathfrak{Z}}(x) = \sum_i \left\| \left( \sum_j \lambda_{i,j} g_j(x) \right) \right\|^2$$

In particular, the equation holds for the instance  $\mathfrak{Z}_0$  where SoS fails to achieve  $(c, s)$ -approximation.

Let  $\tilde{\mathbb{E}}$  be the psuedo expectation functional defined by the SoS solution on  $\mathfrak{Z}_0$ , by linearity of  $\tilde{\mathbb{E}}$ , we have

$$\tilde{\mathbb{E}}P = \tilde{\mathbb{E}}c' - \tilde{\mathbb{E}}\text{val}_{\mathfrak{Z}} = c' - \text{SoS}(\mathfrak{Z}) \leq c - \text{SoS}(\mathfrak{Z}) < 0$$

However on the other hand, by positivity of  $\tilde{\mathbb{E}}$

$$\tilde{\mathbb{E}}P = \sum_i \tilde{\mathbb{E}} \left\| \left( \sum_j \lambda_{i,j} g_j \right) \right\|^2 \geq 0$$

Contradition.

□

## 6.5 Instance Optimal Symmetric LP for Traveling Salesman Problem

In this section we show that for every constant  $k$ , there exists a symmetric LP of size  $O(n^k)$ , such that on every instance of the TRAVELLING SALESMAN PROBLEM problem, its integrality gap is no worse than that of every symmetric LP of size  $\binom{n}{k}$ .

We say that an LP relaxation is a  $(c, s)$ -approximation for TRAVELLING SALESMAN PROBLEM if, for every instance  $\mathfrak{I}$ , the following implication holds true:

$$\text{opt}(\mathfrak{I}) \geq s \implies \min_{y \in \mathfrak{S}} \langle \mathfrak{I}, y \rangle \geq c.$$

To prove this result, we will need the following tailored version of Theorem 1 in [9] and Theorem 2.2 of [12]

**Theorem 6.5.1.** *Let  $\mathfrak{S}_n$  denote the permutation group on  $n$  elements. For an instance  $\mathfrak{I}$  of the TSP problem with  $n$  vertices, there exists a symmetric LP of size  $n^k$  that  $(c, s)$ -approximates  $\mathfrak{I}$  if and only if there exists a family of  $n^k$  functions  $\{f_i : \mathfrak{S}_n \mapsto \mathbb{R}^{\geq 0}\}$  with the following properties:*

- *There exists non-negative constants  $\lambda_i$  such that for every  $\sigma \in \mathfrak{S}_n$ ,  $\text{val}(\sigma) - c = \sum_i \lambda_i f_i(\sigma)$ .*
- *The family  $\{f_i\}$  is invariant under permutations of vertices.*

We remark that in case of minimization problem, we have  $c \leq s$ .

Also we show an analogue to Lemma 6.4.5. Roughly speaking, if a family of functions on  $\mathfrak{S}_n$  is invariant under permutations, then each function only depends on few locations of the tour, and possibly the parity of the tour.

To show that, we need a lemma from [72]

**Lemma 6.5.2.** ([72], Claim 2) *Let  $H$  be a group of permutations whose index in  $\mathfrak{S}_n$  is at most  $\binom{n}{k}$  for some  $k < n/4$ . Then there exists a set  $J$  of size at most  $k$  such that  $H$  contains all even permutations that fix the elements of  $J$ .*

**Lemma 6.5.3.** *Suppose a family of  $\binom{n}{k}$  functions  $\{f_i : \mathfrak{S}_n \mapsto \mathbb{R}^{\geq 0}\}$  is invariant under permutation of its inputs, then for each  $f_i$ , there exists a set of indices  $J_i$  such that  $f_i$  only depends on the positions of  $J_i$  and the parity of the input permutation.*

*Proof.* Let  $\text{Orb}(f)$  denote the orbit of a function  $f$  under permutation of its inputs. Therefore we have  $|\text{Orb}(f_i)| \leq \binom{n}{k}$ . Hence for each  $f_i$  the automorphism group that preserves  $f_i$  is large. By Lemma 6.5.2, the automorphism group  $\text{Aut}(f_i)$

contains all even permutations that fix a subset of coordinates  $J_i$  with  $|J_i| \leq k$ . Therefore the function only depends on the positions of the indices in  $J$  and the parity of the permutation.

□

It is easy to see from Theorem 6.5.1 that the task of finding symmetric LP of size  $m$  that approximates a TSP instance  $\mathfrak{I}$  optimally is equivalent to finding a symmetric family of  $m$  functions  $f_i : \mathfrak{S}_n \mapsto \mathbb{R}_{\geq 0}$  such that the optimum of the program

$$\begin{aligned} & \text{maximize } c \\ & \text{subject to } \text{val}_{\mathfrak{I}} - c \in \text{cone}(f_1, \dots, f_m) \end{aligned}$$

is maximized.

Given a family of functions  $\{f_i\}$ , let  $c_{\mathfrak{I}}(\{f_i\})$  denote the optimum to the program above.

We show that we can explicitly construct a symmetric family  $\{g_i\}$  of size  $O(n^{2k})$ , such that  $c(\{g_i\}) \geq c(\{f_i\})$  for any symmetric family  $\{f_i\}$  of size at most  $\binom{n}{k}$ .

**Definition 6.5.4.** Let  $S$  and  $T$  be ordered tuples of at most  $k$  vertices such that  $|S| = |T|$ . Let  $I_{S,T,\text{odd}} : \mathfrak{S}_n \mapsto \{0, 1\}$  be the indicator function such that  $I_{S,T,\text{odd}}(\sigma) = 1$  if and only if  $\sigma(S) = T$  and  $\sigma$  is an odd permutation. Similarly let  $I_{S,T,\text{even}}$  be the indicator function such that  $I_{S,T,\text{even}}(\sigma) = 1$  if and only if  $\sigma(S) = T$  and  $\sigma$  is an even permutation.

Let  $\{g_i\}$  be the family of functions that consists of  $I_{S,T,\text{odd}}$  and  $I_{S,T,\text{even}}$  for every  $|S| = |T| \leq k$ . It's easy to see that  $\{g_i\}$  has size  $O(n^{2k})$ .

**Lemma 6.5.5.** Let  $\{f_i\}$  be a symmetric family of functions with size at most  $\binom{n}{k}$ , then for any TSP instance  $\mathfrak{I}$ ,  $c_{\mathfrak{I}}(\{g_i\}) \geq c_{\mathfrak{I}}(\{f_i\})$

*Proof.* By Lemma 6.5.3, each function in  $\{f_i\}$  only depends on the positions of at most  $k$  indices and the parity of the permutation, therefore the function can be written as non-negative combination of the indicator functions in  $\{g_i\}$ , which implies  $\text{cone}(\{f_i\}) \subseteq \text{cone}(\{g_i\})$ . □

Hence the following linear program of size  $n^{2k}$  achieves at least as good an approximation guarantee as all symmetric linear programs of size  $\binom{n}{k}$ ,

$$\begin{aligned} & \text{maximize } c \\ & \text{subject to } \text{val}_{\mathfrak{I}} - c \in \text{cone}(g_1, \dots, g_M) \end{aligned}$$

Notice that there are  $O(n^k)$  variables, but  $n!$  equations. Now, we will show how to find a succinct representation of the linear program with  $O(n^k)$  variables

and  $O(n^k)$  constraints. To this end, let us write  $\text{val}_3(\sigma)$  as a sum of the indicator functions of pairwise events,

$$\text{val}_3(\sigma) = \sum_{i=1}^n \sum_{a,b} D(a,b) 1[\sigma(i) = a \wedge \sigma(i+1 \bmod n) = b],$$

where  $D(a,b)$  is the cost of traversing the edge  $(a,b)$ . Rewriting the above linear program in terms of functions,

maximize  $c$

$$\text{subject to } \sum_{i=1}^n \sum_{a,b} D(a,b) 1[\sigma(i) = a \wedge \sigma(i+1 \bmod n) = b] - c - \sum_i \lambda_i g_i = 0$$

$$\lambda_i \geq 0$$

Note that the indicator functions in the objective function can also be written as follows,

$$1[\sigma(i) = a \wedge \sigma(i+1 \bmod n) = b] = I_{(i,i+1),(a,b),0} + I_{(i,i+1),(a,b),1}.$$

Hence in the above linear program there are  $O(n^{2k})$  functions over  $\mathfrak{S}_n$  and we wish to find non-negative  $\lambda_i$  and  $c$  that ensures that  $\sum_{i=1}^n \sum_{a,b} D(a,b) 1[\sigma(i) = a \wedge \sigma(i+1 \bmod n) = b] - c - \sum_i \lambda_i g_i = 0$  while maximizing  $c$ . We would like to rewrite this linear program in an alternate basis, so as to reduce  $n!$  equations to  $n^{2k}$  equations. To achieve this, we begin by making the following observation:

**Observation 6.5.6.** *The inner product  $\langle g_i, g_j \rangle$  between any pair of indicator functions can be computed in time  $O(n^3)$ .*

*Proof.* Consequence of the simple combinatorial structure of the indicator functions. In particular, there are explicit formulae for the inner products of the indicator functions.  $\square$

Hence we can compute the matrix  $V$  whose  $ij^{\text{th}}$  entry is  $\langle g_i, g_j \rangle$  in time  $O(n^{O(k)})$ . Given  $V$ , for a vector  $\Lambda = (\lambda_1, \dots, \lambda_M)$  we will have,

$$\sum_i \lambda_i g_i = 0 \iff V\Lambda = 0$$

Therefore, once matrix  $V$  is computed, it is straightforward to write down a linear program of size  $n^{2k}$  as follows:

maximize  $c$

$$\text{subject to } V(w - \Lambda) = 0$$

$$\Lambda \geq 0$$

wherein  $\Lambda, w$  are vectors indexed by functions  $g_i$ . Here  $w(I_{(i,i+1 \bmod n),(a,b),0}) = w(I_{(i,i+1 \bmod n),(a,b),1}) = -D(a, b)$ ,  $w(1) = c$  and all the remaining coordinates of  $w$  are zero.

This gives us the main theorem of this section.

**Theorem 6.5.7.** *For any constant  $k$ , there exists a symmetric LP that can be generated in time  $n^{O(k)}$  and is of size  $O(n^{2k})$  such that the following holds: the linear program gives an approximation to TSP problem that is at least as good as any symmetric LP of size at most  $\binom{n}{k}$  on every instance of the problem.*

# Chapter 7

## Future Directions

Following this thesis several open questions in approximation algorithms and hardness of approximation immediately arise.

**Optimal Algorithm and Hardness Result for CSPs with Global Cardinality Constraints.** Toward the end of Chapter 5, we showed a construction of dictatorship test using Sum-of-Squares gap instances. However, unlike Max-CSPs, dictatorship tests no longer translate to hardness result. This is due to the non-locality of the constraints. To see this, suppose we want to prove hardness result for MIN BISECTION problem. If we did the usual gadget reduction as in [61], we would end up with a graph of which the underlying structure is very similar to the original Unique Game instance. In particular, if the original Unique Game instance affords a bisection with no edges between them, so will the MAX BISECTION instance, irrespective of the Unique Game instance value! Thus, such reductions would not work.

Therefore it would probably be more hopeful if instead of Unique Game, we start the reduction from some instances with guaranteed expansion in the underlying graph. One problem that might be a good place to start the reduction with is the Small Set Expansion conjecture by Raghavendra and Steurer [63].

Another possible direction is to prove that CSPs with global cardinality constraints are exactly as hard as their counter part without global cardinality constraints. This would immediately imply optimal hardness results for all Max-CSPs (under UGC). In fact, Austrin et al. conjectured that this is indeed the case[6].

**Sum-of-Squares SDP for Unique Games.** As we showed in this thesis, Sum-of-Squares SDP is an extremely powerful tool in designing approximation algorithms. Therefore, the natural question to ask is whether Sum-of-Squares gives delivers a better approximation guarantee for Unique Games. This has been an important open question over the past few years. In fact, Barak et al. showed

that 8-rounds of Sum-of-Squares SDP hierarchy solves all the gap instances that we're aware of for Unique Games. Therefore, it is entirely possible that a constant rounds of Sum-of-Squares SDP hierarchy would be able to provide a better approximation algorithm for MAX CUT problem and therefore refute the Unique Games Conjecture.

**Optimality of Sum-of-Squares among General SDP relaxations.** We showed that among all the possible symmetric SDP relaxations for Max-CSPs of the similar size, Sum-of-Squares achieves the best possible approximation guarantee. Therefore a natural question is whether we can remove the symmetry constraint.



# Bibliography

- [1] Farid Alizadeh. “Interior point methods in semidefinite programming with applications to combinatorial optimization.” In: *SIAM Journal on Optimization* 5.1 (1995), pp. 13–51.
- [2] Noga Alon and Assaf Naor. “Approximating the cut-norm via Grothendieck’s inequality.” In: *SIAM Journal on Computing* 35.4 (2006), pp. 787–803.
- [3] Sanjeev Arora and Eden Chlamtac. “New approximation guarantee for chromatic number.” In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM. 2006, pp. 215–224.
- [4] Sanjeev Arora, Satish Rao, and Umesh Vazirani. “Expander flows, geometric embeddings and graph partitioning.” In: *Journal of the ACM (JACM)* 56.2 (2009), p. 5.
- [5] Per Austrin. “Balanced max 2-sat might not be the hardest.” In: *ACM Symposium on Theory of Computing (STOC)*. Ed. by David S. Johnson and Uriel Feige. ACM, 2007, pp. 189–197. ISBN: 978-1-59593-631-8.
- [6] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. “Better balance by being biased: a 0.8776-approximation for max bisection.” In: *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2013, pp. 277–294.
- [7] Boaz Barak, Prasad Raghavendra, and David Steurer. “Rounding Semidefinite Programming Hierarchies via Global Correlation.” In: *IEEE Foundations of Computer Science (FOCS) (to appear)*. 2011.
- [8] Boaz Barak et al. “Hypercontractivity, sum-of-squares proofs, and their applications.” In: *Proceedings of the 44th symposium on Theory of Computing*. ACM. 2012, pp. 307–326.
- [9] Gábor Braun et al. “Approximation limits of linear programs (beyond hierarchies).” In: *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*. IEEE. 2012, pp. 480–489.

- [10] Mark Braverman and Ankur Moitra. “An information complexity approach to extended formulations.” In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM. 2013, pp. 161–170.
- [11] Jop Briët, Daniel Dadush, and Sebastian Pokutta. “On the existence of 0/1 polytopes with high semidefinite extension complexity.” In: *Algorithms–ESA 2013*. Springer, 2013, pp. 217–228.
- [12] Siu On Chan et al. “Approximate Constraint Satisfaction Requires Large LP Relaxations.” In: *arXiv preprint arXiv:1309.0563* (2013).
- [13] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. “Near-optimal algorithms for maximum constraint satisfaction problems.” In: *ACM-SIAM Symposium on Discrete Algorithms, SODA*. SIAM, 2007, pp. 62–68. ISBN: 978-0-898716-24-5. URL: <http://doi.acm.org/10.1145/1283383.1283391>.
- [14] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. “Near-optimal algorithms for maximum constraint satisfaction problems.” In: *ACM Transactions on Algorithms (TALG)* 5.3 (2009), p. 32.
- [15] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. “Near-optimal algorithms for unique games.” In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM. 2006, pp. 205–214.
- [16] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. “On the advantage over random for maximum acyclic subgraph.” In: *null*. IEEE. 2007, pp. 625–633.
- [17] Moses Charikar and Anthony Wirth. “Maximizing quadratic programs: extending Grothendieck’s inequality.” In: *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*. IEEE. 2004, pp. 54–60.
- [18] Eden Chlamtac. “Approximation algorithms using hierarchies of semidefinite programming relaxations.” In: *Foundations of Computer Science, 2007. FOCS’07. 48th Annual IEEE Symposium on*. IEEE. 2007, pp. 691–701.
- [19] Eden Chlamtac, Konstantin Makarychev, and Yury Makarychev. “How to play unique games using embeddings.” In: *Foundations of Computer Science, 2006. FOCS’06. 47th Annual IEEE Symposium on*. IEEE. 2006, pp. 687–696.
- [20] Eden Chlamtac and Gyanit Singh. “Improved approximation guarantees through higher levels of SDP hierarchies.” In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2008, pp. 49–62.

- [21] Eden Chlamtac and Gyanit Singh. “Improved Approximation Guarantees through Higher Levels of SDP Hierarchies.” In: *APPROX-RANDOM*. Ed. by Ashish Goel et al. Vol. 5171. Lecture Notes in Computer Science. Springer, 2008, pp. 49–62. ISBN: 978-3-540-85362-6.
- [22] Benny Chor and Madhu Sudan. “A geometric approach to betweenness.” In: *SIAM Journal on Discrete Mathematics* 11.4 (1998), pp. 511–523.
- [23] Hamza Fawzi, James Saunderson, and Pablo A Parrilo. “Equivariant semidefinite lifts and sum-of-squares hierarchies.” In: *arXiv preprint arXiv:1312.6662* (2013).
- [24] Uriel Feige and Michel Goemans. “Approximating the value of two power proof systems, with applications to max 2sat and max dicut.” In: *Theory of Computing and Systems, 1995. Proceedings., Third Israel Symposium on the*. IEEE. 1995, pp. 182–189.
- [25] Uriel Feige and Michael Langberg. “The RPR<sup>2</sup> rounding technique for semidefinite programs.” In: *J. Algorithms* 60.1 (2006), pp. 1–23.
- [26] Samuel Fiorini et al. “Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds.” In: *Proceedings of the 44th symposium on Theory of Computing*. ACM. 2012, pp. 95–106.
- [27] Alan Frieze and Mark Jerrum. “Improved approximation algorithms for maxk-cut and max bisection.” In: *Algorithmica* 18.1 (1997), pp. 67–81.
- [28] Alan M. Frieze and Mark Jerrum. “Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION.” In: *Algorithmica* 18.1 (1997), pp. 67–81.
- [29] Michel X Goemans and David Williamson. “Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming.” In: *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM. 2001, pp. 443–452.
- [30] Michel X Goemans and David P Williamson. “. 879-approximation algorithms for MAX CUT and MAX 2SAT.” In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. ACM. 1994, pp. 422–431.
- [31] Michel X. Goemans and David P. Williamson. “Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming.” In: *Journal of the ACM* 42.6 (1995), pp. 1115–1145. URL: <http://portal.acm.org/citation.cfm?id=227684orhttp://www-math.mit.edu/~goemans/orhttp://www-math.mit.edu/~goemans/maxcut-jacm.pdf>.

- [32] João Gouveia, Pablo A Parrilo, and Rekha R Thomas. “Lifts of convex sets and cone factorizations.” In: *Mathematics of Operations Research* 38.2 (2013), pp. 248–264.
- [33] Dima Grigoriev. “Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity.” In: *Theoretical Computer Science* 259.1 (2001), pp. 613–622.
- [34] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*. Vol. 2. Springer Science & Business Media, 2012.
- [35] Venkatesan Guruswami et al. “Finding Almost Perfect Graph Bisections.” In: *Innovations in Computer Science*. Tsinghua University Press, 2011, pp. 321–337.
- [36] Eran Halperin and Uri Zwick. “A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems.” In: *Random Struct. Algorithms* 20.3 (2002), pp. 382–402.
- [37] Eran Halperin and Uri Zwick. “Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs.” In: *Journal of Algorithms* 40.2 (2001), pp. 184–211.
- [38] Gustav Hast. “Beating a random assignment.” In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2005, pp. 134–145.
- [39] Johan Håstad. “Some optimal inapproximability results.” In: *Journal of the ACM (JACM)* 48.4 (2001), pp. 798–859.
- [40] Johann Håstad. “Some optimal inapproximability results.” In: *Journal of the ACM* 48.4 (2001), pp. 798–859.
- [41] Jonas Holmerin and Subhash Khot. “A new PCP outer verifier with applications to homogeneous linear equations and max-bisection.” In: *ACM Symposium on Theory of Computing (STOC)*. 2004, pp. 11–20.
- [42] Marcus Isaksson and Elchanan Mossel. “Maximally Stable Gaussian Partitions with Discrete Applications.” In: *arXiv:0903.3362*. (2009).
- [43] David Karger, Rajeev Motwani, and Madhu Sudan. “Approximate graph coloring by semidefinite programming.” In: *Journal of the ACM (JACM)* 45.2 (1998), pp. 246–265.
- [44] Anna R Karlin, Claire Mathieu, and C Thach Nguyen. “Integrality gaps of linear and semi-definite programming relaxations for knapsack.” In: *Integer Programming and Combinatorial Optimization*. Springer, 2011, pp. 301–314.

- [45] Howard Karloff and Uri Zwick. “A  $7/8$ -approximation algorithm for MAX 3SAT?” In: *Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*. IEEE. 1997, pp. 406–415.
- [46] Howard Karloff and Uri Zwick. “A  $7/8$ -approximation algorithm for MAX 3SAT?” In: *IEEE Foundations of Computer Science, (FOCS)*. 1997, pp. 406–415.
- [47] Subhash Khot. “On the power of unique 2-prover 1-round games.” In: *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*. ACM. 2002, pp. 767–775.
- [48] Subhash Khot and Assaf Naor. “Approximate kernel clustering.” In: *Mathematika* 55.1-2 (2009), pp. 129–165.
- [49] Subhash Khot et al. “Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs?” In: *SIAM J. Comput.* 37.1 (2007), pp. 319–357.
- [50] Y. Komatsu. “Elementary inequalities for Mills’ ratio.” In: *Rep. Statist. Appl. Res. Un. Japan. Sci. Engrs* (1955).
- [51] J. B. Lasserre. “An explicit exact SDP relaxation for nonlinear 0-1 programs.” In: *IPCO 2001*. Ed. by K. Aardel and A. M. H. Gerards. Vol. 2081. Lecture Notes in Computer Science. Berlin: Springer, 2001, pp. 293–303.
- [52] Michael Lewin, Dror Livnat, and Uri Zwick. “Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems.” In: *Integer Programming and Combinatorial Optimization*. Springer, 2002, pp. 67–82.
- [53] László Lovász. “On the Shannon capacity of a graph.” In: *Information Theory, IEEE Transactions on* 25.1 (1979), pp. 1–7.
- [54] László Lovász and Alexander Schrijver. “Cones of matrices and set-functions and 0-1 optimization.” In: *SIAM Journal on Optimization* 1.2 (1991), pp. 166–190.
- [55] Shiro Matuura and Tomomi Matsui. *0.863-approximation algorithm for MAX DICUT*. Springer, 2001.
- [56] Yu Nesterov. “Semidefinite relaxation and nonconvex quadratic optimization.” In: *Optimization methods and software* 9.1-3 (1998), pp. 141–160.
- [57] Yurii Nesterov, Arkadii Nemirovskii, and Yinyu Ye. *Interior-point polynomial algorithms in convex programming*. Vol. 13. SIAM, 1994.
- [58] Pablo A Parrilo. “Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.” PhD thesis. Citeseer, 2000.

- [59] S. Poljak and Z. Tuza. “The Expected Relative Error Of The Polyhedral Approximation Of The Max-Cut Problem.” In: *Operations Research Letters* 16 (1994), pp. 191–198.
- [60] Svatopluk Poljak and Zsolt Tuza. “The expected relative error of the polyhedral approximation of the max-cut problem.” In: *Operations Research Letters* 16.4 (1994), pp. 191–198.
- [61] Prasad Raghavendra. “Optimal Algorithms and Inapproximability Results for Every CSP?” In: *ACM Symposium on Theory of Computing (STOC)* (2008), pp. 245–254.
- [62] Prasad Raghavendra. “Optimal algorithms and inapproximability results for every CSP?” In: *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM. 2008, pp. 245–254.
- [63] Prasad Raghavendra and David Steurer. “Graph expansion and the unique games conjecture.” In: *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM. 2010, pp. 755–764.
- [64] Prasad Raghavendra and David Steurer. “How to Round Any CSP.” In: *IEEE Foundations of Computing (FOCS)*. 2009, pp. 586–594.
- [65] Thomas Rothvoß. “The matching polytope has exponential extension complexity.” In: *arXiv preprint arXiv:1311.2369* (2013).
- [66] Grant Schoenebeck. “Linear Level Lasserre Lower Bounds for Certain k-CSPs.” In: *IEEE Foundations of Computer Science*. IEEE Computer Society, 2008.
- [67] 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*. 2007, pp. 205–216.
- [68] H. D. Sherali and W. P. Adams. “A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems.” In: *Discrete Appl. Math.* 52.1 (1994), pp. 83–106. ISSN: 0166-218X.
- [69] Luca Trevisan et al. “Gadgets, Approximation, and Linear Programming.” In: *SIAM J. Comput* 29.6 (2000), pp. 2074–2097.
- [70] Madhur Tulsiani. “CSP gaps and reductions in the Lasserre Hierarchy.” In: *STOC*. To appear. 2009.
- [71] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.
- [72] Mihalis Yannakakis. “Expressing combinatorial optimization problems by linear programs.” In: *Journal of Computer and System Sciences* 43.3 (1991), pp. 441–466.

- [73] Yinyu Ye. “A .699-approximation algorithm for Max-Bisection.” In: *Mathematical Programming* 90 (2001), pp. 101–111.
- [74] Uri Zwick. “Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables per Constraint.” In: *SODA*. Vol. 98. 1998, pp. 201–210.
- [75] Uri Zwick. “Finding almost-satisfying assignments.” In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM. 1998, pp. 551–560.
- [76] Uri Zwick. “Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems.” In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. ACM. 1999, pp. 679–687.