Polynomial Proof Systems, Effective Derivations, and their Applications in the Sum-of-Squares Hierarchy

Benjamin Weitz

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-38
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-38.html

May 9, 2017

Polynomial Proof Systems, Effective Derivations, and their Applications in the Sum-of-Squares Hierarchy

by

Benjamin Weitz

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
Professor Luca Trevisan

Spring 2017

Polynomial Proof Systems, Effective Derivations, and their Applications in the Sum-of-Squares Hierarchy

Abstract

Polynomial Proof Systems, Effective Derivations, and their Applications in the Sum-of-Squares Hierarchy

by

Benjamin Weitz

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Prasad Raghavendra, Chair

Semidefinite programming (SDP) relaxations have been a popular choice for approximation algorithm design ever since Goemans and Williamson used one to improve the best approximation of Max-Cut in 1992. In the effort to construct stronger and stronger SDP relaxations, the Sum-of-Squares (SOS) or Lasserre hierarchy has emerged as the most promising set of relaxations. However, since the SOS hierarchy is relatively new, we still do not know the answer to even very basic questions about its power. For example, we do not even know when the SOS SDP is guaranteed to run correctly in polynomial time!

In this dissertation, we study the SOS hierarchy and make positive progress in understanding the above question, among others. First, we give a sufficient, simple criteria which implies that an SOS SDP will run in polynomial time, as well as confirm that our criteria holds for a number of common applications of the SOS SDP. We also present an example of a Boolean polynomial system which has SOS certificates that require $2^{O(\sqrt{n})}$ time to find, even though the certificates are degree two. This answers a conjecture of [54].

Second, we study the power of the SOS hierarchy relative to other symmetric SDP relaxations of comparable size. We show that in some situations, the SOS hierarchy achieves the best possible approximation among every symmetric SDP relaxation. In particular, we show that the SOS SDP is optimal for the Matching problem. Together with an SOS lower bound due to Grigoriev [32], this implies that the Matching problem has no subexponential size symmetric SDP relaxation. This can be viewed as an SDP analogy of Yannakakis' original symmetric LP lower bound [72].

As a key technical tool, our results make use of low-degree certificates of ideal membership for the polynomial ideal formed by polynomial constraints. Thus an important step in our proofs is constructing certificates for arbitrary polynomials in the corresponding constraint ideals. We develop a meta-strategy for exploiting symmetries of the underlying combinatorial problem. We apply our strategy to get low-degree certificates for Matching, Balanced CSP, TSP, and others.

To my wonderful parents, brother, and girlfriend.

Contents

Acknowledgments

First, I would like to thank my advisor Prasad Raghavendra for the invaluable help and time he has given me over the last five years at UC Berkeley. He is undoubtedly the person second-most responsible for the creation of this document. My stay at Berkeley was not always easy, and I passed long periods of time without achieving results, which ended with me feeling frustrated and hopeless. Despite this, Prasad was always willing to devote time and talk with me, on subjects both mathematical and otherwise. His mind is truly incredible, and I was always amazed by how quickly he could imagine new and different approaches to problems we were stuck on. Thanks to his assistance, I am now graduating and I am excited for the future. Thank you so much, Prasad!

I would also like to thank Chris Umans (also a Berkeley alum), who was one of my favorite professors at Caltech. Chris was the first person who showed me what research in Theoretical Computer Science was like, and he helped cultivate my interest in it during my time at Caltech. He was also a fantastic teacher, and his classes were a joy to attend. He is probably third-most responsible for this thesis, as without him I may not have even come to graduate school.

Thanks are also in order for Satish Rao, Nikhil Srivastava, and Luca Trevisan for serving on both my Qualifying Committee and my Dissertation Committee. Thank you for taking time out your schedule to listen to me ramble.

I would also like to thank my many graduate student peers at Berkeley (and one technically from Stanford) for sharing many of the experiences of the past five years with me. I would particularly like to thank Jonah Brown-Cohen, Rishi Gupta, Alex Psomas, Aviad Rubinstein, Jarett Schwartz, Ning Tan and Tselil Schramm. Each of you has been a great friend to me, and contributed in no small part to my happiness during my time at Berkeley. I will miss all of you greatly.

My girlfriend Irene deserves more than I could possibly write in the small space provided to me. You have happily supported me in the pursuit of all of my goals, both academic and otherwise. That means everything to me, and I love you very much. I cannot wait to see what the future has in store for us.

Finally, I want to thank my loving mother, father, and brother. Each of you has provided me with plenty of advice when times were tough for me, and it is thanks to you that I was even in the position to begin to attempt this thesis in the first place. In many respects, you are partly responsible for all of my achievements because you raised me. Luckily, I think you did a pretty darn good job of it. I know I do not call as often as you would like, but know that I love each of you to a large degree.

Chapter 1

Introduction

1.1 Combinatorial Optimization and Approximation

Combinatorial optimization problems have been intensely studied by mathematicians and computer scientists for many years. Here we mean any computational task which involves maximizing a function over some discrete set of feasible solutions. The function to maximize is given as input to an algorithm, which attempts to find the feasible solution which achieves the best value. Here are a few examples of problems which will appear repeatedly throughout this thesis:

Example 1.1.1. The Matching problem is, given a graph $G = (V, E)$, compute the size of the largest subset $F \subseteq E$ such that any two edges $e_1, e_2 \in F$ are disjoint.

Example 1.1.2. The Traveling Salesperson, or TSP problem is, given a set of points X and a distance function $d : X \times X \to \mathbb{R}_+$, compute the least distance traveled by any tour which visits every point in X exactly once and returning to the starting point.

Example 1.1.3. The c-Balanced CSP problem is, given Boolean formulas $\phi_1, \ldots, \phi_m$, compute the largest number of $\phi_1, \ldots, \phi_m$ that can be simultaneously satisfied by an assignment with a c-fraction of variables assigned true.

Computer scientists initially began studying combinatorial optimization problems because they appear frequently in both practice and theory. For example, TSP naturally arises when trying to plan school bus routes and Matching clearly emerges when trying to match medical school graduates to hospitals for residency. Unfortunately for the school bus driver, solving TSP has proven to be exceedingly difficult because optimizing such routes is **NP**-hard [43]. Indeed, almost all combinatorial problems of interest are **NP**-hard (Matching is a notable exception), and are thus believed to be computationally intractable. The barrier of **NP**-hardness for solving these problems has been in place since the 1970s.

In an attempt to overcome this roadblock, the framework of approximation algorithms emerged a few years later in 1976 [64]. Rather than trying to exactly solve TSP by finding the route that minimizes the distance traveled, an approximation algorithm attempts to find a route that is not too much longer than the minimum possible route. For example, maybe the algorithm finds a route that is guaranteed to be at most twice the length of the minimum route, even though the minimum route itself is impossible to efficiently compute. A wide variety of algorithmic techniques have been brought to bear on approximation problems. In this work we will focus on writing *convex relaxations* for combinatorial problems in order to approximate them.

1.2 Convex Relaxations

A popular strategy in approximation algorithm design is to develop convex relaxations for combinatorial problems, as can be seen for example in [28, 70, 2, 48]. Since the solution space for combinatorial problems is discrete, we frequently know of no better maximization technique than to simply evaluate a function on every point in the space. However, if we embed the combinatorial solutions somehow in a continuous space and the combinatorial function as a continuous function f, we can enlarge the solution space to make it convex. The enlarged space is called the *feasible region* of the convex relaxation. If we choose our feasible region carefully, then standard convex optimization techniques can be applied to optimize f over it. Because the new solution space is larger than just the set of discrete solutions, the value we receive will be an overestimate of the true discrete maximum of f. We want the convex relaxation to be a *good approximation* in the sense that this overestimate is not too far from the true maximum of f.

Example 1.2.1 (Held-Karp relaxation for TSP)**.** Given an instance of TSP, i.e. distance function $d : [n] \times [n] \to \mathbb{R}$, for every tour τ (a cycle which visits each $i \in [n]$ exactly once), let $(\chi_\tau)_{ij} = 1$ if τ visits j immediately after or before i. Each χ_τ is an embedding of a tour τ in $\mathbb{R}^{\binom{n}{2}}$. Then

$$K = \left\{ x \,\middle|\, \forall S \subset [n] : \sum_{(i,j)\in S\times\overline{S}} x_{ij} \geq 2, \forall ij : 0 \leq x_{ij} \leq 1 \right\}$$

and the function $f = \sum_{ij} x_{ij}$ is a convex relaxation for TSP. In fact, when d is a metric, $\min_K f$ is at least $2/3$ the true minimum.

Proving that a relaxation is a good approximation is usually highly non-trivial, and is frequently done by exhibiting a *rounding scheme*. A rounding scheme is an algorithm that takes a point in the relaxed body and maps it to one of the original feasible solutions for the combinatorial problem. Rounding schemes are designed so that they output a point with approximately the same value, i.e. within a multiplicative factor of ρ. This implies that minimizing over the relaxed body gives an answer that is within a factor of ρ of minimizing over the discrete solutions. As an example, Christofides' approximation for TSP [19] can be interpreted as a rounding algorithm for the Held-Karp relaxation which achieves an approximation factor of 3/2.

In this thesis we will consider a particular kind of convex relaxation, called a semidefinite program (SDP). In an SDP, the enlarged convex body is the intersection of an affine plane with the cone of positive semidefinite (PSD) matrices, that is, the set of symmetric matrices which have all non-negative eigenvalues. The Ellipsoid Algorithm (a detailed history of which can be found in [1]) can be used to optimize a linear function over convex bodies in time polynomial in the dimension[1] so long as there is an efficient procedure to find a separating hyperplane for points outside the body. If a matrix is not PSD, then it must have an eigenvector with a negative eigenvalue. This eigenvector forms a separating hyperplane, and since eigenvector computations can be performed efficiently, the Ellipsoid Algorithm can be used to efficiently optimize linear functions over the feasible regions of SDPs.

SDPs are generalizations of linear programs (LPs), which are convex relaxations whose feasible regions are the intersection of an affine plane with the non-negative orthant. LPs have enjoyed extensive use in approximation algorithms (see [71] for an in-depth discussion). Since the non-negative orthant can be obtained as a linear subspace of the PSD cone (the diagonal of the matrices), SDPs should be able to provide stronger approximation algorithms than LPs.

SDPs first appeared in [49] as a method to study approximation of the Independent Set problem. The work of [28] catapulted SDPs to the cutting edge of approximation algorithms research when the authors wrote an SDP relaxation with a randomized rounding algorithm for the Max Cut problem, achieving the first non-trivial, polynomial-time approximation. We now know that this result separates SDPs from LPs, as [15] implies that any LP relaxation achieving such approximation for Max Cut must be exponential size. In fact, the SDP of [28] is so effective for this problem that it remains the best polynomial-time approximation for Max Cut we know, even decades years later. Since then SDPs have seen a huge amount of success in the approximation world for a wide variety of problems, including clustering [56], tensor decomposition [68], Vertex Cover [41], Sparsest Cut [2], graph coloring [17], and *especially* constraint satisfaction problems (CSPs) [26, 33, 16]. In fact, if a complexity assumption called the Unique Games Conjecture [44] is true, then the work of Raghavendra [59] implies that SDP relaxations provide optimal approximation algorithms for CSPs; to develop a better algorithm would prove **P**=**NP**.

The success of SDPs has prompted significant investigation into the limits of their power. For Boolean combinatorial problems, in principle one could write an SDP with an exponential number of variables that exactly solves the problem. However, such an SDP would not be of much use since even the Ellipsoid Algorithm would require an exponential amount of time to solve the SDP. The study of lower bounds for SDPs has thus been focused on proving that approximating a combinatorial problem requires an SDP with a large number of

[1]Actually the runtime of the Ellipsoid Algorithm also depends polynomially on $\log R$, where R is the radius of the smallest ball intersecting the feasible region of the SDP. This technical requirement is usually not an issue for most SDPs, but it will turn out to be important for studying the SOS hierarchy, which will see a bit later.

variables. This can be seen as a continuation of the work on LP lower bounds of Yannakakis [72], in which he proved that the MATCHING problem has no symmetric LP relaxation of subexponential size. The symmetric requirement was finally dropped 25 years later by [63], and more asymmetric lower bounds were given in [24] for TSP and [15] for CSPs. However, SDP relaxations are fairly new compared to LP relaxations, and there are significantly fewer examples of strong SDP lower bounds. The existence of 0/1 polytopes that require exponential-size exact SDP relaxations is proven in [11]. In [23] and [47] the authors provide exponential symmetric SDP lower bounds for CSPs, and [46] are able to drop the symmetry requirement, again for CSPs.

1.3 Sum-of-Squares Relaxations

There is a particular family of SDP relaxations which has received a great deal of attention recently as a promising tool for approximation algorithms. Called the Sum-of-Squares (SOS) or Lasserre relaxations, they first appeared in [66, 55, 45] as a sequence of SDPs that eventually exactly converge to any 0/1 polytope. They have recently formed the foundation of algorithms for many different problems, ranging from tensor problems [68, 4, 37, 57] to Independent Set [18], Knapsack [42], and CSPs and TSP [61, 47]. There has even been hope among computer scientists that the SOS relaxations could represent a single, unified algorithm which achieves optimal approximation guarantees for many, seemingly unrelated problems [5]. We give a brief description of the SOS relaxations here and give a more precise definition in Section 2.6.

We consider a polynomial embedding of a combinatorial optimization problem, i.e. there are sets of polynomials $\mathcal{P}$ and $\mathcal{Q}$ such that solving

$$\begin{aligned} &\max r(x) \\ \text{s.t. } &p(x) = 0, \forall p \in \mathcal{P} \\ &q(x) \geq 0, \forall q \in \mathcal{Q} \end{aligned}$$

is equivalent to solving the original combinatorial problem. This is not unusual, and indeed every combinatorial optimization problem has such an embedding. One way to solve such an optimization problem is to pick some θ and check if $\theta - r(x)$ is non-negative on the feasible points. If we can do this, then by binary search we can compute the maximum of r quickly. But how can we check if $\theta - r(x)$ is non-negative? The SOS relaxations attempt to express $\theta - r(x)$ as a sum-of-squares polynomial, modulo the constraints of the problem. In other words, they try to find a polynomial identity of the form

$$\theta - r(x) = \sum_i h_i^2(x) + \sum_{q \in \mathcal{Q}} \left(\sum_j h_{qj}^2(x) \right) q(x) + \sum_{p \in \mathcal{P}} \lambda_p(x) p(x)$$

for some polynomials $\{h_i\}, \{h_{qj}\}, \{\lambda_p\}$. We call such an identity an SOS proof of non-negativity for $\theta - r(x)$. If such an identity exists, then certainly $\theta - r(x)$ is non-negative on any x satisfying the constraints. Unless we hope to break the **NP**-hardness barrier, looking for *any* such identity is intractable, so we consider relaxing the problem to checking for the existence of such an identity that uses only polynomials up to degree $2d$. The existence of a degree $2d$ identity turns out to be equivalent to the feasibility of a certain SDP of size $n^{O(d)}$ (see Section 2.6 for specifics), which we call the *degree-d* or *dth SOS SDP*.

While the SOS relaxations have been popular and successful, they are still relatively new, and so our knowledge about them is far from complete. There are even very basic questions about them for which we do not know the answer. In particular, we do not even know when we can solve the SOS relaxations in polynomial time! Because the dth SOS relaxation is a semidefinite program of size $n^{O(d)}$, it is often claimed that any degree-d proof can be found in time polynomial in $n^{O(d)}$ via the Ellipsoid algorithm. However, this claim was debunked very recently by Ryan O'Donnell in [54]. He noted that complications could arise if every proof of non-negativity involves polynomials with extremely large coefficients, and furthermore, he gave an explicit example showing that it is possible for this to occur. Resolving this issue is of paramount importance, as the SOS relaxations lie at the heart of so many approximation algorithms. In this dissertation, we continue this line of work with some positive and negative results discussed in Section 1.5.

Another open area of research is investigating the true power of the SOS relaxations. Since we know SOS relaxations provide good approximation for so many computational problems, it is natural to continue to apply them to new problems. This is a worthy pursuit, but not one that will be explored in this work. An alternative approach would be to try to identify for which problems the SOS relaxations *do not* provide good approximations. For any Boolean optimization problem, if d is large enough, then the dth SOS relaxation will solve the problem exactly. However, if d is too large, then the SDP will have a super-polynomial number of variables, so that even the Ellipsoid Algorithm cannot solve it in polynomial time. Thus as for general SDP lower bounds, it is common to rephrase this question by giving a lower bound on the degree of the SOS relaxation required to achieve a good enough approximation. The degree-d SOS relaxation is size $n^{O(d)}$, so if d is super-constant then the size of the SDP is super-polynomial. This area of research has been much more fruitful than general SDP lower bounds, as the SOS relaxations are concrete objects which are more easily reasoned about. In [32], Grigoriev gives a linear degree lower bound against the MATCHING problem. A sequence of results [52, 21, 60, 36, 6] all give lower bounds against the PLANTED CLIQUE problem. SOS lower bounds for DENSEST k-SUBGRAPH are given in [8]. Different CSP problems are considered in [65, 58, 27, 69, 46]. Proving more lower bounds is also a noble goal, but this thesis will focus on a slightly different evaluation of the effectiveness of the SOS relaxations.

Rather than evaluating the SOS relaxations by how good an approximation they achieve in the absolute sense, we will be evaluating them relative to other SDPs. In particular, we will explore whether or not there exist other SDPs which perform better than the SOS relaxations. Previously, [47, 46] proved that the SOS relaxations provide the best approximation among SDPs of a comparable size for CSPs. We will explore the more restricted setting of [47], where the other SDP relaxations we measure against must be symmetric in some sense, i.e.

respect the natural symmetries of the corresponding combinatorial optimization problem (see Section 2.7 for details).

1.4 Polynomial Ideal Membership and Effective Derivations

In order to study the SOS relaxations, in this dissertation we use as a technical tool polynomial proof systems and the existence of low-degree polynomial proofs. Here we introduce a bit of background on these tools. The polynomial ideal membership problem is the following computational task: Given a set of polynomials $\mathcal{P} = \{p_1, \dots, p_n\}$ and a polynomial r, we want to determine if r is in the ideal generated by $\mathcal{P}$ or not, denoted $\langle \mathcal{P} \rangle$. This problem was first studied by Hilbert [35], and has applications in solving polynomial systems [20] and polynomial identity testing [3]. The theory of Gröbner bases [12] originated as a method to solve the membership problem. Unfortunately, the membership problem is **EXPSPACE**-hard to solve in general [51, 38]. Luckily this will not impede us too much, since we will be studying this problem for the very special instances that correspond to common combinatorial optimization problems.

The membership problem is easily solvable if there exist low-degree proofs of membership for the ideal $\langle \mathcal{P} \rangle$. Note that $r \in \langle \mathcal{P} \rangle$ if and only if there exists a polynomial identity

$$r(x) = \sum_{p \in \mathcal{P}} \lambda_p(x) p(x)$$

for some polynomials $\{\lambda_p \mid p \in \mathcal{P}\}$. We call such a polynomial identity a *derivation* or *proof of membership* for r from $\mathcal{P}$. If we had an *a priori* bound on the degree required for this identity, we could simply solve a system of linear equations to determine the coefficients of each λ_p. The Effective Nullstellensatz [34] tells us that we can take $d \leq (\deg r)^{2^{|\mathcal{P}|}}$. This bound is not terribly useful, because we would need to solve an enormous linear system. This is unavoidable in general because of the **EXPSPACE**-hardness, but in specific cases we could hope for a better bound on d. In particular, the polynomial ideals that arise from combinatorial optimization problems frequently have nice properties that make them much more reasonable than arbitrary polynomial ideals. For example, these ideals are often Boolean (and thus have finite solution spaces) and highly symmetric. In these cases, we could hope for a much better degree bound.

This problem has been studied in [7, 14, 31, 13], mostly in the context of lower bounds. In these works the problem is referred to as the degree of *Nullstellensatz proofs* of membership for r. In this work we will continue to study this problem, however we will be mostly interested in upper bounds on the required degree. We will be able to use the existence of low-degree Nullstellensatz proofs for combinatorial ideals to study the SOS relaxations.

1.5 Contribution of Thesis

In the first part of this thesis, to set the stage for analyzing the SOS relaxations, we give upper bounds on the required degree for Nullstellensatz proofs for many ideals arising from a number of combinatorial problems, including Matching, TSP, and Balanced CSP. In particular, we prove that if $\mathcal{P}$ is a set of polynomials corresponding to Matching, TSP, or Balanced CSP and $r \in \langle \mathcal{P} \rangle$,[2] then r has a Nullstellensatz proof degree at most $k \deg r$ for $k \leq 3$. This implies that the polynomial ideal membership problem for these ideals has a polynomial-time solution. We achieve results for so many different ideals because we develop a meta-strategy which exploits the symmetries present in the underlying combinatorial optimization problems in order to give good upper bounds.

Recall that the Ellipsoid Algorithm can be used to solve the SOS relaxations in polynomial time so long as the bit complexity of the SOS proof it is trying to find is polynomially bounded. The second part of the thesis is devoted to studying this problem of bit complexity in SOS proofs. Our main contribution is to show that SOS proofs for Planted Clique, Max CSP, TSP, Bisection, and some others can be taken to have polynomial bit complexity. This implies that SOS relaxations for these problems do indeed run in polynomial time, patching a potential problem with several known approximation algorithms. We prove this result by giving a set of criteria for the constraints $\mathcal{P}$ and $\mathcal{Q}$, one of which is that any polynomial $r \in \langle \mathcal{P} \rangle$ has Nullstellensatz proofs of bounded degree. This partially motivates our study of Nullstellensatz proofs. On the negative side, we provide an example of a set of polynomials $\mathcal{P}$ *containing Boolean constraints* and a polynomial r which has a degree-two SOS proof of non-negativity, but every SOS proof up until degree $\Omega(\sqrt{n})$ requires coefficients of size roughly $2^{2^{\sqrt{n}}}$. This refutes a conjecture of [54] that simply containing Boolean constraints suffices for polynomial bit complexity.

The final part of the thesis investigates the power of the SOS relaxations, especially for the Matching problem. In particular, we prove the SDP version of Yannakakis' LP lower bound for Matching: The Matching problem has no subexponential-size symmetric SDP achieving a non-trivial approximation. We prove this by extending the techniques of [47] to Matching and show that the degree-d SOS relaxation achieves the best approximation of any symmetric SDP of about the same size. Together with the SOS lower bound of Grigoriev [32], this gives us the exponential lower bound. We prove a similar SOS optimality result for TSP and Balanced CSP, but there are no currently known SOS lower bounds for these problems.

1.6 Organization of Thesis

Chapter 2 will contain preliminary and background discussion on mathematical concepts needed. We will precisely define many of the objects discussed in this introduction, including

[2] Technically, for c-Balanced CSP we also require that $\deg r \leq c$. This is due to a specific obstruction for higher degree polynomials, and the details are in Section 3.5.

combinatorial optimization problems, SDP relaxations, and the SOS relaxations themselves. In Chapter 3 we will discuss low-degree proofs of membership and compile a (non-exhaustive) list of combinatorial optimization problems which admit such proofs. In Chapter 4, we discuss the bit complexity of SOS proofs, and show how low-degree proofs can be used to prove the existence of SOS proofs with small bit complexity. In Chapter 5 we discuss the optimality of the SOS relaxations, and show how this implies an exponential size lower bound for approximating the Matching problem. Finally, in Chapter 6 we discuss a few open problems continuing the lines of research of this thesis.

Chapter 2

Preliminaries

In this chapter we define and discuss the basic mathematical concepts needed for this dissertation.

2.1 Notation

In this section we clarify the basic notation that will be used throughout this thesis. We will use $[n]$ for the set $\{1, 2, \ldots, n\}$, and $[\binom{n}{2}]$ for the set $\{(i, j) \mid i, j \in [n], i \neq j\}$. For two vectors u and v, we use $u \cdot v$ to denote the inner product $\sum_i u_i v_i$. For a matrix A, we use A^T to denote the transpose of A. For matrices A and B with the same dimensions, we use $A \cdot B$ or $\langle A, B\rangle$ to denote the inner product $\mathrm{Tr}[AB^T] = \sum_{ij} A_{ij}B_{ij}$. We will also use $\cdot$ to emphasize multiplication. We use $\mathbb{R}_+$ to denote the space of positive reals, and $\mathbb{R}^{m\times n}$ to denote the space of $m \times n$ matrices. We use $\mathbb{S}^n$ to denote the space of $n \times n$ symmetric matrices.

Definition 2.1.1. A matrix $A \in \mathbb{S}^n$ is called *positive semidefinite* (PSD) if any of the following equivalent conditions holds:

- $v^T Av \geq 0$ for every $v \in \mathbb{R}^n$.
- $A = \sum_i \lambda_i v_i v_i^T$ for some $\lambda_i \geq 0$ and $v_i \in \mathbb{R}^n$.
- Every eigenvalue of A is non-negative.
- $A \cdot B \geq 0$ for every positive semidefinite B.

We use $\mathbb{S}^n_+$ to denote the space of positive semidefinite $n \times n$ matrices, and we write $A \succeq 0$ interchangeably with $A \in \mathbb{S}^n_+$. If every eigenvalue of A is *positive*, then we say A is *positive definite* and write $A \succ 0$. The set of positive definite matrices is the interior of the set of positive semidefinite matrices.

For any matrix or vector, we use $\|\cdot\|$ to denote the maximum entry of that matrix or vector, often represented $\|\cdot\|_\infty$ in other works. In this thesis we will not use any other norms, so we find it most convenient to just omit the subscript.

We use $\mathbb{R}[x_1, \ldots, x_n]$ to denote the space of polynomials on variables $x_1, \ldots, x_n$, and $\mathbb{R}[x_1, \ldots, x_n]_d$ for the space of degree d polynomials. For a fixed integer d to be understood from context and a polynomial p of degree at most d, let $N = \binom{n+d-1}{d}$. We use $\tilde{p}$ for the element of $\mathbb{R}^N$ which is the vector of coefficients of p up to degree d. We use $\mathbf{x}^{\otimes \mathbf{d}}$ to denote the vector of monomials such that $p(x) = \tilde{p} \cdot \mathbf{x}^{\otimes \mathbf{d}}$.

If p is a polynomial of degree at most $2d$, then we also use $\hat{p}$ for an element of $\mathbb{R}^{N \times N}$ such that $p(x) = \hat{p} \cdot \mathbf{x}^{\otimes \mathbf{d}}(\mathbf{x}^{\otimes \mathbf{d}})^T$. Since multiple entries of $\mathbf{x}^{\otimes \mathbf{d}}(\mathbf{x}^{\otimes \mathbf{d}})^T$ are equal, there are multiple choices for $\hat{p}$, for concreteness we choose the one that evenly distributes the coefficient over the equal entries. Now $p = \sum_i q_i^2$ for some polynomials q_i if and only if $\hat{p} = \sum_i \tilde{q}_i \tilde{q}_i^T$, i.e. $\hat{p} \in \mathbb{S}_+^N$. We use $\|p\|$ to denote the largest absolute value of a coefficient of p. If $\mathcal{P}$ is a set of polynomials, then $\|\mathcal{P}\| = \max_{p \in \mathcal{P}} \|p\|$.

2.2 Semidefinite Programming and Duality

In order to explore the power of the Sum-of-Squares relaxations, first we need to explain what a semidefinite program is. In this section we define semidefinite programs and their duals, which are also semidefinite programs.

Definition 2.2.1. A *semidefinite program (SDP) of size* d is a tuple $(C, \{A_i, b_i\}_{i=1}^m)$ where $C, A_i \in R^{d \times d}$ for each i, and $b_i \in \mathbb{R}$ for each i. The *feasible region* of the SDP is the set $S = \{X \mid \forall i : A_i \cdot X = b_i, X \in \mathbb{S}_+^d\}$. The *value* of the SDP is $\max_{X \in S} C \cdot X$.

Fact 2.2.2. *There is an algorithm (referred to as the Ellipsoid Algorithm in this thesis) that, given an SDP* $(C, \{A_i, b_i\}_{i=1}^m)$ *whose feasible region* S *intersects a ball of radius* R*, computes the value of that SDP up to accuracy* ϵ *in time polynomial in* d*,* $\max_i (\log \|A_i\|, \log |b_i|)$*,* $\log \|C\|$*,* $\log R$*, and* $\frac{1}{\epsilon}$*.*

Definition 2.2.3. The *dual* of an SDP $(C, \{A_i, b_i\}_{i=1}^m)$ is the optimization problem (with variables (y, S)):

$$\begin{aligned} &\min_{y,S} b \cdot y \\ \text{s.t. } &\sum_i A_i y_i - C = S \\ &S \succeq 0. \end{aligned}$$

The *value* of the dual is the value of the optimum $b \cdot y^*$.

The following is a well-known fact about strong duality for SDPs, due to Slater [67].

Lemma 2.2.4 (Slater's Condition)**.** *Let* P *be the SDP* $(C, \{A_i, b_i\}_{i=1}^m)$ *and let* D *be its dual. If* X *is feasible for* P *and* (y, S) *is feasible for* D*, then* $C \cdot X \leq b \cdot y$*. Moreover, if there exists a* strictly feasible *point* X *for* P *or* (y, S) D*, that is, a feasible* X *with* $X \succ 0$ *or a feasible* (y, S) *with* $S \succ 0$*, then* $\operatorname{val} P = \operatorname{val} D$*.*

2.3 Polynomial Ideals and Polynomial Proof Systems

We write $p(x)$ or sometimes just p for a polynomial in $\mathbb{R}[x_1, \ldots, x_n]$, and $\mathcal{P}$ for a set of polynomials. We will often also use q and r for polynomials and $\mathcal{Q}$ for a second set of polynomials.

Definition 2.3.1. Let $\mathcal{P}, \mathcal{Q}$ be any sets of polynomials in $\mathbb{R}[x_1, \ldots, x_n]$, and let S be any set of points in $\mathbb{R}^n$.

- We call $V(\mathcal{P}) = \{x \in \mathbb{R}^n \mid \forall p \in \mathcal{P} : p(x) = 0\}$ the *real variety* of $\mathcal{P}$.
- We call $H(\mathcal{Q}) = \{x \in \mathbb{R}^n \mid \forall q \in \mathcal{Q} : q(x) \geq 0\}$ the *positive set* of $\mathcal{Q}$.
- We call $I(S) = \{p \in R[x_1, \ldots, x_n] \mid \forall x \in S : p(x) = 0\}$ the *vanishing ideal* of S.
- We denote $\langle \mathcal{P} \rangle = \{q \in R[x_1, \ldots, x_n] \mid \exists \lambda_p(x) : q = \sum_{p \in \mathcal{P}} \lambda_p \cdot p\}$ for the *ideal generated by* $\mathcal{P}$.
- We call $\mathcal{P}$ *complete* if $\langle \mathcal{P} \rangle = I(V(\mathcal{P}))$.
- If $\mathcal{P}$ is complete, then we write $p_1 \cong p_2 \mod \langle \mathcal{P} \rangle$ if $p_1 - p_2 \in \langle \mathcal{P} \rangle$ or, equivalently, if $p_1(\alpha) = p_2(\alpha)$ for each $\alpha \in V(\mathcal{P})$.

Gröbner bases are objects first considered in [12] as a way to determine if a polynomial r is an element of $\langle \mathcal{P} \rangle$. We define them here and include some of their important properties.

Definition 2.3.2. Let $\succ$ be an ordering on monomials such that, for three monomials u, v, and w, if $u \succeq v$ then $uw \succeq vw$. We say that $\mathcal{P}$ is a *Gröbner Basis* for $\langle \mathcal{P} \rangle$ (with respect to $\succ$) if, for every $r \in \langle \mathcal{P} \rangle$, there exists a $p \in \mathcal{P}$ such that the leading term of r is divisible by the leading term of p.

Example 2.3.3. Consider the polynomials on n variables $x_1, \ldots, x_n$ and let $\succ$ be the degree-lexicographic ordering, so that for two monomials u, and v, $u \succeq v$ if the vector of degrees of u is larger than the vector of v in the lexicographic ordering. Then $\mathcal{P} = \{x_i^2 - x_i \mid i \in [n]\}$ is a Gröbner Basis. The proof is in the proof of Corollary 3.1.2.

If $\mathcal{P}$ is a Gröbner basis, then it is a nice generating set for $\langle \mathcal{P} \rangle$ in the sense that it is possible to define a multivariate division algorithm for $\langle \mathcal{P} \rangle$ with respect to $\mathcal{P}$.

Definition 2.3.4. Let $\succ$ be an ordering of monomials such that if $x_U \succ x_V$ then $x_U x_W \succ x_V x_W$. We say a polynomial q is *reducible* by a set of polynomials $\mathcal{P}$ if there exists a $p \in \mathcal{P}$ such that some monomial of q, say $c_Q x_Q$, is divisible by the *leading* term of p, $c_P x_P$. Then a *reduction* of q by $\mathcal{P}$ is $q - \frac{c_Q}{c_P} x_{Q \setminus P} \cdot p$. We say that a *total reduction* of q by $\mathcal{P}$ is a polynomial obtained by iteratively applying reductions until we reach a polynomial which is not reducible by $\mathcal{P}$.

In general the total reductions of a polynomial q by a set of polynomials $\mathcal{P}$ is not unique and depends on which polynomials one chooses from $\mathcal{P}$ to reduce by, and in what order. So it does not make much sense to call this a division algorithm since there is not a unique remainder. However, when $\mathcal{P}$ is a Gröbner basis, there is indeed a unique remainder.

Proposition 2.3.5. *Let $\mathcal{P}$ be a Gröbner basis for $\langle \mathcal{P} \rangle$ with respect to $\succ$. Then for any polynomial q, there is a unique total reduction of q by $\mathcal{P}$. In particular if $q \in \langle \mathcal{P} \rangle$, then the total reduction of q by $\mathcal{P}$ is 0. The converse is also true, so if $\mathcal{P}$ is a set of polynomials such that any polynomial $q \in \langle \mathcal{P} \rangle$ has unique total reduction by $\mathcal{P}$ equal to 0, then $\mathcal{P}$ is a Gröbner basis.*

Proof. When we reduce a polynomial q by $\mathcal{P}$, the resulting polynomial does not contain one term of q, since it was canceled via a multiple of p for some polynomial $p \in \mathcal{P}$. Because it was canceled via the leading term of p, no higher monomials were introduced in the reduction. Thus as we apply reduction, the position of the terms of q monotonically decrease. This has to terminate at some point, so there is a remainder r which is not reducible by $\mathcal{P}$. To prove that r is unique, first notice that the result of total reduction is a polynomial identity $q = p + r$, where $p \in \langle \mathcal{P} \rangle$ and r is not reducible by $\mathcal{P}$. If there are multiple remainders $q = p_1 + r_1$ and $q = p_2 + r_2$, then clearly $r_1 - r_2 = p_2 - p_1 \in \langle \mathcal{P} \rangle$. By the definition of Gröbner Basis, $r_1 - r_2$ must have its leading term divisible by the leading term of some $p \in \mathcal{P}$. But the leading term of $r_1 - r_2$ must come from either r_1 or r_2, neither of which contain terms divisible by leading terms of any polynomial in $\mathcal{P}$. Thus $r_1 - r_2 = 0$.

For the converse, let $q \in \langle \mathcal{P} \rangle$, and note again that any reduction of q by a polynomial in $\mathcal{P}$ does not include higher monomials than the one canceled. Since the only total reduction of q is 0, its leading term has to be canceled eventually, so it must be divisible by the leading term of some polynomial in $\mathcal{P}$. □

Testing Zero Polynomials

This section discusses how to certify that a polynomial $r(x)$ is zero on all of some set S. In the main context of this thesis, we have access to some polynomials $\mathcal{P}$ such that $S = V(\mathcal{P})$. When $\mathcal{P}$ is complete, testing if r is zero on S is equivalent to testing if $r \in \langle \mathcal{P} \rangle$. One obvious way to do this is to simply brute-force over the points of $V(\mathcal{P})$ and evaluate r on all of them. However, we are mostly interested in situations where the points of $V(\mathcal{P})$ are in bijection with solutions to some combinatorial optimization problem. In this case, there are frequently an exponential number of points in $V(\mathcal{P})$ and this amounts to a brute-force search over this space. If $\mathcal{P}$ is a Gröbner basis, then we could also simply compute a total reduction of r by $\mathcal{P}$ and check if it is 0. However, Gröbner bases are often very complicated and difficult to compute, and we do not always have access to one. We want a more efficient certificate for membership in $\langle \mathcal{P} \rangle$.

Definition 2.3.6. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ be a set of polynomials. We say that r *is derived from* $\mathcal{P}$ *in degree* d if there is a polynomial identity of the form

$$r(x) = \sum_{i=1}^{n} \lambda_i(x) \cdot p_i(x),$$

and $\max_i \deg(\lambda_i \cdot p_i) \leq d$. We often call this polynomial identity a Nullstellensatz (HN) proof, derivation, or certificate from $\mathcal{P}$. We also write $r_1 \cong_d r_2$ if $r_1 - r_2$ has a derivation from $\mathcal{P}$ in degree d. We write $\langle \mathcal{P} \rangle_d$ for the polynomials with degree d derivations from $\mathcal{P}$ (*not* the degree d polynomials in $\langle \mathcal{P} \rangle$!).

The following is an important result which connects derivations to the feasibility of a polynomial system of equations.

Lemma 2.3.7 (Hilbert's Weak Nullstellensatz [35])**.** $1 \in \langle \mathcal{P} \rangle$ *if and only if there is no* $\alpha \in \mathbb{C}^n$ *such that* $p(\alpha) = 0$ *for every* $p \in \mathcal{P}$*, i.e.* $\mathcal{P}$ *is infeasible.*

A derivation of 1 from $\mathcal{P}$ is called an *HN refutation of* $\mathcal{P}$. It is a study of considerable interest to bound the degree of refutations for various systems of polynomial equations [7, 14, 31, 13]. However, in this thesis we will primarily concern ourselves with feasible systems of polynomial equations, so we will mostly use the Nullstellensatz to argue that the only constant polynomial in $\langle \mathcal{P} \rangle$ is the zero polynomial.

The following lemma is an easy but important fact, which we will use to construct derivations in Chapter 3.

Lemma 2.3.8. *If* $q_0 \cong_{d_1} q_1$ *and* $q_1 \cong_{d_2} q_2$*, then* $q_0 \cong_d q_2$ *where* $d = \max(d_1, d_2)$*.*

Proof. We have the polynomial identities

$$q_i - q_{i+1} = \sum_{p \in \mathcal{P}} \lambda_{ip} \cdot p$$

for $i = 0$ and $i = 1$. Adding the two identities together gives a derivation for $q_0 - q_2$. The degrees of the polynomials appearing in derivation are clearly bounded by $\max(d_1, d_2)$. $\square$

The problem of finding a degree-d HN derivation for r can be expressed as a linear program with $n^d|\mathcal{P}|$ variables, since the polynomial identity is linear in the coefficients of the λ_i. Thus if such a derivation exists, it is possible to find efficiently in time polynomial in $n^d|\mathcal{P}|$, $\log \|\mathcal{P}\|$ and $\log \|r\|$. These parameters are all polynomially related to the size required to specify the input: $(r, \mathcal{P}, d)$.

Definition 2.3.9. We say that $\mathcal{P}$ is k*-effective* if $\mathcal{P}$ is complete and every polynomial $p \in \langle \mathcal{P} \rangle$ of degree d has a HN proof from $\mathcal{P}$ in degree kd.

When $\mathcal{P}$ is k-effective for constant k, if we ever wish to test membership in $\langle \mathcal{P} \rangle$ for some polynomial r, we need only search for a HN proof up to degree $k \deg r$, yielding an efficient algorithm for the membership problem (this is polynomial time because the size of the input r is $O(n^{\deg r})$).

Testing Non-negative Polynomials with Sum of Squares

Testing non-negativity for polynomials on a set S has an obvious application to optimization. If one is trying to solve the polynomial optimization problem

$$\begin{aligned} &\max r(x) \\ \text{s.t. } &p(x) = 0, \forall p \in \mathcal{P} \\ &q(x) \geq 0, \forall q \in \mathcal{Q}, \end{aligned}$$

then one way to do so is to iteratively pick a θ and test whether $\theta - r(x)$ is positive on $S = V(\mathcal{P}) \cap H(\mathcal{Q})$. If we perform binary search on θ, we can compute the maximum of r very quickly. One way to try and certify non-negative polynomials is to express them as sums of squares.

Definition 2.3.10. A polynomial $s(x) \in \mathbb{R}[x_1, \dots, x_n]$ is called a *sum-of-squares (or SOS)* polynomial if $s(x) = \sum_i h_i^2(x)$ for some polynomials $h_i \in \mathbb{R}[x_1, \dots, x_n]$. We often use $s(x)$ to denote SOS polynomials.

Clearly an SOS polynomial is non-negative on all of $\mathbb{R}^n$. However, the converse is not always true.

Fact 2.3.11 (Motzkin's Polynomial [53])**.** *The polynomial $p(x, y) = x^4y^2 + x^2y^4 - 3x^2y^2 + 1$ is non-negative on $\mathbb{R}^n$ but is not a sum of squares.*

Because our goal is to optimize over some set S, we actually only care about the non-negativity of a polynomial on S rather than on all of $\mathbb{R}^n$.

Definition 2.3.12. A polynomial $r(x) \in \mathbb{R}[x_1, \dots, x_n]$ is called *SOS modulo S* if there is an SOS polynomial $s \in I(S)$ such that $r \cong s \mod I(S)$. If $\deg s = k$ then we say r is *k-SOS modulo S*. If $S = V(\mathcal{P})$ and $\mathcal{P}$ is complete, we sometimes use *modulo $\mathcal{P}$* instead.

If a polynomial r is SOS modulo S then r is non-negative on S. For many optimization problems, $S \subseteq \{0, 1\}^n$. In this case, the converse holds.

Fact 2.3.13. *If $S \subseteq \{0, 1\}^n$ and r is a polynomial which is non-negative on S, then r is n-SOS modulo S.*

When we have access to two sets of polynomials $\mathcal{P}$ and $\mathcal{Q}$ such that $S = V(\mathcal{P}) \cap H(\mathcal{Q})$, as in our main context of polynomial optimization, we can define a certificate of non-negativity:

Definition 2.3.14. Let $\mathcal{P}$ and $\mathcal{Q}$ be two sets of polynomials. A polynomial $r(x)$ is said to have a *degree d proof of non-negativity from $\mathcal{P}$ and $\mathcal{Q}$* if there is a polynomial identity of the form

$$r(x) = s(x) + \sum_{q \in \mathcal{Q}} s_q(x) \cdot q(x) + \sum_{p \in \mathcal{P}} \lambda_p(x) \cdot p(x),$$

where $s(x)$, and each $s_q(x)$ are SOS polynomials, and $\max_{pq}(\deg s, \deg s_q q, \deg \lambda_p p) \leq d$. We often call this polynomial identity a Positivestellensatz Calculus ($\mathrm{PC}_>$) proof of non-negativity, derivation, or certificate from $\mathcal{P}$ and $\mathcal{Q}$. We often identify the proof with the set of polynomials $\Pi = \{s\} \cup \{s_q \mid q \in \mathcal{Q}\} \cup \{\lambda_p \mid p \in \mathcal{P}\}$.

If $\mathcal{Q} = \emptyset$ and both $r(x)$ and $-r(x)$ have $\mathrm{PC}_>$ proofs of non-negativity from $\mathcal{P}$, then we say that r has a $\mathrm{PC}_>$ derivation.

If r has a $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}$ and $\mathcal{Q}$, then r is non-negative on $S = V(\mathcal{P}) \cap H(\mathcal{Q})$. This can be seen by noticing that the first two terms in the proof are non-negative because they are sums of products of polynomials which are non-negative on S, and the final term is of course zero on S because it is in $\langle \mathcal{P} \rangle$.

The problem of finding a degree-d proof of non-negativity can be expressed as a semidefinite program of size $O(n^d(|\mathcal{P}| + |\mathcal{Q}|))$ since a polynomial is SOS if and only if its matrix of coefficients is PSD. Then the Ellipsoid Algorithm can be used to find a degree-d proof of non-negativity Π in time polynomial in $n^d(|\mathcal{P}| + |\mathcal{Q}|), \log \|r\|, \log \|\mathcal{P}\|, \log \|\mathcal{Q}\|$, and $\log \|\Pi\|$. Nearly all of these parameters are bounded by the size required to specify the input of $(r, \mathcal{P}, \mathcal{Q}, d)$. However, the quantity $\|\Pi\|$ is worrisome; Π is not part of the input and we have no *a priori* way to bound its size. One way to argue r has proofs of bounded norm is of course to simply exhibit one. If we suspect there are no proofs with small norm, there are also certificates we can find:

Lemma 2.3.15. *Let $\mathcal{P}$ and $\mathcal{Q}$ be sets of polynomials and $r(x)$ be a polynomial. Pick any $p^* \in \mathcal{P}$. If there exists a linear functional $\phi : \mathbb{R}[x_1, \ldots, x_n] \to \mathbb{R}$ such that*

(1) $\phi[r] = -\epsilon < 0$,

(2) $\phi[\lambda p] = 0$ *for every* $p \in \mathcal{P}$ *except* p^* *and* λ *such that* $\deg(\lambda p) \leq 2d$,

(3) $\phi[s^2 q] \geq 0$ *for every* $q \in \mathcal{Q}$ *and polynomial* s *such that* $\deg(s^2 q) \leq 2d$,

(4) $\phi[s^2] \geq 0$ *for every polynomial* s *such that* $\deg(s^2) \leq 2d$,

(5) $|\phi[\lambda p^*]| \leq \delta \|\lambda\|$ *for every* λ *such that* $\deg(\lambda p^*) \leq 2d$.

then every degree-d $\mathrm{PC}_>$ proof of non-negativity Π for r from $\mathcal{P}$ and $\mathcal{Q}$ has $\|\Pi\| \geq \frac{\epsilon}{\delta}$.

Proof. The proof is very simple. Any degree-d proof of non-negativity for r is a polynomial identity

$$r(x) = s(x) + \sum_{q \in \mathcal{Q}} s_q(x) \cdot q(x) + \sum_{\substack{p \in \mathcal{P} \\ p \neq p^*}} \lambda_p(x) \cdot p(x) + \lambda^*(x) \cdot p^*(x)$$

with polynomial degrees appropriately bounded. If we apply ϕ to both sides, we have

$$\begin{aligned} -\epsilon = \phi[r] &= \phi[s] + \sum_{q \in \mathcal{Q}} \phi[s_q q] + \sum_{\substack{p \in \mathcal{P} \\ p \neq p^*}} \phi[\lambda_p p] + \phi[\lambda^* p] \\ &= a_1 + a_2 + 0 + \phi[\lambda^* p^*], \end{aligned}$$

where $a_1, a_2 \geq 0$ by properties (3) and (4) of ϕ. Thus $\phi[\lambda^* p^*] \leq -\epsilon$, but by property (5), we must have $\|\lambda^*\| \geq \frac{\epsilon}{\delta}$, and thus $\|\Pi\|$ is at least this much as well. $\square$

Strong duality actually implies that the converse of Lemma 2.3.15 is true as well (the reader might notice that ϕ is actually a hyperplane separating r from the set of polynomials with bounded coefficients for p^*), but as we only need this direction in this thesis we omit the proof of the converse. Also note that if $\delta = 0$, i.e. ϕ satisfies $\phi[\lambda p^*] = 0$, then the lemma implies there are no proofs of non-negativity for r.

2.4 Combinatorial Optimization Problems

We follow the framework of [9] for combinatorial problems. We define only maximization problems here, but it is clear that the definition extends easily to minimization problems as well.

Definition 2.4.1. A *combinatorial maximization problem* $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ consists of a finite set $\mathcal{S}$ of feasible solutions and a set $\mathcal{F}$ of non-negative objective functions. An exact algorithm for such a problem takes as input an $f \in \mathcal{F}$ and computes $\max_{\alpha \in \mathcal{S}} f(\alpha)$.

We can also generalize to approximate solutions: Given two functions $c, s \colon \mathcal{F} \to \mathbb{R}$ called approximation guarantees, we say an algorithm (c, s)-approximately solves $\mathcal{M}$ or achieves approximation (c, s) on $\mathcal{M}$ if given any $f \in \mathcal{F}$ with $\max_{s \in \mathcal{S}} f(s) \leq s(f)$ as input, it computes $\mathrm{val} \in \mathbb{R}$ satisfying $\max_{\alpha \in \mathcal{S}} f(\alpha) \leq \mathrm{val} \leq c(f)$. If $c(f) = \rho s(f)$, we also say the algorithm ρ-approximately solves $\mathcal{M}$ or achieves approximation ratio ρ on $\mathcal{M}$.

We think of the functions $f \in \mathcal{F}$ as defining the problem instances and the feasible solutions $\alpha \in \mathcal{S}$ as defining the combinatorial objects we are trying to maximize over. The functions c and s can be thought of as the usual approximation parameters *completeness* and *soundness*. If $c(f) = s(f) = \max_{\alpha \in \mathcal{S}} f(\alpha)$, then a (c, s)-approximate algorithm for $\mathcal{M}$ is also an exact algorithm. Here are few concrete examples of combinatorial maximization problems:

Example 2.4.2 (Maximum Matching)**.** Recall that the Maximum Matching problem is, given a graph $G = (V, E)$, find a maximum set of disjoint edges. We can express this as a combinatorial optimization problem for each even n as follows: K_n be the complete graph on n vertices. The set of feasible solutions $\mathcal{S}_n$ is the set of all maximum matchings on K_n. The objective functions will be indexed by edge subsets of K_n and defined $f_E(M) = |E \cap M|$. It is clear that for a graph $G = (V, E)$ with $|V| = n$, the size of the maximum matching in G is exactly either $\max_{M \in \mathcal{S}_n} f_E(M)$ or $\max_{M \in \mathcal{S}_{n+1}} f_E(M)$, depending on if n is even or odd respectively.

Example 2.4.3 (Traveling Salesperson Problem)**.** Recall that the Traveling Salesperson Problem (TSP) is, given a set X and a function $d : X \times X \to \mathbb{R}^+$, find a tour τ of X that minimizes the total cost of adjacent pairs in the tour (including the first and last elements).

This can be cast in this framework easily: the set of feasible solutions $\mathcal{S}$ is the set of all permutations of n elements. The objective functions are indexed by the function d and can be written $f_d(\tau) = \sum_{i=1}^{n} d(\tau(i), \tau(i+1))$, where $n+1$ is taken to be 1. TSP is a minimization problem rather than a maximization problem, so we ask for the algorithm to compute $\min_{\tau\in\mathcal{S}} f(\tau)$ instead. We could set $s(f) = \min_{\alpha\in\mathcal{S}} f(\alpha)$ and $c(f) = \frac{2}{3}\min_{\alpha\in\mathcal{S}} f(\alpha)$ and ask for an algorithm that (c, s)-approximately solves TSP instead (Christofides' algorithm [19] is one such algorithm when d is a metric).

Definition 2.4.4. For a problem $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ and approximation guarantees c and s, the (c, s)-*Slack Matrix* M is an operator that takes as input an $\alpha \in \mathcal{S}$ and an $f \in \mathcal{F}$ such that $\max_{\alpha\in\mathcal{S}} f(\alpha) \leq s(f)$ and returns $M(\alpha, f) = c(f) - f(\alpha)$.

The slack matrix encodes some combinatorial properties of $\mathcal{M}$, and we will see in the next section that certain properties of the slack matrix correspond to the existence of specific convex relaxations that (c, s)-approximately solve $\mathcal{M}$. In particular, we will see that the existence of SDP relaxations for $\mathcal{M}$ depends on certain factorizations of the slack matrix.

2.5 SDP Relaxations for Optimization Problems

A popular method for solving combinatorial optimization problems is to formulate them as SDPs, and use generic algorithms such as the Ellipsoid Method for solving SDPs.

Definition 2.5.1. Let $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ be a combinatorial maximization problem. Then an SDP relaxation of $\mathcal{M}$ of size d consists of

1. *SDP:* Constraints $\{A_i, b_i\}_{i=1}^{m}$ with $A_i \in \mathbb{R}^{d\times d}$ and $b_i \in \mathbb{R}$ and a set of affine objective functions $\{w^f \mid f \in \mathcal{F}\}$ with each $w^f : \mathbb{R}^{d\times d} \to \mathbb{R}$,

2. *Feasible Solutions:* A set $\{X^\alpha \mid \alpha \in S\}$ in the feasible region of the SDP satisfying $w^f(X^\alpha) = f(\alpha)$ for each f.

We say that the SDP relaxation is a (c, s)-approximate relaxation or that it achieves (c, s)-approximation if, for each $f \in \mathcal{F}$ with $\max_{\alpha\in\mathcal{S}} f(\alpha) \leq s(f)$,

$$\max_X \left\{ w^f(X) \mid \forall i : A_i \cdot X = b_i, X \in \mathbb{S}_+^d \right\} \leq c(f).$$

If the SDP relaxation achieves a $(\max_{\alpha\in\mathcal{S}} f(\alpha), \max_{\alpha\in\mathcal{S}} f(\alpha))$-approximation, we say it is *exact*. If $c(f) = \rho s(f)$, then we also say the SDP relaxation achieves a ρ-approximation.

Given a (c, s)-approximate SDP formulation for $\mathcal{M}$, we can (c, s)-approximately solve $\mathcal{M}$ on input f simply by solving the SDP $\max w^f(X)$ subject to $X \in \mathbb{S}_+^d$ and $\forall i : A_i(X) = b_i$.

Example 2.5.2. We can embed any polytope in n dimensions with d facets into the PSD cone of size $2n + d$ and get an exact SDP relaxation for the optimization problem that maximizes linear functions over the vertices of P. Let V be the vertices and (A, b) determine the facets of P, so that

$$P = \mathrm{conv}\{\alpha \mid \alpha \in V\} = \{x : Ax \leq b\},$$

where A is a $d \times n$ matrix. Then we can define new variables x_i^+ and x_i^- for each $i \in [n]$ and z_j for each $j \in [d]$. Let $\mathrm{diag}(()X, Y)$ denote the block-diagonal matrix whose blocks are X and Y. Then for any vector $l \in \mathbb{R}^n$, let $l' = \mathrm{diag}(a_i, -a_i, 0, 0, \dots, 1, 0, 0, \dots, 0)$, where the 1 is where the ith zero would be. In other words,

$$l' \cdot (x_1^+, x_2^+, \dots, x_n^+, x_1^-, x_2^-, \dots, x_n^-, z_1, z_2, \dots, z_d) = l \cdot (x^+ - x^-) + z_i.$$

Now for any vertex α of P, there is a $(x_\alpha^+, x_\alpha^-, z_\alpha)$ such that $(x_\alpha^+ - x_\alpha^-) = \alpha$. Thus the SDP $\mathrm{diag}(a_i') \cdot X = b_i$, $X = \mathrm{diag}(x^+, x^-, z)$, $X \succeq 0$ with feasible solutions $X^\alpha = \mathrm{diag}(x_\alpha^+, x_\alpha^-, z_\alpha)$ and objective functions $w^l(X) = \mathrm{diag}(l') \cdot X$ is an SDP relaxation for maximizing any linear function over the vertices of P. It is easy to see that it is exact.

Now that we have defined SDP relaxations, we wish to know when we can use them to get good approximations for combinatorial problems. This question has been studied for some time, originally in the context of linear relaxations. Yannakakis was able to prove a connection between the (c, s)-approximate slack matrix and the existence of (c, s)-approximate linear relaxations [72]. His work laid the foundation for a number of extensions to other kinds of convex relaxations. Here we give the generalization for the existence of (c, s)-approximate SDP relaxations.

Theorem 2.5.3 (Generalization of Yannakakis' Factorization Theorem)**.** *Let $\mathcal{M}$ be a combinatorial optimization problem with (c, s)-Slack Matrix $M(\alpha, f)$. There exists an (c, s)-approximate SDP relaxation of size d for $\mathcal{M}$ if and only if there exists $X^\alpha, Y_f \in \mathbb{S}_+^d$ and $\mu_f \in \mathbb{R}_+$ such that $X^\alpha \cdot Y_f + \mu_f = M(\alpha, f)$ for each $\alpha \in \mathcal{S}$ and $f \in \mathcal{F}$ with $\max_{\alpha \in \mathcal{S}} f(\alpha) \leq s(f)$. Such X^α and Y_f are called a* PSD factorization *of size d.*

(Proof from [9]). First, we prove that if $\mathcal{M}$ has such a size d relaxation, then M has a factorization of size at most d. Let $\{A_i, b_i\}$ be the constraints of the SDP, X^α be the feasible solutions, and w^f be the affine objective functions. We assume that there exists an X such that $A_i \cdot X = b_i$ and $X_i \succ 0$ is strictly feasible. Otherwise, the feasible region lies entirely on a face of $\mathbb{S}_+^d$, which itself is a PSD cone of smaller dimension, and we could take an SDP relaxation of smaller size. For an $f \in \mathcal{F}$ such that $\max_{\alpha \in \mathcal{S}} f(\alpha) \leq s(f)$, let $w^f(X)$ have maximum w^* on the feasible region of the SDP. By Lemma 2.2.4, there exists (y_f, Y_f) such that

$$w^* - w_f(X) = Y_f \cdot X - \sum_i (y_f)_i (A_i \cdot X - b_i).$$

Substituting X^α and adding $\mu_f = c(f) - w^* \geq 0$ (the inequality follows because the SDP relaxation achieves approximation (c, s)), we get

$$M(\alpha, f) = Y_f \cdot X^\alpha + \mu_f.$$

For the other direction, let $w^f(X) = c(f) - \mu_f - Y_f \cdot X$, let the X^α be the feasible solutions, and let the constraints be empty, so the SDP is simply $X \succeq 0$. Then for any f satisfying the soundness guarantee,

$$\max_{X \succeq 0} w^f(X) = c(f) - \mu_f - \min_{X \succeq 0} Y_f \cdot X = c(f) - \mu_f \leq c(f).$$

Clearly the X^α are feasible because they are PSD, and so we have constructed a (c, s)-approximate SDP relaxation. □

2.6 Polynomial Formulations, Theta Body and SOS SDP Relaxations

In this section we first define what a polynomial formulation for a combinatorial optimization problem $\mathcal{M}$ is, and then use that formulation to derive two families of SDP relaxations for $\mathcal{M}$: The Theta Body and Sum-of-Squares relaxations.

Definition 2.6.1. A *degree-d polynomial formulation on n variables* for a combinatorial optimization problem $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ is three sets of degree-d polynomials $\mathcal{P}, \mathcal{Q}, \mathcal{O} \subseteq \mathbb{R}[x_1, \ldots, x_n]_d$ and a bijection ϕ between $\mathcal{S}$ and $V(\mathcal{P}) \cap H(\mathcal{Q})$ such that for each $f \in \mathcal{F}$ and $\alpha \in \mathcal{S}$, there exists a polynomial $o^f \in \mathcal{O}$ with $o^f(\phi(\alpha)) = f(\alpha)$. We will often abuse notation by suppressing the bijection ϕ and writing α for both an element of $\mathcal{S}$ and the corresponding one in $V(\mathcal{P}) \cap H(\mathcal{Q})$. We call $\mathcal{P}$ the equality constraints, $\mathcal{Q}$ the inequality constraints, and $\mathcal{O}$ the objective polynomials. The polynomial formulation is called *Boolean* if $V(\mathcal{P}) \cap H(\mathcal{Q}) \subseteq \{0, 1\}^n$.

Example 2.6.2. Matching on n vertices has a degree two polynomial formulation on $\binom{n}{2}$ variables. Let

$$\mathcal{P} = \left\{x_{ij}^2 - x_{ij} \mid i, j \in [n]\right\} \cup \left\{\sum_i x_{ij} - 1 \mid j \in [n]\right\} \cup \left\{x_{ij}x_{ik} \mid i, j, k \in [n], j \neq k\right\}.$$

For a matching M, let $(\chi_M)_{ij} = 1$ if $(i, j) \in M$ and 0 otherwise. Then clearly $\phi(M) = \chi_M$ is a bijection, and it is easily verified that every $\chi_M \in V(\mathcal{P})$. Finally, for an objective function $f_E(M) = |M \cap E|$, we define $o^{f_E}(x) = \sum_{ij:(i,j) \in E} x_{ij}$.

A polynomial formulation for $\mathcal{M}$ defines a polynomial optimization problem: given input o^f,

$$\begin{aligned} &\max o^f(x) \\ \text{s.t. } & p(x) = 0, \forall p \in \mathcal{P} \\ & q(x) \geq 0, \forall q \in \mathcal{Q}. \end{aligned}$$

Solving this optimization problem is equivalent to solving the problem $\mathcal{M}$.

In Section 2.3 we discussed how polynomial optimization problems could sometimes be solved by searching for $\mathrm{PC}_{>}$ proofs of non-negativity. Furthermore, these proofs can be found using semidefinite programming. It should come as no surprise then that, given a polynomial formulation for $\mathcal{M}$, there are SDP relaxations based on finding certificates of non-negativity. The Theta Body relaxation, first considered in [29], is defined only for formulations without inequality constraints. It finds a certificate of non-negativity for $r(x)$ which is an SOS polynomial $s(x)$ together with a polynomial $g \in \langle \mathcal{P} \rangle$ such that $r(x) = s(x) + g(x)$.

Let $(\mathcal{P}, \mathcal{O}, \phi)$ be a degree-d polynomial formulation for $\mathcal{M}$. Recall that every polynomial p of degree at most $2d$ has a $d \times d$ matrix of coefficients $\widehat{p}$ such that $p(\alpha) = \widehat{p} \cdot \left(\mathbf{x}^{\otimes \mathbf{d}}(\alpha)(\mathbf{x}^{\otimes \mathbf{d}}(\alpha))^T\right)$. This includes the constant polynomial 1, whose matrix we denote $\hat{1}$.

Definition 2.6.3. For $D \geq d$, the *degree-D or Dth Theta-Body Relaxation* of $(\mathcal{P}, \mathcal{O}, \phi)$ is an SDP relaxation for $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ consisting of:

1. *Semidefinite program:*
   - $\widehat{p} \cdot X = 0$ for every $p \in \langle \mathcal{P} \rangle$ of degree at most $2D$,
   - $\widehat{1} \cdot X = 1$, and
   - $X \succeq 0$.
   - For each polynomial $o^f \in \mathcal{O}$, we define the affine function $w^f(X) = \widehat{o^f} \cdot X$.
2. Feasible solutions: For any $\alpha \in \mathcal{S}$, let $X^\alpha = \mathbf{x}^{\otimes \mathbf{D}}(\phi(\alpha))(\mathbf{x}^{\otimes \mathbf{D}}(\phi(\alpha)))^T$.

This definition of the Theta Body Relaxation makes it obvious that it is an SDP relaxation for $\mathcal{M}$, but we will frequently find it more convenient to work with the dual SDP. Working with the dual exposes the connection between the Theta Body relaxation and polynomial proof systems.

Lemma 2.6.4. *The dual of the Theta Body SDP Relaxation with objective function $\widehat{o^f} \cdot X$ can be expressed* $\min c$ *subject to $c - o^f(x)$ is $2D$-SOS modulo $\langle \mathcal{P} \rangle$.*

Proof. The dual is

$$\begin{aligned} &\min y_1 \\ \text{s.t. } \; & y_1 \cdot \widehat{1} - \widehat{o^f} = \widehat{s} + \sum_{p \in \langle P \rangle} y_p \cdot \widehat{p} \\ & \widehat{s} \succeq 0 \end{aligned}$$

The equality constraint of the dual is a constraint on matrices, but we can also think of it as a constraint on degree $2D$ polynomials via the map $\widehat{p} \leftrightarrow p$. Recall that $\widehat{s} \succeq 0$ if and only if s is a sum-of-squares polynomial. Thus this constraint is equivalent to asking that the polynomial $y_1 - o^f(x)$ be $2D$-SOS modulo $\langle \mathcal{P} \rangle$. $\square$

The Theta Body does not find PC$_>$ proofs of non-negativity, but instead finds a different kind of proof that uses any low degree $g \in \langle \mathcal{P} \rangle$. To write down the Dth Theta Body relaxation, we need to know all the degree-D polynomials in $\langle \mathcal{P} \rangle$. In particular, we need a basis of polynomials for the vector space of degree-D polynomials in $\langle \mathcal{P} \rangle$. To get our hands on this, we would need to be able to *at least* solve the membership problem for $\langle \mathcal{P} \rangle$ up to degree D. Unfortunately, this problem is frequently intractable, and so even trying to formulate the Dth Theta Body is intractable. We can define a weaker SDP relaxation which does not merely use an arbitrary $g \in \langle \mathcal{P} \rangle$, but provides a derivation for g from $\mathcal{P}$, i.e. it finds a PC$_>$ proof. More generally, even when $\mathcal{Q} \neq \emptyset$, we can define the Lasserre or SOS relaxations as follows:

Definition 2.6.5. Let $(\mathcal{P}, \mathcal{Q}, \mathcal{O}, \phi)$ be a degree-d polynomial formulation for $\mathcal{M} = (\mathcal{S}, \mathcal{F})$, and let $\mathcal{Q} = \{q_1, \ldots, q_k\}$. For $D \geq d$, the *degree-D or Dth Lasserre Relaxation* or *degree-D or Dth Sum-of-Squares Relaxation (SOS)* is an SDP relaxation for $\mathcal{M}$ consisting of:

1. Semidefinite program: For clarity, we define q_0 to be the constant polynomial 1, $D_i = D - \deg q_i$, and $N_i = \binom{n+D_i-1}{D_i}$. Let $\mathrm{diag}(M_1, M_2, \ldots, M_k)$ denote the block-diagonal matrix whose blocks are $M_1, \ldots, M_k$. Then the constraints are
   - $X = \mathrm{diag}(X_{q_0}, X_{q_1}, X_{q_2}, \ldots, X_{q_k})$, where X_{q_i} is an $N_i \times N_i$ matrix whose rows and columns are indexed by the monomials up to degree D_i.
   - For every i and pair of monomials x_U and x_V of degree at most D_i, $(X_{q_i})_{UV} = X_{q_0} \cdot \widehat{q_i x_U x_V}$. This implies that if $\tilde{c}$ is the vector of coefficients of a polynomial $c(x)$ of degree at most D_i and $X_{q_0} = \mathbf{x}^{\otimes \mathbf{D_0}}(\mathbf{x}^{\otimes \mathbf{D_0}})^T$, then $\tilde{c}^T X_i \tilde{c} = q(x)c(x)^2$.
   - For every $p \in \mathcal{P}$ and polynomial λ such that λp has degree at most $2d$, we have the constraint $\widehat{\lambda p} \cdot X_{q_0} = 0$.
   - $\widehat{1} \cdot X_{q_0} = 1$
   - $X \succeq 0$.
   - For each polynomial $o^f \in \mathcal{O}$, we define the affine objective function $w^f(X) = \widehat{o^f} \cdot X_{q_0}$.
2. Feasible solutions: For any $\alpha \in \mathcal{S}$, let $X^\alpha_{q_i} = \mathbf{x}^{\otimes \mathbf{D_i}}(\phi(\alpha))(\mathbf{x}^{\otimes \mathbf{D_i}}(\phi(\alpha)))^T q_i(\alpha)$ for each $0 \leq i \leq k$. Then let
$$X^\alpha = \mathrm{diag}(X^\alpha_{q_0}, X^\alpha_{q_1}, \ldots, X^\alpha_{q_k}).$$

Once again, we will find it much more convenient to work with the dual to make the connections to polynomial proof systems more explicit.

Lemma 2.6.6. *The dual of the degree $2D$ Sum-of-Squares SDP Relaxation with objective function $\widehat{o^f}$ can be expressed as $\min c$ subject to $c - o^f(x)$ has a degree $2D$ PC$_>$ proof of non-negativity from $\mathcal{P}$ and $\mathcal{Q}$.*

Proof. Recall we use $\mathrm{diag}(M_1, \dots, M_k)$ to denote the block-diagonal matrix whose blocks are $M_1, \dots, M_k$. Then the dual of the SOS relaxation is $\min y_1$ subject to $\hat{s} \succeq 0$ and

$$\mathrm{diag}(y_1 \cdot \widehat{1}, 0, \dots, 0) - \mathrm{diag}(\widehat{o^f}, 0, \dots, 0) = \widehat{s} + \sum_{\substack{p \in \mathcal{P} \\ \lambda}} y_{\lambda p} \cdot \mathrm{diag}(\widehat{\lambda p}, 0, \dots, 0) + \\ + \sum_{\substack{i \in [k] \\ U,V}} y_{iUV} \cdot \mathrm{diag}(\widehat{q_i x_U x_V}, 0, \dots, 0, \widehat{-x_U x_V}, 0, \dots, 0)$$

where in the last sum the second nonzero diagonal block is in the ith place. Clearly $\widehat{s}$ must be block-diagonal since everything else is block-diagonal. Furthermore, we know that the ith block of $\widehat{s}$ is equal to $\sum_{UV} \widehat{x_U x_V} y_{iUV}$ since the LHS is zero in every block but the first. Since $S \succeq 0$, this block must also be PSD, and thus must correspond to a sum-of-squares polynomial s_i. The constraint on the first block is then

$$y_1 \cdot \widehat{1} - \widehat{o^f} = \widehat{s_1} + \sum_{\substack{p \in \mathcal{P} \\ \lambda}} y_{\lambda p} \cdot \widehat{\lambda p} + \sum_{i=1}^{k} \widehat{s_i q_i}.$$

As a constraint on polynomials, this simply reads that $y_1 - o^f(x)$ must have a degree $2D$ $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}$ and $\mathcal{Q}$. □

Remark 2.6.7. The observant reader may notice that as presented, the Theta Body and SOS relaxations do not satisfy Slater's condition for strong duality (see Lemma 2.2.4), so it may not be valid to consider their duals instead of their primals. One can handle this by taking $\mathbf{x}^{\otimes \mathbf{D}}$ to be a basis for the low-degree elements of $\mathbb{R}[x_1, \dots, x_n]/\langle \mathcal{P} \rangle$, rather than a basis for every low-degree polynomial. Then [39] show that if $V(\mathcal{P}) \cap H(\mathcal{Q})$ is compact, there is no duality gap. From this point on we work exclusively with the duals, so we will not worry about it too much.

The Dth Theta Body and SOS relaxations are each relaxations of size $N = \binom{n+D-1}{D}$, since their feasible solutions have one coordinate for each monomial up to total degree D. For both hierarchies, it is clear that by projecting onto the coordinates up to degree $D' < D$, the feasible region of the D'th relaxation is contained in the feasible region of the D'th relaxation. Thus if the D'th relaxation achieves a (c, s)-approximation, so does the Dth. Furthermore, sometimes if we take the degree large enough, the relaxation becomes exact.

Lemma 2.6.8. *If the polynomial formulation is Boolean, then the nth Theta Body and nth SOS relaxation are both exact.*

Proof. Follows immediately from Fact 2.3.13 □

Relations Between Theta Body and Lasserre Relaxations

Here we compare and contrast the two different relaxations. Let $(\mathcal{P}, \mathcal{O}, \phi)$ be a polynomial formulation for $\mathcal{M} = (\mathcal{S}, \mathcal{F})$.

Lemma 2.6.9. *If the Dth Lasserre relaxation achieves (c, s)-approximation of $\mathcal{M}$, then the Dth Theta Body relaxation does as well.*

Proof. The lemma follows immediately by noticing that any degree $2D$ $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}$ for a polynomial $r(x)$ implies that $r(x)$ is $2D$-SOS modulo $\langle \mathcal{P} \rangle$. □

We can also prove a partial converse in some cases:

Proposition 2.6.10. *If $\mathcal{P}$ is k-effective and the Dth Theta Body relaxation achieves (c, s) approximation of $\mathcal{M}$, then the kDth Lasserre relaxation does as well.*

Proof. Because the Theta Body relaxation is a (c, s)-approximation of $\mathcal{M}$, we have that, for every $f \in \mathcal{F}$ with $\max f \leq s(f)$, there exists a number $c^* \leq c(f)$ such that $c^* - o^f(x)$ is $2D$-SOS modulo $\langle \mathcal{P} \rangle$. In other words, there is a polynomial identity $c^* - o^f(x) = s(x) + g(x)$, where s is an SOS polynomial and $g \in \langle \mathcal{P} \rangle$. Because $\mathcal{P}$ is k-effective, g has a degree $2kD$ derivation from $\mathcal{P}$, so we have a polynomial identity

$$c^* - o^f(x) = s(x) + \sum_{p \in \mathcal{P}} \lambda_p(x) p(x).$$

This implies that $(c^*, s(x), \lambda_p(x))$ are feasible solutions for the kDth Lasserre relaxation, and since $c^* \leq c(f)$, it achieves a (c, s)-approximation. □

Example 2.6.11. For CSP, $\mathcal{P} = \{x_i^2 - 1 \mid i \in [n]\}$. By Corollary 3.1.2, $\mathcal{P}$ is 1-effective. Thus the Dth Theta Body and Lasserre Relaxations are identical in this case.

Proposition 2.6.10 allows us to translate results about the Theta Body relaxations to Lasserre relaxations. In particular, in Chapter 5 we will see how easy it is to prove that Theta Body relaxations are optimal among symmetric relaxations of a given size. If the constraints are effective, this allows us to conclude that Lasserre relaxations which are not too much larger achieve the same guarantees. This allows us to lower bound the size of *any* symmetric SDP relaxation by finding lower bounds for Lasserre relaxations.

2.7 Symmetric Relaxations

Often, the solutions to a combinatorial optimization problem exhibit many symmetries. For example, in the Matching problem, a maximum matching of K_n is still a maximum matching even if the vertices are permuted arbitrarily. This additional structure allows for easier analysis. It is natural, then, to consider relaxations that exhibit similar symmetries. Rounding these relaxations is often more straightforward and intuitive. In this section we formally define what we mean by symmetric versions of all the problem formulations we have presented above. First, we recall some basic group theory.

Definition 2.7.1. Let G be a group and X be a set. We say G *acts on* X if there is a map $\phi : G \to (X \to X)$ satisfying $\phi(1)(x) = x$ and $\phi(g_1)(\phi(g_2)(x)) = \phi(g_1 g_2)(x)$. In practice we omit the ϕ and simply write gx for $\phi(g)(x)$.

Definition 2.7.2. Let G act on X. Then $\mathrm{Orbit}(x) = \{y \mid \exists g : gx = y\}$ is called the *orbit* of x, and $\mathrm{Stab}(x) = \{g \mid gx = x\}$ is called the *stabilizer* of x.

Fact 2.7.3 (Orbit-Stabilizer Theorem)**.** *Let G act on X. Then $|G : Stab(x)| = |Orbit(x)|$.*

We will use S_n to denote the symmetric group on n letters, and A_n for the alternating group on n letters. For $I \subseteq [n]$, we use $S([n] \setminus I)$ for the subgroup of S_n which stabilizes every $i \in I$, and similarly for $A([n] \setminus I)$.

Optimization problems often have natural symmetries, which we can represent by the existence of a group action.

Definition 2.7.4. A combinatorial optimization problem $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ is G-symmetric if there are actions of G on $\mathcal{S}$ and $\mathcal{F}$ such that, for each $\alpha \in \mathcal{S}$ and $f \in \mathcal{F}$, $gf(g\alpha) = f(\alpha)$.

Example 2.7.5 (Maximum Matching)**.** Let $\mathcal{M}$ be the Matching problem on n vertices from Example 2.4.2. For an element $g \in S_n$ and a matching M of K_n, let gM be the matching where $(i, j) \in gM$ if and only if $(g^{-1}i, g^{-1}j) \in M$. For a subset of edges E, let $gf_E(M) = f_{gE}(M)$, where $gE = \{(gi, gj) \mid (i, j) \in E\}$. Then $\mathcal{M}$ is S_n-symmetric under these actions.

Definition 2.7.6. An SDP relaxation $(\{X^\alpha\}, \{(A_i, b_i)\}, \{w^f\})$ for a G-symmetric problem $\mathcal{M}$ is G-symmetric if there is an action of G on $\mathbb{S}^d_+$ such that $gX^\alpha = X^{g\alpha}$ for every α, and $w^{gf}(gX) = w^f(X)$, and $A_i \cdot X = b$ for all i if and only if $A_i \cdot gX = b$ for all i. We say the relaxation is G-coordinate-symmetric if the action of G is by permutation of the coordinates, in other words G has an action on $[d]$ and $(gX)_{ij} = X_{gi,gj}$.

Example 2.7.7. The usual linear relaxation for the Matching problem on n vertices is

$$K = \left\{ x \in \mathbb{R}^{\binom{n}{2}} \,\middle|\, \forall i : \sum_j x_{ij} \le 1, \forall ij : 0 \le x_{ij} \le 1 \right\},$$

with objective functions $w^{f_E}(x) = \sum_{(i,j)\in E} x_{ij}$. This relaxation is S_n-coordinate-symmetric under the action $(g\alpha)_{ij} = \alpha_{gi,gj}$ for any $\alpha \in \mathbb{R}^{\binom{n}{2}}$. This action essentially represents the permutation of the vertices of the underlying graph. It is simple to confirm that this action satisfies the above requirements.

Asymmetric relaxations are harder to come by, since they are unintuitive to design. The above example has the nice interpretation that x_{ij} is a variable that is supposed to represent the presence of the edge (i, j) in the matching. Asymmetric relaxations do not have such simple interpretations. We do not have many examples of cases where asymmetry actually helps, but the reader can refer to [40] for one example for the MATCHING problem with only $\log n$ edges.

Definition 2.7.8. A polynomial formulation $(\mathcal{P}, \mathcal{Q}, \mathcal{O}, \phi)$ on n variables for a G-symmetric problem $\mathcal{M}$ is G-symmetric if there is an action of G on $[n]$, yielding an action on polynomials simply by extending $gx_i = x_{gi}$ multiplicatively and linearly, such that $gp \in \mathcal{P}$ for each $p \in \mathcal{P}$, $gq \in \mathcal{Q}$ for each $q \in \mathcal{Q}$, and $go \in \mathcal{O}$ for each $o \in \mathcal{O}$. Note that this implies that G fixes $\langle \mathcal{P} \rangle$ as well, and that the natural action of G on $\mathbb{R}^n$ also fixes $V(\mathcal{P}) \cap H(\mathcal{Q})$. Finally, we require $g\phi(\alpha) = \phi(g\alpha)$.

Lemma 2.7.9. *If a G-symmetric problem $\mathcal{M}$ has a G-symmetric polynomial formulation, then the Theta Body and SOS SDP relaxations are G-coordinate-symmetric.*

Proof. The Dth Theta Body and SOS SDP relaxations are determined by $N = \binom{n+D-1}{D}$ coordinates, one for every monomial up to degree D. We index the coordinates by these monomials. We define an action of G on $\mathbb{S}_+^N$ which permutes $[N]$ simply by its action on monomials inherited by the G-symmetric polynomial formulation. Under this action, for any polynomial p, we have $\hat{p} \cdot (gX) = \widehat{g^{-1}p} \cdot X$. Since $\langle \mathcal{P} \rangle, \mathcal{Q}, \mathcal{O}$ are fixed by G and $\widehat{g^{-1}1} = \hat{1}$, it is clear that the constraints and objective functions of both the Theta Body and SOS relaxations are invariant under G. The feasible solutions are also invariant:

$$gX^{\alpha} = \mathbf{x}^{\otimes \mathbf{d}}(g\phi(\alpha))(\mathbf{x}^{\otimes \mathbf{d}}(g\phi(\alpha)))^T = \mathbf{x}^{\otimes \mathbf{d}}(\phi(g\alpha))(\mathbf{x}^{\otimes \mathbf{d}}(\phi(g\alpha)))^T = X^{g\alpha},$$

concluding the proof. □

When we have a G-symmetric combinatorial optimization problem, it is sensible to write symmetric SDP relaxations for it. The structure and symmetries of the problem are reflected in the relaxation, and it can often be interpreted more easily.

Chapter 3

Effective Derivations

In this section we prove that many natural sets of polynomials $\mathcal{P}$ arising from polynomial formulations for combinatorial optimization problems are k-effective with k a constant. This means that the problem of determining if a polynomial p is in the ideal $\langle \mathcal{P} \rangle$ can be solved simply by solving a linear system with $n^{O(\deg p)}$ variables. This can be done in time polynomial in $n^{\deg p}$ and $\log \|p\|$, which is the size required to specify the input p. When $\mathcal{P}$ is complete, this is equivalent to determining if $p(\alpha) = 0$ for every $\alpha \in V(\mathcal{P})$. In Chapter 4 and Chapter 5 we will see that determining if $\mathcal{P}$ is k-effective has important consequences for studying the Sum-of-Squares relaxations on the polynomial optimization problem determined by the equality constraints $\mathcal{P}$.

3.1 Gröbner Bases

Recall the definition of Gröbner basis from Definition 2.3.2, and in particular recall that we can define a multivariate division algorithm with respect to a Gröbner basis such that every polynomial has a unique remainder. The following lemma is an obvious consequence of the division algorithm.

Lemma 3.1.1. *Let $\mathcal{P}$ be a Gröbner basis. Then $\mathcal{P}$ is 1-effective.*

Proof. Let $r \in \langle \mathcal{P} \rangle$ be of degree d, and consider the remainder of r after dividing by $\mathcal{P}$. Because $r \in \langle \mathcal{P} \rangle$, and remainders are unique, the only possible remainder is the zero polynomial. If we enumerate the polynomials that are produced by the iterative reductions $r = r_0, r_1, \dots, r_N = 0$, then $r_i = r_{i+1} + q_{i+1}p_{i+1}$, where $p_{i+1} \in \mathcal{P}$, $\deg r_{i+1} \leq \deg r_i$, and $\deg q_{i+1}p_{i+1} \leq \deg r_i$. Combining all these sums into one, we get $r = \sum_i q_i p_i$, which is a derivation of degree d. $\square$

Lemma 3.1.1 is unsurprising, as Gröbner bases first originated as a method to solve the polynomial ideal problem [12]. While Gröbner bases yield positive results, they are often unwieldy, complicated, and above all extremely expensive to compute. Even so, there are several important combinatorial optimization problems that have constraints which are Gröbner bases, one of which we used in Example 2.3.3.

Corollary 3.1.2. *The* CSP *formulation* $\mathcal{P}_{\text{CSP}} = \{x_i^2 - x_i \mid i \in [n]\}$ *is* 1*-effective.*

Proof. We prove that $\mathcal{P}_{\text{CSP}}$ is a Gröbner basis. Let $p \in \langle \mathcal{P}_{\text{CSP}} \rangle$. If p is not multilinear, we can divide p by elements of $\mathcal{P}_{\text{CSP}}$ until we have a multilinear remainder r. Because $p \in \langle \mathcal{P}_{\text{CSP}} \rangle$ and each element of $\mathcal{P}_{\text{CSP}}$ is zero on the hypercube $\{0,1\}^n$, r must also be zero on the hypercube. But the multilinear polynomials form a basis for functions on the hypercube, so if r is a multilinear polynomial which is zero, then it must be the zero polynomial. □

Corollary 3.1.3. *The* Clique *formulation* $\mathcal{P}_{\text{CLIQUE}} = \{x_i^2 - x_i \mid i \in V\} \cup \{x_i x_j \mid (i,j) \notin E\}$ *is* 1*-effective.*

Proof. We prove that $\mathcal{P}_{\text{CLIQUE}}$ is a Gröbner basis. Let $p \in \langle \mathcal{P}_{\text{CLIQUE}} \rangle$. If p is not multilinear, we can divide it until we have a multilinear remainder r_1. Now by dividing r_1 by the non-edge polynomials in the second part of $\mathcal{P}_{\text{CLIQUE}}$, we can remove all monomials containing $x_i x_j$ where $(i,j) \notin E$ to get r_2. Thus r_2 contains only monomials which are cliques of varying sizes in the graph (V, E). Let C be the smallest clique with a nonzero coefficient r_C in r_2. Let χ_C be the characteristic vector of C, i.e. $(\chi_C)_i = 1$ if $i \in C$, and $(\chi_C)_i = 0$ otherwise. Then $r_2(\chi_C) = r_C$. But $p(\chi_C) = 0$ for every $p \in \mathcal{P}$, and $r_2 \in \langle \mathcal{P} \rangle$. Thus $r_C = 0$, a contradiction, and so r_2 is the zero polynomial. □

Of course, not every problem is so neatly described by a small Gröbner basis. There are many natural problems whose solution spaces have a small set of generating polynomials which are not Gröbner bases, and indeed their Gröbner basis can be exponentially. Even though the generating polynomials are not a Gröbner basis, they can still be k-effective for constant k, and thus admit a good algorithm for membership.

3.2 Proof Strategy for Symmetric Solution Spaces

In this section we describe our main proof strategy to show that a set of polynomials $\mathcal{P}$ is effective. We apply this strategy to combinatorial optimization problems which have a natural symmetry to their solution spaces $V(\mathcal{P})$. For each of these problems, we will define an S_m-action on $[n]$, which extends to an action on $\mathbb{R}[x_1, \ldots, x_n]$ as well as $\mathbb{R}^n$ by permutation of variable names and indices respectively. The action will be a natural permutation of the solutions. For example, for Matching, the group action will correspond to simply permuting the vertices of the graph.

After the group action is defined, our proof strategy follows in three steps:

(1) Prove that $\mathcal{P}$ is complete. This is usually done by exhibiting a degree n derivation from $\mathcal{P}$ for any polynomial p which is zero on $V(\mathcal{P})$. This step is essential for the induction in step (3).

(2) Prove that for every $p \in \langle \mathcal{P} \rangle$, the polynomial $\frac{1}{n!}\sum_{\sigma \in S_m} \sigma p$ has a derivation from $\mathcal{P}$ in degree $\deg p$. This is usually fairly easy because of the large amount of structure this symmetrization forces on the polynomial.

(3) Prove that for every $\sigma \in S_m$, $p - \sigma p$ has a derivation from $\mathcal{P}$ in degree at most $k \deg p$, for some constant k. This is performed by induction on a natural parameter of the combinatorial optimization problem. Generally speaking, the more complicated the solution space for the problem, the worse bounds we can prove for the constant k.

We use this general strategy to prove that many polynomial formulations for different natural combinatorial optimization problems admit effective derivations. Our efforts to find a unifying theory that explains the effectiveness of this strategy on the different problems have failed, so we have to prove that each set of polynomials is effective on a case-by-case basis.

3.3 Effective Derivations for Matching

Fix an even integer n. Then the Matching problem has a polynomial formulation on $\binom{n}{2}$ variables with constraints

$$\begin{aligned} \mathcal{P}_{\mathrm{M}}(n) = &\left\{x_{ij}^2 - x_{ij} \mid i, j \in [n]\right\} \\ &\cup \left\{\sum_i x_{ij} - 1 \mid j \in [n]\right\} \\ &\cup \left\{x_{ij}x_{ik} \mid i, j, k \in [n], j \neq k\right\}. \end{aligned} \tag{3.1}$$

The first group of polynomials ensures the variables are Boolean. The second group of polynomials ensures each vertex is matched to another vertex. The last group of polynomials ensures that each vertex is not matched to multiple vertices. We abuse notation slightly and use x_{ij} and x_{ji} equivalently. We omit the dependence on n when it is clear from context. For an element σ of the symmetric group S_n, we define the action of σ on a variable by $\sigma x_{ij} = x_{\sigma(i)\sigma(j)}$. We define the action of σ on a monomial by extending this action multiplicatively, and the action of σ on a full polynomial by extending linearly. Note that $\mathcal{P}_{\mathrm{M}}$ is fixed by the action of every σ, as are its solutions $V(\mathcal{P}_{\mathrm{M}})$ corresponding to the matchings of K_n. Thus for any $p \in \mathcal{P}_{\mathrm{M}}$, we also have $\sigma p \in \mathcal{P}_{\mathrm{M}}$. For a partial matching M, i.e. a set of disjoint pairs from $[\binom{n}{2}]$, define $x_M = \prod_{e \in M} x_e$ with the convention that $x_\emptyset = 1$. First, we note an easy lemma on the structure of polynomials in $\langle \mathcal{P}_{\mathrm{M}} \rangle$:

Lemma 3.3.1. *Let p be any polynomial. Then there is a multilinear polynomial q such that every monomial of q is a partial matching monomial, and $p - q \in \langle \mathcal{P}_{\mathrm{M}} \rangle_{\deg p}$.*

Proof. It suffices to prove the lemma when p is a monomial. Let $p = \prod_{e \in A} x_e^{k_e}$ for a set A of edges with multiplicities $k_e \geq 1$. From the constraint $x_e^2 - x_e$, it follows that $\prod_{e \in A} x_e^{k_e} - C \prod_{e \in A} x_e$ has a derivation from $\mathcal{P}_{\mathrm{M}}$ in degree $\deg p$, for some constant C. Now if A is a partial matching we are done, otherwise there exist edges $f, g \in A$ which are not disjoint. But then $x_f x_g \in \mathcal{P}_{\mathrm{M}}$, and so $\prod_{e\in A} x_e$ has a derivation from $\mathcal{P}_{\mathrm{M}}$ in degree $|A|$, which implies the statement. $\square$

With Lemma 3.3.1 in hand, we complete step (1) of our strategy:

Lemma 3.3.2. *$\langle \mathcal{P}_{\mathrm{M}}(n)\rangle$ is complete for any even n.*

Proof. Let p be a polynomial such that $p(\alpha) = 0$ for each $\alpha \in V(\mathcal{P}_{\mathrm{M}})$. By Lemma 3.3.1, we can assume that $p(x)$ is a multilinear polynomial whose monomials correspond to partial matchings. For such a partial matching M, clearly $x_M - x_M \prod_{u\notin M}\sum_v x_{uv}$ has a derivation in degree n using the constraints $\sum_v x_{uv} - 1 \in \mathcal{P}_{\mathrm{M}}$. By eliminating terms which do not correspond to partial matchings, we get $x_M - \sum_{M':M\subset M'} x_{M'} \in \langle \mathcal{P}_{\mathrm{M}}\rangle$. Doing this to every monomial, we determine there is a polynomial p' which is homogeneous of degree n such that $p - p' \in \langle \mathcal{P}_{\mathrm{M}}\rangle$. Now since the coefficients of p' correspond exactly to perfect matchings, for each monomial in p', there is an $\alpha \in V(\mathcal{P}_{\mathrm{M}})$ such that the coefficient of the monomial is $p'(\alpha)$. Since $p'(\alpha) = 0$ for every $\alpha \in V(\mathcal{P}_{\mathrm{M}})$, it must be that $p' = 0$, and so $p \in \langle \mathcal{P}_{\mathrm{M}}\rangle$. $\square$

Now we move on to the second step of our proof.

Symmetric Polynomials

We will prove the following lemma:

Lemma 3.3.3. *Let p be a polynomial in $\mathbb{R}^{\binom{n}{2}}$. Then there is a constant c_p such that $\sum_{\sigma\in\mathcal{S}_n} \sigma p - c_p \in \langle \mathcal{P}_{\mathrm{M}}\rangle_{\deg p}$.*

To do so, it will be useful to first prove a few lemmas on how we can simplify the structure of p. Any partial matching monomial may be extended as a sum over partial matching monomials containing that partial matching using the constraint $\sum_j x_{ij} - 1 \in \mathcal{P}_{\mathrm{M}}$, as we did in the proof of Lemma 3.3.2. The first lemma here shows how to extend by a single edge, and the second iteratively applies this process to extend by multiple edges.

Lemma 3.3.4. *For any partial matching M on $2d$ vertices and a vertex u not covered by M,*

$$x_M \cong_{d+1} \sum_{\substack{M_1 = M\cup\{i,j\}:\\ j\in[n]\setminus(M\cup\{i\})}} x_{M_1}. \tag{3.2}$$

Proof. We use the constraints $\sum_v x_{ij} - 1$ to add variables corresponding to edges at u, and then use $x_{uv}x_{uw}$ to remove monomials not corresponding to a partial matching:

$$x_M \cong x_M \sum_{v\in K_n} x_{ij} \cong \sum_{\substack{M_1 = M\cup\{i,j\}:\\ j\in K_n\setminus(M\cup\{i\})}} x_{M_1}.$$

It is easy to see that these derivations are done in degree $d+1$. $\square$

Lemma 3.3.5. *For any partial matching M of $2d$ vertices and $d \le k \le n/2$, we have*

$$x_M \cong_k \frac{1}{\binom{n/2-d}{k-d}} \sum_{\substack{M' \supset M \\ |M'|=k}} x_{M'} \tag{3.3}$$

Proof. We use induction on $k-d$. The start of the induction is with $k=d$, when the sides of (3.3) are actually equal. If $k>d$, let u be a fixed vertex not covered by M. Applying Lemma 3.3.4 to M and u followed by the inductive hypothesis gives:

$$\begin{aligned} x_M &\cong_{d+1} \sum_{\substack{M_1=M\cup\{i,j\}: \\ j\in K_n\setminus(M\cup\{i\})}} x_{M_1} \\ &\cong_k \frac{1}{\binom{n/2-d-1}{k-d-1}} \sum_{\substack{M'\supset M_1 \\ |M'|=k \\ M_1=M\cup\{i,j\}: \\ j\in K_n\setminus(M\cup\{i\})}} x_{M'}. \end{aligned}$$

Averaging over all vertices i not covered by M, we obtain:

$$\begin{aligned} x_M &\cong_k \frac{1}{n-2d}\frac{1}{\binom{n/2-d-1}{k-d-1}} \sum_{\substack{M'\supset M_1 \\ |M'|=k \\ M_1=M\cup\{i,j\}: \\ \{i,j\}\in K_n\setminus M}} x_{M'} \\ &= \frac{1}{n-2d}\frac{1}{\binom{n/2-d-1}{k-d-1}} 2(k-d) \sum_{\substack{M'\supset M \\ |M'|=k}} x_{M'}. \\ &= \frac{1}{\binom{n/2-d}{k-d}} \sum_{\substack{M'\supset M \\ |M'|=k}} x_{M'} \end{aligned}$$

where in the second step the factor $2(k-d)$ accounts for the different choices of $\{i,j\}$ that can lead to extending M to M'. □

Finally, we can prove the first main lemma:

Proof of Lemma 3.3.3. Given Lemma 3.3.1, it suffices to prove the claim for $p = x_M$ for some partial matching M. Let $\deg p = |M| = k$. Note that S_n acts transitively on the monomials of degree k, and thus by the Orbit-Stabilizer theorem, $2^k k!(n-2k)!$ elements of S_n stabilize p. Thus $\sum_{\sigma\in S_n} \sigma x_M = 2^k k!(n-2k)! \sum_{M':|M'|=k} x_{M'}$. Finally, apply Lemma 3.3.5 with $d=0$:

$$\begin{aligned} \sum_{\sigma\in S_n} \sigma x_M &= 2^k k!(n-2k)! \sum_{M':\,|M'|=k} x_{M'} \\ &\cong_k 2^k k!(n-2k)! \binom{n/2}{k}. \end{aligned}$$

□

Corollary 3.3.6. *If $p \in \langle \mathcal{P}_{\mathrm{M}} \rangle$, then $\sum_{\sigma \in S_n} \sigma p$ has a derivation from $\mathcal{P}_{\mathrm{M}}$ in degree $\deg p$.*

Proof. Apply Lemma 3.3.3 to obtain a constant c_p such that $\sum_{\sigma \in \mathcal{S}_n} \sigma p \cong c_p$. Now since $p \in \langle \mathcal{P}_{\mathrm{M}} \rangle$, $c_p \in \langle \mathcal{P}_{\mathrm{M}} \rangle$ as well. But the only constant polynomial in $\langle \mathcal{P}_{\mathrm{M}} \rangle$ is 0 by Lemma 2.3.7. □

Getting to a Symmetric Polynomial

Now by Lemma 2.3.8 and Lemma 3.3.3, it suffices to exhibit a derivation of the difference polynomial $p - \sum_{\sigma \in S_n} \sigma p$ from $\mathcal{P}_{\mathrm{M}}$ in low degree. Our proof will be by an induction on the number of vertices n. Because the number of vertices will be changing in this section, we will stop omitting the dependence on n. The next lemma will allow us to apply induction:

Lemma 3.3.7. *Let $L \in \langle \mathcal{P}_{\mathrm{M}}(n) \rangle_d$. Then $L \cdot x_{n+1,n+2} \in \langle \mathcal{P}_{\mathrm{M}}(n+2) \rangle_{d+1}$.*

Proof. It suffices to prove the statement for $L \in \mathcal{P}_{\mathrm{M}}(n)$. If $L = x_{ij}^2 - x_{ij}$ or $L = x_{ij}x_{ik}$, the claim is clearly true because $L \in \mathcal{P}_{\mathrm{M}}(n+2)$. So consider $L = \sum_j x_{ij} - 1$ for some $i \in [n]$, and note that

$$\begin{aligned} L \cdot x_{n+1,n+2} - \left(\sum_{j=1}^{n+2} x_{ij} - 1 \right) x_{n+1,n+2} &= -x_{i,n+1}x_{n+1,n+2} - x_{i,n+2}x_{n+1,n+2} \\ &\cong_2 0. \end{aligned}$$

□

We are now ready to prove the main theorem of this section.

Theorem 3.3.8. *Let $p \in \langle \mathcal{P}_{\mathrm{M}}(n) \rangle$, and let $d = \deg p$. Then p has a derivation from $\mathcal{P}_{\mathrm{M}}(n)$ in degree $2d$.*

Proof. By Lemma 3.3.1, we can assume without loss of generality that p is a multilinear polynomial whose monomials correspond to partial matchings. As promised, our proof is by induction on n. Consider the base case of $n = 2$. Then $V(\mathcal{P}_{\mathrm{M}}(2)) = \{1\}$ and since there is only one variable, either p is a constant or linear polynomial. The only such polynomials that are zero on $V(\mathcal{P}_{\mathrm{M}}(2))$ are 0 and scalar multiples of $x_{12} - 1$. The former case has the trivial derivation, and the latter case is simply an element of $\mathcal{P}_{\mathrm{M}}(2)$.

Now assume that for any d, the theorem statement holds for polynomials in $\langle \mathcal{P}_{\mathrm{M}}(n') \rangle$ for any $n' < n$. Let $p \in \langle \mathcal{P}_{\mathrm{M}}(n) \rangle$ be multilinear of degree d whose monomials correspond to partial matchings, and let $\sigma = (i, j)$ be a transposition of two vertices. We consider the polynomial $\Delta = p - \sigma p$. Note that $\Delta \in \langle \mathcal{P}_{\mathrm{M}}(n) \rangle$, and any monomial which does not match either i or j, or a monomial which matches i to j, will not appear in Δ as it will be canceled by the subtraction. Thus we can write

$$\Delta = \sum_{e : i \in e \text{ or } j \in e} L_e x_e,$$

with each L_e having degree at most $d-1$. Our goal is to remove two of the variables in these matchings in order to apply induction. In order to do that, we will need each term to depend not only on either i or j, but both. To this end, we multiply each term by the appropriate polynomial $\sum_k x_{ik}$ or $\sum_k x_{jk}$ (recall that $\sum_k x_{ik} - 1 \in \mathcal{P}(n)$) to obtain

$$\Delta \cong_{d+1} \sum_{k_1 k_2} L_{k_1 k_2} x_{ik_1} x_{jk_2}.$$

We can think of the RHS polynomial as being a partition over the possible different ways to match i and j. Furthermore, because of the elements of $\mathcal{P}$ of type $x_{ij}x_{ik}$, we can take $L_{k_1k_2}$ to be independent of x_e for any e incident to any of i, j, k_1, k_2. We argue that $L_{k_1k_2} \in \mathcal{P}_{\mathrm{M}}(n-4)$. We know that $\Delta(\alpha) = 0$ for any $\alpha \in V(\mathcal{P}_{\mathrm{M}}(n))$. Let $\alpha \in V(\mathcal{P}_{\mathrm{M}}(n))$ such that $\alpha_{ik_1} = 1$ and $\alpha_{jk_2} = 1$. Then it must be that $\alpha_{ik} = 0$ and $\alpha_{jk} = 0$ for any other k, since otherwise $\alpha \notin V(\mathcal{P}_{\mathrm{M}}(n))$. Thus $\Delta(\alpha) = L_{k_1k_2}(\alpha)$. Since $L_{k_1k_2}$ is independent of any edge incident to i, j, k_1, k_2, it does not involve those variables, so $L_{k_1k_2}(\alpha) = L_{k_1k_2}(\beta)$, where β is the restriction of α to the $\binom{n-4}{2}$ variables which $L_{k_1k_2}$ depends on. But such a β is simply an element of $V(\mathcal{P}_{\mathrm{M}}(n-4))$, and all elements of $V(\mathcal{P}_{\mathrm{M}}(n-4))$ can be obtained this way. Thus $L_{k_1k_2}$ is zero on all of $V(\mathcal{P}_{\mathrm{M}}(n-4))$, and by Lemma 3.3.2, $L_{k_1k_2} \in \langle \mathcal{P}_{\mathrm{M}}(n-4) \rangle$. Now by the inductive hypothesis, $L_{k_1k_2}$ has a derivation from $\mathcal{P}_{\mathrm{M}}(n-4)$ of degree at most $2d-2$. By two applications of Lemma 3.3.7, $L_{k_1k_2}x_{ik_1}x_{jk_2}$ has a derivation from $\mathcal{P}_{\mathrm{M}}(n)$ of degree at most $2d$, and thus so does Δ.

Because transpositions generate the symmetric group, the above argument implies that $p - \frac{1}{n!}\sum_{\sigma \in S_n} \sigma p$ has a derivation from $\mathcal{P}_{\mathrm{M}}(n)$ of degree at most $2d$. Combined with Corollary 3.3.6, this is enough to prove the theorem statement. □

3.4 Effective Derivations for TSP

For each integer n, a polynomial formulation with n^2 variables for TSP on n vertices uses the following polynomials:

$$\begin{aligned} \mathcal{P}_{\mathrm{TSP}}(n) = & \left\{ x_{ij}^2 - x_{ij} \mid i, j \in [n] \right\} \\ & \cup \left\{ \sum_i x_{ij} - 1 \mid j \in [n] \right\} \\ & \cup \left\{ x_{ij}x_{ik}, x_{ji}x_{ki} \mid i, j, k \in [n], j \neq k \right\}. \end{aligned} \tag{3.4}$$

The first group of polynomials ensures the variables are Boolean, the second group of polynomials ensures that each city i is visited at some point in the tour, and the last set of polynomials ensures that no city is visited multiple times in the tour. A tour $\tau \in S_n$ (which is a feasible solution for TSP) is identified with the vector $\chi_\tau(i,j) = 1$ if $\tau(i) = j$ and 0 otherwise. We omit the dependence on n if it is clear from context. For an element σ of the symmetric group S_n, we define the action of σ on a variable by $\sigma x_{ij} = x_{\sigma(i)j}$. We define the action of σ on a monomial by extending this action multiplicatively, and the action of σ on a full polynomial by extending linearly. Then $\mathcal{P}_{\mathrm{TSP}}$ is fixed by the action of every σ, as are its solutions $V(\mathcal{P}_{\mathrm{TSP}})$ corresponding to the tours.

Note that $V(\mathcal{P}_{\mathrm{TSP}})$ corresponds to a matching on $K_{n,n}$, the complete bipartite graph on $2n$ vertices. Thus it should come as no surprise that the same proof strategy as the one we used for matchings on the complete graph K_n should work just fine. This section will be extremely similar to the previous one, and the reader loses very little by skipping ahead to Section 3.5. It would be more elegant if we could just reduce $\mathcal{P}_{\mathrm{TSP}}(n)$ to $\mathcal{P}_{\mathrm{M}}(2n)$. This requires proving that any polynomial which is zero on $V(\mathcal{P}_{\mathrm{TSP}}(n))$ is the projection of a polynomial of similar degree which is zero on $V(\mathcal{P}_{\mathrm{M}}(2n))$. Unfortunately we do not know how to prove this except by proving that $\mathcal{P}_{\mathrm{TSP}}$ is effective, so the reader will have to live with some repetition.

For a partial matching M of $K_{n,n}$, i.e. a set of disjoint pairs from $[n] \times [n]$, define $x_M = \prod_{e \in M} x_e$ with the convention that $x_\emptyset = 1$. We also define $M_L = \{i \in [n] \mid \exists j : (i,j) \in M\}$ and $M_R = \{j \in [n] \mid \exists i : (i,j) \in M\}$.

Lemma 3.4.1. *Let p be any polynomial. Then there is a multilinear polynomial q such that every monomial of q is a partial matching monomial, and $p - q$ has a derivation from $\mathcal{P}$ of degree* $\deg p$.

Proof. The statement follows easily by using the elements of $\mathcal{P}_{\mathrm{TSP}}$ of the form $x_{ij}^2 - x_{ij}$ to make a multilinear polynomial, then eliminating any monomial which is not a partial matching by using elements of the form $x_{ij}x_{ik}$ or $x_{ji}x_{ki}$. □

With Lemma 3.4.1 in hand, we prove the following easy result:

Lemma 3.4.2. *$\langle \mathcal{P}_{\mathrm{TSP}}(n) \rangle$ is complete for any n.*

Proof. Let p be a polynomial such that $p(\alpha) = 0$ for each $\alpha \in V(\mathcal{P}_{\mathrm{TSP}})$. By Lemma 3.4.1, we can assume that $p(x)$ is a multilinear polynomial whose monomials correspond to partial matchings. For such a partial matching M, clearly $x_M - x_M \prod_{i \notin M} \sum_j x_{ij}$ has a derivation in degree n using the constraints $\sum_j x_{ij} - 1 \in \mathcal{P}_{\mathrm{TSP}}$. By eliminating terms which do not correspond to partial matchings, we get $x_M - C_M \sum_{M' : M \subset M'} x_{M'} \in \langle \mathcal{P} \rangle$, for some constant C_M. Doing this to every monomial, we determine there is a polynomial p' which is homogeneous of degree n such that $p - p' \in \langle \mathcal{P} \rangle$. Now since the monomials of p' correspond to perfect matchings, each monomial has an α such that the coefficient of that monomial is exactly $p'(\alpha)$. Since $p'(\alpha) = 0$ for every $\alpha \in V(\mathcal{P}_{\mathrm{TSP}})$, it must be that $p' = 0$, and so $p \in \langle \mathcal{P}_{\mathrm{TSP}} \rangle$. □

Now we move on to the second step of our proof.

Symmetric Polynomials

We will complete this step of our proof using the same helper lemmas as for Matching. The numbers appearing are slightly different due to the difference in the number of partial matchings for K_n and $K_{n,n}$, and the action of S_n is slightly different, but they are all basically the same.

Lemma 3.4.3. *For any partial matching M on $2d$ vertices and a vertex $i \in [n] \setminus M_L$,*

$$x_M \cong \sum_{\substack{M_1 = M \cup \{i,j\}: \\ j \in [n] \setminus (M_R)}} x_{M_1}, \tag{3.5}$$

and the derivation can be done in degree $d+1$.

Proof. We use the constraints $\sum_v x_{uv} - 1$ to add variables corresponding to edges at u, and then use $x_{uv}x_{uw}$ to remove monomials not corresponding to a partial matching:

$$x_M \cong x_M \sum_{j \in [n]} x_{ij} \cong \sum_{\substack{M_1 = M \cup \{i,j\}: \\ j \in [n] \setminus M_R}} x_{M_1}.$$

It is easy to see that these derivations are done in degree $d+1$. □

Lemma 3.4.4. *For any partial matching M of $2d$ vertices and $d \le k \le n$, we have*

$$x_M \cong \frac{1}{\binom{n-d}{k-d}} \sum_{\substack{M' \supset M \\ |M'| = k}} x_{M'} \tag{3.6}$$

Proof. We use induction on $k-d$. The start of the induction is with $k = d$, when the sides of (3.6) are actually equal. If $k > d$, let i be a fixed vertex not in M_L. Applying Lemma 3.4.3 to M and i followed by the inductive hypothesis gives:

$$\begin{aligned} x_M &\cong \sum_{\substack{M_1 = M \cup \{i,j\}: \\ j \in [n] \setminus M_R}} x_{M_1} \\ &\cong \frac{1}{\binom{n-d-1}{k-d-1}} \sum_{\substack{M' \supset M_1 \\ |M'| = k \\ M_1 = M \cup \{i,j\}: \\ j \in [n] \setminus M_R}} x_{M'}. \end{aligned}$$

Averaging over all vertices i not in M_L, we obtain:

$$\begin{aligned} x_M &\cong \frac{1}{n-d} \frac{1}{\binom{n-d-1}{k-d-1}} \sum_{\substack{M' \supset M_1 \\ |M'| = k \\ M_1 = M \cup \{i,j\}: \\ \{i,j\} \in [n] \times [n] \setminus M}} x_{M'} \\ &= \frac{1}{n-d} \frac{1}{\binom{n-d-1}{k-d-1}} (k-d) \sum_{\substack{M' \supset M \\ |M'| = k}} x_{M'}. \\ &= \frac{1}{\binom{n-d}{k-d}} \sum_{\substack{M' \supset M \\ |M'| = k}} x_{M'} \end{aligned}$$

where in the second step the factor $(k-d)$ accounts for the different choices of $\{i,j\}$ that can lead to extending M to M'. $\square$

Lemma 3.4.5. *Let p be a polynomial in $\mathbb{R}^{n^2}$. Then there is a constant c_p such that $\sum_{\sigma\in S_n}\sigma p - c_p$ has a derivation from $\mathcal{P}_{\mathrm{TSP}}$ in degree at most $\deg p$.*

Proof. Given Lemma 3.4.1, it suffices to prove the claim for $p = x_M$ for some partial matching M. Let $\deg p = |M| = k$. There are $(n-k)!$ elements of S_n that stabilize a given partial matching M, so $\sum_{\sigma\in S_n}\sigma x_M = (n-k)!\sum_{M':|M'|=k} x_{M'}$. Finally, apply Lemma 3.4.4 with $d=0$:

$$\begin{aligned}\sum_{\sigma\in S_n}\sigma x_M &= (n-k)!\sum_{M' :\, |M'|=k} x_{M'}\\ &\cong (n-k)!\binom{n}{k}.\end{aligned}$$

$\square$

Corollary 3.4.6. *If $p\in\langle\mathcal{P}_{\mathrm{TSP}}\rangle$, then $\sum_{\sigma\in S_n}\sigma p$ has a derivation from $\mathcal{P}_{\mathrm{TSP}}$ in degree $\deg p$.*

Proof. Apply Lemma 3.4.5 to obtain a constant c_p such that $\sum_{\sigma\in S_n}\sigma p\cong c_p$. Now since $p\in\langle\mathcal{P}_{\mathrm{TSP}}\rangle$, $c_p\in\langle\mathcal{P}_{\mathrm{TSP}}\rangle$ as well. But by Lemma 2.3.7, the only constant polynomial in $\langle\mathcal{P}_{\mathrm{TSP}}\rangle$ is 0. $\square$

Getting to a Symmetric Polynomial

The third step also proceeds in an almost identical manner.

Lemma 3.4.7. *Let L be a polynomial with a degree d derivation from $\mathcal{P}_{\mathrm{TSP}}(n)$. Then $Lx_{n+1,n+2}x_{n+2,n+1}$ has a degree $d+2$ derivation from $\mathcal{P}_{\mathrm{TSP}}(n+2)$.*

Proof. It suffices to prove the statement for $L\in\mathcal{P}_{\mathrm{TSP}}(n)$. If $L = x_{ij}^2 - x_{ij}$, $L = x_{ij}x_{ik}$, or $L = x_{ji}x_{ki}$, the claim is clearly true because $L\in\mathcal{P}_{\mathrm{TSP}}(n+2)$. So consider $L=\sum_j x_{ij}-1$ for some i, and note that

$$\begin{aligned}Lx_{n+1,n+2}x_{n+2,n+1} - \left(\sum_{j=1}^{n+2} x_{ij} - 1\right)x_{n+1,n+2}x_{n+2,n+1} &= (x_{i,n+1}+x_{i,n+2})\,x_{n+1,n+2}x_{n+2,n+1}\\ &= (x_{i,n+1}x_{n+2,n+1})\,x_{n+1,n+2} + (x_{i,n+2}x_{n+1,n+2})\,x_{n+2,n+1}\\ &\cong_3 0\end{aligned}$$

The case for $L=\sum_i x_{ij}-1$ is symmetric. $\square$

We are now ready to prove the main theorem of this section.

Theorem 3.4.8. *Let $p \in \langle \mathcal{P}_{\text{TSP}}(n) \rangle$ for any n, and let $d = \deg p$. Then p has a derivation from $\mathcal{P}_{\text{TSP}}(n)$ in degree $2d$.*

Proof. By Lemma 3.4.1, we can assume that p is a multilinear polynomial whose monomials correspond to partial matchings on $K_{n,n}$. As before, our proof is by induction on n. Consider the base case of $n = 1$. Then $V(\mathcal{P}_{\text{TSP}}(1)) = \{1\}$ and either p is a constant or linear polynomial (since there is only one variable, x_{11}). The only such polynomials that are zero on $V(\mathcal{P}_{\text{TSP}}(1))$ are 0 and scalar multiples of $x_{11} - 1$. The former case has the trivial derivation, and the latter case is simply an element of $\mathcal{P}_{\text{TSP}}(1)$.

Now assume that for any d, the theorem statement holds for polynomials in $\langle \mathcal{P}_{\text{TSP}}(n') \rangle$ for any $n' < n$. Let $p \in \langle \mathcal{P}_{\text{TSP}}(n) \rangle$ be multilinear of degree d whose monomials correspond to partial matchings, and let $\sigma = (i, j)$ be a transposition of two left indices. We consider the polynomial $\Delta = p - \sigma p$. Note that $\Delta \in \langle \mathcal{P}_{\text{TSP}}(n) \rangle$, and any monomial which does not match either i or j will not appear in Δ as it will be canceled by the subtraction. Thus we can write

$$\Delta = \sum_{e : e = (i,k) \text{ or } e = (j,k)} L_e x_e,$$

with each L_e having degree at most $d - 1$. Proceeding as before, we multiply each term by the appropriate constraint $\sum_k x_{ik}$ or $\sum_j x_{jk}$ to obtain a decomposition

$$\Delta \cong_{d+1} \sum_{k_1, k_2} L_{k_1 k_2} x_{ik_1} x_{jk_2}.$$

We can think of the RHS polynomial as being a partition over the possible different ways to match i and j. Furthermore we can take $L_{k_1 k_2}$ to be independent of x_e for any e incident to any of i, j, k_1, k_2. We argue that $L_{k_1 k_2} \in \mathcal{P}_{\text{TSP}}(n-2)$. We know that $\Delta(\alpha) = 0$ for any $\alpha \in V(\mathcal{P}_{\text{TSP}}(n))$. Let $\alpha \in V(\mathcal{P}_{\text{TSP}}(n))$ such that $\alpha_{ik_1} = 1$ and $\alpha_{jk_2} = 1$. Then it must be that $\alpha_{ik} = 0$ and $\alpha_{jk} = 0$ for any other k, since otherwise $\alpha \notin V(\mathcal{P}_{\text{TSP}}(n))$. Thus $\Delta(\alpha) = L_{k_1 k_2}(\alpha)$. Since $L_{k_1 k_2}$ is independent of any edge incident to i, j, k_1, k_2, it does not involve those variables, so $L_{k_1 k_2}(\alpha) = L_{k_1 k_2}(\beta)$, where β is the restriction of α to the $(n-2)^2$ variables which $L_{k_1 k_2}$ depends on. But such a β is simply an element of $V(\mathcal{P}_{\text{TSP}}(n-2))$, and all elements of $V(\mathcal{P}_{\text{TSP}}(n-2))$ can be obtained this way. Thus $L_{k_1 k_2}$ is zero on all of $V(\mathcal{P}_{\text{TSP}}(n-2))$, and by Lemma 3.4.2, $L_{k_1 k_2} \in \langle \mathcal{P}_{\text{TSP}}(n-2) \rangle$. Now by the inductive hypothesis, $L_{k_1 k_2}$ has a derivation from $\mathcal{P}_{\text{TSP}}(n-2)$ of degree at most $2d - 2$. By Lemma 3.4.7, $L_{k_1 k_2} x_{ik_1} x_{jk_2}$ has a derivation from $\mathcal{P}_{\text{TSP}}(n)$ of degree at most $2d$, and thus so does Δ.

Because transpositions generate the symmetric group, the above argument implies that $p - \frac{1}{n!} \sum_{\sigma \in S_n} \sigma p$ has a derivation from $\mathcal{P}_{\text{TSP}}(n)$ of degree at most $2d$. Combined with Corollary 3.4.6 and Lemma 2.3.8, this is enough to prove the theorem statement. $\square$

3.5 Effective Derivations for Balanced-CSP

Fix integers n and $c \leq n$. Then the BALANCED-CSP problem has a polynomial formulation on n variables with constraints

$$\mathcal{P}_{\mathrm{BCSP}}(n,c) = \left\{x_i^2 - x_i \mid i \in [n]\right\} \cup \left\{\sum_i x_i - c\right\}. \tag{3.7}$$

The first set of polynomials ensures that the variables are Boolean, and the final polynomial is a balance constraint that forces a specific number of variables to be 1. The BISECTION constraints are the special case when n is even and $c = n/2$. As before, we need to define the appropriate symmetric action. For an element $\sigma \in S_n$, we define $\sigma x_i = x_{\sigma(i)}$ and extend this action multiplicatively and linearly to get an action on every polynomial. Once again, note that $\mathcal{P}_{\mathrm{BCSP}}$ and $V(\mathcal{P}_{\mathrm{BCSP}})$ are fixed by S_n under this action, and thus if $p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$, then $\sigma p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$. We will begin by proving 1-effectiveness for the special case of BISECTION, as we will encounter an obstacle for general c. Because $\mathcal{P}_{\mathrm{BCSP}}$ contains the Boolean constraints $\{x_i^2 - x_i \mid i \in [n]\}$, we will take p to be a multilinear polynomial without loss of generality. For a set $A \subseteq [n]$, let $x_A = \prod_{i \in A} x_i$. Our proof strategy is the same three-step strategy referenced in Section 3.2.

Lemma 3.5.1. *$\langle \mathcal{P}_{\mathrm{BCSP}}(n,c) \rangle$ is complete for any n and $c \leq n$.*

Proof. Let p be a multilinear polynomial which is zero on all of $V(\mathcal{P}_{\mathrm{BCSP}})$. First, we argue that if $A \subseteq [n]$ is such that $|A| > c$ then $x_A \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$. We prove this by backwards induction from n to $c+1$. For the base case of $|A| = n$, note that

$$x_A \cong \frac{1}{n-c} x_A \left(\sum_i x_i - c\right),$$

and the RHS is clearly an element of $\langle \mathcal{P} \rangle$. Now if $|A| = k$ with $c + 1 \leq k < n$, we have

$$(k-c)\, x_A + \sum_{i \notin A} x_{A \cup \{i\}} \cong x_A \left(\sum_i x_i - c\right).$$

By the inductive hypothesis, the second term of the LHS is in $\langle \mathcal{P}_{\mathrm{BCSP}} \rangle$, and obviously the RHS is in $\langle \mathcal{P}_{\mathrm{BCSP}} \rangle$, and thus so is x_A. Thus we can assume that the monomials of p are all of degree at most c. For any monomial x_A of p, we have $x_A(\sum_i x_i - 1) \cong \sum_{i \notin A} x_{A \cup \{i\}} - (c - |A|)x_A$, and so $x_A - \frac{1}{c-|A|} \sum_{i \notin A} x_{A \cup \{i\}} \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$, and so we can replace x_A with monomials of one higher degree. Repeatedly applying this up to degree c (at which point we must stop to avoid dividing by zero), we determine there is a polynomial p' which is homogenous of degree c such that $p - p' \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$. Now let $p'_{i_1,\ldots,i_c}$ be the coefficient of the monomial $x_{i_1} \ldots x_{i_c}$ in p' and let α be the element of $V(\mathcal{P}_{\mathrm{BCSP}})$ with $i_1, \ldots, i_c$ coordinates equal to 1 and all other coordinates equal to zero. Then $p'(\alpha) = p'_{i_1,\ldots,i_c}$, but $p'(\alpha) = 0$. Thus in fact $p' = 0$, and so $p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$. ∎

Symmetric Polynomials

The second step is to show that any symmetrized polynomial can be derived from a constant polynomial in low degree. It is considerably simpler than MATCHING in this case, as the fundamental theorem of symmetric polynomials tells us that powers of $\sum_i x_i$ generate all the symmetric polynomials.

Lemma 3.5.2. *Let p be a multilinear polynomial in $\mathbb{R}^n$. Then there exists a constant c_p such that $p' = \sum_{\sigma \in \mathcal{S}_n} \sigma p - c_p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle_{\deg p}$. If $p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$, then $p' \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle_{\deg p}$.*

Proof. It is sufficient to prove the lemma for monomials $x_A = \prod_{i \in A} x_i$. We will induct on the degree of the monomial $|A|$. If $|A| = 1$, then $p = x_i$ for some $i \in [n]$, and $p' = \sum_{\sigma \in \mathcal{S}_n} \sigma x_i = (n-1)! \sum_i x_i \cong (n-1)! \cdot c$, which can clearly be performed in degree one. Now assume $|A| = k$, so that $p' = \sum_{\sigma \in \mathcal{S}_n} \sigma x_A = (n-k)! \sum_{|B|=k} x_B$. Then $p'' = p' - \frac{(n-k)!}{k!} \left(\sum_i x_i - c\right)^k$ is a polynomial which, after multilinearizing by reducing by the Boolean constraints, has degree at most $k-1$ (the coefficient $\frac{(n-k)!}{k!}$ was chosen to cancel the highest degree term of p'). Furthermore, p'' is in $\langle \mathcal{P}_{\mathrm{BCSP}} \rangle$ because p and p' are, and $(\sum_i x_i - c)^k$ is an element of $\langle \mathcal{P} \rangle$. Finally, p'' is fixed by every σ. Thus by the inductive hypothesis, p'' has a derivation from some constant in degree $k-1$. Since $p' \cong_k p''$, this implies the statement for $|A| = k$ and completes the proof by induction.

The second line of the lemma follows immediately, since if $p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$ then $c_p \in \langle \mathcal{P}_{\mathrm{BCSP}} \rangle$, but the only constant polynomial in $\langle \mathcal{P}_{\mathrm{BCSP}} \rangle$ is 0 by Lemma 2.3.7. $\square$

Now we move on to the third and final step, where we specialize to the BISECTION constraints $\mathcal{P}_{\mathrm{BCSP}}(n, n/2)$.

Getting to a Symmetric Polynomial

Recall the third step of our strategy is to show that $p - \sigma p$ can be derived from $\mathcal{P}_{\mathrm{BCSP}}$ in low degree. It will be easier in this case as compared to MATCHING because we do not have to increase the degree of $p - \sigma p$ in order to isolate a variable to remove and do the induction. Because of this, we will be able to show that BISECTION is actually 1-effective and not lose a factor of two. We need a lemma to help us do the induction:

Lemma 3.5.3. *Let $L \in \langle \mathcal{P}_{\mathrm{BCSP}}(n, c) \rangle_d$. Then $L \cdot (x_{n+1} - x_{n+2}) \in \langle \mathcal{P}_{\mathrm{BCSP}}(n+2, c+1) \rangle_{d+1}$.*

Proof. It is sufficient to prove the lemma for $L \in \mathcal{P}_{\mathrm{BCSP}}(n, c)$. If $L = x_i^2 - x_i$ for some i, then $L \in \mathcal{P}_{\mathrm{BCSP}}(n+2, c+1)$ and so the lemma is clearly true. If $L = \sum_{i=0}^{n} x_i - c$, then

$$
\begin{aligned}
L \cdot (x_{n+1} - x_{n+2}) - \left(\sum_{i=0}^{n+2} x_i - (c+1)\right)(x_{n+1} - x_{n+2}) &= (1 - x_{n+1} - x_{n+2}) \cdot (x_{n+1} - x_{n+2}) \\
&= x_{n+1} - x_{n+2} - x_{n+1}^2 - x_{n+1}x_{n+2} + x_{n+1}x_{n+2} + x_{n+2}^2 \\
&\cong_2 0.
\end{aligned}
$$

$\square$

We are now ready to prove that the BISECTION constraints admit effective derivations.

Theorem 3.5.4. *Let n be even and $p \in \langle \mathcal{P}_{\mathrm{BCSP}}(n, n/2) \rangle$ and $d = \deg p$. Then p has a derivation from $\mathcal{P}_{\mathrm{BCSP}}(n, n/2)$ in degree d.*

Proof. By reducing by the Boolean constraints, we can assume p is a multilinear polynomial. We will induct on the number of vertices n, so first we must handle the base case of $n = 2$ (recall n is even). The only degree zero polynomial in $\mathcal{P}_{\mathrm{BCSP}}(2, 1)$ is the zero polynomial which has the trivial derivation. If $p = ax_1 + bx_2 + c$, we know $p(0, 1) = p(1, 0) = 0$. This implies $a = b = -c$, and so p is a multiple of $\sum_i x_i - 1$, which clearly has a derivation of degree 1. Finally, x_1x_2 has the derivation $x_1x_2 = x_1(x_1 + x_2 - 1) + (-1) \cdot (x_1^2 - x_1)$. So any quadratic polynomial in $\langle \mathcal{P}_{\mathrm{BCSP}}(2, 1) \rangle$ can be reduced to a linear polynomial in degree 2, but we already showed that every linear polynomial has a degree 1 derivation. This proves the base case.

Now assume the theorem statement for $\mathcal{P}_{\mathrm{BCSP}}(n', n'/2)$ with $n' < n$. Let $\sigma = (i, j)$ be a transposition between two vertices. We consider the polynomial $\Delta = p - \sigma p$. We can decompose $p = r_ix_i + r_jx_j + r_{ij}x_ix_j + q_{ij}$, where each of the polynomials r_i, r_j, r_{ij}, and q_{ij} depend on neither x_i nor x_j, and r_i and r_j are degree $d-1$. Then $\Delta = (r_i - r_j)(x_i - x_j)$. Now since $\Delta \in \langle \mathcal{P}_{\mathrm{BCSP}}(n, n/2) \rangle$, we know that $\Delta(x) = 0$ for any $x \in \{0, 1\}^n$ with exactly $n/2$ indices which are 1. In particular, if we set $x_i = 1$ and $x_j = 0$, we know that $(r_i - r_j)$ must be zero if the remaining variables are set so that they have exactly $n/2 - 1$ indices which are 1. In other words, $(r_i - r_j)$ is zero on $V(\mathcal{P}_{\mathrm{BCSP}}(n - 2, (n - 2)/2))$. By Lemma 3.5.1, we have $(r_i - r_j) \in \langle \mathcal{P}_{\mathrm{BCSP}}(n - 2, (n - 2)/2) \rangle$, and thus by the inductive hypothesis $(r_i - r_j)$ has a derivation from $\mathcal{P}_{\mathrm{BCSP}}(n-2, (n-2)/2)$ in degree $d-1$. By Lemma 3.5.3, $\Delta = (r_i - r_j)(x_i - x_j)$ has a derivation from $\mathcal{P}_{\mathrm{BCSP}}(n, n/2)$ in degree d.

Now since the transpositions generate the entire symmetric group, we have

$$p \cong_d \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} \sigma p \cong_d 0,$$

where the last congruence is by Lemma 3.5.2. Thus p has a derivation from $\mathcal{P}_{\mathrm{BCSP}}(n, n/2)$ in degree d. □

Obstacles for General c

We will consider general $c < n/2$ since the case of $c > n/2$ is symmetric via the linear transformation $y_i = 1 - x_i$. We would like to argue that $\mathcal{P}_{\mathrm{BCSP}}(n, c)$ is effective even in the case of general c. What goes wrong if we just try to imitate the proof of Theorem 3.5.4? If we do so, eventually we arrive at the base case of the induction: $\mathcal{P}_{\mathrm{BCSP}}(n - 2c, 0)$. The problem is that the linear monomials x_i are in $\langle \mathcal{P}_{\mathrm{BCSP}}(n - 2c, 0) \rangle$ but it is not obvious how to derive x_i from $\mathcal{P}_{\mathrm{BCSP}}(n - 2c, 0)$. In fact, it turns out that derivations of x_i require degree $(n - 2c + 1)/2$. If $c = \Omega(n)$, the gap between this and the degree of x_i, namely 1, is as large as $\Omega(n)$!

This obstacle is not an artifact of our proof strategy, but an intrinsic obstacle. There are essentially two kinds of polynomials in $\langle \mathcal{P}_{\mathrm{BCSP}}(n,c) \rangle$: Polynomials of degree at most c, and polynomials of degree $c+1$ or greater. The former have efficient derivations:

Lemma 3.5.5. *Let $p \in \langle \mathcal{P}_{\mathrm{BCSP}}(n,c) \rangle$ have degree at most c. Then p has a derivation from $\mathcal{P}_{\mathrm{BCSP}}(n,c)$ in degree $\deg p$.*

We delay the proof of this lemma until the next section. However, the polynomials of degree $c+1$ or greater actually have no derivations until degree $(n-c+1)/2$, so if $c << n$, then $\mathcal{P}_{\mathrm{BCSP}}(n,c)$ is not k-effective for any constant k. We will see that this phenomenon is because of the fact that the Pigeonhole Principle requires high degree for HN derivations. The negation of the Pigeonhole Principle is the following set of constraints:

$$\begin{aligned} \neg\mathcal{PHP}(m,n) = &\left\{x_{ij}^2 - x_{ij} \mid i \in [m], j \in [n]\right\} \\ &\cup \left\{\sum_j x_{ij} - 1 \mid i \in [m]\right\} \\ &\cup \{x_{ij}x_{ik} \mid i \in [m], j,k \in [n], j \neq k\} \\ &\cup \{x_{ij}x_{kj} \mid i,k \in [m], j \in [n], i \neq k\} \end{aligned}$$

$\neg\mathcal{PHP}(m,n)$ asserts the existence of an injective mapping from $[m]$ into $[n]$. If $m > n$, then clearly there is no such mapping, so the set of polynomials is infeasible. This implies that $1 \in \langle \neg\mathcal{PHP}(m,n) \rangle$ by Lemma 2.3.7. However, Razborov proved that any derivation of 1 from $\neg\mathcal{PHP}(m,n)$ has degree at least $n/2+1$ [62]. This allows us to prove the following by reduction:

Lemma 3.5.6. *Let $p = x_1x_2\dots x_cx_{c+1}$. Then $p \in \langle \mathcal{P}_{\mathrm{BCSP}}(n,c) \rangle$, but any derivation of p from $\mathcal{P}_{\mathrm{BCSP}}(n,c)$ has degree at least $(n-c+1)/2$.*

Proof. We argued $p \in \langle \mathcal{P}_{\mathrm{BCSP}}(n,c) \rangle$ in Lemma 3.5.1, and essentially used a Pigeonhole Principle argument where the pigeons are the $n-c$ zeros, and the holes are the $n-c-1$ variables not appearing in p. More formally, we show how to manipulate any derivation of p from $\mathcal{P}_{\mathrm{BCSP}}(n,c)$ to get a refutation of $\neg\mathcal{PHP}(n-c, n-c-1)$.

Any derivation of p from $\mathcal{P}_{\mathrm{BCSP}}(n,c)$ is a polynomial identity of the following form:

$$x_1x_2\dots x_{c+1} = \lambda \cdot \left(\sum_i x_i - c\right) + \sum_i \lambda_i \cdot (x_i^2 - x_i).$$

Now set $x_1 = x_2 = \dots = x_{c+1} = 1$ to get

$$1 = \lambda' \cdot \left(\sum_{i>c+1} x_i + 1\right) + \sum_{i>c+1} \lambda_i' \cdot (x_i^2 - x_i).$$

We define variables y_{ij} with the intention that $y_{ij} = 1$ if the ith variable is the jth zero. Thus we replace $x_i \to 1 - \sum_{j=1}^{n-c} y_{ij}$ and get

$$\begin{aligned}
1 &= \lambda'(y) \cdot \left(\sum_{i>c+1} \left(1 - \sum_{j=1}^{n-c} y_{ij} \right) + 1 \right) + \sum_{i>c+1} \lambda'_i(y) \cdot \left(\left(1 - \sum_{j=1}^{n-c} y_{ij} \right)^2 - 1 + \sum_{j=1}^{n-c} y_{ij} \right) \\
&= \lambda'(y) \cdot \left(n - c - 1 - \sum_{i>c+1,j} y_{ij} + 1 \right) + \sum_{i>c+1} \lambda'_i(y) \cdot \left(\sum_{j=1}^{n-c} y_{ij}^2 - \sum_{j=1}^{n-c} y_{ij} + 2 \sum_{j \neq j'} y_{ij} y_{ij'} \right) \\
&= \sum_{j=1}^{n-c} -\lambda'(y) \cdot \left(\sum_{i>c+1} y_{ij} - 1 \right) + \sum_{i>c+1} \lambda'_i(y) \cdot \left(\sum_{j} \left(y_{ij}^2 - y_{ij} \right) + 2 \sum_{j \neq j'} y_{ij} y_{ij'} \right).
\end{aligned}$$

Note that each term in the last equation contains a constraint in $\neg\mathcal{PHP}(n-c, n-c-1)$. Thus the degree of this derivation must be at least $(n-c+1)/2$. Fixing $x_1, \ldots, x_{c+1}$ can only reduce the degree, so the degree of the original derivation must be at least $(n-c+1)/2$ as well. □

Effective PC$_>$ Derivations for High Degree Polynomials

Lemma 3.5.6 tells us that we cannot hope to prove that $\mathcal{P}_{\mathrm{BCSP}}(n, c)$ has effective HN proofs, but we are not soley interested in HN proofs. In particular, because the applications we consider in this thesis are primarily focused on Semidefinite Programming, we have access to the more powerful PC$_>$ proof system. In this system, $\neg\mathcal{PHP}$ is not difficult to refute, and indeed once we allow ourselves PC$_>$ proofs we can show that Balanced-CSP admits effective derivations.

Lemma 3.5.7. *Let $p = x_1 x_2 \ldots x_c x_{c+1}$. Then p has a PC$_>$ proof Π from $\mathcal{P}_{\mathrm{BCSP}}(n, c)$ in degree $2(c+1)$ with $\|\Pi\| \leq 1$.*

Proof. Recall that a PC$_>$ proof consists of two proofs of non-negativity: one for p and one for $-p$. The first is trivial: every monomial is the multilinearization of itself squared. Thus every monomial has a proof of non-negativity in twice its degree. For the second, we observe the following identity

$$-x_1 x_2 \ldots x_{c+1} = x_1 x_2 \ldots x_c \left(c - \sum_i x_i \right) + \sum_{i \leq c} -(x_i^2 - x_i) \prod_{j \leq c, j \neq i} x_j + \left(\prod_{i \leq c} x_i \right) \sum_{i > c+1} x_i.$$

The first two terms each have factors in $\mathcal{P}_{\mathrm{BCSP}}(n, c)$, and the last term is a sum of monomials with non-negative coefficients. These monomials all have proofs of non-negativity, and thus so does $-p$. It is simple to check that these proofs involve coefficients of unit size. □

Effective HN Derivations for Low Degree Polynomials

It remains to prove that the low-degree polynomials in $\mathcal{P}_{\text{BCSP}}(n, c)$ have efficient derivations. We will be able to use simple HN derivations for these polynomials. The proof is very similar to the one for Bisection, but we have to do a double induction on n and c since the balance changes in the inductive step. We will take $c \leq n/2$, since the other case is symmetric.

Lemma 3.5.8. *Fix $c \leq n/2$. Let $p \in \langle \mathcal{P}_{\text{BCSP}}(n, c) \rangle$ with $\deg p \leq c$. Then p has a derivation from $\mathcal{P}_{\text{BCSP}}(n, c)$ in degree $\deg p$.*

Proof. The proof is by double induction on n and c. The base case is the lemma statement for $\mathcal{P}_{\text{BCSP}}(n, 0)$ for all n. In this case p is a constant polynomial, and the only constant polynomial in $\mathcal{P}_{\text{BCSP}}(n, 0)$ is the zero polynomial, which has the trivial derivation. Now consider the case when $p \in \mathcal{P}_{\text{BCSP}}(n, c)$ for $c \leq n/2$. Then following the same argument as in Theorem 3.5.4, we define $\Delta = p - \sigma p = (r_i - r_j) \cdot (x_i - x_j)$, where r_i and r_j do not depend on x_i or x_j. Setting $x_i = 1$ and $x_j = 0$, we again conclude that $(r_i - r_j) \in \langle \mathcal{P}_{\text{BCSP}}(n-2, c-1) \rangle$. Since $c \leq n/2$, clearly $c - 1 \leq (n-2)/2$. Also, since $r_i - r_j$ has degree $\deg p - 1$, we still have $\deg(r_i - r_j) \leq c - 1$. Thus we can apply the inductive hypothesis to get a derivation for $r_i - r_j$ from $\mathcal{P}_{\text{BCSP}}(n-2, c-1)$ in degree $\deg p - 1$. Then Lemma 3.5.3 tells us that Δ has a derivation from $\mathcal{P}_{\text{BCSP}}(n, c)$ in degree $\deg p$, completing the induction. Taken with Lemma 3.5.2, this implies the statement. $\square$

3.6 Boolean Sparse PCA

The last example we give for our proof strategy is the Boolean Sparse PCA problem, which has a formulation on n variables with constraints

$$\mathcal{P}_{\text{SPCA}}(n, c) = \left\{x_i^3 - x_i \mid i \in [n]\right\} \cup \left\{\sum_i x_i^2 - c\right\}.$$

The first set of polynomials ensures that the variables are *ternary*, and lie in $\{0, \pm 1\}$. The last polynomial is another balance constraint, and sets the number of variables that can be nonzero. We define a symmetric action by $\sigma x_i = x_{\sigma(i)}$ for $\sigma \in S_n$ and extend it appropriately. These constraints arise when trying to reconstruct a planted sparse vector from noisy samples, see for example [50].

These constraints are similar to the Balanced-CSP constraints of the previous section with one crucial difference: The variables are ternary instead of binary. This will complicate the analysis, but it turns out that with a bit more casework we will be able to push it through. However, the Pigeonhole Principle obstacle remains, and we will once again only be able to prove that low-degree polynomials have effective HN derivations.

Lemma 3.6.1. *$\mathcal{P}_{\text{SPCA}}(n, c)$ is complete for every n and $c \leq n$.*

Proof. Recall that by pigeonhole principle any monomial that involves $c+1$ or more distinct variables will be zero over $V(\mathcal{P}_{\text{SPCA}})$. Our first step is to show that these are in $\langle \mathcal{P}_{\text{SPCA}} \rangle$. The proof will be a reverse induction on the number of distinct variables, going from n to $c+1$. For the base case, let x_A be any monomial with n distinct variables. Then $x_A \cong \frac{1}{n-c} x_A(\sum_i x_i^2 - c)$, so clearly $x_A \in \langle \mathcal{P}_{\text{SPCA}} \rangle$. Now let x_A be any monomial with $c+1 \leq k < n$ distinct variables. Then

$$(k-c)\, x_A + \sum_{i \notin A} x_{A \cup \{i,i\}} \cong x_A \left(\sum_i x_i^2 - c \right).$$

By the inductive hypothesis, the second term of the LHS is in $\langle \mathcal{P}_{\text{SPCA}} \rangle$, and thus so is x_A. Now let p be a polynomial such that $p(\alpha) = 0$ for every $\alpha \in V(\mathcal{P}_{\text{SPCA}})$. We can assume that the monomials of p involve at most c distinct variables. For any monomial x_A of p, we have $x_A(\sum_i x_i - 1) \cong \sum_{i \notin A} x_{A \cup \{i,i\}} - (c - |A|)x_A$, and so $x_A - \frac{1}{c-|A|} \sum_{i \notin A} x_{A \cup \{i,i\}} \in \langle \mathcal{P}_{\text{SPCA}} \rangle$, and so we can replace x_A with monomials of one higher degree. Repeatedly applying this up to degree c (at which point we must stop to avoid dividing by zero), we determine there is a polynomial p' which has only monomials involving exactly c distinct variables such that $p - p' \in \langle \mathcal{P}_{\text{SPCA}} \rangle$. Fix two disjoint sets U_1 and U_2 of the variables with $|U_1 \cup U_2| = c$ and let $p'_{U_1 U_2}$ be the coefficient of the monomial of p' corresponding to the variables in $U_1 \cup U_2$ with the variables in U_1 appearing with degree one and the variables in U_2 appearing with degree two. We will prove by induction that $p'_{U_1 U_2} = 0$ for every U_1, U_2. For the base case, let $U_1 = \emptyset$. Then if we average every monomial of p' over the $\alpha \in V(\mathcal{P}_{\text{SPCA}})$ that assign nonzero values exactly to the variables in $U_1 \cup U_2$, every monomial except $p'_{U_1 U_2}$ is zero, and that monomial has value one. Since $p'(\alpha) = 0$ for each $\alpha \in V(\mathcal{P}_{\text{SPCA}})$, this implies that $p'_{U_1 U_2} = 0$. Proceeding by induction, let $|U_1| = k$. Then if we average over all the $\alpha \in V(\mathcal{P}_{\text{SPCA}})$ that assign nonzero values exactly to the variables in $U_1 \cup U_2$ and assigns value 1 to the variables in U_1, every monomial is zero except p'_{UV} with $U \cup V = U_1 \cup U_2$ and $U \subseteq U_1$. By the inductive hypothesis these all have zero coefficients except $p'_{U_1 U_2}$, and now since p' is zero on all these points, we once again have $p'_{U_1 U_2}$. Doing this for every U_1, U_2, we determine $p' = 0$ and thus $p \in \langle \mathcal{P}_{\text{SPCA}} \rangle$. $\square$

Symmetric Polynomials

Once again, we prove a derivation lemma for symmetric polynomials. For this set of constraints, it is not as simple as saying that every symmetric polynomial is equal to some constant on $V(\mathcal{P}_{\text{SPCA}})$ because we only have a constraint on $\sum_i x_i^2$ as opposed to $\sum_i x_i$. In particular, the polynomial $\sum_i x_i$ itself does not reduce to a constant on $V(\mathcal{P}_{\text{SPCA}})$. We will have to make a slightly more general argument.

Lemma 3.6.2. *Let p be a polynomial in $\mathbb{R}^n$. Then there exists a univariate polynomial q of degree* $\deg p$ *such that $p' = \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} \sigma p \cong q(\sum_i x_i)$.*

Proof. We prove that for every elementary symmetric polynomial $e_k(x)$, there exists a univariate polynomial q_k such that $e_k(x) - q_k(\sum_i x_i)$ has a derivation from $\mathcal{P}_{\text{SPCA}}$ in degree k, then the fundamental theorem of symmetric polynomials implies the lemma. For the base case, clearly $q_0(t) = 1$ and $q_1(t) = t$. For the general case, consider the terms of the expansion of $(\sum_i x_i)^k$. They are indexed by the non-increasing partitions of k: $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_\ell)$ and can be written $c_\lambda \sum_{i_1,\ldots,i_\ell} x_{i_1}^{\lambda_1} x_{i_2}^{\lambda_2} \ldots x_{i_\ell}^{\lambda_\ell}$. Now just by reducing by $x_i^3 - x_i$, we can reduce the exponents on each variable to either one or two. If any exponent is two, then by reducing by the constraint $\sum_i x_i^2 - c$, we can replace any of these exponents with a multiplicative constant. Thus after reducing, all of the exponents are one. But now this term is simply a multiple of some $e_{k'}(x)$, with $k' \leq k$. Since one term is exactly $k! e_k(x)$, we have

$$\frac{1}{k!}\left(\sum_i x_i\right)^k - e_k(x) \cong_k \sum_{i=1}^{k-1} a_i e_i(x)$$

for some real numbers a_i. Now by the inductive hypothesis, we know that there exist polynomials q_i such that $e_i(x) - q_i(\sum_i x_i)$ has a derivation from $\mathcal{P}_{\text{SPCA}}$ in degree i. Thus we set $\frac{1}{k!} q_k(t) = t^k - \sum_i a_i q_i(t)$ to complete the induction and the lemma. □

Corollary 3.6.3. *Let $p \in \langle \mathcal{P}_{\text{SPCA}} \rangle$ with $\deg p \leq c$. Then $p' = \frac{1}{n!} \sum_{\sigma \in S_n} \sigma p$ has a derivation from $\mathcal{P}_{\text{SPCA}}$ in degree $\deg p$.*

Proof. By Lemma 3.6.2, we know that there is a univariate polynomial $q(t)$ of degree $\deg p$ such that $p' - q(\sum_i x_i) \in \langle \mathcal{P}_{\text{SPCA}} \rangle_{\deg p}$. Since $p \in \langle \mathcal{P}_{\text{SPCA}} \rangle$, so is p' and $q(\sum_i x_i)$. Since there are $c + 1$ possible values of $\sum_i x_i$ in $V(\mathcal{P}_{\text{SPCA}})$, namely $\{-c, -c+2, \ldots, c-2, c\}$, q has $c+1$ zeros. But $\deg q = \deg p \leq c$, so q must be the zero polynomial. □

Getting to a Symmetric Polynomial

This process should be familiar by now. Since there are more choices for values for the variables we are going to strip off, we are going to need to do a little more casework, but the general strategy is the same. We start with a lemma that allows us to perform induction.

Lemma 3.6.4. *Let L be a polynomial with a degree d derivation from $\mathcal{P}_{\text{SPCA}}(n, c)$. Then $L \cdot (x_{n+1}^2 - x_{n+2}^2)$ has a degree $d+2$ derivation from $\mathcal{P}_{\text{SPCA}}(n+2, c+1)$, and $L \cdot (x_{n+1} x_{n+2})$ has a degree $d+2$ derivation from $\mathcal{P}_{\text{SPCA}}(n+2, c+2)$.*

Proof. It suffices to prove the theorem for $L \in \mathcal{P}_{\text{SPCA}}(n, c)$. If $L = x_i^3 - x_i$ for some i, then clearly the statement is true as $L \in \mathcal{P}_{\text{SPCA}}(n+2, c+1)$ and $L \in \mathcal{P}_{\text{SPCA}}(n+2, c+2)$, so let $L = \sum_i x_i^2 - c$. Now notice that

$$\begin{aligned} L \cdot (x_{n+1}^2 - x_{n+2}^2) - \left(\sum_{i=1}^{n+2} x_i^2 - (c+1)\right)(x_{n+1}^2 - x_{n+2}^2) &= (1 - x_{n+1}^2 + x_{n+2}^2)(x_{n+1}^2 - x_{n+2}^2) \\ &= x_{n+1}^2 - x_{n+2}^2 - x_{n+1}^4 + x_{n+2}^4 \\ &\cong_4 x_{n+1}^2 - x_{n+2}^2 - x_{n+1}^2 + x_{n+2}^2 \\ &= 0 \end{aligned}$$

and

$$
\begin{aligned}
Lx_{n+1}x_{n+2} - \left(\sum_{i=1}^{n+2} x_i^2 - (c+2)\right) x_{n+1}x_{n+2} &= (2 - x_{n+1}^2 + x_{n+2}^2)x_{n+1}x_{n+2} \\
&= 2x_{n+1}x_{n+2} - x_{n+1}^3 x_{n+2} + x_{n+1}x_{n+2}^3 \\
&\cong_4 2x_{n+1}x_{n+2} - x_{n+1}x_{n+2} + x_{n+1}x_{n+2} \\
&= 0
\end{aligned}
$$

to conclude the lemma. □

Now we prove that Boolean Sparse PCA admits effective derivations for low degree polynomials.

Lemma 3.6.5. *Fix $c \leq n/2$. Let $p \in \langle \mathcal{P}_{\mathrm{SPCA}}(n,c) \rangle$ with $\deg p \leq c/2$. Then p has a derivation from $\mathcal{P}_{\mathrm{SPCA}}(n,c)$ in degree at most $3 \deg p$.*

Proof. We do double induction on n and c. For the base case of $\mathcal{P}_{\mathrm{SPCA}}(n,0)$, note that the only polynomial with degree at most 0 is the constant polynomial 0, which has the trivial derivation. Now let p have degree at most $d \leq c/2$. We can assume the individual degree of each variable is at most two by reducing by the ternary constraints. Following the same argument as in Theorem 3.5.4, we define the polynomial $\Delta = p - \sigma p$ for the transposition $\sigma = (i,j)$, but now since p is not multilinear, we write it as

$$
p = r_{10}x_i + r_{01}x_j + r_{20}x_i^2 + r_{02}x_j^2 + r_{11}x_ix_j + r_{21}x_i^2x_j + r_{12}x_ix_j^2 + r_{22}x_i^2x_j^2 + q_{ij}
$$

where none of the r or q polynomials depend on x_i or x_j. Then Δ can be written

$$
\begin{aligned}
\Delta &= (r_{10} - r_{01})(x_i - x_j) + (r_{20} - r_{02})(x_i^2 - x_j^2) + (r_{21} - r_{12})(x_i^2x_j - x_ix_j^2) \\
&= ((r_{10} - r_{01}) + (r_{20} - r_{02})(x_i + x_j) + (r_{21} - r_{12})x_ix_j)(x_i - x_j) \\
&= (R_0 + R_1(x_i + x_j) + R_2x_ix_j)(x_i - x_j)
\end{aligned}
$$

where we define $R_0 = (r_{10} - r_{01})$, $R_1 = (r_{20} - r_{02})$, and $R_2 = (r_{21} - r_{12})$, and note that they are polynomials of degree at most $d-1$. If we set $x_i = 1$ and $x_j = 0$, we obtain a polynomial $R_0 + R_1$ which must be zero on $V(\mathcal{P}_{\mathrm{SPCA}}(n-2, c-1))$. Furthermore, if we set $x_i = -1$ and $x_j = 0$, then $R_0 - R_1$ is zero on $V(\mathcal{P}_{\mathrm{SPCA}}(n-2, c-1))$, and setting $x_i = 1$ and $x_j = -1$, we also get that $R_0 - R_2$ is zero on $V(\mathcal{P}_{\mathrm{SPCA}}(n-2, c-2))$.

Since $c \leq n/2$, clearly $c - 2 \leq c - 1 \leq (n-2)/2$. Since $d \leq c/2$, we also have $d - 1 \leq (c-2)/2$. Since by Lemma 3.6.1 we know $\mathcal{P}_{\mathrm{SPCA}}(n,c)$ is complete, we can apply the inductive hypothesis and so all these polynomials have derivations of degree at most $3(d-1)$ from their constraints. By Lemma 3.6.4, we know $(R_0 + R_1)(x_i^2 - x_j^2)$, $(R_0 - R_1)(x_i^2 - x_j^2)$, and $(R_0 - R_2)x_ix_j$ have derivations from $V(\mathcal{P}_{\mathrm{SPCA}}(n,c))$ in degree $3d - 1$.

From the first two polynomials, it is clear that $R_0(x_i^2 - x_j^2)$ and $R_1(x_i^2 - x_j^2)$ have derivations in degree $3d-1$. We also have

$$\begin{aligned} R_0(x_i^2 - x_j^2) \cdot (x_i + x_j) - (R_0 - R_2)x_i x_j \cdot (x_i - x_j) = \\ &= R_0((x_i^2 - x_j^2)(x_i + x_j) - x_i x_j(x_i - x_j)) + R_2 x_i x_j (x_i - x_j) \\ &\cong (R_0 + R_2 x_i x_j)(x_i - x_j) \end{aligned}$$

and thus $(R_0 + R_2 x_i x_j)(x_i - x_j)$ is derivable in degree $3d$. Together with $R_1(x_i + x_j)(x_i - x_j)$ having a derivation in degree $3d-1$, this implies that Δ has a derivation in degree $3d$. Taken together with Lemma 3.6.2, we conclude that p has a derivation in degree $3d$. $\square$

3.7 Optimization Problems with Effective Derivations

We include a corollary here summarizing all the results of this chapter:

Corollary 3.7.1. *The following polynomial formulations of combinatorial optimization problems admit k-effective derivations:*

- CSP*:* $\mathcal{P}_{\mathrm{CSP}}(n) = \{x_i^2 - x_i \mid i \in [n]\}$*, for* $k = 1$*.*
- Clique*:* $\mathcal{P}_{\mathrm{CLIQUE}}(V, E) = \{x_i^2 - x_i \mid i \in V\} \cup \{x_i x_j \mid (i,j) \notin E\}$*, for* $k = 1$*.*
- Matching*:*

$$\begin{aligned} \mathcal{P}_{\mathrm{M}}(n) = &\left\{x_{ij}^2 - x_{ij} \mid i, j \in [n]\right\} \\ &\cup \left\{\sum_i x_{ij} - 1 \mid j \in [n]\right\} \\ &\cup \left\{x_{ij}x_{ik} \mid i, j, k \in [n], j \neq k\right\}, \end{aligned}$$

  for $k = 2$*.*
- TSP*:*

$$\begin{aligned} \mathcal{P}_{\mathrm{TSP}}(n) = &\left\{x_{ij}^2 - x_{ij} \mid i, j \in [n]\right\} \\ &\cup \left\{\sum_i x_{ij} - 1 \mid j \in [n]\right\} \\ &\cup \left\{x_{ij}x_{ik}, x_{ji}x_{ki} \mid i, j, k \in [n], j \neq k\right\}, \end{aligned}$$

  for $k = 2$*.*
- Bisection*:* $\mathcal{P}_{\mathrm{BCSP}}(n, n/2) = \{x_i^2 - x_i \mid i \in [n]\} \cup \left\{\sum_i x_i - \frac{n}{2}\right\}$*, for* $k = 1$*.*

The following sets of constraints admit k-effective derivations up to degree c:

- Balanced CSP: $\mathcal{P}_{\text{BCSP}}(n, c) = \{x_i^2 - x_i \mid i \in [n]\} \cup \{\sum_{i=1}^{n} x_i - c\}$, *for* $k = 1$.
- Boolean Sparse PCA: $\mathcal{P}_{\text{SPCA}}(n, 2c) = \{x_i^3 - x_i \mid i \in [n]\} \cup \{\sum_i x_i^2 - 2c\}$, *for* $k = 3$.

Chapter 4

Bit Complexity of Sum-of-Squares Proofs

In this chapter we will show how effective derivations can be applied to prove that the Ellipsoid algorithm runs in polynomial time for many practical inputs to the Sum-of-Squares algorithm. First, we recall the Sum-of-Squares relaxation for approximate polynomial optimization. We wish to solve the following optimization problem:

$$\begin{aligned} &\max r(x) \\ \text{s.t. } &p(x) = 0, \forall p \in \mathcal{P} \\ &q(x) \geq 0, \forall q \in \mathcal{Q}. \end{aligned}$$

One natural way to try and solve this optimization problem is to guess a θ and try to prove that $\theta - r(\alpha) \geq 0$ for all α satisfying the constraints. Then we can use binary search to try and find the smallest such θ. One way to try to prove this is to try and find a $\mathrm{PC}_{>}$ proof of non-negativity for $\theta - r(x)$ from $\mathcal{P}$ and $\mathcal{Q}$. As discussed in Section 2.3, any such proof of degree at most d can be found by writing a semidefinite program of size $n^{O(d)}$ whose constraints use numbers which require a number of bits polynomial in $\log \|r\|$, $\log \|\mathcal{P}\|$, and $\log \|\mathcal{Q}\|$. Solving this SDP is called the degree-d Sum-of-Squares relaxation.

The Ellipsoid Algorithm is commonly cited as a tool that will solve SDPs in polynomial time, and thus it is often claimed that the Sum-of-Squares relaxation can be implemented in polynomial time. Unfortunately, as first pointed out by Ryan O'Donnell in [54], the Ellipsoid Algorithm actually has some technical requirements to ensure that it actually runs in polynomial time, one of which is that the feasible region of the SDP must intersect a ball of radius R centered at the origin such that $\log R$ is polynomial. This is often not an issue, but when trying to argue that a $\mathrm{PC}_{>}$ proof can be found in polynomial time *singularly because* it is low-degree, then the situation is not so clear. The potential problem is that $\theta - r(x)$ may have a degree-d proof of non-negativity, but that proof may have to contain coefficients of size so enormous that $\log R$ is not polynomial in $\log \|r\|$, $\log \|\mathcal{P}\|$, and $\log \|\mathcal{Q}\|$. In this case if our intention is to use the SOS SDP to brute-force over all degree-d $\mathrm{PC}_{>}$

proofs of non-negativity, we would have to run the Ellipsoid Algorithm for exponential time. Indeed, O'Donnell gave an example of a polynomial system and a polynomial r which had degree two proofs of non-negativity, but all of them necessarily contained coefficients of doubly-exponential size. In this chapter we develop some of the first results on when the Sum-of-Squares relaxation for the optimization problem described by $(r, \mathcal{P}, \mathcal{Q})$ is guaranteed to run in polynomial time. We show how to use effective derivations to argue that the bit complexity of $\mathrm{PC}_>$ proofs of non-negativity is polynomially bounded.

We will conclude this chapter by strengthening the example of Ryan O'Donnell which showed that there are polynomial optimization problems whose low-degree proofs of non-negativity always contain coefficients of doubly exponential size. We show that, despite his hopes in [54], there are even Boolean polynomial optimization problems exhibiting this phenomenon.

4.1 Conditions, Definitions, and the Main Result

As O'Donnell's example shows, we cannot hope to prove that the Sum-of-Squares relaxation will always run in polynomial time. We must impose some conditions on the optimization problem defined by $(r, \mathcal{P}, \mathcal{Q})$ in order to guarantee a polynomial runtime. First, we will assume that the solution space $S = V(\mathcal{P}) \cap H(\mathcal{Q})$ is reasonably bounded, specifically that $\|S\| \leq 2^{\mathrm{poly}(n)}$. This will be the case for all of the combinatorial problems we consider (they actually have $\|S\| \leq 1$).

Our main theorem is that if there exists a special distribution μ over $V(\mathcal{P})$ satisfying three conditions, then any $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}$ and $\mathcal{Q}$ can be taken to have polynomial bit complexity. The conditions are quite general and we believe they apply to a wide swathe of problems beyond those that we prove here. In fact, they depend only on the solution space of $(\mathcal{P}, \mathcal{Q})$, so we drop the dependence on r. We explain the three conditions we require below.

Definition 4.1.1. For $\epsilon > 0$, we say that μ *ϵ-robustly satisfies* the inequalities $\mathcal{Q}$ if $q(\alpha) \geq \epsilon$ for each $\alpha \in \mathrm{supp}(\mu)$ and $q \in \mathcal{Q}$.

We require ϵ-robustness because our analysis will end up treating the constraints in $\mathcal{P}$ differently from the constraints in $\mathcal{Q}$. Because of this, we can only hope for our analysis to hold under ϵ-robustness, since otherwise one could simulate a constraint from $\mathcal{P}$ simply by having both p and $-p$ in $\mathcal{Q}$.

Definition 4.1.2. Recall we use $\mathbf{x}^{\otimes \mathbf{d}}$ denote the vector whose entries are all the monomials in $\mathbb{R}[x_1, \ldots, x_n]$ up to total degree d. For a point $\alpha \in \mathbb{R}^n$, we use $\mathbf{x}^{\otimes \mathbf{d}}(\alpha)$ to denote the vector whose entries have each been evaluated at α. For a distribution μ on $V(\mathcal{P})$, we define the *μ-moment matrix up to level d*:

$$M_{\mu,d} = \mathop{\mathbb{E}}_{\alpha \sim \mu} \left[\mathbf{x}^{\otimes \mathbf{d}}(\alpha)\mathbf{x}^{\otimes \mathbf{d}}(\alpha)^T\right]$$

Clearly $M_{\mu,d}$ is a PSD matrix, and furthermore it encodes information about the distribution μ. For example, if we let $\tilde{c} \in \mathbb{R}^{\binom{n+d-1}{d}}$, then $\tilde{c}$ corresponds to the polynomial $c(x) = \tilde{c} \cdot \mathbf{x}^{\otimes \mathbf{d}}$, and then $\tilde{c}^T M_{\mu,d} \tilde{c} = \mathbb{E}_{\alpha\sim\mu}\left[c(\alpha)^2\right]$. In particular, if $\tilde{c}$ is a zero eigenvector of $M_{\mu,d}$, then $c(x)$ is zero on all of S.

Definition 4.1.3. We say that μ is *δ-spectrally rich up to degree d* if every nonzero eigenvalue of $M_{\mu,d}$ is at least δ.

If μ is δ-spectrally rich up to degree d and p is an arbitrary polynomial of degree at most d, then there exists a polynomial p' such that $p'(\alpha) = p(\alpha)$ for each $\alpha \in \operatorname{supp}(\mu)$ and $\|p'\| \leq \frac{1}{\delta} \max_\alpha |p'(\alpha)|$. Thus spectral richness can be thought of as ensuring that the polynomials which are not zero on all of $\operatorname{supp}(\mu)$ can be bounded. What about the polynomials that are zero on $\operatorname{supp}(\mu)$? We need to ensure that we can bound those as well, or else a $\mathrm{PC}_>$ proof could require one with enormous coefficients. The key is that, since a bounded degree PC derivation is a *linear* system, its solution can be taken to have bounded coefficients.

Definition 4.1.4. We say that $\mathcal{P}$ is *k-complete for* $\operatorname{supp}(\mu)$ *up to degree* d if, for every zero eigenvector $\tilde{c}$ of $M_{\mu,d}$, the degree-d polynomial $c(x) = \tilde{c}^T \mathbf{x}^{\otimes \mathbf{d}}$ has a derivation from $\mathcal{P}$ in degree k.

If μ has support over all of $V(\mathcal{P})$, then k-completeness up to degree d is implied by $\mathcal{P}$ being k/d-effective. What if the support of μ is some smaller subset? Well, $\operatorname{supp}(\mu)$ had better at least be very close to $V(\mathcal{P})$, otherwise there is no hope that $\mathcal{P}$ is complete for $\operatorname{supp}(\mu)$ up to degree d. In fact, if $\operatorname{supp}(\mu) \neq V(\mathcal{P})$, it is impossible for every polynomial that is zero on $\operatorname{supp}(\mu)$ to have a derivation from $\mathcal{P}$, since in this case $I(\operatorname{supp}(\mu)) \neq \langle \mathcal{P} \rangle$. However, since we are only dealing with $\mathrm{PC}_>$ proofs up to degree d, we only actually care about polynomials up to degree d. In other words, we want $\operatorname{supp}(\mu)$ to be close enough to $V(\mathcal{P})$ that only polynomials of degree higher than d can tell the difference.

Example 4.1.5. Let μ be the uniform distribution over $S = \{0,1\}^n \setminus (0,0,\ldots,0)$. Then $\mathcal{P} = \{x_i^2 - x_i \mid i \in [n]\}$ is 1-complete for S up to degree $n-1$. To see this, let $r(x)$ be a polynomial which is zero on all of S, but $r \notin \langle \mathcal{P} \rangle$. Then $r(0,0,\ldots,0) \neq 0$, and has the unique multilinearization

$$\tilde{r}(x) = r(0,0,\ldots,0) \prod_{i=1}^{n} (1 - x_i),$$

and thus the degree of r must be n.

Example 4.1.6. Let μ be the uniform distribution over $S = \{0,1\}^n \setminus \{(1,y) \mid y \in \{0,1\}^{n-1}\}$. Then $\mathcal{P} = \{x_i^2 - x_i \mid i \in [n]\}$ is not k-complete for S up to degree d for any $k \geq d \geq 1$. To see this, note that the polynomial x_1 is zero on all of S, and thus corresponds to a zero eigenvector of $M_{\mu,d}$. But x_1 is not zero on $V(\mathcal{P})$, so $x \notin \langle \mathcal{P} \rangle$, and thus x has no derivation from $\mathcal{P}$ at all.

In order for μ to be robust, it must have support only in $S = V(\mathcal{P}) \cap H(\mathcal{Q})$. In this case, completeness implies that the additional constraints $q(x) \geq 0$ for each $q \in \mathcal{Q}$ do not themselves imply a low-degree polynomial equality not already derivable from $\mathcal{P}$. We consider this part of the condition to be extremely mild, because one could simply add such a polynomial equality to the constraints $\mathcal{P}$ of the program.

Example 4.1.7. Let $\mathcal{P} = \{x_i^2 - x_i \mid i \in [n]\}$ and $\mathcal{Q} = \{2 - \sum_{i=2}^n x_i\}$. Then $S = V(\mathcal{P}) \cap H(\mathcal{Q})$ is the set of binary strings with at most two ones. $\mathcal{P}$ is not k-complete up to degree 3 for any distribution with $\mathrm{supp}(\mu) = S$ for any k because $x_1x_2x_3$ is zero on S but clearly not on $V(\mathcal{P})$. However, $\mathcal{P}' = \mathcal{P} \cup \{x_ix_jx_k \mid i, j, k \in [n] \text{ and distinct}\}$ is 1-complete for S.

Finally, we compile all of the conditions together:

Definition 4.1.8. We say that $(\mathcal{P}, \mathcal{Q})$ admits a (ϵ, δ, k)-*rich solution space up to degree* d with certificate μ if there exists a distribution μ over $V(\mathcal{P}) \cap H(\mathcal{Q})$ which ϵ-robustly satisfies $\mathcal{Q}$, is δ-spectrally rich, and for which $\mathcal{P}$ is k-complete, all up to degree d. If $1/\epsilon = 2^{\mathrm{poly}(n^d)}$, $1/\delta = 2^{\mathrm{poly}(n^d)}$, and $k = O(d)$, we simply say that $(\mathcal{P}, \mathcal{Q})$ has a *rich* solution space up to degree d.

Armed with all of these definitions, we can finally formally state the main result of this chapter:

Theorem 4.1.9. *Assume that* $\|\mathcal{P}\|, \|\mathcal{Q}\|, \|r\| \leq 2^{\mathrm{poly}(n^d)}$. *Let* $(\mathcal{P}, \mathcal{Q})$ *admit an* (ϵ, δ, k)*-rich solution space up to degree d with certificate* μ. *Then if* $r(x)$ *has a* $PC_>$ *proof of non-negativity from* $\mathcal{P}$ *and* $\mathcal{Q}$ *in degree at most d, it also has a* $PC_>$ *proof of non-negativity from* $\mathcal{P}$ *and* $\mathcal{Q}$ *in degree* $O(d)$ *such that the coefficients of every polynomial appearing in the proof are bounded by* $2^{\mathrm{poly}(n^k, \log\frac{1}{\delta}, \log\frac{1}{\epsilon})}$.

In particular, if $(\mathcal{P}, \mathcal{Q})$ *has a rich solution space up to degree d, then every coefficient in the proof can be written with only* $\mathrm{poly}(n^d)$ *bits, and the degree-*$O(d)$ *Sum-of-Squares relaxation of* $(r, \mathcal{P}, \mathcal{Q})$ *runs in polynomial time via the Ellipsoid Algorithm.*

We delay the proof of Theorem 4.1.9 until Section 4.4. First, we offer some discussion on the restrictiveness of each of the three requirements of richness and collect some example optimization problems which admit rich solution spaces.

4.2 How Hard is it to be Rich?

For the rest of this chapter, we pick μ to be the uniform distribution over $S = V(\mathcal{P}) \cap H(\mathcal{Q})$. For all of the examples we considered, this was sufficient to exhibit a rich certificate. We will abuse terminology a little bit and use μ and S interchangeably. Here we will argue that robustness is easily achieved, and if S lies inside the hypercube $\{0, 1\}^n$, then it is naturally spectrally rich. Because most combinatorial optimization problems have Boolean constraints, their solution spaces lie inside the hypercube. This means that the main interesting property is the completeness of $\mathcal{P}$ for S.

Robust Satisfaction

How difficult is it to ensure that S robustly satisfies the inequalities $\mathcal{Q}$? For one, if $\epsilon = \min_{q\in\mathcal{Q}} \min_{\alpha\in V(\mathcal{P})\setminus H(\mathcal{Q})} |q(\alpha)| > 0$, then we can perturb the constraints in $\mathcal{Q}$ slightly without changing the underlying solution space S so that S $\epsilon/2$-robustly satisfies $\mathcal{Q}$. Simply make $\mathcal{Q}'$ by replacing each $q \in \mathcal{Q}$ with $q' = q + \epsilon/2$. Clearly for $\alpha \in S$, $q'(\alpha) = q(\alpha) + \epsilon/2 \geq \epsilon/2$. Furthermore, we still have $S = V(\mathcal{P}) \cap H(\mathcal{Q}')$ by the definition of ϵ. For many combinatorial optimization problems, their solution spaces are discrete and separated, and so this ϵ is appreciably large, so there is no issue.

Example 4.2.1. Consider the Balanced-Separator constraints: $\mathcal{P} = \{x_i^2 - x_i \mid i \in [n]\}$ and $\mathcal{Q} = \{2n/3 - \sum_i x_i, \sum_i x_i - n/3\}$. The solution space S is the set of binary strings with between $n/3$ and $2n/3$ ones. If n is divisible by 3, then S does not robustly satisfy $\mathcal{Q}$, since there are strings with exactly $n/3$ ones. However there is a very simple fix by setting $\mathcal{Q}' = \{2n/3 + 1/2 - \sum_i x_i, \sum_i x_i + 1/2 - n/3\}$. Then S is 1/2-robust for $\mathcal{Q}'$, and since $\sum_i x_i$ is a sum of Boolean variables, any point in $V(\mathcal{P})$ changes the sum by integer numbers. Thus adding 1/2 to the constraints does not change $V(\mathcal{P}) \cap H(\mathcal{Q})$.

While we do not have a generic theorem that shows most problems satisfy robust satisfaction, we have not yet encountered a situation where it was the bottleneck. The technique described above has always sufficed.

Spectral Richness

Recall that S is δ-spectrally rich if the moment matrix $M_{S,d}$ has only nonzero eigenvalues of size at least δ. When S lies in the hypercube, we can achieve a bound for its spectral richness using this simple lemma:

Lemma 4.2.2. *Let $M \in \mathbb{R}^{N\times N}$ be an integer matrix with $|M_{ij}| \leq B$ for all $i, j \in [N]$. The smallest non-zero eigenvalue of M is at least $(BN)^{-N}$.*

Proof. Because M is PSD, it has a full-rank principal minor A. Without loss of generality, let A be the upper-left block of M. We claim the least eigenvalue of A lower bounds the least nonzero eigenvalue of M. Since M is symmetric, there must be a C such that

$$M = \begin{bmatrix} I \\ C \end{bmatrix} A \begin{bmatrix} I & C^T \end{bmatrix}.$$

Let $P = [I, C^T]$, ρ be the least eigenvalue of A, and x be a vector perpendicular to the zero eigenspace of P. Then we have $x^T M x \geq \rho x^T P^T P x$, but x is perpendicular to the zero eigenspace of $P^T P$. Now $P^T P$ has the same nonzero eigenvalues as $PP^T = I + C^T C \succeq I$, and thus $x^T P^T P x \geq 1$, and so every nonzero eigenvalue of M is at least ρ. Now A is a full-rank bounded integer matrix with dimension at most N. The magnitude of its determinant is at least 1 and all eigenvalues are at most $N \cdot B$. Therefore, its least eigenvalue must be at least $(BN)^{-N}$ in magnitude. $\square$

As a corollary, we get:

Corollary 4.2.3. *Let $\mathcal{P}$ and $\mathcal{Q}$ be such that $S \subseteq \{0, \pm 1\}^n$. Then S is δ-spectrally rich with $\frac{1}{\delta} = 2^{\mathrm{poly}(n^d)}$.*

Proof. Recall $M_{S,d} = \mathbb{E}_{\alpha \in S}[\mathbf{x}^{\otimes \mathbf{d}}(\alpha)\mathbf{x}^{\otimes \mathbf{d}}(\alpha)^T]$, and note that $|S| \cdot M$ is an integer matrix with entries at most 3^n. The result follows by applying Lemma 4.2.2. □

Most combinatorial optimization problems are inherently discrete by nature, and so their polynomial formulations can naturally be taken to have solution spaces in $\mathbb{Z}^n$. In this case some multiple of their moment matrices are integer matrices, and we can use Lemma 4.2.2 to show spectral-richness. Even when not dealing with combinatorial optimization, it is possible to prove spectral richness as we will see with Unit Vector later. For these reasons, we consider spectral richness to be a mild condition.

Completeness

Recall that if $S = V(\mathcal{P})$, then $\mathcal{P}$ being k-complete for S up to degree d is equivalent to $\mathcal{P}$ being k/d-effective. Furthermore, it is easy to see that if there is a polynomial $p \in \langle \mathcal{P} \rangle$ of degree d which does not have a degree-k derivation from $\mathcal{P}$, then $\mathcal{P}$ cannot be complete for any subset $S \subseteq V(\mathcal{P})$. Thus in order to prove that $\mathcal{P}$ is k-complete for some subset S up to degree d, we *must* at least prove that $\mathcal{P}$ is k/d-effective. As we saw in Chapter 3, proving this is often tricky, and there is not yet any general theory for it. On the bright side, because the previous two conditions are so mild, it is often the case that completeness is the only problem to deal with before being able to conclude that the Sum-of-Squares relaxation is efficient. This fact is one of the main motivations behind our study of effective derivations. Because of the lack of a general theory for effective derivations, we also lack a general theory for giving low bit complexity proofs of non-negativity, and so we apply Theorem 4.1.9 on a case-by-case basis. However, in this chapter we at least compile a list of some combinatorial problems to which Theorem 4.1.9 applies. These problems arise from common applications of the Sum-of-Squares relaxations.

4.3 Optimization Problems with Rich Solution Spaces

Before we assemble the list in full, we give two more problems that have rich solution spaces.

Lemma 4.3.1. *The* Unit-Vector *problem has a formulation on n variables with constraints $\mathcal{P}_{\mathrm{UV}} = \{\sum_{i=1}^n x_i^2 - 1\}$. Then the uniform distribution over $S = V(\mathcal{P})$ is rich for $\mathcal{P}$ up to any degree.*

Proof. To prove spectral richness, we note that in [25] the author gives an exact formula for each entry of the matrix $M_{S,d} = \int_S m(x)$ for any monomial p. The formulas imply that

$(n+d)!\pi^{-n/2}M$ is an integer matrix with entries (very loosely) bounded by $(n+d)!d!2^n$. By Lemma 4.2.2, we conclude that S is δ-spectrally rich with $1/\delta = 2^{\mathrm{poly}(n^d)}$.

Since $\langle \mathcal{P} \rangle$ has only a single generator, to prove that $\mathcal{P}$ is d-complete for S, all we have to do is show that every element of $I(S)$ is a multiple of $\sum_i x_i^2 - 1$. Let $p(x)$ be any degree-d polynomial which is zero on the unit sphere $S = V(\mathcal{P})$, and define the even part of p, $p_0(x) = p(x) + p(-x)$. Clearly p_0 is also zero on the unit sphere, with degree $k = 2\lfloor (d+1)/2 \rfloor$. Note that p_0 has only terms of even degree. Define a sequence of polynomials $\{p_i\}_{i \in \{0,\ldots,k/2\}}$ as follows. Define q_i to be the part of p_i which has degree strictly less than k, and let $p_{i+1} = p_i + q_i \cdot (\sum_i x_i^2 - 1)$. Then each p_i is zero on the unit sphere and has no monomials of degree strictly less than $2i$. Thus $p_{k/2}$ is homogeneous of degree k. But then $p_{k/2}(tx) = t^k p_{k/2}(x) = 0$ for any unit vector x and $t > 0$, and thus $p_{k/2}(x)$ must be the zero polynomial. This implies that p_0 is a multiple of $\sum_i x_i^2 - 1$, since each $p_{i+1} - p_i$ is a multiple of $\sum_i x_i^2 - 1$. The same logic shows that the odd part of p, $p(x) - p(-x)$, is also a multiple of $\sum_i x_i^2 - 1$, and thus so is $p(x)$. Now since every element of $\langle \mathcal{P} \rangle$ must be a multiple of $\sum_i x_i^2 - 1$, obviously $\mathcal{P}$ is 1-effective, so $\mathcal{P}$ is d-complete for S up to degree d for any d. $\square$

Lemma 4.3.2. *Consider the* BALANCED SEPARATOR *formulation* $\mathcal{P} = \{x_i^2 - x_i \mid i \in [n]\}$ *and* $\mathcal{Q} = \{1/100 + 2n/3 - \sum_i x_i, 1/100 + \sum_i x_i - n/3\}$. *Then the uniform distribution over* $S = V(\mathcal{P}) \cap H(\mathcal{Q})$ *is rich for* $(\mathcal{P}, \mathcal{Q})$ *up to degree* $n/3$.

Proof. First, S is clearly 1/100-robust for $\mathcal{Q}$, even if n is divisible by three. Second, $S \subseteq \{0,1\}^n$, so by Corollary 4.2.3 it is spectrally rich. To prove completeness, we note that $\mathcal{P}$ is 1-effective by Corollary 3.1.2. It remains to prove that $\mathcal{Q}$ does not introduce additional low-degree polynomial equalities. Suppose r is a polynomial that is zero on S. Without loss of generality, we may assume that r is multilinear by using the constraints $\{x_i^2 - x_i \mid i \in [n]\}$. Then consider the symmetrized polynomial $r^* = \frac{1}{n!}\sum_{\sigma \in S_n} \sigma r$, where σ acts by $\sigma x_i = x_{\sigma(i)}$. Then because $\mathcal{P}$ and $\mathcal{Q}$ are fixed by this action, r^* must also evaluate to zero on S. Because r^* is symmetric and multilinear, it is a linear combination of the elementary symmetric polynomials $e_k(x)$. However, a simple induction shows that there is a univariate polynomial q_k of degree k for each k such that $e_k(x) - q_k(\sum_i x_i) \in \langle \mathcal{P} \rangle$. In particular this implies there is a univariate polynomial $q(t)$ with $\deg q \le r^* = \deg r$ such that $q(\sum_i x_i)$ is zero on S. This univariate polynomial has $n/3$ zeros since S has points with $n/3$ different possible values for $\sum_i x_i$. But q cannot be the zero polynomial because it is non-zero on $V(\mathcal{P})$, so q has degree at least $n/3$, and so does r. Thus every non-zero multilinear polynomial that is zero on S but not in $\langle \mathcal{P} \rangle$, has degree at least $n/3$, and $\mathcal{P}$ is 1-complete for S up to degree $n/3$. $\square$

Finally, we collect all the problems discussed:

Corollary 4.3.3. *For the following combinatorial optimization problems, the uniform distribution over* $S = V(\mathcal{P}) \cap H(\mathcal{Q})$ *is a rich certificate up to any degree:*

- CSP*:* $\mathcal{P}_{\mathrm{CSP}}(n) = \{x_i^2 - x_i \mid i \in [n]\}$.
- CLIQUE*:* $\mathcal{P}_{\mathrm{CLIQUE}}(V, E) = \{x_i^2 - x_i \mid i \in V\} \cup \{x_i x_j \mid (i,j) \notin E\}$.

- MATCHING*:*

$$\begin{aligned}\mathcal{P}_{\mathrm{M}}(n) = &\left\{x_{ij}^2 - x_{ij} \mid i,j \in [n]\right\}\\ &\cup \left\{\sum_i x_{ij} - 1 \mid j \in [n]\right\}\\ &\cup \{x_{ij}x_{ik} \mid i,j,k \in [n], j \neq k\}\,.\end{aligned}$$

- TSP*:*

$$\begin{aligned}\mathcal{P}_{\mathrm{TSP}}(n) = &\left\{x_{ij}^2 - x_{ij} \mid i,j \in [n]\right\}\\ &\cup \left\{\sum_i x_{ij} - 1 \mid j \in [n]\right\}\\ &\cup \{x_{ij}x_{ik}, x_{ji}x_{ki} \mid i,j,k \in [n], j \neq k\}\,.\end{aligned}$$

- BISECTION*:* $\mathcal{P}_{\mathrm{BCSP}}(n, n/2) = \{x_i^2 - x_i \mid i \in [n]\} \cup \left\{\sum_i x_i - \frac{n}{2}\right\}$.
- UNIT-VECTOR*:* $\mathcal{P}_{\mathrm{UV}} = \{\sum_i x_i^2 - 1\}$.

For the following optimization problems, S is a rich certificate up to degree c:

- BALANCED SEPARATOR*:* $\mathcal{P}_{\mathrm{BS}}(3c) = \{x_i^2 - x_i \mid i \in [3c]\}$*,* $\mathcal{Q}_{\mathrm{BS}}(3c, c) = \{1/100 + 2c - \sum_{i=1}^{3c} x_i, 1/100 + \sum_{i=1}^{3c} x_i - c\}$*.*
- BALANCED CSP*:* $\mathcal{P}_{\mathrm{BCSP}}(n, c) = \{x_i^2 - x_i \mid i \in [n]\} \cup \{\sum_{i=1}^n x_i - c\}$*.*
- BOOLEAN SPARSE PCA*:* $\mathcal{P}_{\mathrm{SPCA}}(n, 2c) = \{x_i^3 - x_i \mid i \in [n]\} \cup \{\sum_i x_i^2 - 2c\}$*.*

Proof. UNIT-VECTOR and BALANCED SEPARATOR were discussed above. For all the other problems, $S \subseteq \{0, \pm 1\}^n$, so by Corollary 4.2.3, S is spectrally rich. Furthermore, for these problems, $\mathcal{P}$ was proven to admit effective derivations in Chapter 3 (see Corollary 3.7.1), and $\mathcal{Q}$ is empty, so $S = V(\mathcal{P})$. Thus $\mathcal{P}$ is k-complete for S up to the appropriate degree d, with $k = O(d)$. □

4.4 Proof of the Main Theorem

(Proof of Theorem 4.1.9). For convenience, we write $\mathcal{P} = \{p_1, \dots, p_m\}$ and $\mathcal{Q} = \{q_1, \dots, q_\ell\}$. Let μ be the certificate for (ϵ, δ, k)-richness of $(\mathcal{P}, \mathcal{Q})$, let $S = \mathrm{supp}(\mu)$, and let $r(x)$ be a degree-d polynomial which has a $\mathrm{PC}_{>}$ proof of non-negativity from $(\mathcal{P}, \mathcal{Q})$. In other words, there is a polynomial identity

$$r(x) = \sum_{i=1}^{t_0} h_i^2 + \sum_{i=1}^{\ell} \left(\sum_{j=1}^{t_i} h_{ij}^2\right) q_i + \sum_{i=1}^{m} \lambda_i p_i.$$

Our goal is to find a potentially different $\mathrm{PC}_>$ proof of non-negativity for r which uses only polynomials of bounded norm.

First, we rewrite the original $\mathrm{PC}_>$ proof into a more convenient form before proving bounds on each individual term. Because the elements of $\mathbf{x}^{\otimes \mathbf{d}}$ are a basis for $\mathbb{R}[x_1, \dots, x_n]_d$, every polynomial in the proof can be expressed as $\tilde{c}^T \mathbf{x}^{\otimes \mathbf{d}}$, where $\tilde{c}$ is a vector of reals:

$$\begin{aligned} r(x) &= \sum_{i=1}^{t_0} (\tilde{h}_i^T \mathbf{x}^{\otimes \mathbf{d}})^2 + \sum_{i=1}^{\ell} \left(\sum_{j=1}^{t_i} (\tilde{h}_{ij}^T \mathbf{x}^{\otimes \mathbf{d}})^2 \right) q_i + \sum_{i=1}^{m} \lambda_i p_i \\ &= \langle H, \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \rangle + \sum_{i=1}^{\ell} \langle H_i, \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \rangle q_i + \sum_{i=1}^{m} \lambda_i p_i \end{aligned}$$

for PSD matrices H, $H_1, \dots, H_\ell$. Next, we average this polynomial identity via the distribution μ:

$$\mathop{\mathbb{E}}_{\alpha \sim \mu} [r(\alpha)] = \left\langle H, \mathop{\mathbb{E}}_{\alpha \sim \mu} \left[\mathbf{x}^{\otimes \mathbf{d}}(\alpha) \mathbf{x}^{\otimes \mathbf{d}}(\alpha)^T \right] \right\rangle + \sum_{i=1}^{\ell} \left\langle H_i, \mathop{\mathbb{E}}_{\alpha \sim \mu} \left[q_i(\alpha) \mathbf{x}^{\otimes \mathbf{d}}(\alpha) \mathbf{x}^{\otimes \mathbf{d}}(\alpha)^T \right] \right\rangle + 0$$

The LHS is at most $\mathrm{poly}(\|r\|, \|S\|)$. The RHS is a sum of positive numbers, since the inner products are over pairs of PSD matrices (recall $q_i(\alpha) \geq \epsilon > 0$). Thus the LHS is an upper bound on each term of the RHS. We would like to say that since S is δ-spectrally rich, the first term is at least $\delta \operatorname{Tr}(H)$. Unfortunately the averaged matrix may have zero eigenvectors, and it is possible that H could have very large eigenvalues in these directions. However, because $\mathcal{P}$ is k-complete for S, these can be absorbed into the final term at the cost of increasing the degree to k. More formally, let $\Pi = \sum_u u u^T$ be the projector onto the zero eigenspace of $M_{\mu,d} = \mathbb{E}_{\alpha \sim \mu}[\mathbf{x}^{\otimes \mathbf{d}}(\alpha) \mathbf{x}^{\otimes \mathbf{d}}(\alpha)^T]$. Because $\mathcal{P}$ is k-complete for S, for each u there is a degree-k derivation $u^T \mathbf{x}^{\otimes \mathbf{d}} = \sum_i \sigma_{ui} p_i$. Then $\Pi \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T = \sum_u (u^T \mathbf{x}^{\otimes \mathbf{d}}) \cdot u (\mathbf{x}^{\otimes \mathbf{d}})^T$. Thus we can write

$$\begin{aligned} \langle H, \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \rangle &= \left\langle H, (\Pi + \Pi^\perp) \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T (\Pi + \Pi^\perp) \right\rangle \\ &= \left\langle H, \Pi^\perp \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \Pi^\perp \right\rangle + \sum_u u^T \mathbf{x}^{\otimes \mathbf{d}} \left(\left\langle H, \Pi^\perp \mathbf{x}^{\otimes \mathbf{d}} u^T + \mathbf{x}^{\otimes \mathbf{d}} u^T \Pi^\perp + \mathbf{x}^{\otimes \mathbf{d}} u^T \Pi \right\rangle \right) \\ &= \left\langle \Pi^\perp H \Pi^\perp, \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \right\rangle + \sum_i \sigma_i p_i, \end{aligned}$$

for some polynomials σ_i. Doing the same for the other terms and setting $H' = \Pi^\perp H \Pi^\perp$ and similarly for H_i', we get a new proof:

$$r(x) = \langle H', \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \rangle + \sum_{i=1}^{\ell} \langle H_i', \mathbf{x}^{\otimes \mathbf{d}} (\mathbf{x}^{\otimes \mathbf{d}})^T \rangle q_i + \sum_{i=1}^{m} \lambda_i' p_i.$$

Now the zero eigenspace of H' is contained in the zero eigenspace of $M_{\mu,d}$. Furthermore, the δ-spectral richness of μ implies that each nonzero eigenvalue of $M_{\mu,d}$ is at least δ, so $\langle H', M_{\mu,d}\rangle \geq \delta\,\mathrm{Tr}(H')$. Also, the ϵ-robustness of μ implies that $q_i(\alpha) \geq \epsilon$ for each i and α. Thus

$$\left\langle H'_i, \mathop{\mathbb{E}}_{\alpha\sim\mu}\left[q_i(\alpha)\mathbf{x}^{\otimes\mathbf{d}}(\alpha)(\mathbf{x}^{\otimes\mathbf{d}}(\alpha))^T\right]\right\rangle \geq \left\langle H'_i, \mathop{\mathbb{E}}_{\alpha\sim\mu}\left[\epsilon\mathbf{x}^{\otimes\mathbf{d}}(\alpha)(\mathbf{x}^{\otimes\mathbf{d}}(\alpha))^T\right]\right\rangle \geq \epsilon\delta\,\mathrm{Tr}(H'_i).$$

Thus, after averaging we have

$$\mathrm{poly}(\|r\|, \|S\|) \geq \delta\,\mathrm{Tr}(C) + \sum_{i=1}^{\ell} \delta\epsilon\,\mathrm{Tr}(H'_i).$$

Every entry of a PSD matrix is bounded by the trace, so H' and each H'_i have entries bounded by $\mathrm{poly}(\|r\|, \|S\|, \frac{1}{\delta}, \frac{1}{\epsilon})$.

The only thing left to do is to bound the coefficients λ'_i. This turns out to be easy because the $\mathrm{PC}_>$ proof is linear in these coefficients. If we imagine the coefficients of the λ'_i as variables, then the linear system induced by the polynomial identity

$$r(x) - \langle H', \mathbf{x}^{\otimes\mathbf{d}}(\mathbf{x}^{\otimes\mathbf{d}})^T\rangle - \sum_{i=1}^{\ell}\langle H'_i, \mathbf{x}^{\otimes\mathbf{d}}(\mathbf{x}^{\otimes\mathbf{d}})^T\rangle = \sum_{i=1}^{m}\lambda'_i p_i$$

is clearly feasible, and the coefficients of the LHS are bounded by $\mathrm{poly}(\|r\|, \|S\|, \frac{1}{\delta}, \frac{1}{\epsilon})$. There are $O(n^k)$ variables, so by Cramer's rule, the coefficients of the λ'_i can be taken to be bounded by $\mathrm{poly}(\|\mathcal{P}\|^{n^k}, \frac{1}{\delta}, \frac{1}{\epsilon}, \|r\|, \|S\|, n!)$. By assumption, $\|\mathcal{P}\|, \|r\|, \|S\| \leq 2^{\mathrm{poly}(n^d)}$. Thus this bound is at most $2^{\mathrm{poly}(n^k, \log\frac{1}{\delta}, \log\frac{1}{\epsilon})}$. □

4.5 A Polynomial System with No Efficient Proofs

In [54], Ryan O'Donnell gave the first example of a set of polynomials $\mathcal{P}$ and a polynomial r which has a degree two $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}$, but *any* such degree two proof must necessarily contain polynomials with doubly-exponential coefficients. In his paper, he was the first to point out the trouble with the Sum-of-Squares relaxation that we have endeavored to address in this chapter. He also hoped that if $\mathcal{P}$ was a *Boolean* system, i.e. $\{x_i^2 - x_i \mid i \in [n]\} \subseteq \mathcal{P}$, then any $\mathrm{PC}_>$ proof from $\mathcal{P}$ could be taken to have polynomial bit complexity. Unfortunately, in this section we answer this question in the negative. We develop a polynomial system containing the Boolean constraints, but which still has polynomials with proofs of non-negativity that require polynomials of doubly-exponential size. Furthermore, our construction also holds even for proofs of high degree. In O'Donnell's original example, the polynomial r has proofs of low bit complexity at degree four. In our example, the polynomial r has no proofs of low bit complexity until degree $\Omega(\sqrt{n})$, thus scuttling any hope of solving the bit complexity problem by simply using a higher degree Sum-of-Squares relaxation.

A First Example

The original example given in [54] essentially contains the following system whose repeated squaring is responsible for the blowup of the coefficients in the proofs:

$$\mathcal{P} = \quad \{x_1^2 - x_2, \quad x_2^2 - x_3, \quad \dots, \quad x_{n-1}^2 - x_n, \quad x_n^2\}.$$

The solution space is simply $V(\mathcal{P}) = \{(0, 0, \dots, 0)\}$, and therefore the polynomial $\epsilon - x_1$ must be non-negative over $V(\mathcal{P})$ for any $\epsilon > 0$. However, it is not obvious as to whether or not a low-degree $\mathrm{PC}_>$ proof of this non-negativity exists.

Lemma 4.5.1. *The polynomial $\epsilon - x_1$ has a degree two $PC_>$ proof of non-negativity from $\mathcal{P}$.*

Proof. The following polynomial identity implies the lemma statement:

$$\epsilon - x_1 \cong \left(\sqrt{\frac{\epsilon}{n}} - \left(\frac{n}{4\epsilon}\right)^{1/2} x_1\right)^2 + \left(\sqrt{\frac{\epsilon}{n}} - \left(\frac{n}{4\epsilon}\right)^{3/2} x_2\right)^2 + \left(\sqrt{\frac{\epsilon}{n}} - \left(\frac{n}{4\epsilon}\right)^{7/2} x_3\right)^2 + \\ + \cdots + \left(\sqrt{\frac{\epsilon}{n}} - \left(\frac{n}{4\epsilon}\right)^{(2^n-1)/2} x_n\right)^2. \qquad (*)$$

To explain a little, let the ith term in the proof be $(A_i - B_i x_i)^2$. First notice that $\sum_i A_i^2 = \epsilon$. Second, notice that $-2A_iB_ix_i = -\left(\frac{n}{4\epsilon}\right)^{2^{i-1}-1} x_i$. Third, $(B_ix_i)^2 = \left(\frac{n}{4\epsilon}\right)^{2^i-1} x_i^2 \cong \left(\frac{n}{4\epsilon}\right)^{2^i-1} x_{i+1}$. Everything has been carefully set up so that $(B_ix_i)^2 \cong -2A_{i+1}B_{i+1}$. Finally, clearly $B_n^2x_n^2 \cong 0$. Thus every term cancels out except $\sum_i A_i^2 - 2A_1B_1x_1 = \epsilon - x_1$. □

Of course, the above proof involves coefficients of doubly-exponential size, which means that it will not be found by running a polynomial time version of the Ellipsoid Algorithm. Is it possible to find a proof for $\epsilon - x_1$ that does not use coefficients of such huge size?

Lemma 4.5.2. *Let $\epsilon < 1/2$. Then any $PC_>$ proof of $\epsilon - x_1$ from $\mathcal{P}$ of degree d must involve polynomials with coefficients of size at least $\Omega\left(\frac{1}{n^d}\left(\frac{1}{2\epsilon}\right)^{2^n}\right)$.*

Proof. We will define a linear functional $\phi : \mathbb{R}[X]_d \to \mathbb{R}$ as in Lemma 2.3.15. Recall we want ϕ to satisfy the following:

(1) $\phi[\epsilon - x_1] = -\epsilon$

(2) $\phi[\sigma(x_i^2 - x_{i+1})] = 0$ for any $i \leq n-1$ and σ of degree at most $d-2$

(3) $|\phi[\lambda x_n^2]| \leq (2\epsilon)^{2^n} n^d \|\lambda\|$.

(4) $\phi[p^2] \geq 0$ for any p^2 of degree at most d

Note that any monomial is equivalent to some power of x_1. For example, $x_1x_2x_3 \cong x_1^7$. More generally, it is clear from $\mathcal{P}$ that

$$\prod_{i=1}^{n} x_i^{\beta_i} \cong x_1^{\sum_{j=1}^{n} 2^{j-1}\beta_j}.$$

Define ϕ by linearly extending its action on monomials, defined by:

$$\phi\left[\prod_{i=1}^{n} x_i^{\beta_i}\right] = (2\epsilon)^{\sum_i 2^{i-1}\beta_i}.$$

Clearly $\phi[\epsilon - x_1] = -\epsilon$, thus satisfying condition (1). Condition (2) is obviously satisfied if σ is a monomial, and linearity of ϕ implies that it holds for any polynomial σ. For condition (3), if λ is a monomial, then $\phi[\lambda x_n^2] \leq \phi[x_n^2] = (2\epsilon)^{2^n}$. If λ is not a monomial, it has at most n^d monomials, and maximum coefficient at most $\|\lambda\|$. Then by linearity of ϕ, we have $\phi[\lambda x_n^2] \leq (2\epsilon)^{2^n} n^d \|\lambda\|$. For condition (4), note that ϕ is multiplicative. Then clearly $\phi[p^2] = \phi[p]^2 \geq 0$. $\square$

Even though r does not have any efficient $\mathrm{PC}_>$ proofs of non-negativity, this example does not achieve our goal of exhibiting a system that contains all the Boolean constraints. We show how to modify it in the following section.

A Boolean System

One simple way to try to make the system Boolean is to just add the constraints $x_i^2 - x_i$ to $\mathcal{P}$. Unfortunately, this introduces new proofs for $\epsilon - x_i$, and they have low bit complexity. To see this, it is clear that $x_i^2 - x_i \cong x_{i+1} - x_i$, and by adding these together, we can get a telescoping sum and derive $x_n - x_1$. But now $x_n - x_1 \cong x_n^2 - x_1 \cong -x_1$, and thus $x_1 \in \langle \mathcal{P} \rangle_2$. Because HN proofs are linear, x_1 has a derivation $\sum_p \lambda_p p$ with low bit complexity, which can be used to write a $\mathrm{PC}_>$ proof for

$$\epsilon - x_1 = \sqrt{\epsilon}^2 + \lambda_p p.$$

By constraining the variables x_i we add new ways to formulate proofs. We want to add constraints in a way that $\mathrm{PC}_>$ proofs do not realize that the x_i are actually constrained further.

We draw inspiration from the Knapsack problem, which is known to be difficult to refute with $\mathrm{PC}_>$ proofs. We replace each instance of the variable x_i with a sum of $2k$ Boolean variables: $x_i \to \sum_j w_{ij} - k$. The new set of constraints is

$$\mathcal{P}' = \left\{ \left(\sum_j w_{ij} - k\right)^2 - \left(\sum_j w_{i+1,j} - k\right) \mid i \in [n-1] \right\}$$
$$\cup \left\{ \left(\sum_j w_{nj} - k\right)^2 \right\}$$
$$\cup \left\{ w_{ij}^2 - w_{ij} \mid i \in [n], j \in [2k] \right\}.$$

The solution space $V(\mathcal{P}')$ is the set of n bit strings of $2k$ bits, each with exactly k ones.

Lemma 4.5.3. *The polynomial $r = \epsilon - \left(\sum_j w_{1j} - k\right)$ is non-negative on $V(\mathcal{P}')$, and has a degree two $PC_>$ proof of non-negativity from $\mathcal{P}'$.*

Proof. The polynomial r is non-negative because there are exactly k ones among the w_{1j}, so $r(\alpha) = \epsilon > 0$ on $V(\mathcal{P}')$. Furthermore, r has a proof of non-negativity since we can just replace each instance of x_i with $\left(\sum_j w_{ij} - k\right)$ in ($*$). $\square$

Before we prove that the doubly-exponential coefficients are necessary, we need the following technical lemma, due to [30]:

Lemma 4.5.4. *Let $0 < \delta < 1$. Then there exists a linear function $\phi_\delta : \mathbb{R}[X]_d \to \mathbb{R}$ and a constant C satisfying, for any λ up to degree Ck,*

(1) $\phi_\delta[\lambda \cdot (w_{ij}^2 - w_{ij})] = 0$,

(2) $\phi_\delta[\lambda \cdot ((\sum_j w_{ij} - k) - \delta)] = 0$,

(3) $\phi_\delta[p^2] \geq 0$ *for any polynomial p of degree at most $Ck/2$.*

The lemma is equivalent to claiming that a PC$_>$ refutation for that system of equations (Knapsack) requires degree Ck. Since δ is not an integer and each w_{ij} is Boolean, obviously $\sum_j w_{ij} - k - \delta = 0$ is unsatisfiable, but because there is no PC$_>$ proof until degree Ck, the linear function ϕ_δ exists. We will use these linear functions to pretend that $\sum_j w_{ij} - k = (2\epsilon)^{2^{i-1}}$ and mimic the proof in Lemma 4.5.2.

Lemma 4.5.5. *Let $r = \epsilon - \left(\sum_j w_{ij} - k\right)$ and $\epsilon < 1/2$. Then any degree-Ck $PC_>$ proof of non-negativity for r from $\mathcal{P}'$ contains a polynomial of norm at least $\Omega\left(\frac{1}{(nk)^d} \cdot \left(\frac{1}{2\epsilon}\right)^{2^n}\right)$.*

Proof. Let $d \leq Ck$, $W_i = \{w_{i1}, w_{i2}, \ldots, w_{i,2k}\}$, and $W = \bigcup_i W_i$. We will use $\sigma(W)$ to denote an arbitrary monomial, and $\sigma_1(W_1), \ldots, \sigma_n(W_n)$ to be the monomials whose product is σ. We will use λ to denote an arbitrary polynomial. We will define a linear functional satisfying the requirements of Lemma 2.3.15, which will prove the theorem. Define a linear functional $\Phi : \mathbb{R}[W_1, W_2, \dots, W_n]_d \to \mathbb{R}$ by linearly extending its action on monomials to the monomial σ:

$$\Phi[\sigma] = \phi_1(\sigma_1)\phi_2(\sigma_2)\dots\phi_n(\sigma_n),$$

where each ϕ_i is the linear function $\phi_{(2\epsilon)^{2^{i-1}}}$ guaranteed to exist by Lemma 4.5.4.

First, clearly

$$\Phi\left[\epsilon - \left(\sum_j w_{1j} - k\right)\right] = \phi_1\left[\epsilon - \left(\sum_j w_{1j} - k\right)\right] = -\epsilon.$$

Second,

$$\Phi\left[\sigma \cdot (w_{ij}^2 - w_{ij})\right] = \phi_i\left[\sigma_i \cdot (w_{ij}^2 - w_{ij})\right]\prod_{j \neq i}\phi_j[\sigma_j] = 0.$$

Linearity of Φ implies the same is true for any polynomial of degree at most Ck. Similarly,

$$\begin{aligned}\Phi\left[\sigma \cdot \left(\sum_j w_{ij} - k\right)\right] &= \phi_i\left[\sigma_i \cdot \left(\sum_j w_{ij} - k\right)\right]\prod_{j \neq i}\phi_j[\sigma_j] \\ &= \phi_i\left[\sigma_i \cdot (2\epsilon)^{2^{i-1}}\right]\prod_{j \neq i}\phi_j[\sigma_j] \\ &= (2\epsilon)^{2^{i-1}}\Phi[\sigma].\end{aligned}$$

Again, linearity implies that the same holds for any polynomial of degree at most Ck. This implies that for any polynomial λ,

$$\Phi\left[\lambda \cdot \left(\left(\sum_j w_{ij} - k\right)^2 - \left(\sum_j w_{i+1,j} - k\right)\right)\right] = 0,$$

as well as

$$\left|\Phi\left[\lambda \cdot \left(\sum_j w_{nj} - k\right)^2\right]\right| = (2\epsilon)^{2^n}|\Phi[\lambda]| \leq (2\epsilon)^{2^n}(nk)^d\|\lambda\|,$$

where the $(nk)^d$ appears because there are at most that many monomials of degree d, and since every variable is Boolean, Φ is at most 1 on any monomial.

The only remaining condition to prove is that Φ is non-negative on squares. Define the linear operator $T_i : \mathbb{R}[W_1, W_2, \dots, W_i] \to \mathbb{R}[W_1, \dots, W_{i-1}]$ with $T_i[\prod_{j \leq i}\sigma_j] = \phi_i[\sigma_i] \cdot \prod_{j<i}\sigma_j$. Clearly $\Phi[\lambda] = T_1 T_2 \dots T_n[\lambda]$. We claim that for any i, and any λ, $T_i[\lambda^2]$ is a sum-of-squares polynomial. This, together with the fact that each ϕ_i is non-negative on squares, implies that Φ is non-negative on squares.

It is sufficient to prove the claim for T_2. For multisets U with elements from W_1 and V with elements from W_2, and we define $w_U = \prod_{w\in U} w$ and similarly for w_V. Write $\lambda = \sum_{UV} \alpha_{UV} w_U w_V$. Then

$$\begin{aligned} T_2[\lambda^2] &= T_2\left[\sum_{UVU'V'} \alpha_{UV}\alpha_{U'V'} w_U w_V w_{U'} w_{V'}\right] \\ &= \sum_{UVU'V'} \alpha_{UV}\alpha_{U'V'} w_U w_{U'} \phi_2[w_V w_{V'}]. \end{aligned}$$

If we define a matrix $M(V,V') = \phi_2[w_V w_{V'}]$, then because ϕ_2 is non-negative on squares, this matrix is PSD. Furthermore, define $\mathbf{w}(V) = \sum_U \alpha_{UV} w_U$. Then $T_2[\lambda^2] = \mathbf{w}^T M \mathbf{w}$. Since M is PSD, it can be written $\sum_u uu^T$ for some vectors u. Then $T_2[\lambda^2] = \sum_u \mathbf{w}^T uu^T \mathbf{w} = \sum_u (u^T\mathbf{w})^2$ is a sum of squares. □

Finally, we prove our main theorem.

Theorem 4.5.6. *There exists a set of quadratic polynomials $\mathcal{P}'$ on n variables and a polynomial r non-negative on $V(\mathcal{P}')$ such that*

- *$\mathcal{P}'$ contains the polynomial $x_i^2 - x_i$ for every $i \in [n]$.*
- *r has a degree two $PC_>$ proof of non-negativity from $\mathcal{P}'$.*
- *Every $PC_>$ proof of non-negativity for r from $\mathcal{P}'$ of degree at most $O(\sqrt{n})$ has a polynomial with a coefficient of size at least $\Omega(\frac{1}{n^d} 2^{\exp\sqrt{n}})$.*

Proof. We take the polynomial system $\mathcal{P}'$ discussed in this section with $k = n$. Then there are $N = n^2$ variables total, and the properties follow directly from Lemma 4.5.3 and Lemma 4.5.5. □

Chapter 5

Optimal Symmetric SDP Relaxations

The main result of this section is to show that when the solution space of a polynomial formulation for a combinatorial optimization problem satisfies certain symmetry properties, then the Theta Body SDP relaxation (see Section 2.6) achieves the best approximation among all symmetric SDPs of a comparable size. This is proven using an old technique of Yannakakis on the size of certain permutation groups that has been used time and time again to find optimal symmetric LP and SDP relaxations.

We combine this result with some of our results in Chapter 3 to prove that the Sum-of-Squares SDP relaxation (see Section 2.6 again) performs no worse than the Theta Body relaxation, thus showing that the SOS SDP is optimal for problems including Matching, TSP, and Balanced CSP. Furthermore, this allows us to translate lower bounds against the SOS SDP to lower bounds against any symmetric SDP formulation. We apply this to the Matching problem using the lower bound of Grigoriev [32] and get an exponential lower bound for the size of any symmetric SDP relaxation for the Matching problem.

5.1 Theta Body Optimality

Recall that a S_m-symmetric combinatorial optimization problem $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ has a symmetric polynomial formulation if two conditions hold. First, there is a polynomial optimization problem $(\mathcal{P}, \mathcal{Q}, \mathcal{O}, \phi)$ on n variables such that solving the associated optimization problem solves $\mathcal{M}$ as well. Second, there is an action of S_m on the coordinates $[n]$ (extending naturally to an action on $\mathbb{R}^n$ and $\mathbb{R}[x_1, \ldots, x_n]$) that is compatible with the action on $\mathcal{S}$: $\sigma\phi(\alpha) = \phi(\sigma\alpha)$.

Definition 5.1.1. We say that the symmetric polynomial formulation is (k_1, k_2)-*block transitive* if, for each $I \subseteq [m]$ of size at most k_1, there exists a $J \subseteq [n]$ of size at most k_2 such that $A([m] \setminus U)$ acts transitively on each $S_{J,c} = \{x \in \mathbb{R}^n : x \in V(\mathcal{P}), x|_J = c\}$, i.e. each set of solutions in $V(\mathcal{P})$ which agree on J.

Example 5.1.2. The usual formulation for the Matching problem is $\left(k, \binom{k}{2}\right)$-block transitive for each $k < m/2$. Recall the constraints of the polynomial formulation for the Matching problem on $\binom{m}{2}$ variables from (3.1). The map ϕ is defined so that for a matching M, $\phi(M) = \chi_M$, where $(\chi_M)_{ij} = 1$ if $(i,j) \in M$ and 0 otherwise. Then S_m acts by permuting the vertices of the graph.

For a subset $I \subseteq [m]$ with $|I| < m/2$, we set $J = E(I, I)$, the set of edges that lie entirely in I. Let M_1 and M_2 be two matchings that agree on J. We define a permutation σ as follows: Set σ to fix I. Because M_1 and M_2 are perfect matchings, they must have the same number of edges in both $E(I, \overline{I})$ and $E(\overline{I}, \overline{I})$. For a vertex $v \in \overline{I}$, if $M_1(v) \in I$, then we set $\sigma(v) = M_2(M_1(v))$. Otherwise, we set σ to be an arbitrary bijection between the edges of M_1 in $E(\overline{I}, \overline{I})$ and the edges of M_2 in $E(\overline{I}, \overline{I})$. Clearly $\sigma \in S([m] \setminus I)$ and $\sigma(\chi_{M_1}) = \chi_{M_2}$. If σ is even, we are done. Otherwise, since $|I| < m/2$, there is an edge $(u, v) \in M_2 \cap E(\overline{I}, \overline{I})$. Then if σ_{uv} is the transposition of u and v, $\sigma_{uv}\sigma$ is an even permutation which still fixes J and maps χ_{M_1} to χ_{M_2}.

Example 5.1.3. The usual formulation for Balanced CSP is (k, k)-block transitive for every $k \leq m-3$. Recall the constraints for the polynomial formulation for Balanced CSP on m variables from (3.7). The map ϕ is defined so that for an assignment A, $\phi(A) = \chi_A$, where $(\chi_A)_i = 1$ if $A(i) = 1$ and 0 otherwise. Then S_n acts by permuting the labels of the variables.

For a subset $I \subseteq [m]$, we set $J = I$. Let two assignments A_1 and A_2 that agree on J, and define a permutation σ as follows: χ_{A_1} and χ_{A_2} have the same number of indices which are zero, and indices which are one. Let σ be any pair of bijections between the indices which are one in χ_{A_1} and the indices which are one in χ_{A_2}, and likewise for the indices which are zero. Furthermore, since χ_{A_1} and χ_{A_2} agree on J, we can choose σ to be a pair of bijections which are the identity on J, so $\sigma \in S([m] \setminus J)$. Clearly $\sigma(\chi_{A_1}) = \chi_{A_2}$. Finally, if σ is not already even, since $|I| \leq m - 3$, there are two indices ℓ_1 and ℓ_2 outside of J such that $A_2(\ell_1) = A_2(\ell_2)$. Then $(\ell_1, \ell_2) \cdot \sigma$ is even and still fixes J and maps χ_{A_1} to χ_{A_2}.

The point of this definition is that if a polynomial formulation is block-transitive, then it is easy to show that invariant functions can be represented with low-degree polynomials. Going from arbitrary functions to low-degree polynomials is crucial to showing optimality for the Theta Body.

Lemma 5.1.4. *Let $(\mathcal{P}, \mathcal{O}, \phi)$ be a A_m-symmetric, Boolean, (k_1, k_2)-block transitive polynomial formulation and $h : V(\mathcal{P}) \to \mathbb{R}$ be a function. If there is a set I of size $|I| \leq k_1$ such that h is stabilized by $A([m] \setminus I)$ under the group action $\sigma h(\alpha) = h(\sigma^{-1}\alpha)$, then there is a polynomial $h'(x)$ such that $h'(\phi(\alpha)) = h(\phi(\alpha))$ and the degree of h' is at most k_2.*

Proof. For any $\sigma \in A([m] \setminus I)$ and $\alpha \in V(\mathcal{P})$, we know $h(\alpha) = \sigma h(\alpha) = h(\sigma^{-1}\alpha)$. By block-transitivity, there exists a set J of size $|J| \leq k_2$ such that $A([m] \setminus I)$ acts transitively on elements of $V(\mathcal{P})$ which agree on J. Thus if $\alpha, \beta \in V(\mathcal{P})$ such that $\alpha|_J = \beta|_J$, $h(\alpha) = h(\beta)$. Thus h depends only on the coordinates J, and since the polynomial formulation is Boolean, any such function can be expressed as a degree-$|J|$ polynomial. □

Before we state our main theorem, we recall that the dth Theta Body relaxation with objective $o(x)$ is

$$\begin{aligned} &\min c \\ \text{s.t. } & c - o(x) \text{ is } d\text{-SOS modulo } \langle \mathcal{P} \rangle. \end{aligned}$$

Theorem 5.1.5. *Let $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ have a S_m-symmetric (k_1, k_2)-block transitive Boolean polynomial formulation $(\mathcal{P}, \mathcal{O}, \phi)$ on n variables. Then if $\mathcal{M}$ has any (c, s)-approximate, S_m-symmetric SDP relaxation of size $r < \sqrt{\binom{m}{k_1}}$ the k_2th Theta Body relaxation is a (c, s)-approximate relaxation as well.*

Recall that the size of the k_2th Theta Body relaxation is $n^{O(k_2)}$, so if $k_2 = O(k_1)$, then the size of the Theta Body relaxation is polynomial in the size of the original symmetric formulation. Before we prove the main theorem, we need two lemmas. One has to do with obtaining sum-of-squares representations for the objective functions given a small SDP formulation:

Lemma 5.1.6. *If $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ has a (c, s)-approximate SDP formulation of size at most k, then there exist a family of $\binom{k+1}{2}$ functions $\mathcal{H}$ from $\mathcal{S}$ into $\mathbb{R}$ such that for every $f \in \mathcal{F}$, with $\max_{\alpha \in \mathcal{S}} f(\alpha) \leq s(f)$,*

$$c(f) - f = \sum_i g_i^2$$

where each $g_i \in \langle \mathcal{H} \rangle$. Furthermore, if the SDP formulation is G-coordinate-symmetric for some group G, then $\mathcal{H}$ is G-invariant under the action $\sigma h(s) = h(\sigma^{-1} s)$.

Proof. Consider the slack matrix for $\mathcal{M}$: $M(\alpha, f) = c(f) - f(\alpha)$. By Theorem 2.5.3, if there exists an SDP formulation for $\mathcal{M}$ of size k_1, then there are $k_1 \times k_1$ PSD matrices X^α and Y_f such that $M(\alpha, f) = X^\alpha \cdot Y_f + \mu_f$ for some $\mu_f > 0$. Let $\sqrt{\cdot}$ denote the unique PSD square root. We define a set of functions $\mathcal{H}$ by $h_{ij}(\alpha) = (\sqrt{X^\alpha})_{ij}$. Since $h_{ij} = h_{ji}$ there are only $\binom{k_1+1}{2}$ functions in $\mathcal{H}$. We have

$$\begin{aligned} c(f) - f(\alpha) &= X^\alpha \cdot Y_f + \mu_f \\ &= \mathrm{Tr}[\sqrt{X^\alpha}\sqrt{X^\alpha}\sqrt{Y_f}\sqrt{Y_f}] + \mu_f \\ &= \mathrm{Tr}[(\sqrt{X^\alpha}\sqrt{Y_f})^T \sqrt{X^\alpha}\sqrt{Y_f}] + \mu_f \\ &= \sum_{ij} \left(\sum_k (\sqrt{X^\alpha})_{ik} (\sqrt{Y_f})_{kj} \right)^2 + \mu_f \\ &= \sum_{ij} \left(\sum_k (\sqrt{Y_f})_{kj} h_{ik}(\alpha) \right)^2 + \mu_f. \end{aligned}$$

Lastly, $\sigma h_{ij}(\alpha) = h_{ij}(\sigma^{-1}\alpha) = \sqrt{X^{\sigma^{-1}\alpha}}_{ij} = \sqrt{\sigma^{-1}X^{\alpha}}_{ij}$. Because σ^{-1} is a coordinate permutation, its action on X^α can be written $\sigma^{-1}X^\alpha = P(\sigma)X^\alpha P(\sigma)^T$. Then since

$$\left(P(\sigma)\sqrt{X^\alpha}P(\sigma)^T\right)^2 = P(\sigma)X^\alpha P(\sigma)^T = \sigma^{-1}X^\alpha,$$

and the PSD square root is unique, we have $\sqrt{\sigma^{-1}X^\alpha} = \sigma^{-1}\sqrt{X^\alpha}$. Thus $h_{ij}(\sigma^{-1}\alpha) = \sigma^{-1}\sqrt{X^\alpha}_{ij} = \sqrt{X^\alpha}_{\sigma^{-1}i\sigma^{-1}j} = h_{\sigma^{-1}i\sigma^{-1}j}(\alpha)$, so indeed $\mathcal{H}$ is G-invariant. □

The second lemma we need is an old group-theoretic result. It has been used frequently in the context of symmetric LP and SDP formulations, see for example [47, 40, 10].

Lemma 5.1.7. *[([22], Theorems 5.2A and 5.2B)] Let $m \geq 10$ and let $G \leq S_m$. If $|S_m : G| < \binom{m}{k}$ for some $k < m/4$, then there is a subset $I \subseteq [m]$ such that $|I| < k$, and $A([m] \setminus I)$ is a subgroup of G.*

We are ready to prove the main theorem.

Proof of Theorem 5.1.5. We start with the family of $\binom{r+1}{2} < \binom{m}{k_1}$ functions $\mathcal{H}$ with the properties specified in Lemma 5.1.6. We abuse notation slightly and just continue to write $\mathcal{H}$ for the family of functions whose domain is $V(\mathcal{P})$ instead of $\mathcal{S}$. There is no real difference since they are in bijection. For $h \in \mathcal{H}$, we have $|\mathrm{Orb}(h)| \leq |\mathcal{H}| < \binom{m}{k_1}$. By the orbit-stabilizer theorem, $|S_m : \mathrm{Stab}(h)| = |\mathrm{Orb}(h)| < \binom{m}{k_1}$, so by Lemma 5.1.7, there is a $I \subseteq [m]$ of size at most k_1 such that $A([m] \setminus I) \leq \mathrm{Stab}(h)$. Applying Lemma 5.1.4, we obtain polynomials $h'(x)$ of degree at most k_2 which agree with h on $V(\mathcal{P})$. Then for each f satisfying $\max_{\alpha\in\mathcal{S}} f(\alpha) \leq s(f)$,

$$c(f) - o^f(\phi(\alpha)) = \sum_i \left(\sum_{h\in\mathcal{H}} \alpha_h \cdot h'(\phi(\alpha))\right)^2 + \mu_f$$

for every $\alpha \in \mathcal{S}$. This is an equality on every point of $V(\mathcal{P})$ and each h' is degree at most k_2, so $C(f) - o^f(x)$ is $2k_2$-SOS modulo $\langle \mathcal{P} \rangle$. Thus the k_2th Theta Body relaxation is a (c, s)-approximate SDP relaxation of $\mathcal{M}$. □

Theorem 5.1.5 and Proposition 2.6.10 immediately imply the following corollary:

Corollary 5.1.8. *Let $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ have a S_m-symmetric (k_1, k_2)-block transitive Boolean polynomial formulation $(\mathcal{P}, \mathcal{O}, \phi)$ on n variables. If $\mathcal{P}$ is ℓ-effective, then if $\mathcal{M}$ has any (c, s)-approximate, S_m-symmetric SDP relaxation of size $r < \sqrt{\binom{m}{k_1}}$ the ℓk_2th Lasserre relaxation is a (c, s)-approximate relaxation as well.*

We also have collected several examples of combinatorial problems that we can apply Corollary 5.1.8 to:

Corollary 5.1.9.

- *If the* MATCHING *problem has a S_m-symmetric SDP relaxation of size $r < \sqrt{\binom{m}{k}}$ achieving (c, s)-approximation, the $2k^2$ Lasserre relaxation is a (c, s)-approximate relaxation as well.*
- *If* BALANCED CSP *has a S_m-symmetric SDP relaxation of size $r < \sqrt{\binom{m}{k}}$ achieving (c, s)-approximation, the kth Lasserre relaxation is a (c, s)-approximate relaxation as well.*

Proof. Follows from Example 5.1.2, Example 5.1.3, Theorem 3.3.8, Lemma 3.5.8, Lemma 3.5.7 and Corollary 5.1.8. □

5.2 Optimality for TSP

While block-transitivity is a useful categorization for capturing the symmetries of many problems, sometimes it is not sufficient. Unfortunately, TSP is not block-transitive, so we are unable to exactly apply the framework of the previous section. However, we will find out that is very nearly block-transitive, and a few modifications are enough to prove that the SOS relaxations are optimal for TSP. Recall the polynomial formulation of TSP on m^2 variables from (3.4). The map ϕ is defined so that for a tour τ, $\phi(\tau) = \chi_\tau$, where $(\chi_\tau)_{ij} = 1$ if $\tau(i) = j$ and 0 otherwise.

This polynomial formulation is S_m-symmetric under the action $\sigma(x_{ij}) = x_{\sigma(i)j}$, which represents simply composing a tour τ with σ on the left. Under this action, the above formulation for TSP is almost block-transitive:

Lemma 5.2.1. *If $I \subseteq [m]$, then let $J = I \times [m]$. Then $A([m] \setminus I)$ acts transitively on the elements of $V(\mathcal{P})$ that agree on J* and *have the same parity (as tours).*

Proof. If τ, τ' are tours with $\phi(\tau)|_J = \phi(\tau')|_J$ and $\text{sign}(\tau) = \text{sign}(\tau')$, then let $\sigma = \tau'\tau^{-1}$. Clearly $\sigma\tau = \tau'$. For $i \in I$, $\sigma(i) = \tau'(\tau^{-1}(i)) = i$ since τ' and τ agree on I, thus $\sigma \in S([m]\setminus I)$. Because both τ' and τ have the same parity, σ must be even, so $\sigma \in A([m] \setminus I)$. □

If we naively attempt the same strategy as in the previous section, we will show that the functions in $\mathcal{H}$ depend only on the placement of a small number of vertices in the tour, *and* the parity of the tour. Unfortunately, the parity of the tour is a high degree function in this polynomial formulation. To handle this dependence, we embed any tour as an even tour on a larger set of vertices, then find a good approximation for TSP on the set of larger vertices. To this end, define the function $T : S_m \to A_{2m}$ by $T(\tau) = \tau\tau'$, where τ' fixes $[m]$, and $\tau'(i) = \tau(i) + m$ for $i \in \{m + 1, \ldots, 2m\}$. Since $\text{sign}(\tau) = \text{sign}(\tau')$, $T(\tau)$ is indeed an even permutation. Also note that $T(\tau)|_{[m]}$ is a permutation of $[m]$, and indeed equal to τ. Now we are ready to prove our main theorem:

Theorem 5.2.2. *If* TSP *on $2m$ vertices has an A_{2m}-coordinate symmetric SDP relaxation of size $r < \sqrt{\binom{2m}{k}}$ with approximation guarantees $s(f) = \min_{\alpha \in \mathcal{S}} f(\alpha)$ and $c(f) = \rho s(f)$, then the $2k$th Lasserre relaxation is a (c, s)-approximate relaxation for* TSP *on m vertices.*

Proof. Let f be an objective function for TSP on m vertices, and let F be the objective function for TSP on $2m$ vertices which has

$$d_F(i, j+m) = d_F(i+m, j) = d_F(i+m, j+m) = d_F(i, j) = d_f(i, j).$$

Then $F(T(\tau)) = 2f(\tau)$. Furthermore, if $\Pi \in S_{2m}$, then there exist tours τ_1 of $[m]$ and τ_2 of $\{m+1, \ldots, 2m\}$ such that $F(\Pi) = F(\tau_1\tau_2) = f(\tau_1) + f(\tau_2)$. This can be seen just by setting $\tau_1(i) = \Pi(i)$ or $\Pi(i) - m$, whichever is in $[m]$, and $\tau_2(i+m) = \Pi(i)$ or $\Pi(i) + m$, whichever is in $\{m+1, \ldots, 2m\}$. Clearly from the definition of F this does not change the value. This implies that $\min_\Pi F(\Pi) = 2\min_\tau f(\tau)$.

Now if TSP has a symmetric SDP relaxation as in the theorem statement, by starting identically to Theorem 5.1.5, we obtain a family of $\binom{r+1}{2}$ functions $\mathcal{H}$ which are A_{2m}-invariant and

$$F(\Pi) - \rho \min_\Pi F(\Pi) = \sum_i g_i^2(\Pi)$$

where each $g_i \in \langle \mathcal{H} \rangle$. Furthermore, each $h \in \mathcal{H}$ has a subset I_h such that h is stabilized by $A([2m] \setminus I_h)$ and $|I_h| \leq k$. Then by Lemma 5.2.1, the function h depends only on the variables in $I_h \times [2m]$ and the sign of the permutation. The restriction of h to the image of T must then depend only on the variables in $I_h \times [2m]$, since every image of T is an even permutation. Thus there exists a polynomial $h'(x)$ which depends only on the variables in $I_h \times [2m]$ which agrees with h on the image of T. Because the polynomial formulation for TSP is Boolean and we can eliminate monomials of the form $x_{ij}x_{i\ell}$ for $j \neq \ell$, the polynomial $h'(x)$ can be taken to have degree at most $|I_h| \leq k$. Finally, we note that $x_{ij} = x_{i+m,j+m}$ and $x_{i,j+m} = x_{i+m,j} = 0$ for every $i, j \in [m]$ on the image of T. Thus we can replace each instance of $x_{i+m,j+m}$ in $h'(x)$ with x_{ij}, and each instance of $x_{i,j+m}$ or $x_{i+m,j}$ with 0 and not change the value of $h'(x)$ on the image of T. Now h' depends only on variables with indices in $[m] \times [m]$, and since $T(\tau)$ restricted to these variables is τ, we have the following polynomial identity:

$$F(T(\tau)) - \rho \min_\Pi F(\Pi) = \sum_i \left(\sum_{h \in \mathcal{H}} \alpha_{ih} h'(\phi(\tau)) \right)^2.$$

Now the LHS is equal to $2f(\tau) - 2\rho \min_\tau f(\tau)$, and thus $o^f(x) - \rho \min_\tau f(\tau)$ is $2k$-SOS modulo $\langle \mathcal{P} \rangle$. Since this is true for every objective f, this implies that the kth Theta Body on m vertices is a ρ-approximate SDP relaxation. Finally, by Theorem 3.4.8, we know that $\mathcal{P}$ is 2-effective, so the $2k$th Lasserre relaxation is also a ρ-approximate SDP relaxation. □

5.3 Lower Bounds for the Matching Problem

In Section 5.1 we proved that the SOS relaxation provides the best approximation for the MATCHING problem among small symmetric SDPs. However, it is also known that the SOS relaxations, which certify non-negativity via $\mathrm{PC}_>$ proofs, do not perform well on the MATCHING problem. In particular, they are incapable of certifying that the number of edges in the matching of an m-clique with m odd is at most $(m-1)/2$ until $\Omega(m)$ rounds:

Theorem 5.3.1 (Due to [32]). *If m is odd, $V(\mathcal{P}_{\mathrm{M}}(m)) = \emptyset$, but every $\mathrm{PC}_>$ refutation of $\mathcal{P}_{\mathrm{M}}(m)$ has degree $\Omega(m)$.*

Since the SOS relaxations do poorly on matchings, we can prove that every small symmetric SDP formulation must do poorly.

Theorem 5.3.2. *Assume the* MATCHING *problem has an S_m-coordinate-symmetric SDP relaxation of size d that achieves a (c,s)-approximation with $c(f) = \max f + \epsilon/2$ and $s(f) = \max f$ for some $0 \leq \epsilon < 1$. Then $d \geq 2^{\Omega(m)}$.*

Proof. Let k be the smallest integer such that $d < \sqrt{\binom{m}{k}}$. Taking Example 5.1.2, Theorem 3.3.8, and Corollary 5.1.8 together, the $2\binom{k}{2}$th Lasserre relaxation is a (C,S)-approximate SDP formulation for the MATCHING problem. Actually if we are slightly more careful in our application of Lemma 5.1.4, we can show that the kth Lasserre relaxation suffices. For a set $I \subseteq [m]$, the associated subset of $\left[\binom{m}{2}\right]$ that satisfies the block-transitivity is $E(I,I)$, the set of edges lying entirely in I. This has size $\binom{k}{2}$, and so we can conclude that the polynomials h' have degree at most $\binom{k}{2}$. However, by eliminating monomials containing $x_{ij}x_{i\ell}$ for $\ell \neq j$ (which are zero on $V(\mathcal{P}_{\mathrm{M}})$), we can actually take the polynomials h' to have degree at most $k/2$.

Now let $n = m/2$ or $m/2-1$, whichever is odd. Let $A = [n]$, $B = \{n,\dots,2n\}$, and if $n = m/2-1$, let $C = \{2n+1, 2n+2\}$, otherwise $C = \emptyset$. Note that $A \cup B \cup C = [m]$ and they are all disjoint. Consider the objective function $f = f_{E(A,A)}$ and its associated polynomial $o^f(x) = \sum_{ij \in E(A,A)} x_{ij}$. Because the kth Lasserre relaxation achieves a $(\max f + \epsilon/2, \max f)$-approximation, and by choice of $s(f)$, every f satisfies the soundness condition, we know

$$\begin{aligned} c(f) - o^f(x) &= \frac{n-1}{2} + \frac{\epsilon}{2} - \sum_{ij \in E(A,A)} x_{ij} \\ &\cong_1 \frac{1}{2} \sum_{\substack{i \in A \\ j \in B,C}} x_{ij} - \frac{1-\epsilon}{2} \end{aligned}$$

has a $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}_{\mathrm{M}}(m)$ of degree at most $2k$. Now we make a substitution in the polynomial identity: replace each instance of x_{ij} with either $x_{i-n,j-n}$ if $i,j \in B$, or 0 if $(i,j) \in (A,B), (A,C), (B,C)$. If $(i,j) \in (C,C)$, replace x_{ij} with 1. Note that under this substitution, every polynomial in $\mathcal{P}_{\mathrm{M}}(m)$ is mapped to either 0 or a polynomial in

$\mathcal{P}_{\mathrm{M}}(n)$. So if this substitution is made on a $\mathrm{PC}_>$ proof from $\mathcal{P}_{\mathrm{M}}(m)$, the result is a $\mathrm{PC}_>$ proof from $\mathcal{P}_{\mathrm{M}}(n)$, clearly of no higher degree. Since this substitution maps $\sum_{ij:(i,j)\in(A,B) \text{ or } (A,C)} x_{ij}$ to the zero polynomial, this implies that $-\frac{1-\epsilon}{2}$ has a degree-$2k$ $\mathrm{PC}_>$ proof of non-negativity from $\mathcal{P}_{\mathrm{M}}(n)$. By Theorem 5.3.1, this means that $k = \Omega(n)$, and since $n = \Theta(m)$, clearly $d \geq 2^{\Omega(m)}$. $\square$

Chapter 6

Future Work

Despite the progress made in this thesis, there are still plenty of directions for future efforts.

Effective Derivations

In Chapter 3 we developed a proof strategy for proving that sets of polynomials admit effective derivations. However this strategy is by no means universally applicable, and it had to be applied on a case-by-case basis. Is there a criterion for combinatorial ideals that suffices to show that a set of polynomials admits k-effective derivations for constant k? Failing that, knowing more combinatorial optimization problems that have polynomial formulations which admit effective derivations would be especially useful, for example for applying Theorem 4.1.9. What about combinatorial optimization problems without the strong symmetries discussed in this thesis? As an example, does the Vertex Cover formulation $\mathcal{P}_{\mathrm{VC}}(V, E) = \{x_i^2 - x_i \mid i \in V\} \cup \{(1 - x_i)(1 - x_j) \mid (i, j) \in E\}$ admit effective derivations?

Bit Complexity of SOS proofs

In Chapter 4 we provided a sufficient criteria to check if $\mathrm{PC}_{>}$ proofs of non-negativity can be taken to have only polynomial bit complexity. However, it has some significant shortcomings. The most glaring example is its inapplicability to $\mathrm{PC}_{>}$ refutations, i.e. proofs that $-1 \geq 0$ from an infeasible system of polynomial equations. Because an infeasible set of polynomials has *no* solutions, it certainly cannot have a rich solution space, and our criteria does not apply. For example, we are unable to prove that the SOS refutations of Knapsack use only small coefficients, even though it is clear from simply examining these known refutations that they do not have enormous coefficients. It is also known that adding the objective function as a constraint to the SDP, i.e. adding $c - o(x) = 0$ and checking feasibility is a tighter relaxation. Our results do not extend to this case as it is requires finding refutations when the constraints are infeasible. More generally, our criteria are sufficient but not necessary. Exactly categorizing the sets of polynomials that have $\mathrm{PC}_{>}$ proofs with small bit complexity is of great importance for applying the SOS relaxations as approximation algorithms, and it is a potential direction for future research.

Optimal SDPs

In Chapter 5 we gave some results on when the SOS relaxations provide optimal approximation among any symmetric SDP of comparable size. One obvious open problem is to drop the symmetry requirement. The SOS relaxations are known to be optimal for constraint satisfaction problems [46] even among asymmetric SDPs, and our results give some evidence that the same might be true for the Matching problem. This would be an important result, as it would mean that the Matching problem cannot be solved using SDPs, even though the problem lies in **P**! This would show that SDPs do not provide optimal approximations for every combinatorial optimization problem.

Bibliography

[1] M. Akgül. *Topics in relaxation and ellipsoidal methods*. Research notes in mathematics. Pitman Advanced Pub. Program, 1984. ISBN: 9780273086345. URL: https://books.google.com/books?id=VCqCAAAAIAAJ.

[2] Sanjeev Arora, Satish Rao, and Umesh Vazirani. "Expander Flows, Geometric Embeddings and Graph Partitioning". In: *J. ACM* 56.2 (Apr. 2009), 5:1–5:37. ISSN: 0004-5411. DOI: 10.1145/1502793.1502794. URL: http://doi.acm.org/10.1145/1502793.1502794.

[3] V. Arvind and Partha Mukhopadhyay. "The Ideal Membership Problem and Polynomial Identity Testing". In: *Inf. Comput.* 208.4 (Apr. 2010), pp. 351–363. ISSN: 0890-5401. DOI: 10.1016/j.ic.2009.06.003. URL: http://dx.doi.org/10.1016/j.ic.2009.06.003.

[4] Boaz Barak, Jonathan A. Kelner, and David Steurer. "Dictionary Learning and Tensor Decomposition via the Sum-of-Squares Method". In: *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*. STOC '15. New York, NY, USA: ACM, 2015, pp. 143–151. ISBN: 978-1-4503-3536-2. DOI: 10.1145/2746539.2746605. URL: http://doi.acm.org/10.1145/2746539.2746605.

[5] Boaz Barak and David Steurer. "Sum-of-squares proofs and the quest toward optimal algorithms". In: *arXiv preprint arXiv:1404.5236* (2014).

[6] Boaz Barak et al. "A Nearly Tight Sum-of-Squares Lower Bound for the Planted Clique Problem". In: *CoRR* abs/1604.03084 (2016). URL: http://arxiv.org/abs/1604.03084.

[7] P. Beame et al. "Lower bounds on Hilbert's Nullstellensatz and propositional proofs". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 794–806. DOI: 10.1109/SFCS.1994.365714.

[8] Aditya Bhaskara et al. "Polynomial Integrality Gaps for Strong SDP Relaxations of Densest K-subgraph". In: *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '12. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2012, pp. 388–405. URL: http://dl.acm.org/citation.cfm?id=2095116.2095150.

[9] Gábor Braun, Sebastian Pokutta, and Daniel Zink. "Inapproximability of Combinatorial Problems via Small LPs and SDPs". In: *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*. STOC '15. New York, NY, USA: ACM, 2015, pp. 107–116. ISBN: 978-1-4503-3536-2. DOI: 10.1145/2746539.2746550. URL: http://doi.acm.org/10.1145/2746539.2746550.

[10] Gabor Braun et al. "Approximation Limits of Linear Programs (Beyond Hierarchies)". In: *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. FOCS '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 480–489. ISBN: 978-0-7695-4874-6. DOI: 10.1109/FOCS.2012.10. URL: http://dx.doi.org/10.1109/FOCS.2012.10.

[11] Jop Briët, Daniel Dadush, and Sebastian Pokutta. "On the existence of 0/1 polytopes with high semidefinite extension complexity". In: *Mathematical Programming* 153.1 (2015), pp. 179–199. ISSN: 1436-4646. DOI: 10.1007/s10107-014-0785-x. URL: http://dx.doi.org/10.1007/s10107-014-0785-x.

[12] Bruno Buchberger. "Bruno Buchberger's PhD Thesis 1965: An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal". In: *J. Symb. Comput.* 41.3-4 (Mar. 2006), pp. 475–511. ISSN: 0747-7171. DOI: 10.1016/j.jsc.2005.09.007. URL: http://dx.doi.org/10.1016/j.jsc.2005.09.007.

[13] Sam Buss et al. "Linear Gaps Between Degrees for the Polynomial Calculus Modulo Distinct Primes". In: *J. Comput. Syst. Sci.* 62.2 (Mar. 2001), pp. 267–289. ISSN: 0022-0000. DOI: 10.1006/jcss.2000.1726. URL: http://dx.doi.org/10.1006/jcss.2000.1726.

[14] Samuel R Buss and Toniann Pitassi. "Good Degree Bounds on Nullstellensatz Refutations of the Induction Principle". In: *J. Comput. Syst. Sci.* 57.2 (Oct. 1998), pp. 162–171. ISSN: 0022-0000. DOI: 10.1006/jcss.1998.1585. URL: http://dx.doi.org/10.1006/jcss.1998.1585.

[15] Siu On Chan et al. "Approximate Constraint Satisfaction Requires Large LP Relaxations". In: *J. ACM* 63.4 (Oct. 2016), 34:1–34:22. ISSN: 0004-5411. DOI: 10.1145/2811255. URL: http://doi.acm.org/10.1145/2811255.

[16] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. "Near-optimal Algorithms for Maximum Constraint Satisfaction Problems". In: *ACM Trans. Algorithms* 5.3 (July 2009), 32:1–32:14. ISSN: 1549-6325. DOI: 10.1145/1541885.1541893. URL: http://doi.acm.org/10.1145/1541885.1541893.

[17] Eden Chlamtac. "Approximation Algorithms Using Hierarchies of Semidefinite Programming Relaxations". In: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 691–701. ISBN: 0-7695-3010-9. DOI: 10.1109/FOCS.2007.13. URL: http://dx.doi.org/10.1109/FOCS.2007.13.

[18] Eden Chlamtac and Gyanit Singh. "Improved Approximation Guarantees Through Higher Levels of SDP Hierarchies". In: *Proceedings of the 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008 on Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques*. APPROX '08 / RANDOM '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 49–62. ISBN: 978-3-540-85362-6. DOI: 10.1007/978-3-540-85363-3_5. URL: http://dx.doi.org/10.1007/978-3-540-85363-3_5.

[19] N. Christofides and CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENCES RESEARCH GROUP. *Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*. Management sciences research report. Defense Technical Information Center, 1976. URL: https://books.google.com/books?id=2A7eygAACAAJ.

[20] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007. ISBN: 0387356509.

[21] Yash Deshpande and Andrea Montanari. "Improved Sum-of-Squares Lower Bounds for Hidden Clique and Hidden Submatrix Problems". In: *Proceedings of The 28th Conference on Learning Theory*. Ed. by Peter Grnwald, Elad Hazan, and Satyen Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, 2015, pp. 523–562. URL: http://proceedings.mlr.press/v40/Deshpande15.html.

[22] J.D. Dixon and B. Mortimer. *Permutation Groups*. Graduate Texts in Mathematics. Springer New York, 1996. ISBN: 9780387945996. URL: https://books.google.com/books?id=4QDpFN6k61EC.

[23] H. Fawzi, J. Saunderson, and P. A. Parrilo. "Equivariant semidefinite lifts and sum-of-squares hierarchies". In: *ArXiv e-prints* (Dec. 2013). arXiv: 1312.6662 [math.OC].

[24] Samuel Fiorini et al. "Exponential Lower Bounds for Polytopes in Combinatorial Optimization". In: *J. ACM* 62.2 (May 2015), 17:1–17:23. ISSN: 0004-5411. DOI: 10.1145/2716307. URL: http://doi.acm.org/10.1145/2716307.

[25] Gerald B. Folland. "How to Integrate a Polynomial over a Sphere". In: *The American Mathematical Monthly* 108.5 (2001), pp. 446–448. ISSN: 00029890, 19300972. URL: http://www.jstor.org/stable/2695802.

[26] A. Frieze and M. Jerrum. "Improved approximation algorithms for MAXk-CUT and MAX BISECTION". In: *Algorithmica* 18.1 (1997), pp. 67–81. ISSN: 1432-0541. DOI: 10.1007/BF02523688. URL: http://dx.doi.org/10.1007/BF02523688.

[27] Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. "Optimal Sherali-Adams Gaps from Pairwise Independence". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 12th International Workshop, APPROX 2009, and 13th International Workshop, RANDOM 2009, Berkeley, CA, USA, August 21-23, 2009. Proceedings*. Ed. by Irit Dinur et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 125–139. ISBN: 978-3-642-03685-9. DOI: 10.1007/978-3-642-03685-9_10. URL: http://dx.doi.org/10.1007/978-3-642-03685-9_10.

[28] Michel X. Goemans and David P. Williamson. "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming". In: *J. ACM* 42.6 (Nov. 1995), pp. 1115–1145. ISSN: 0004-5411. DOI: 10.1145/227683.227684. URL: http://doi.acm.org/10.1145/227683.227684.

[29] João Gouveia, Pablo A. Parrilo, and Rekha R. Thomas. "Theta Bodies for Polynomial Ideals". In: *SIAM J. on Optimization* 20.4 (Mar. 2010), pp. 2097–2118. ISSN: 1052-6234. DOI: 10.1137/090746525. URL: http://dx.doi.org/10.1137/090746525.

[30] D. Grigoriev. "Complexity of Positivstellensatz proofs for the knapsack". In: *computational complexity* 10.2 (2001), pp. 139–154. ISSN: 1420-8954. DOI: 10.1007/s00037-001-8192-0. URL: http://dx.doi.org/10.1007/s00037-001-8192-0.

[31] D. Grigoriev. "Tseitin's tautologies and lower bounds for Nullstellensatz proofs". In: *Proceedings 39th Annual Symposium on Foundations of Computer Science*. 1998, pp. 648–652. DOI: 10.1109/SFCS.1998.743515.

[32] Dima Grigoriev. "Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity". In: *Theoretical Computer Science* 259.1 (2001), pp. 613–622.

[33] Eran Halperin and Uri Zwick. "Approximation Algorithms for MAX 4-SAT and Rounding Procedures for Semidefinite Programs". In: *Proceedings of the 7th International IPCO Conference on Integer Programming and Combinatorial Optimization*. London, UK, UK: Springer-Verlag, 1999, pp. 202–217. ISBN: 3-540-66019-4. URL: http://dl.acm.org/citation.cfm?id=645589.757889.

[34] G. Hermann. "Die Frage der endlich vielen Schritte in der Theorie der Polynomideale. (Unter Benutzung nachgelassener Stze von K. Hentzelt)". In: *Mathematische Annalen* 95 (1926), pp. 736–788. URL: http://eudml.org/doc/159153.

[35] D. Hilbert. "Ueber die vollen Invariantensysteme". In: *Mathematische Annalen* 42 (1893), pp. 313–373. URL: http://eudml.org/doc/157652.

[36] Samuel B. Hopkins, Pravesh K. Kothari, and Aaron Potechin. "SoS and Planted Clique: Tight Analysis of MPW Moments at all Degrees and an Optimal Lower Bound at Degree Four". In: *CoRR* abs/1507.05230 (2015). URL: http://arxiv.org/abs/1507.05230.

[37] Samuel B. Hopkins, Jonathan Shi, and David Steurer. "Tensor principal component analysis via sum-of-squares proofs". In: *CoRR* abs/1507.03269 (2015). URL: http://arxiv.org/abs/1507.03269.

[38] Dung T. Huynh. "Complexity of the word problem for commutative semigroups of fixed dimension". In: *Acta Informatica* 22.4 (1985), pp. 421–432. ISSN: 1432-0525. DOI: 10.1007/BF00288776. URL: http://dx.doi.org/10.1007/BF00288776.

[39] Cédric Josz and Didier Henrion. "Strong duality in Lasserre's hierarchy for polynomial optimization". In: *Optimization Letters* 10.1 (2016), pp. 3–10. ISSN: 1862-4480. DOI: 10.1007/s11590-015-0868-5. URL: http://dx.doi.org/10.1007/s11590-015-0868-5.

[40] Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. "Symmetry Matters for the Sizes of Extended Formulations". In: *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization*. IPCO'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 135–148. ISBN: 3-642-13035-6, 978-3-642-13035-9. DOI: 10.1007/978-3-642-13036-6_11. URL: http://dx.doi.org/10.1007/978-3-642-13036-6_11.

[41] George Karakostas. "A Better Approximation Ratio for the Vertex Cover Problem". In: *ACM Trans. Algorithms* 5.4 (Nov. 2009), 41:1–41:8. ISSN: 1549-6325. DOI: 10.1145/1597036.1597045. URL: http://doi.acm.org/10.1145/1597036.1597045.

[42] Anna R. Karlin, Claire Mathieu, and C. Thach Nguyen. "Integrality Gaps of Linear and Semi-definite Programming Relaxations for Knapsack". In: *CoRR* abs/1007.1283 (2010). URL: http://arxiv.org/abs/1007.1283.

[43] Richard M. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*. Ed. by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Boston, MA: Springer US, 1972, pp. 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9. URL: http://dx.doi.org/10.1007/978-1-4684-2001-2_9.

[44] Subhash Khot. "On the Power of Unique 2-prover 1-round Games". In: *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*. STOC '02. New York, NY, USA: ACM, 2002, pp. 767–775. ISBN: 1-58113-495-9. DOI: 10.1145/509907.510017. URL: http://doi.acm.org/10.1145/509907.510017.

[45] Jean B. Lasserre. "An Explicit Exact SDP Relaxation for Nonlinear 0-1 Programs". In: *Integer Programming and Combinatorial Optimization: 8th International IPCO Conference Utrecht, The Netherlands, June 13–15, 2001 Proceedings*. Ed. by Karen Aardal and Bert Gerards. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 293–303. ISBN: 978-3-540-45535-6. DOI: 10.1007/3-540-45535-3_23. URL: http://dx.doi.org/10.1007/3-540-45535-3_23.

[46] James R. Lee, Prasad Raghavendra, and David Steurer. "Lower Bounds on the Size of Semidefinite Programming Relaxations". In: *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*. STOC '15. New York, NY, USA: ACM, 2015, pp. 567–576. ISBN: 978-1-4503-3536-2. DOI: 10.1145/2746539.2746599. URL: http://doi.acm.org/10.1145/2746539.2746599.

[47] James R. Lee et al. "On the Power of Symmetric LP and SDP Relaxations". In: *Proceedings of the 2014 IEEE 29th Conference on Computational Complexity*. CCC '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 13–21. ISBN: 978-1-4799-3626-7. DOI: 10.1109/CCC.2014.10. URL: http://dx.doi.org/10.1109/CCC.2014.10.

[48] Shi Li. "A 1.488 approximation algorithm for the uncapacitated facility location problem". In: *Information and Computation* 222 (2013). 38th International Colloquium on Automata, Languages and Programming (ICALP 2011), pp. 45 –58. ISSN: 0890-5401. DOI: http://doi.org/10.1016/j.ic.2012.01.007. URL: http://www.sciencedirect.com/science/article/pii/S0890540112001459.

[49] L. Lovasz. "On the Shannon Capacity of a Graph". In: *IEEE Trans. Inf. Theor.* 25.1 (Sept. 2006), pp. 1–7. ISSN: 0018-9448. DOI: 10.1109/TIT.1979.1055985. URL: http://dx.doi.org/10.1109/TIT.1979.1055985.

[50] Tengyu Ma and Avi Wigderson. "Sum-of-squares Lower Bounds for Sparse PCA". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 1612–1620. URL: http://dl.acm.org/citation.cfm?id=2969239.2969419.

[51] Ernst W Mayr and Albert R Meyer. "The complexity of the word problems for commutative semigroups and polynomial ideals". In: *Advances in Mathematics* 46.3 (1982), pp. 305 –329. ISSN: 0001-8708. DOI: http://dx.doi.org/10.1016/0001-8708(82)90048-2. URL: http://www.sciencedirect.com/science/article/pii/0001870882900482.

[52] Raghu Meka, Aaron Potechin, and Avi Wigderson. "Sum-of-squares Lower Bounds for Planted Clique". In: *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*. STOC '15. New York, NY, USA: ACM, 2015, pp. 87–96. ISBN: 978-1-4503-3536-2. DOI: 10.1145/2746539.2746600. URL: http://doi.acm.org/10.1145/2746539.2746600.

[53] T.S. Motzkin. *The arithmetic-geometric inequality*. Wright-Patterson, Air Force Base, Ohio, 1967.

[54] Ryan O'Donnell. "SOS is not obviously automatizable, even approximately". In: *Electronic Colloquium on Computational Complexity (ECCC)* 23 (2016), p. 141. URL: http://eccc.hpi-web.de/report/2016/141.

[55] Pablo A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. Tech. rep. 2000.

[56] Jiming Peng and Yu Wei. "Approximating K-means-type Clustering via Semidefinite Programming". In: *SIAM J. on Optimization* 18.1 (Feb. 2007), pp. 186–205. ISSN: 1052-6234. DOI: 10.1137/050641983. URL: http://dx.doi.org/10.1137/050641983.

[57] Aaron Potechin and David Steurer. "Exact tensor completion with sum-of-squares". In: *CoRR* abs/1702.06237 (2017). URL: http://arxiv.org/abs/1702.06237.

[58] P. Raghavendra and D. Steurer. "Integrality Gaps for Strong SDP Relaxations of UNIQUE GAMES". In: *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. 2009, pp. 575–585. DOI: 10.1109/FOCS.2009.73.

[59] Prasad Raghavendra. "Optimal Algorithms and Inapproximability Results for Every CSP?" In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. New York, NY, USA: ACM, 2008, pp. 245–254. ISBN: 978-1-60558-047-0. DOI: 10.1145/1374376.1374414. URL: http://doi.acm.org/10.1145/1374376.1374414.

[60] Prasad Raghavendra and Tselil Schramm. "Tight Lower Bounds for Planted Clique in the Degree-4 SOS Program". In: *CoRR* abs/1507.05136 (2015). URL: http://arxiv.org/abs/1507.05136.

[61] Prasad Raghavendra and Ning Tan. "Approximating CSPs with Global Cardinality Constraints Using SDP Hierarchies". In: *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '12. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2012, pp. 373–387. URL: http://dl.acm.org/citation.cfm?id=2095116.2095149.

[62] A.A. Razborov. "Lower bounds for the polynomial calculus". In: *computational complexity* 7.4 (1998), pp. 291–324. ISSN: 1420-8954. DOI: 10.1007/s000370050013. URL: http://dx.doi.org/10.1007/s000370050013.

[63] Thomas Rothvoss. "The Matching Polytope Has Exponential Extension Complexity". In: *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*. STOC '14. New York, NY, USA: ACM, 2014, pp. 263–272. ISBN: 978-1-4503-2710-7. DOI: 10.1145/2591796.2591834. URL: http://doi.acm.org/10.1145/2591796.2591834.

[64] Sartaj Sahni and Teofilo Gonzalez. "P-Complete Approximation Problems". In: *J. ACM* 23.3 (July 1976), pp. 555–565. ISSN: 0004-5411. DOI: 10.1145/321958.321975. URL: http://doi.acm.org/10.1145/321958.321975.

[65] Grant Schoenebeck. "Linear Level Lasserre Lower Bounds for Certain k-CSPs". In: *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*. FOCS '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 593–602. ISBN: 978-0-7695-3436-7. DOI: 10.1109/FOCS.2008.74. URL: http://dx.doi.org/10.1109/FOCS.2008.74.

[66] N. Z. Shor. "An approach to obtaining global extremums in polynomial mathematical programming problems". In: *Cybernetics* 23.5 (1987), pp. 695–700. ISSN: 1573-8337. DOI: 10.1007/BF01074929. URL: http://dx.doi.org/10.1007/BF01074929.

[67] Morton Slater. "Lagrange Multipliers Revisited". In: *Traces and Emergence of Nonlinear Programming*. Ed. by Giorgio Giorgi and Tinne Hoff Kjeldsen. Basel: Springer Basel, 2014, pp. 293–306. ISBN: 978-3-0348-0439-4. DOI: 10.1007/978-3-0348-0439-4_14. URL: http://dx.doi.org/10.1007/978-3-0348-0439-4_14.

[68] Gongguo Tang and Parikshit Shah. "Guaranteed Tensor Decomposition: A Moment Approach". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 1491–1500. URL: http://dl.acm.org/citation.cfm?id=3045118.3045277.

[69] Madhur Tulsiani. "CSP Gaps and Reductions in the Lasserre Hierarchy". In: *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*. STOC '09. New York, NY, USA: ACM, 2009, pp. 303–312. ISBN: 978-1-60558-506-2. DOI: 10.1145/1536414.1536457. URL: http://doi.acm.org/10.1145/1536414.1536457.

[70] Santosh Vempala and Mihalis Yannakakis. "A Convex Relaxation for the Asymmetric TSP". In: *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '99. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999, pp. 975–976. ISBN: 0-89871-434-6. URL: http://dl.acm.org/citation.cfm?id=314500.314955.

[71] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. 1st. New York, NY, USA: Cambridge University Press, 2011. ISBN: 0521195276, 9780521195270.

[72] Mihalis Yannakakis. "Expressing Combinatorial Optimization Problems by Linear Programs". In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. STOC '88. New York, NY, USA: ACM, 1988, pp. 223–228. ISBN: 0-89791-264-0. DOI: 10.1145/62212.62232. URL: http://doi.acm.org/10.1145/62212.62232.