New lower bounds for Approximation Algorithms in the Lovász-Schrijver Hierarchy

Iannis Tourlakis

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
By the Department of
Computer Science

June 2006

Abstract

Determining how well we can efficiently compute approximate solutions to NP-hard problems is of great theoretical and practical interest. Typically the famous PCP theorem is used for showing that a problem has no algorithms computing good approximations. Unfortunately, for many problem this approach has failed. Nevertheless, for such problems, we may instead be able to show that a large subclass of algorithms cannot compute good approximations.

This thesis takes this approach, concentrating on subclasses of algorithms defined by the LS and LS_+ Lovász-Schrijver hierarchies. These subclasses define hierarchies of algorithms where algorithms in higher levels (also called "rounds") require more time, but may compute better approximations. Algorithms in the LS hierarchy are based on linear programming relaxations while those in the more powerful LS_+ hierarchy are based on semidefinite programming relaxations. Most known approximation algorithms lie within the first two–three levels of the LS_+ hierarchy, including the recent celebrated approximation algorithms of Goemans-Williamson [27] and Arora-Rao-Vazirani [7] for MAX-CUT and SPARSEST-CUT, respectively. So understanding the power of these algorithm families seems important.

We obtain new lower bounds for the LS and LS_+ hierarchies for several important problems. In all cases the approximations we rule out in these hierarchies are not ruled out by known PCP-based arguments. Moreover, unlike PCP-based inapproximability results, all our results are unconditional and do not rely on any computational complexity assumptions.

The lower bounds we prove are as follows:

1. For VERTEX COVER we show that $\Omega(\log n)$ rounds of LS are needed to obtain $2-\epsilon$ approximations and $\Omega(\log^2 n)$ rounds are needed for $1.5-\epsilon$ approximations.

2. For MAX-3SAT and SET COVER we show that $\Omega(n)$ rounds of LS_+ are needed for any non-trivial approximation.

3. For VERTEX COVER on rank-k hypergraphs we show that $\Omega(n)$ rounds of LS_+ are needed for $k-1-\epsilon$ approximations.

4. For VERTEX COVER on rank-k hypergraphs we show that $\Omega(\log\log n)$ rounds of LS are needed for $k-\epsilon$ approximations.

Acknowledgments

First, I would like to warmly thank my advisor Sanjeev Arora for his guidance over the past few years. I have learned so much from him not only about the Theory of Computing, but also on how to be a successful researcher and how to effectively present my work. This thesis would not have been possible without his help!

I thank my thesis readers Moses Charikar and Boaz Barak as well as my non-readers Alexander Razborov and Nicholas Pippenger for their helpful comments and suggestions.

Many thanks also to Mikhail Alekhnovich, Joshua Buresh-Oppenheim, Avner Magen and Toniann Pitassi for numerous discussions about lower bounds for lift-and-project methods. Their help (especially that of my co-author Misha) steered me towards many of the results in this thesis.

I would like to thank Satyen Kale for permission to include his proof of an integrality gap for VERTEX COVER after one round of LS_+.

I would also like to thank Melissa Lawson for all her help and advice while at Princeton.

A special thanks to my office-mates as well as other friends in and out of the department for making life at Princeton enjoyable: Amal Ahmed, Nir Ailon, Tony Capra, Dan "Danilo Tavares" Dantas, Milda Darguzaite, Loukas Georgiadis, Elad Hazan, Don Johnson, Satyen Kale, Carl Kingsford, Jay Ligatti, Patrick Min, Ananya Misra, Diego Nehab, Spyros Triantafyllis, Brent Waters, Kevin Wayne, and Tony Wirth.

I gratefully acknowledge funding through Sanjeev Arora's David and Lucile Packard Fellowship and NSF grants CCR 0205594, CCR 0098180, MSPA-MCS 0528414, CCF 0514993, ITR 0205594.

I thank Allison Doerr for her love and support which have made the past few years so memorable!

Finally, I would like to thank my family for all the encouragement and support they've given me over my graduate studies. This thesis is dedicated to them.

Contents

List of Figures

Chapter 1

Introduction

Determining how well we can efficiently compute approximate solutions to **NP**-hard optimization problems is of both theoretical and practical interest. The *approximation ratio* of an algorithm is a number α such that the algorithm is guaranteed to compute a solution whose cost is within a factor α of the optimal cost. The main goal of this thesis is to identify which α's cannot be achieved by algorithms based on linear and semidefinite programming.

In the last dozen or so years, using techniques based on the famous PCP theorem from complexity theory [8, 6], researchers have shown that for many problems there are strong lower bounds on the approximation ratios achievable by *any* polynomial time algorithm (i.e., not just those based on linear semidefinite programming). Let us briefly recall the notion of *Probabilistically Checkable Proofs* (PCPs) and the PCP theorem, and their use in proving inapproximability results. Recall that a language L is in the complexity class **NP** if there exists a polynomial time verifier V for L such that for all strings $x \in L$ there exists a succinct (i.e., polynomial size in $|x|$) proof convincing V that x is in L, whereas for all $x \notin L$ no succinct proof convinces V that x is in L. Surprisingly, the PCP theorem states that if verifiers are in addition allowed to use $O(\log n)$ bits of randomness while checking a membership proof, then for the **NP**-complete language 3SAT there exists such a verifier V that can decide the satisfiability of a formula by checking only a *constant* number of bits of a candidate membership proof. The catch is that with some small probability (over the $O(\log n)$ random bits the verifier uses) the verifier may accept an unsatisfiable formula; however, V will never reject a satisfiable formula.

The PCP theorem thus gives a new characterization for **NP** as the class of all languages where membership can be probabilistically checked using $O(\log n)$ bits of randomness and examining a constant number of bits of the proof. Feige et al. [23] had noted before the PCP theorem was discovered that such a characterization could be used to show that for many optimization problems, attaining a certain approximation ratio is just as hard as computing 3SAT exactly. In general, a PCP-based hardness of approximation result for a specific optimization problem L is proved by coming up with an appropriate reduction from 3SAT to L and then designing and optimizing a PCP verifier tailored for this reduction. Indeed, over the past dozen or so years, a rich tapestry of inapproximability results has been discovered using PCPs in this way. For compendia of such results, see Arora and Lund [5], Feige [21] and Trevisan [53].

However, even with recent advances in designing on the one hand stronger PCPs and discovering on the other hand better and better approximation algorithms, there remain two nagging gaps in our knowledge of which problems can and cannot be efficiently well-approximated. First, there are several optimization problems for which there remain gaps between the approximation ratios achieved by known algorithms and those ruled out by PCP-based techniques. For example, this is true for VERTEX COVER and metric TRAVELLING SALESMAN, two very basic optimization problems (indeed VERTEX COVER is in Karp's original list of **NP**-complete problems). In particular, the respective approximation ratios for metric TRAVELLING SALESMAN are 1.5 and 1.02, while for VERTEX COVER they are are 2 and 1.36 [16].

The second gap is that current PCP-based results do not rule out the existence of slightly subexponential-time approximation algorithms. Algorithms are called *slightly subexponential* if they run in time 2^{n^δ} for some fixed $0 < \delta < 1$. The gap in our knowledge for such algorithms results from the fact that known PCP-based inapproximability results use reductions that greatly increase instance size. For example, consider the reduction from 3SAT to VERTEX COVER used by Dinur and Safra [16] to show that 1.36-approximations of VERTEX COVER in polynomial time is impossible unless $\mathbf{P} = \mathbf{NP}$. Their proof reduces the polynomial-time computation of 3SAT on instances of size n to the polynomial-time computation of 1.36-approximations for VERTEX COVER on graphs of size n^C for some astronomically large constant C. Their reduction therefore implies that computing 1.36-approximations of VERTEX COVER in time $T(n)$ gives an algorithm for 3SAT running in time $O(T(n^C))$. Since we believe computing 3SAT requires exponential time, the Dinur-Safra reduction can be used to rule out 1.36-approximations of VERTEX COVER by any algorithm taking time asymptotically less than $2^{n^{1/C}}$; however, their reduction, cannot rule out, say, a 1.2-approximation to VERTEX COVER in $2^{n^{0.01}}$ time—an interesting possibility.

PCP-based results are used (in conjunction with some plausible computational complexity assumption such as $\mathbf{P} \neq \mathbf{NP}$) to rule out approximation ratios for *all* polynomial algorithms. Moreover, as alluded above they can be used to rule out even approximation algorithms running in super-polynomial time by appropriately strengthening the computational complexity assumption used. When PCP-based approaches stall as for the two gaps in our knowledge mentioned above, Arora, Bollobás and Lovász [3] suggested pursuing what might be an easier goal: instead of ruling out *all* polynomial (slightly subexponential) time algorithms, rule out large *subclasses* of polynomial (slightly subexponential, respectively) time algorithms.

This thesis takes this approach, concentrating on subclasses of algorithms defined by the LS and LS_+ Lovász-Schrijver procedures [43]. These procedures define what are known as the LS and LS_+ hierarchies of algorithms by systematically tightening linear and positive semidefinite relaxations over many rounds. Algorithms in the LS hierarchy are based on linear programming relaxations while those in the more powerful LS_+ hierarchy are based on semidefinite programming relaxations. A relaxation lies in the *rth level* of the LS (respectively, LS_+) hierarchy if it can be derived by tightening an initial relaxation r times using the LS (respectively, LS_+) procedure. Lovász and Schrijver observe that one can optimize a linear function over the relaxations produced after r rounds of either the LS or LS_+ procedures in $n^{O(r)}$ time.

The LS and LS_+ procedures are often described as *lift-and-project* or *matrix cut* operators. They are called cut operators since (over repeated application) they systematically cut away slices of an initial polytope to obtain the polytope's integral hull (see Section 1.1.3 for formal definitions). They are called lift-and-project (or matrix) operators since the cuts are obtained by "lifting" the initial polytope to a higher dimensional space, adding new constraints to the lifted polytope, and then projecting the resulting polytope back to the original space. We will describe this process as well as the intuition behind it in detail in Chapter 2.

It is instructive to view the LS and LS_+ lift-and-project procedures as defining restricted models of computation. For example, in the LS model, an optimization problem is in "time" n^r if it can be computed by a linear relaxation in the rth level of the LS hierarchy. In particular, relaxations in rounds $r = O(1)$ corresponds to "polynomial time", while relaxations in rounds $r = n^\delta$ where $0 < \delta < 1$ correspond to "slightly subexponential time".

These computation models seem quite powerful: It is known that both of these procedures yield the integral hull after n rounds. In particular, all of **NP** is computable in "exponential time" even in the weaker LS model. On the other hand, many recent celebrated approximation algorithms such as the Goemans-Williamson [27] algorithm for MAX-CUT and the Arora-Rao-Vazirani [7] algorithm for SPARSEST-CUT can be derived using a constant number of LS_+ rounds and hence, are computable in "polynomial time" in the LS_+ model. Thus, it seems important to study the power of the LS and LS_+ procedures. In particular, proving (unconditional) inapproximability results in these computation models may give evidence about a problem's true inapproximability. Indeed, such lower bounds may be viewed as analogous to lower bounds proved for other restricted computation models such as for monotone circuits [48] or specific proof systems [10].

Lift-and-project procedures also provide a tool for studying the approximation ratios achievable by slightly subexponential algorithms: Proving strong inapproximability results for slightly subexponential algorithms in the LS and LS_+ hierarchies may give evidence that a problem does not have even slightly subexponential non-trivial approximation algorithms.

Initially research on lift-and-project methods concentrated on studying how many rounds were required by such methods to derive specific inequalities given an initial linear relaxation [43, 50, 26]. How the integrality gap improved with each round of lift-and-project was not analyzed. Motivated by the weak PCP-based inapproximability results for VERTEX COVER mentioned above, Arora et al. [3] were the first to explicitly study integrality gaps for the LS hierarchy. They showed that the integrality gap remains $2 - o(1)$ even after tightening the standard linear relaxation for VERTEX COVER with $\Omega(\sqrt{\log n})$ rounds of LS lift-and-project. Contemporarily with [3], Feige and Krauthgamer [24] showed that a large integrality gap remains for relaxations of INDEPENDENT SET derived after even $\Omega(\log n)$ rounds of tightening with the stronger LS_+ method (Feige and Krauthgamer do not explicitly state their results as integrality gaps but such gaps are an immediate corollary). Buresh-Oppenheim et al. [11] showed that $\Omega(n)$ rounds of LS_+ lift-and-project are needed to achieve non-trivial approximations for MAX-kSAT for $k \geq 5$. While strong PCP-based inapproximability results are known for MAX-kSAT, the results in [11] can be interpreted as showing that in the LS_+ computation model there do not exist even slightly subexponential non-trivial approximation algorithms for these problems. We discuss work previous to this thesis more thoroughly in Section 2.4.

In this thesis we build on previous work as well as introduce new techniques to prove several new inapproximability results in both the LS and LS_+ hierarchies. We now give a high-level description of our results together with an outline of this thesis; more details on our results will be given in Section 2.5 after first giving a full technical description of lift-and-project systems in Chapter 2. In Chapter 2 we will also describe a related lift-and-project system due to Sherali and Adams [49] which was introduced contemporaneously to that of Lovász and Schrijver and for which there are still no known non-trivial inapproximability results.

In Chapter 3 we use the "expanding constraints" method to show that non-trivial approximations for MAX-3SAT, SET COVER and rank-k hypergraph VERTEX COVER require $\Omega(n)$ rounds of LS_+ lift-and-project. In particular, no nontrivial slightly subexponential approximation algorithms exist for these problems in the LS_+ "computation model".

In Chapter 4 we extend the results in [3] to show both that (1) the integrality gap for VERTEX COVER relaxations remains $2-o(1)$ even after tightening the initial relaxation with $\Omega(\log n)$ rounds of LS lift-and-project and that (2) the integrality gap for VERTEX COVER on rank-k hypergraphs remains $k-o(1)$ even after tightening the initial relaxation with $\Omega(\log \log n)$ rounds of LS lift-and-project. This may suggest that the true inapproximability factors for graph and rank-k hypergraph VERTEX COVER are $2 - o(1)$ and $k - o(1)$, respectively, even though the strongest PCP-based hardness results only rule out 1.36 [16] and $k-1-\epsilon$ [15] approximations, respectively, for these problems.

In Chapter 5 we introduce the "fence method" and use it to show that the integrality gap for (graph) VERTEX COVER remains at least $1.5 - o(1)$ even after tightening the initial relaxation with $\Omega(\log^2 n)$ rounds of LS lift-and-project.

In Chapter 6 we show that any linear relaxation for INDEPENDENT SET where each inequality uses at most $n^{\epsilon(1-\gamma)}$ of the input variables has an integrality gap of $n^{1-\epsilon}$. While this result does not obviously translate into an integrality gap for INDEPENDENT SET in the LS or LS_+ hierarchies, it is in the same "spirit" as it also proves an integrality gap for a large class of linear relaxations.

Finally, in Chapter 7 we discuss some limitations to our approaches as well as describing several directions for future work.

1.1 Preliminaries

The notation $[n]$ will denote the set $\{1, \ldots, n\}$. We will use e_i to denote the ith unit vector and define the vector f_i, $i \geq 1$, to be $e_0 - e_i$. The dimensions of e_i and f_i will always be clear from context.

1.1.1 Graphs: Theory and optimization problems

Given a graph $G = (V, E)$, we will let n denote the number of vertices in G and will assume $V = [n]$. The set of vertices adjacent to a vertex i in a graph will be denoted by $\Gamma(i)$. We extend this notation to sets $S \subseteq V$ by having $\Gamma(S) = \cup_{i \in S} \Gamma(i)$. A graph G is drawn from the random graph model $\mathcal{G}(n, p)$ by choosing each of the $\binom{n}{2}$ possible edges for G with probability p.

An *independent set* in G is a subset of vertices such that no two vertices in the subset are adjacent. A *vertex cover* in G is a subset $S \subseteq V$ of vertices such that every edge $e \in E$ contains an endpoint in S. Note that if S is an independent set, then $V \backslash S$ is a vertex cover and vice versa. The *maximum independent set* problem (INDEPENDENT SET) is to find the an independent set of maximum size in the input graph G. The *independence number* of G, denoted $\alpha(G)$, is the maximum size of an independent set in G. The *minimum vertex cover* problem (VERTEX COVER) is to find a vertex cover of minimum size in the input graph G. By the above observation, the size of a minimal vertex cover in a graph G is $n - \alpha(G)$.

A *hypergraph* $H = (V, E)$ consists of a set V of vertices (again assumed without loss of generality to be $[n]$) and a set E of subsets of V. That is, E is a subset of the power-set of V. If all sets in E have size k, then we say that H is a *k-uniform* hypergraph, or alternatively, a *rank-k hypergraph*. Note that a rank-2 hypergraph is of course a standard graph. The concepts of independent sets and vertex covers are naturally extended to hypergraphs as are the maximum independent set and minimum vertex cover problems. The size of a maximal independent set in a hypergraph H is denoted by $\alpha(H)$ and it is not hard to verify that the size of a minimum vertex cover in an n-node hypergraph H is $n - \alpha(H)$, just as in the graph case.

1.1.2 Computational complexity and approximability

The reader is assumed to be familiar with basic concepts from Computational Complexity theory. For completeness, we will review the complexity classes used in this thesis. More details on Computational Complexity can be found in standard texts on complexity theory such as Garey and Johnson [25] and Papadimitriou [45].

A *computational problem* L is a subset of $\{0,1\}^*$; the "problem" is to decide whether a given string belongs to L. We denote the length of a string $x \in \{0,1\}^*$ by $|x|$. A problem $L \subseteq \{0,1\}^*$ is in the complexity class $\mathbf{DTIME}(f(n))$ if there exists a deterministic Turing machine M that accepts L and runs in time $f(n)$ on inputs of size n. The complexity class $\mathbf{P}$ is then defined to be $\cup_{c\geq 1}\mathbf{DTIME}(n^c)$. A problem $L \subseteq \{0,1\}^*$ is in the complexity class $\mathbf{NTIME}(f(n))$ if there exists a non-deterministic Turing machine M that accepts L and runs in time $f(n)$ on inputs of size n. Alternatively, $L \in \mathbf{NTIME}(f(n))$ if there exists $L' \in \mathbf{DTIME}(n)$ such that $L = \left\{x : \exists y \in \{0,1\}^{f(n)} \text{ such that } (x,y) \in L'\right\}$ (here "(x,y)" denotes the concatenation of the strings x and y). The class $\mathbf{NP}$ is then defined to be $\cup_{c\geq 1}\mathbf{NTIME}(n^c)$.

Recall that a language L is *polynomial-time reducible* to another language L' if there exists a polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that $x \in L$ iff $f(x) \in L'$. A problem L is $\mathbf{NP}$-*hard* if *all* languages in $\mathbf{NP}$ are polynomial time reducible to L. A problem L is $\mathbf{NP}$-*complete* if it is $\mathbf{NP}$-hard and also in $\mathbf{NP}$.

An important class of problems we will be concerned with are the class of $\mathbf{NP}$-*hard optimization problems*. An optimization problem L is either a minimization or maximization problem and is defined by a polynomial-time computable predicate $P(x,y)$, a polynomial p, and a cost function $f : \{0,1\}^* \to \mathbb{Z}$. Given an input x, a minimization problem outputs a string y, $|y| \leq p(|x|)$, such that $P(x,y)$ holds and such that $f(y)$ is minimized; a maximization instead seeks to maximize $f(y)$. An optimization problem L is $\mathbf{NP}$-hard if for every problem L' in $\mathbf{NP}$ there exists a polynomial time algorithm that decides L' provided access to L as an oracle.

Example 1.1. The problem of finding a vertex cover of minimum size in a graph is an $\mathbf{NP}$-hard optimization problem: The predicate $P(x,y)$ takes as input a graph x and a candidate vertex cover y and checks that y is a vertex cover in x. Since the problem "A graph G has a vertex cover of size k" is known to be $\mathbf{NP}$-complete, it follows that minimum vertex cover is indeed $\mathbf{NP}$-hard.

Since computing an optimal solution for an instance of an $\mathbf{NP}$-hard optimization problem is intractable unless $\mathbf{P} = \mathbf{NP}$, we can ask if it is possible to efficiently find a solution whose cost is not much worse than the optimal cost. A minimization optimization problem L has a *polynomial time α-approximation algorithm*, $\alpha \geq 1$, if there exists a polynomial time algorithm which for *every* instance produces a solution with cost at most αOPT where OPT is the value of the optimal solution. Note that α can depend on n. A maximization problem has an α-approximation algorithm if the solutions have cost at most OPT/α. For more about approximation algorithms see Vazirani [55] and Hochbaum [33].

Probabilistically checkable proofs (PCPs) and the PCP theorem [8, 6] have been used to show for several **NP**-hard optimization problems that for certain factors $\alpha > 1$, computing an α-approximation of the problem is also **NP**-hard. That is, for many problems finding an approximate solution is just as hard as computing the optimum exactly. Compendia of such results can be found in Arora and Lund [5], and more recently, Feige [21] and Trevisan [53].

1.1.3 Integer and linear programming

Two important tools in combinatorial optimization for obtaining approximate solutions for **NP**-complete problems are *linear programming* and *positive semidefinite programming relaxations* (the latter is the topic of Section 1.1.5).

Before describing these methods, we begin with some definitions. A set S is called *convex* if for all $x, y \in S$, $\frac{1}{2}(x + y) \in S$. The *convex hull* of a set $S \subseteq \mathbb{R}^n$, denoted $conv(S)$, is the smallest convex set in $\mathbb{R}^n$ containing S. A set P is a *polytope* if it can be defined by $\{x : Ax \geq b\}$ for some matrix A and vector b. Let P be a polytope. If $a^T x \geq b$ is an inequality such that a point x satisfies the inequality iff $x \in P$ then the inequality is called a *facet* of P. The polytope P_I consisting of all points in the convex hull of the set $P \cap \{0, 1\}^n$ is called the *integral hull* for P.

A set is called a *cone* if it is closed under multiplication by a nonnegative number. A *convex cone* is therefore a cone that is closed under nonnegative linear combinations. Given a closed convex cone K, its *polar cone* K^* is defined as

$$K^* = \left\{a \in \mathbb{R}^{n+1} : a^T x \geq 0 \quad \forall x \in K\right\}. \tag{1.1}$$

Intuitively, K^* consists of all valid linear constraints for K.

The motivation behind using linear (and positive semidefinite) programming relaxations is the fact the Ellipsoid method can be used to efficiently optimize any linear function arbitrarily well over *any* convex set K that has a polynomial-time *weak separation oracle* which we define shortly (see Grötschel, Lovász and Schrijver [28, 29] for details).[1] In particular, the set need not have an explicit polynomial size description. To keep the notation clean we will define separation oracles for convex cones rather than polytopes; extending the definition to polytopes is straightforward.

[1] In addition, K must satisfy some technical properties, namely that K both contains and is contained within two balls whose sizes are known.

Definition 1.2. A *strong separation oracle* for a convex cone $K \subseteq \mathbb{R}^n$ is an running in polynomial time that given a vector $x \in \mathbb{R}^n$ either states that $x \in K$ or returns a vector $w \in \mathbb{R}^n$ such that $w^T y \geq 0$ for all $y \in K$, but $w^T x < 0$. The vector w is called a *separating hyperplane*. A *weak separation oracle* allows numerical errors: The input now consists of a vector $x \in \mathbb{R}^n$ and a rational number $\epsilon > 0$. The oracle then states in time polynomial in n and $1/\epsilon$ either that the Euclidean distance between x and K is at most ϵ or it returns a vector w such that $|w| \geq 1$, $w^T x \leq \epsilon$ and the Euclidean distance between w and K^* is at most ϵ.

An n-variable *linear program* is the following optimization problem:

$$\begin{aligned} &\min c^T x \\ \text{s.t. } & Ax \geq b, \\ & x \geq 0, \end{aligned} \tag{1.2}$$

where x is an n-dimensional vector, and $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. When the variables are restricted to be integers we get an *integer linear program* or simply *integer program* for short. By definition, the set of feasible solutions for a linear program forms a polytope. The central object of study in this thesis is to determine for a variety of linear programs how the polytope of feasible solutions compares with its *integral hull*, the convex hull of all integral solutions.

The problem of checking if a 3CNF formula ϕ is satisfiable can be expressed as an integer program of size polynomial in $|\phi|$ and hence integer programming is **NP**-hard. On the other hand, the feasible region of a linear program with polynomially many constraints (in n) trivially has a strong separation oracle. Hence, (polynomial size) linear programming is in **P**.

This suggests the following approach for approximating **NP**-hard optimization problems: We wish to optimize a linear function f over some set $F \subseteq \{0,1\}^n$. First we obtain a description for the convex hull polytope $conv(F)$. By the above discussion, the convex hull generally need not have a polynomial description (or even a weak separation oracle), so we instead try to find a (polynomial size) linear program whose integral hull $P_I = conv(F)$. The polytope P of feasible solutions to the linear program is called a *linear programming relaxation* or simply a *linear relaxation* for P_I. The quality of the relaxation P is measured by its *integrality gap* which is the ratio $\frac{\text{optimum value of } f \text{ over } P_I}{\text{optimum value of } f \text{ over } P}$.[2]

Example 1.3. For VERTEX COVER on graphs, represent each vertex cover as an incidence vector on the set of vertices and denote the convex hull of all such vectors

[2] When designing approximation algorithms, one also needs some kind of *rounding algorithm* which converts fractional solutions to integer ones in polynomial time. However, it has been empirically observed that no matter which rounding algorithm is used, the resulting approximation algorithm never has an approximation ratio better than the integrality gap. Hence, we will not worry about rounding algorithms in this thesis.

by $VC(G)$. Then $VC(G) = P_I$ where P is the polytope defined by the following constraints:

$$x_i + x_j \geq 1 \quad \forall \{i, j\} \in E, \tag{1.3}$$

$$0 \leq x_i \leq 1 \quad \forall i \in V. \tag{1.4}$$

It can be shown that each $x_i \in \left\{0, \frac{1}{2}, 1\right\}$ in any optimal solution. Hence, the integrality gap for this relaxation of $VC(G)$ is 2 since we can obtain a vertex cover from a fractional solution by taking all vertices i such that $x_i \in \left\{\frac{1}{2}, 1\right\}$ (for details see [32]).

Given a linear relaxation we could try to "tighten it" by adding to it inequalities satisfied by the integral hull. Extensive research has been done for finding tighter relaxations for specific optimization problems. Moreover, several techniques have been developed for systematically tightening relaxations over many rounds. This is the topic of Chapter 2 where we discuss "lift-and-project" methods and how they are used to tighten relaxations.

1.1.4 Farkas's lemma and Duality

For proofs of the results given in this section see Papadimitriou and Steiglitz [46] or Vazirani [55].

Recall the general linear program given in (1.2). Farkas's lemma tells us precisely when the set of conditions for such an LP are satisfiable:

Lemma 1.4 (Farkas's Lemma). *The set of conditions in* (1.2) *is infeasible iff there exist an m-dimensional vector $\lambda \geq \mathbf{0}$ such that $\lambda^T A < \mathbf{0}$ but $\lambda \cdot b > 0$.*

Above, the notation $u < v$ (respectively, $u \leq v$) for two vectors u, v means u is componentwise strictly less (respectively, less than or equal) than v.

Aside from giving a useful condition for testing whether a set of constraints is feasible (indeed we will use Farkas's lemma for this purpose in Chapter 4), Farkas's lemma is also used to prove the (strong) LP duality theorem which we state shortly. First we define the *dual* of the linear program given in (1.2). Our original linear program (1.2) (called the *primal*) had n variables and m constraints. The dual of (1.2) has m variables $y_1, \ldots, y_m$, and n constraints and is given by the following linear program:

$$\begin{aligned} \max y^T b \\ \text{s.t. } y^T A \leq c^T, \\ y \geq 0. \end{aligned} \tag{1.5}$$

Note that if the primal contains an equality constraint instead of an inequality, then the variable in the dual corresponding to that constraint is unconstrained (i.e., it need not be non-negative). Note also that the dual of the dual is the primal.

Theorem 1.5 (LP duality theorem). *If the primal and dual are both feasible, then their optima are identical.*

1.1.5 Semidefinite programming

One powerful technique for tightening relaxations is positive semidefinite programming. Recall that an $n \times n$ matrix Y is *positive semidefinite* (notated $Y \succeq 0$) if it is symmetric and $x^T Y x \geq 0$ for all $x \in \mathbb{R}^n$.

Lemma 1.6. *Let Y be an $n \times n$ symmetric matrix. Then the following are equivalent:*

1. *Y is positive semidefinite*
2. *All the eigenvalues of Y are nonnegative*
3. *There exist real-valued vectors $v_1, \ldots, v_n$ such that $Y_{ij} = v_i^T v_j$.*
4. *For all $i \in [n]$ the upper-left $i \times i$ submatrix of Y has a positive determinant.*

Corollary 1.7. *The set of positive semidefinite matrices is a convex cone.*

A *semidefinite program* is the following optimization problem:

$$\begin{aligned} \max C \cdot Y& \\ \text{s.t. } A_1 \cdot Y \geq b_1&, \\ \vdots& \\ A_m \cdot Y \geq b_m&, \\ Y \succeq 0&, \end{aligned}$$

where Y is an $n \times n$ matrix of variables, $C, A_1, \ldots, A_m \in \mathbb{R}^{n \times n}$, $b_1, \ldots, b_m \in \mathbb{R}$ and $X \cdot Y$ is interpreted as $\sum_{i,j \in [n]} X_{ij} Y_{ij}$. Note that the semidefinite program is just a linear program on the n^2 variables Y_{ij} together with linear constraints $A_i \cdot Y \geq b_i$, but with the extra condition that the variables Y_{ij} must also form a positive semidefinite matrix.

On account of Corollary 1.7, if there exists a weak separation oracle for the constraints of a positive semidefinite program, then the Ellipsoid method can be used to solve the program to arbitrary precision in polynomial time (see Grötschel, Lovász and Schrijver [28, 29] for details).

Example 1.8. We can often use semidefinite programming to try to "simulate" the power of quadratic programming. For example, consider the following quadratic integer programming formulation for graph VERTEX COVER:

$$\begin{aligned} \min \sum_{i=1}^{n} (1 + x_i x_0)/2 & \\ (x_0 - x_i)(x_0 - x_j) = 0 & \qquad \forall \{i, j\} \in E \\ x_i \in \{-1, 1\} & \qquad \forall i \in \{0, 1, \ldots, n\}\,. \end{aligned}$$

The set of vertices i with $x_i = x_0$ correspond to the vertex cover. Suppose we relax the variables $x_i \in \{-1, 1\}$ to vectors v_i of norm 1:

$$\begin{aligned} \min \sum_{i=1}^{n} (1 + v_i \cdot v_0)/2 & \\ (v_0 - v_i) \cdot (v_0 - v_j) = 0 & \qquad \forall \{i, j\} \in E \\ \|v_i\| = 1 & \qquad \forall i \in \{0, 1, \ldots, n\}\,. \end{aligned} \tag{1.6}$$

To see that the above is in fact a semidefinite relaxation, we use Lemma 1.6 which shows that (1.6) is equivalent to the following (more explicit) semidefinite program:

$$\begin{aligned} \min \sum_{i=1}^{n} (1 + Y_{i0})/2 & \\ Y_{00} - Y_{0j} - Y_{0i} + Y_{ij} = 0 & \qquad \forall \{i, j\} \in E \\ Y_{ii} = 1 & \qquad \forall i \in \{0, 1, \ldots, n\} \\ Y \succeq 0. & \end{aligned}$$

Semidefinite programming relaxations are at the root of many recent breakthrough approximation algorithms such as the Goemans-Williamson [27] algorithm for MAX-CUT, the Karloff-Zwick [35] algorithm for MAX-3SAT, and the Arora-Rao-Vazirani [7] algorithm for SPARSEST-CUT.

Chapter 2

Lift-and-project methods

2.1 Motivation

Consider the standard linear relaxation for VERTEX COVER:

$$\min \sum_{i\in[n]} x_i \tag{2.1}$$

$$x_i + x_j \geq 1 \qquad \forall \{i,j\} \in E \qquad (\textit{Edge constraints}) \tag{2.2}$$

$$0 \leq x_i \leq 1 \qquad \forall i \in [n] \qquad (\textit{Non-negativity constraints}) \tag{2.3}$$

In this relaxation the x_i's are real numbers in $[0,1]$. Suppose we wish to tighten the relaxation to force the x_i's to be 0/1 in any optimal solution. To this end, we could introduce any constraints satisfied by 0/1 vertex covers. For instance, the x_i's can be required for every odd-cycle C to satisfy the following constraint:

$$\sum_{i\in C} x_i \geq \frac{|C|+1}{2} \qquad (\textit{Odd-cycle constraint}) \tag{2.4}$$

Many other families of inequalities satisfied by 0/1 vertex covers are known, but a complete listing will probably never be found because of complexity reasons: since VERTEX COVER is **NP**-complete, such a list cannot be polynomial size unless $\mathbf{P} = \mathbf{NP}$.

Lovász and Schrijver [43] and Sherali and Adams [49] give automatic methods for generating over many rounds *all* valid inequalities. In particular, they give methods for obtaining tighter and tighter relaxations for any 0/1 optimization problem starting from an arbitrary relaxation. The idea is to "lift" to n^2 dimensions, add constraints in this high-dimensional space, and then project back to n-space. This is why their procedures are called "lift-and-project" or sometimes simply "lifting".

The motivation is two-fold. On the one hand, while a polytope may have exponentially many facets, it may be that the polytope is a projection of a high-dimensional polytope with fewer facets. The second (related) motivation is to try to simulate the power of quadratic programs. Solving quadratic programs is of course **NP**-hard since adding the quadratic constraints $x_i(1-x_i)=0$ to a linear relaxation forces 0/1 answers. For example, all 0/1 vertex covers satisfy

$$\begin{aligned} x_i^2 &= x_i \\ (1-x_i)(1-x_j) &= 0 \qquad \forall\{i,j\}\in E \end{aligned}$$

To linearly simulate these constraints, the methods of Lovász and Schrijver introduce new linear variables Y_{ij} to "represent" the products x_ix_j and then demand that the "lifted" variables satisfy $1-x_i-x_j+Y_{ij}=0$ for all edges $\{i,j\}$. In addition, since $x_i=x_i^2$ for 0-1 variables, we can demand that $Y_{ii}=x_i$. Using this rule, we can then take positive linear combinations of these constraints to eliminate all "quadratic" terms and obtain constraints using only the original variables x_i.

Note that the relaxation we obtain in this way is at least as good as the trivial linear relaxation for VERTEX COVER. Indeed since the lifted variables Y_{ij} must lie in $[0,1]$, the constraint $1-x_i-x_j+Y_{ij}=0$ implies the original edge constraint $x_i+x_j\geq 1$.

Since the variables Y_{ij} are supposed to "represent" the products x_ix_j, we can also add constraints that force the matrix Y of variables Y_{ij} to "behave" like the matrix X whose (i,j)th entry is the product x_ix_j. For example, since $x^2=x$ for boolean variables, we should have $x_i=Y_{ii}$. Moreover, since $x_ix_j=x_jx_i$, the matrix Y should be symmetric. This justifies our use above of the notation Y_{ij} instead of $Y_{i,j}$. Finally, note that Lemma 1.6 implies that X is positive semidefinite since the (i,j)th entry of X is equal to the dot-product of the (one-dimensional) vectors x_i and x_j. Hence, we can also demand that $Y\succeq 0$. The more constraints we put on Y, the tighter the relaxation we should get. Indeed, Lovász and Schrijver obtain three progressively stronger methods for tightening relaxations by adding more and more such constraints to Y.

2.2 The Lovász-Schrijver method

The notation uses homogenized inequalities. In particular, we introduce a new variable x_0 and replace each constraint $a^Tx\geq b$ in our relaxation with the constraint $a^Tx\geq bx_0$. Hence, if P is an n-dimensional polytope contained in $[0,1]^n$, we work instead with the (closed) convex cone

$$K^P=\left\{\begin{pmatrix}\lambda\\ \lambda x\end{pmatrix}\in\mathbb{R}^{n+1}: x\in P,\lambda\in\mathbb{R}\right\}.$$

In practice, the polytope P will always be clear from context so we will simplify our notation and simply write K for K^P. Let K_I denote the convex cone generated by all 0-1 vectors in K. Note that the projection of K_I on the hyperplane $x_0 = 1$ is precisely the convex hull of all integral vectors in P which we denote by P_I.

Let $Q \subseteq \mathbb{R}^{n+1}$ denote the closed convex cone generated from the polytope $[0,1]^n$ using the above homogenization procedure (i.e., $Q = K^{[0,1]^n}$). Note that since the polytope $[0,1]^n$ is defined by the constraints $e_i^T x \geq 0$ and $(e_0 - e_i)^T x \geq 0$, the polar cone Q^* for Q is therefore the cone spanned by the $2n$ vectors e_i and $f_i = e_0 - e_i$.

Now suppose K_1 is a closed convex cone contained in Q. Then $Q^* \subseteq K_1^*$ and hence, the constraints $x^T e_i \geq 0$ and $x^T f_i \geq 0$ are satisfied by all $x \in K_1$. Suppose moreover that $u^T x \geq 0$ is some constraint satisfied by all $x \in K_1$, that is, $u \in K_1^*$. Then the quadratic constraints $(u^T x)(x^T e_i) \geq 0$ and $(u^T x)(x^T f_i) \geq 0$ are also satisfied by all $x \in K_1$. Intuitively, this can be interpreted as follows when we project back onto the hyperplane $x_0 = 1$: multiplying the constraints defining a polytope $P \subseteq [0,1]^n$ by the constraints $x_i \geq 0$ and $x_i \leq 1$ gives quadratic constraints valid for P.

More generally, we could multiply the constraints defining K_1 by any constraints satisfied by K_1 to obtain valid quadratic constraints on K_1. Formally, suppose K_1 and K_2 are closed convex cones contained in Q, and consider the cone $K_1 \cap K_2$ (in practice, we will always have $K_1 \subseteq K_2$ so that $K_1 \cap K_2 = K_1$). Then for all $u \in K_1^*$ and all $v \in K_2^*$, the constraint $(u^T x)(x^T v) \geq 0$ is valid for $K_1 \cap K_2$. Indeed, it is easy to see that

$$K_1 \cap K_2 = \left\{x : (u^T x)(x^T v) \geq 0 \quad \forall u \in K_1^*, \forall v \in K_2^*\right\}.$$

Let us focus on all 0-1 vectors $x \in K_1 \cap K_2$ such that $x_0 = 1$, i.e., on vectors $x \in (K_1 \cap K_2)_I$. For such a vector x, if we set $Y = xx^T$ and use the fact that $x_i^2 = x_i$ for 0-1 variables, then the following properties hold:

1. $u^T Y v \geq 0$ for all $u \in K_1^*$ and all $v \in K_2^*$. Equivalently, $YK_2^* \subseteq K_1$.

2. $Ye_0 = diag(Y)$, i.e., $Y_{i0} = Y_{ii}$.

3. Y is symmetric.

4. Y is positive semidefinite.

Lovász and Schrijver derive their lift-and-project procedure by relaxing the above constraints to include *all* matrices $Y \in \mathbb{R}^{(n+1)\times(n+1)}$ that satisfy the above properties.

Formally, given K_1 and K_2, define the following cones:

$$M(K_1, K_2) = \left\{Y \in \mathbb{R}^{(n+1)\times(n+1)} : Y \text{ satisfies (1)–(3)}\right\},$$
$$M_+(K_1, K_2) = \left\{Y \in \mathbb{R}^{(n+1)\times(n+1)} : Y \text{ satisfies (1)–(4)}\right\}.$$

The above represent the "lifting" part of the construction since the cones M and M_+ are over $(n+1)^2$ dimensions.

The cones M and M_+ "simulate" quadratic constraints with linear constraints in the following sense: Consider the set of quadratic constraints on x obtained by multiplying a constraint $u^T x \geq 0$ for K_1 with a constraint $v^T x \geq 0$ for K_2. Next "linearize" these quadratic constraints by replacing quadratic products $x_i x_j$ with linear variables Y_{ij}. Then these linear constraints are satisfied by all matrices Y in $M(K_1, K_2)$ and $M_+(K_1, K_2)$.

The projections of the cones $M(K_1, K_2)$ and $M_+(K_1, K_2)$ are defined as follows:

$$N(K_1, K_2) = \{Ye_0 : Y \in M(K_1, K_2)\},$$
$$N_+(K_1, K_2) = \{Ye_0 : Y \in M_+(K_1, K_2)\}.$$

By the above discussion it immediately follows that,

$$(K_1 \cap K_2)_I \subseteq N_+(K_1, K_2) \subseteq N(K_1, K_2).$$

On the other hand, suppose $x \in N(K_1, K_2)$. Then there exists $Y \in M(K_1, K_2)$ such that $Ye_0 = x$. Since $Q^* \subseteq K_2^*$, it follows from property (1) that $u^T Y e_i \geq 0$ and $u^T Y f_i \geq 0$ for every $u \in K_1^*$. But then, adding these inequalities we have that $u^T Y e_0 = u^T x \geq 0$ for all $u \in K_1^*$, and hence, $x \in K_1$. Similarly, $x \in K_2$. Hence,

$$(K_1 \cap K_2)_I \subseteq N_+(K_1, K_2) \subseteq N(K_1, K_2) \subseteq K_1 \cap K_2. \tag{2.5}$$

In this thesis we will only be concerned with the case $K_2 = Q$ in which case we will simply write $N(K) = N(K, Q)$ and $N_+(K) = N_+(K, Q)$. Note that given a polytope P we will often abuse notation and write $N(P)$ or $N_+(P)$ to denote the cones $N(K^P)$ and $N_+(K^P)$.

The operators N and N_+ may be iterated. Define $N^r(K)$ recursively by having $N^0(K) = K$ and letting $N^r(K) = N(N^{r-1}(K))$ for $r \geq 1$. The iterated N^+ operator is defined analogously. The hierarchies defined by these iterated operators are called the LS and LS_+ *hierarchies*, respectively.

By definition of Q, we have the following lemma characterizing the N and N_+ operators:

Lemma 2.1. *Let K be a closed convex cone in $\mathbb{R}^{n+1}$. Then $x \in N(K)$ iff there exists a symmetric matrix $Y \in \mathbb{R}^{(n+1)\times(n+1)}$ satisfying*

1. *$Ye_0 = \mathrm{Diag}(Y) = x$.*

2. *For $1 \leq i \leq n$, both Ye_i and $Y(e_0 - e_i)$ are in K.*

Moreover, $x \in N_+(K)$ iff Y is in addition positive semidefinite.

Following Buresh-Oppenheim et al. [11] we will often call the matrix Y witnessing that x is in $N(K)$ (or $N_+(K)$) a *protection matrix* since it "protects" x for one round of lift-and-project.

Lemma 2.1 will be at the root of all lower bounds proved in this thesis. Recall that a linear relaxation has an integrality gap of α if there exists a feasible vector x for the relaxation for which the objective is a factor α better than the cost of the best integral solution. Such a vector x *witnesses* the integrality gap of α. So to show that the integrality gap of a relaxation remains α even after r applications of, say, the N operator, it suffices to show that some vector witnessing that integrality gap is "protected" for r rounds of the N operator. Lemma 2.1 then suggests using an inductive argument to prove such a result. This is indeed what we will do when proving integrality gaps for polytopes in the LS and LS_+ hierarchies.

In practice, we will only be concerned with showing that vectors $x \in \mathbb{R}^{n+1}$ with $x_0 = 1$ survive a round of lifting. For such points, we have the following corollary of Lemma 2.1:

Corollary 2.2. *Let K be a cone in $\mathbb{R}^{n+1}$ and suppose $x \in \mathbb{R}^{n+1}$ where $x_0 = 1$. Then $x \in N(K)$ iff there is a symmetric matrix $Y \in \mathbb{R}^{(n+1)\times(n+1)}$ satisfying*

1. *$Ye_0 = diag(Y) = x$.*
2. *For $1 \leq i \leq n$: If $x_i = 0$ then $Ye_i = \vec{0}$; If $x_i = 1$ then $Ye_i = x$; Otherwise, Ye_i/x_i, $Y(e_0 - e_i)/(1 - x_i)$ both lie in the projection of $N(K)$ onto the hyperplane $x_0 = 1$.*

Moreover, $x \in N_+(K)$ iff Y is in addition positive semidefinite.

Lovász and Schrijver [43] show that at most n iterations of even the weaker N operator suffice to obtain the integral hull.

Theorem 2.3 ([43]). *Let $K \subseteq Q$ be a closed convex cone in $\mathbb{R}^{n+1}$. Then $N^n(K) = K_I$.*

Proof. Suppose first that $x \in N(K)$ and let $Y \in M(K, K)$ be a protection matrix for x. Since Y is a protection matrix for x, it follows that $(Ye_i)_i = (Ye_i)_0$ and $(Yf_i)_i = 0$. Hence, $Ye_i \in K|_{x_i=x_0}$ and $Yf_i \in K|_{x_i=0}$. But then, since $x = Ye_0 = Ye_i + Yf_i$, it follows that $x \in K|_{x_i=x_0} + K|_{x_i=x_0}$. Since this holds for all $i \in [n]$, it follows that,

$$N(K) \subseteq \bigcap_{i\in[n]} \left(K|_{x_i=x_0} + K|_{x_i=x_0}\right).$$

Iterating, it then follows that,

$$N^r(K) \subseteq \bigcap_{\{i_1,i_2,\ldots,i_r\}\subseteq[n]} \left(\sum_{T\in\{0,x_0\}^r} K|_{(x_{i_1},x_{i_2},\ldots,x_{i_r})=T}\right).$$

For $r = n$ this becomes,

$$N^n(K) \subseteq \sum_{T \in \{0, x_0\}^n} K|_{(x_1, x_2, \ldots, x_n) = T} = K_I.$$

On the other hand, equation (2.5) implies that $K_I \subseteq N^n(K)$. The theorem follows. □

Corollary 2.4. *Suppose $N^r(K) \neq K_I$ where $0 \leq r < n$. Then $N^{r+1} \subsetneqq N^r(K)$.*

Lovász and Schrijver [43] define a third lift-and-project operator N_0 for a cone K by having

$$N_0(K) = \bigcap_{i \in [n]} \left(K|_{x_i = x_0} + K|_{x_i = x_0} \right).$$

The hierarchy corresponding to this operator is called the LS_0 *hierarchy*. As the proof of Theorem 2.3 shows, even though this operator is weaker than the N operator, it still derives the integral hull after at most n rounds. Note that the operator N_0 has the same characterization as the one given for N by Lemma 2.1 with the exception that a protection matrix Y witnessing that a point survives one round of the N_0 operator does not need to be symmetric.

The cones $N_0(K)$, $N(K)$ and $N_+(K)$ have the following important algorithmic property:

Theorem 2.5. *Suppose K has a polynomial (in n) time separation oracle. Then there exists an $n^{O(r)}$ time separation oracle for $N_+^r(K)$ (and hence, for $N_0^r(K)$, $N^r(K)$ also). Hence, using the ellipsoid one can optimize a linear function over $N_0^r(K)$, $N^r(K)$ and $N_+^r(K)$ in time $n^{O(r)}$.*

Proof. By induction on r. The base case $r = 0$ follows trivially, so assume there is an $n^{O(r-1)}$ time algorithm for optimizing over $N_+^{r-1}(K)$. Let Y be any matrix and suppose we want to check that Y satisfies the conditions of Lemma 2.1. If Y violates condition (1) in the lemma or is not positive semidefinite, then this is trivially detected and a separating hyperplane is also trivially given. To check that Y satisfies condition (2) of the lemma it suffices to check that each of the Ye_i and Yf_i are in $N_+^{r-1}(K)$. But this can be done by calling the separation oracle for $N_+^{r-1}(K)$ which runs in time $n^{O(r-1)}$ by the induction hypothesis. □

In contrast, a polynomial time separation oracle for K is not known to imply the existence of a polynomial time algorithm for optimizing over the (potentially tighter) polytopes $N(K, K)$ and $N_+(K, K)$. The problem is that instead of condition (2) of Lemma 2.1 we would now have to check whether $YK^* \subseteq K$, and K^* (unlike Q^*) may require exponentially many vectors to generate.

2.2.1 Deriving inequalities in LS and LS_+

Given a relaxation

$$\begin{aligned} a_r^T x \geq b \qquad & r = 1, 2, \ldots, m \\ 0 \leq x_i \leq 1 \qquad & i = 1, 2, \ldots, n, \end{aligned}$$

one round of N produces a system of inequalities in $(n+1)^2$ variables Y_{ij} for $i, j = 0, 1, \ldots, n$. As already mentioned, the intended "meaning" is that $Y_{ij} = x_i x_j$ and $Y_{00} = 1$, $Y_{0i} = x_i x_0 = x_i$, and $Y_{00} = 1$ so every quadratic expression in the x_i's can be viewed as a linear expression in the Y_{ij}'s. This is how the quadratic inequalities below should be interpreted. The following inequalities are derived in one round:

$$\begin{aligned} (1 - x_i) a_r^T x \geq (1 - x_i) b \qquad & \forall i = 1, \ldots, n, \quad \forall r = 1, \ldots, m \\ x_i a_r^T x \geq x_i b \qquad & \forall i = 1, \ldots, n, \quad \forall r = 1, \ldots, m \\ x_i x_i = x_i \qquad & \forall i = 1, 2, \ldots, n \end{aligned}$$

The last constraint corresponds to the fact that $x_i^2 = x_i$ for 0/1 variables. In the LS_+ system we also require that the matrix with entries Y_{ij} is positive semidefinite. Since any positive combination of the above inequalities is also implied, we can use such combinations to eliminate all non-linear terms.

Example 2.6. Consider the relaxation for VERTEX COVER given by the edge constraints (2.2) and the bounding constraints $0 \leq x_i \leq 1$ for all $i \in [n]$. We show that the linearization $Y_{00} - Y_{i0} - Y_{j0} + Y_{ij} = 0$ of the quadratic constraint $(x_i - 1)(x_j - 1) = 0$ for all edges $\{i, j\} \in E$ is derived by one application of the N operator.

Suppose $\{i, j\} \in E$. Then given the constraint $x_i + x_j \geq 1$, one round of N derives the constraint $(1 - x_i)(x_i + x_j) \geq 1 - x_i$. Simplifying using the rule $x_i = x_i^2$ and replacing products $x_i x_j$ with Y_{ij}, we get that $Y_{00} - Y_{i0} - Y_{j0} + Y_{ij} \leq 0$. On the other hand, starting from the constraint $x_i \leq 1$, one round of N derives the constraint $(1 - x_j) x_i \leq 1 - x_j$, whose linearization is $Y_{00} - Y_{i0} - Y_{j0} + Y_{ij} \geq 0$. Hence, $Y_{00} - Y_{i0} - Y_{j0} + Y_{ij} = 0$ is indeed derived after one round of the N operator.

Example 2.7. Lovász and Schrijver [43] show that the set of inequalities derivable in one round of N for the trivial VERTEX COVER relaxation (i.e., the one given by the edge constraints (2.2) and the constraints $0 \leq x_i \leq 1$, $i \in V$) are exactly the odd-cycle inequalities. To illustrate, we show how to derive in one round the odd-cycle inequality $x_1 + x_2 + x_3 \geq 2$ for a triangle on nodes $\{1, 2, 3\}$ starting from the edge constraints (2.2). One round of N generates the following inequalities (amongst others):

$$\begin{aligned}(1-x_1)(x_1+x_2) &\geq 1-x_1 &(2.6)\\ (1-x_2)(x_2+x_3) &\geq 1-x_2 &(2.7)\\ (1-x_3)(x_1+x_3) &\geq 1-x_3 &(2.8)\\ x_1(x_2+x_3) &\geq x_1 &(2.9)\\ x_2(x_1+x_3) &\geq x_2 &(2.10)\end{aligned}$$

Adding inequality (2.6) twice to the sum of the remaining four inequalities and then simplifying using the rule $x_i^2 = x_i$ gives $x_1 + x_2 + x_3 \geq 2$ as desired.

In the next three subsections we derive some standard SDP relaxations using N_+. These derivations appear to be "folklore" results (the latter two derivations were noted in [1]).

Deriving the standard SDP relaxation for Vertex Cover

Recall the standard LP relaxation for VERTEX COVER given by (2.1)–(2.3) and let K be the cone obtained by homogenizing the polytope corresponding to this relaxation. We show that the relaxation obtained after tightening K with one round of N_+ is at least as good as the standard SDP relaxation for VERTEX COVER given in (1.6) in the following sense: We will show that for every point $x \in N_+(K)|_{x_0=1}$ there exists a feasible set of vectors v_i for (1.6) such that the value of (2.1) at x is the same as the value of the SDP objective for the vectors v_i.

Suppose $x \in N_+(K)|_{x_0=1}$. Then there exists a matrix $Y \in M_+(K)$ such that $x = Ye_0 = diag(Y)$ and $Y_{00} = 1$. Since Y is positive semidefinite, there exist vectors $u_0, u_1, \ldots, u_n$ such that $Y_{ij} = u_i \cdot u_j$ for all $i, j \in \{0, 1, \ldots, n\}$. For $i = 0, 1, \ldots, n$, let $v_i = u_0 - 2u_i$. Then

$$\|v_i\|^2 = u_0 \cdot u_0 - 4u_i \cdot u_0 + 4u_i \cdot u_i = Y_{00} - 4Y_{0i} + 4Y_{ii} = 1.$$

Moreover, if $\{i, j\} \in E$, then

$$(v_0 - v_i) \cdot (v_0 - v_j) = 4(u_0 - u_i) \cdot (u_0 - u_j) = 4(Y_{00} - Y_{i0} - Y_{j0} + Y_{ij}).$$

But the latter is 0 as shown in Example 2.6. Hence, the vectors v_i give a feasible solution for (1.6). Finally, note that the value of this solution is

$$\frac{1}{2}\sum_{i\in V}(v_0 \cdot v_i + 1) = \frac{1}{2}\sum_{i\in V}((u_0 - 2u_0) \cdot (u_0 - 2u_i) + 1) = \sum_{i\in V} u_0 \cdot u_i = \sum_{i\in V} x_i.$$

Deriving the standard SDP relaxation for Max-Cut

Given a graph G, the quadratic integer programming formulation of MAX-CUT is as follows:

$$\max \sum_{\{i,j\}\in E} \frac{1}{4} |x_i - x_j|^2$$
$$x_i \in \{-1, 1\}\,.$$

In the Goemans-Williamson [27] SDP relaxation we instead have to find unit vectors $u_1, u_2, \ldots, u_n$ that maximize

$$\sum_{\{i,j\}\in E} \frac{1}{4} \|u_i - u_j\|^2. \tag{2.11}$$

We show how to derive this relaxation with one round of N_+ on the trivial linear relaxation for MAX-CUT.

The integer programming formulation of MAX-CUT has 0/1 variables x_i and d_{ij} where x_i indicates which side of the cut vertex i is on, and d_{ij} is 1 iff i and j are on opposite sides of the cut. The IP formulation is then:

$$\begin{aligned}
&\max_{\{i,j\}\in E} d_{ij} && \\
&d_{ij} \geq x_i - x_j && \forall i, j = 1, 2, \ldots, n \\
&d_{ij} \leq x_i + x_j && \forall i, j = 1, 2, \ldots, n \\
&d_{ij} \leq 2 - (x_i + x_j) && \forall i, j = 1, 2, \ldots, n
\end{aligned}$$

In the linear programming relaxation we allow the variables d_{ij}, x_i to take values in $[0, 1]$. Let P be the polytope corresponding to this relaxation and let K denote its homogenization.

One round of N_+ generates the following inequalities on d_{ij} (amongst others):

$$x_i d_{ij} \geq x_i (x_i - x_j)$$
$$(1 - x_i) d_{ij} \geq (1 - x_i)(x_j - x_i).$$

(Above $x_i x_j$ is of course short-hand for Y_{ij}.) Adding these inequalities and simplifying using the fact that $x_i^2 = x_i$, we obtain $d_{ij} \geq (x_i - x_j)^2$. Similarly one can obtain $d_{ij} \leq (x_i - x_j)^2$ whereby it follows that

$$d_{ij} = (x_i - x_j)^2 = Y_{ii} + Y_{jj} - 2Y_{ij}. \tag{2.12}$$

Suppose now that $y \in N_+(K)|_{y_0=1}$ where the first n coordinates correspond to the variables x_i and the last n^2 coordinates correspond to the variables d_{ij}. Let $Y \in$

$M_+(K)$ be such that $Ye_0 = diag(Y) = y$. Since Y is positive semidefinite, then by Lemma 1.6, so is the upper-left $(n+1)\times(n+1)$ submatrix of Y. In particular, there exist vectors $v_0, v_1, \ldots, v_n \in \mathbb{R}^{n+1}$ such that $Y_{ij} = v_i \cdot v_j$ for all $i, j \in \{0, 1, \ldots, n\}$. So since the Y_{ij} must satisfy (2.12), it follows that $d_{ij} = \|v_i - v_j\|^2$.

Since the variables in the Goemans-Williamson relaxation are simulating $-1/1$ variables rather than $0/1$ variables as in the LP relaxation, we apply a linear transformation to the vectors v_i to obtain a solution for the Goemans-Williamson SDP: Define vectors $u_1, u_2, \ldots, u_n$ by having $u_i = v_0 - 2v_i$. Then these vactors satisfy

$$
\begin{aligned}
d_{ij} &= \tfrac{1}{4}\|u_i - u_j\|^2 \\
\|u_i\|^2 &= \|v_0\|^2 - 4v_0 \cdot v_i + 4\|v_i\|^2 = Y_{00} - 4Y_{0i} + 4Y_{ii} = 1,
\end{aligned}
$$

where the last equality uses the fact that $Y_{i0} = Y_{ii}$ for all $Y \in M_+(K)$. So the u_i's are a feasible solution to the GW relaxation. We conclude that one round of N_+ produces a relaxation at least as tight as the GW relaxation.

Deriving the Arora-Rao-Vazirani SDP relaxation for Sparsest-Cut

Arora, Rao, and Vazirani [7] derive their $\sqrt{\log n}$-approximation for SPARSEST-CUT using a SDP relaxation similar to the one for MAX-CUT and whose salient feature is the *triangle inequality* (see [7] for details):

$$\|u_i - u_j\|^2 + \|u_j - u_k\|^2 \geq \|u_i - u_k\|^2 \qquad \forall i, j, k. \tag{2.13}$$

In particular, $d_{ij} = \|u_i - u_k\|^2$ forms a metric space. Note that since the u_i are supposed to be unit vectors, (2.13) is equivalent to

$$u_i \cdot u_k \geq u_i \cdot u_j + u_j \cdot u_k - 1 \qquad \forall i, j, k.$$

The ARV relaxation minus the triangle inequality is derived similarly to the GW relaxation above with one round of N_+. We claim now that the triangle inequality is implied after three rounds of N_+. That is, if K is the convex cone corresponding to the SPARSEST-CUT linear relaxation, and $Y \in M_+^3(K)$, then $Y_{ik} \geq Y_{ij} + Y_{jk} - 1$.

Note the following corollary of Theorem 2.3:

Corollary 2.8. *Let $K \subseteq Q$ be a convex cone in $\mathbb{R}^{n+1}$. For each $I \subseteq [n]$, $|I|$ rounds of N on K suffice to derive all inequalities that hold for the integral hull of K and only involve the variables from I.*

Hence, after r rounds the induced solution on subsets of size r lies in the convex hull of integer solutions for the induced problem on subsets of size r. For SPARSEST-CUT, the induced problem on subsets is itself a SPARSEST-CUT problem, and hence, after three rounds the d_{ij} variables restricted to sets of size three lie in the *cut cone*. Since the cut cone is just the set of ℓ_1 (pseudo)metrics, it follows that the d_{ij} variables form a (pseudo)metric. Thus three rounds of N_+ give a relaxation that is at least as strong as the ARV relaxation.

2.3 The Sherali-Adams method

Sherali and Adams [49] introduced their lift-and-project method contemporaneously (and independently) with those of Lovász and Schrijver. Whereas the Lovász-Schrijver systems obtain tighter and tighter relaxations by repeatedly lifting and projecting, the Sherali-Adams system keeps lifting and lifting, but only projects at the end. In fact, it is not even necessary to project in the Sherali-Adam system since we can optimize a linear function directly over the lifted polytope instead of projecting first.

The Sherali-Adams system introduces variables to simulate higher and higher degree products of the basic variables whereas Lovász-Schrijver only introduce variables to simulate quadratic products: the first level simulates quadratic products, the second cubic products, etc. Indeed, we will see that one round of Sherali-Adams tightening gives the same relaxations as one round of LS tightening.

Formally, given a closed convex cone $K \subseteq Q$ in $\mathbb{R}^{n+1}$, Sherali and Adams define for each $r \in [n]$ a hierarchy of cones $SA^r(K)$. The cone $SA^r(K)$ has a coordinate for each $s \subseteq n$, $|s| \leq r+1$ and hence lies in $\mathbb{R}^{V(n,r)}$ where $V(n,r) = \sum_{i=0}^{r+1} \binom{n}{i}$. The idea is for each variable y_s to "simulate" the homogeneous term $(\Pi_{i \in s} x_i) \times x_0^{r-|s|}$. Let $y^{(r)}$ denote the vector of all $V(n,r)$ variables. We require the following notation: For subsets $s, t, u \subseteq [n]$ define the "$\star$" operator by having $y_s \star (a y_t + y_u) = a y_{s \cup t} + y_{s \cup u}$.

The cones $SA^r(K)$ are defined inductively as follows. Let $SA^0(K)$ be the cone K where $y_{\{i\}} = x_i$ and $y_\emptyset = x_0$. The constraints defining $SA^r(K)$ for $r \geq 1$ are the following: For each constraint $a^T y^{(r-1)} \geq 0$ in $SA^{r-1}(K)$ and for each $i \in [n]$, $SA^r(K)$ has the constraints

$$
\begin{aligned}
y_{\{i\}} \star a^T y^{(r-1)} &\geq 0, \\
(1 - y_{\{i\}}) \star a^T y^{(r-1)} &\geq 0.
\end{aligned}
$$

The projected cone $S^r(K)$ in $\mathbb{R}^{n+1}$ is then defined to be the cone obtained by projecting each point $u \in SA^r(K)$ to the point $u|_{s:|s| \leq 1}$.

Note that if $x \subseteq K_I$ and $x_0 = 1$, then the vector $y^{(r)}$ defined by $y_s = \Pi_{i \in s} x_i$ is in $SA^r(K)$ for all r. Hence, $K_I \subseteq S^r(K)$ for all r. Indeed, we will show below that $K_I = S^n(K)$.

The following lemma gives an alternative characterization of $SA^r(K)$ and will play a similar role in our analysis of the Sherali-Adams hierarchy as Lemma 2.1 played in our analysis of the Lovász-Schrijver hierarchies.

Lemma 2.9. *For $u \in \mathbb{R}^{V(n,r)}$ define for all $i \in [n]$ vectors $v^i, w^i \in \mathbb{R}^{V(n,r-1)}$ such that for all $s \subseteq [n]$,$|s| \leq r$, we have $v^i_s = u_{s \cup \{i\}}$ and $w^i_s = u_s - u_{s \cup \{i\}}$. Then $u \in SA^r(K)$ iff for all $i \in [n]$ the vectors v^i, w^i lie in $SA^{r-1}(K)$.*

Proof. Suppose that $SA^{r-1}(K)$ has a constraint $a^T y^{(r-1)} \geq 0$. Then for each $i \in [n]$, $SA^r(K)$ has the two constraints $y_{\{i\}} \star a^T y^{(r-1)} \geq 0$ and $(1 - y_{\{i\}}) \star a^T y^{(r-1)} \geq 0$. If we let $b^T y^{(r)} = y_{\{i\}} \star a^T y^{(r-1)}$ and $c^T y^{(r)} = (1 - y_{\{i\}}) \star a^T y^{(r-1)}$, then

$$b_s = \begin{cases} 0 & \text{if } i \notin s \\ a_s + a_{s \setminus \{i\}} & \text{if } i \in s \end{cases}$$

and,

$$c_s = \begin{cases} a_s & \text{if } i \notin s \\ -a_{s \setminus \{i\}} & \text{if } i \in s \end{cases}$$

Hence, for all $s \subseteq [n]$ such that $i \in s$, we have that $u_s = v_s^i = v_{s \setminus \{i\}}^i$, and so,

$$\begin{aligned} b^T u &= \sum_{s: i \in s} (a_s + a_{s \setminus \{i\}}) u_s \\ &= \sum_{s: i \in s} a_s v_s^i + \sum_{s: i \notin s} a_s v_s^i \\ &= a^T v^i. \end{aligned}$$

On the other hand, we also have that $w_s^i = 0$ whenever $i \in s$ and $w_s^i = u_s - u_{s \setminus \{i\}}$ when $i \notin s$. So,

$$\begin{aligned} c^T u &= \sum_{s: i \in s} -a_{s \setminus \{i\}} u_s + \sum_{s: i \notin s} a_s u_s \\ &= \sum_{s: i \notin s} (-a_s u_{s \cup \{i\}} + a_s u_s) \\ &= a^T w^i. \end{aligned}$$

But then we have shown that u satisfies the constraints for $SA^r(K)$ iff for all $i \in [n]$, the vectors v^i and w^i satisfy the constraints for $SA^{r-1}(K)$. $\square$

We remark that the first level of the Sherali-Adams hierarchy is identical to the first level of the LS hierarchy. Indeed, suppose $x \in S(K)$. Then there exists $u \in SA(K)$ such that $x = u|_{s:|s| \leq 1}$. Let Y be the $(n+1) \times (n+1)$ symmetric matrix where $Y_{ij} = u_{\{i,j\}}$. Note that $Ye_0 = \mathrm{Diag}(Y) = x$. Moreover, if for each $i \in [n]$ we let $v^i, w^i \in SA^0(K) = K$ be the vectors for u given by Lemma 2.9, then $Ye_i = v^i$ and $Yf_i = w^i$. It follows then from Lemma 2.1 that $x \in N(K)$, and hence that $S(K) \subseteq N(K)$. Showing $N(K) \subseteq S(K)$ is similar.

Theorem 2.10. *If $K \subseteq Q$ is a closed convex cone in $\mathbb{R}^{n+1}$, then $K_I \subseteq S^r(K) \subseteq S^{r-1}(K) \subseteq K$ for every $r > 1$.*

Proof. That $K_I \subseteq S^r(K)$ was already noted above. So suppose $x \in S^r(K)$, $r \geq 1$. Hence there exists a vector $u \in SA^r(K)$ such that $x = u|_{s:|s|\leq 1}$. Lemma 2.9 shows that there exist vectors $v, w \in SA^{r-1}(K)$ such that $v_s = u_{s\cup\{1\}}$ and $w_s = u_s - u_{s\cup\{1\}}$ for each $s \subseteq [n]$, $|s| \leq r-1$. In particular, $v_s + w_s = u_s$ for each $s \subseteq [n]$, $|s| \leq r-1$. But $SA^{r-1}(K)$ is a convex cone, and hence, $v + w \in SA^{r-1}(K)$ and so, $x = (v+w)|_{s:|s|\leq 1} \in S^{r-1}(K)$. So $S^r(K) \subseteq S^{r-1}(K)$ and the theorem follows. $\square$

The following theorem shows that $S^n(K) = K_I$. The proof is nearly identical to that of Theorem 2.3 but uses Lemma 2.9 instead of Lemma 2.1 and is thus omitted.

Theorem 2.11. *If $K \subseteq Q$ is a closed convex cone in $\mathbb{R}^{n+1}$ then $S^n(K) = K_I$.*

Finally we note that Lemma 2.9 together with the proof of Theorem 2.5 show that efficient separation oracles exist for cones in the Sherali-Adams hierarchy:

Theorem 2.12. *Suppose K has a polynomial (in n) time separation oracle. Then there exists an $n^{O(r)}$ time separation oracle for $SA^r(K)$*

We have already seen that $S(K) = N(K)$. Laurent [42] proves the following more general relation between the SA and LS hierarchies which shows that the S operator is at least as strong as the N operator.

Theorem 2.13. *[42] If $K \subseteq Q$ is a closed convex cone in $\mathbb{R}^{n+1}$ then $S^r(K) \subseteq N^r(K)$ for all $r \geq 0$.*

2.3.1 Adding a positive semidefiniteness constraint

Whereas the Sherali-Adams hierarchy is at least as strong as the LS hierarchy, its relationship with the LS_+ hierarchy is not clear since the Sherali-Adams hierarchy does not include a positive semidefiniteness constraint. Sanjeev Arora has proposed a way to naturally add a positive semidefiniteness constraint to the definition of the Sherali-Adams hierarchy in order to obtain a hierarchy at least as strong as the LS_+ hierarchy. This definition is based on the alternate characterization of the Sherali-Adams hierarchy given by Lemma 2.9.

Definition 2.14. Given a point $u \in \mathbb{R}^{V(n,r)}$ and a subset $T \subseteq [n]$ of size $|T| = r-1$, let $Y^T \in \mathbb{R}^{(n+1)\times(n+1)}$ be the symmetric matrix where $Y^T_{00} = u_T$, and for all $i, j \in [n]$ we have that $Y^T_{ij} = u_{T\cup\{i,j\}}$ and $Y^T_{0i} = u_{T\cup\{i\}}$.

Definition 2.15 (Sherali-Adams with PSD constraint). Given a cone $K \subseteq Q$ in $\mathbb{R}^{n+1}$ let $SA^0_+(K)$ be the cone K and for $r \geq 1$ let $SA^r_+(K) \subseteq \mathbb{R}^{V(n,r)}$ be the cone consisting of all points $u \in \mathbb{R}^{V(n,r)}$ such that:

1. For all $i \in [n]$, the vectors $v^i, w^i \in \mathbb{R}^{V(n,r-1)}$ defined by $v^i_s = u_{s\cup\{i\}}$ and $w^i_s = u_s - u_{s\cup\{i\}}$ are in $SA^{r-1}_+(K)$.

2. For all subsets $T \subseteq [n]$ of size $|T| = r-1$ the matrix Y^T is positive semidefinite.

Similarly to the standard Sherali-Adams hierarchy, let $S_+^r(K)$ denote the projection of $SA_+^r(K)$ back to $\mathbb{R}^{n+1}$. It is not hard to verify that $N_+(K) = S_+(K)$. Moreover, Theorem 2.13 can be extended to prove that the inclusion $N_+^r(K) \subseteq S_+(K)$ holds for all r. Finally, the same argument that shows that $N_+(K)$ has a weak separation oracle whenever K does also shows that $SA_+(K)$ has a weak separation oracle whenever K does.

2.4 Previous lower bounds

Initially, research on lift-and-project methods concentrated on proving lower bounds on the number of rounds needed to derive specific inequalities valid for the integral hull of the problem in question. Results studying integrality gaps for polytopes derived by lift-and-project methods appear in more recent papers.

We first survey those results proving lower bounds on the number of rounds needed to derive inequalities using lift-and-project methods.

In their paper introducing the LS and LS_+ hierarchies, Lovász and Schrijver [43] show that the relaxation obtained after one round of N on the trivial linear relaxation is exactly the relaxation for INDEPENDENT SET defined by the edge and odd-cycle constraints for INDEPENDENT SET (these are analogous to the edge and odd-cycle constraints for VERTEX COVER given by equations (2.2) and (2.4) above). In contrast, the relaxation for INDEPENDENT SET obtained after one round of N_+ derives several other well-known constraint families, such as the so-called clique and wheel constraints.

Stephen and Tunçel [50] show that $\Omega(\sqrt{n})$ rounds of N_+ are needed to derive some simple inequalities for the MATCHING polytope. Cook and Dash [14] and Goemans and Tunçel [26] independently both show how for some simple relaxations, the full n rounds of N_+ are required to derive some simple inequalities. Laurent [42] shows that for the same example used by Cook and Dash, and Goemans and Tunçel, the Sherali-Adams procedure also requires the full n rounds to derive the integral hull.

Arora et al. [3] were the first to suggest studying the integrality gap of relaxations obtained using lift-and-project methods. They showed that the integrality for VERTEX COVER remains $2-\delta$ after $\Omega(\sqrt{\log n})$ rounds of N. Nearly contemporaneously, Feige and Krauthgamer [24] also showed that large gaps remain for INDEPENDENT SET after $\Omega(\log n)$ rounds of N_+ liftings. As mentioned in the introduction, Feige and Krauthgamer did not state their results in terms of integrality gaps; rather, they were interested in determining how many rounds are needed to derive the INDEPENDENT SET integral hull for a graph drawn at random from the $\mathcal{G}(n, \frac{1}{2})$ model. However, their results readily translate to give integrality gaps for such graphs. Concretely, their results show that for almost all graphs in $\mathcal{G}(n, \frac{1}{2})$ the integrality gap remains $\Omega\left(\frac{\sqrt{n}}{2^{r/2}\log n}\right)$ after tightening the trivial linear relaxation for INDEPENDENT SET with r rounds of N_+ lift-and-project.

Buresh-Oppenheim et al. [11] considered the problem of proving integrality gaps from the angle of propositional proof complexity. In the proof complexity setting, LS-type procedures can be viewed as deduction systems with a prescribed set of derivation rules. The axioms are the polytope constraints and the derivation rules give the inequalities implied by one round of N_+. (For more details on the relation between LS_+ refutations and LS_+ approximation algorithms see Section 7.2.) Their paper [11] shows a linear lower bound on the number of N_+ rounds needed to refute an unsatisfiable linear system for kSAT and kXOR-SAT when $k \geq 5$. In particular, for $k \geq 5$ they prove that a linear number of rounds of N_+ is needed to obtain an integrality gap better than $(2^k - 1)/2^k - \epsilon$ for MAX-kSAT. The cases when $k \leq 4$ were left open.

2.5 Results

In this thesis we prove new inapproximability results for both the LS and LS_+ hierarchies. These results can be divided into two types. On the one hand, we have inapproximability results in these hierarchies that rule out stronger approximation ratios than those ruled out by current PCP-based results. On the other, we have results for several problems which rule out even slightly subexponential non-trivial approximation algorithms in these hierarchies.

In Chapter 3 we prove inapproximability results in the LS_+ hierarchy for three different optimization problems. Before giving the results we require some notation. Let $\mathrm{HVC}(G)$ denotes the polytope corresponding to the standard relaxation for VERTEX COVER on a rank-k hypergraph G (see Section 3.2 for definitions). Let $\mathrm{SAT}(\phi)$ denote the polytope corresponding to the standard relaxation for MAX-3SAT for a 3-CNF formula ϕ (see Section 3.3 for definitions). An instance of SET COVER consists of a tuple (S, C) where C is a collection of n subsets of a finite set S of size m. Given an instance (S, C) of SET COVER, let $\mathrm{MSC}(S, C)$ denote the polytope corresponding to the standard relaxation for the SET COVER problem on (S, C) (see Section 3.4 for definitions). We prove the following three theorems in Chapter 3.

Theorem 3.4. *Let $k \geq 3$. For all $\alpha > 0$ there exists a constant $\gamma > 0$ and a k-uniform hypergraph G such that the integrality gap of $N_+^r(\mathrm{HVC}(G))$ is at least $(k-1)(1-\alpha)$ for all γn.*

Theorem 3.14. *For any constant $\alpha > 0$, there exist constants $\beta, \gamma > 0$ such that if ϕ is a random βn clause 3-CNF formula on n variables, then with high probability* *the integrality gap for $N_+^r(\mathrm{SAT}(\phi))$ is at least $\frac{8}{7} - \alpha$ for all $r \leq \gamma n$.*

Theorem 3.20. *For all $\epsilon > 0$, there exists $\delta > 0$ and an instance (S, C), $|S| = n$, of* SET COVER *for which the integrality gap of $N_+^r\,\mathrm{MSC}(S, C)$ is at least $(1 - \epsilon)\ln n$ for all $r \leq \delta n$.*

These three results are joint work with Mikhail Alekhnovich and Sanjeev Arora and appear in [1]. Note that there are inapproximability results in the PCP setting where all the above factors appear [15, 30, 20, 47]. However, all these results use reductions that greatly blow up the instance size, and hence imply the above integrality gaps—regardless of the computational complexity assumption—for only n^δ rounds (for some small constant $\delta > 0$) and not for $\Omega(n)$ rounds. Moreover, for SET COVER the PCP results are even weaker: an integrality gap of $(1 - \epsilon)\ln n$ for is implied only for $n^{o(1)}$ rounds [20]. Note that the PCP results for SET COVER in [47] do imply an $\Omega(\log n)$ gap for n^δ rounds for some constant $\delta > 0$; however the gap given is at most $c \log n$ for some small constant c.

The integrality gaps given by the above three theorems are all proved using a refined version of the "expanding constraints" method first introduced in Buresh-Oppenheim et al. [11]. This method takes advantage of certain graph expansion properties satisfied by the constraints defining the initial relaxation. It uses these expansion properties to "clean up" the vectors being protected with Lemma 2.1 in the inductive lower bound proof. The main technical contribution in our results is a more refined "cleaning up" strategy (called *expansion correction*; see Chapter 3 for details) necessitated by the far more complicated (yet still positive semidefinite) protection matrices demanded by the optimization problems we consider.

As is clear from the section on related work above, the approximation gap for the minimum VERTEX COVER problem on graphs has focused on this problem a lot of the attention of researchers working with lift-and-project methods. We contribute to this line of work by proving the following integrality gaps for VERTEX COVER in the *LS* hierarchy in Section 4.1 of Chapter 4. Given a graph G, let $\mathrm{VC}(G)$ denote the polytope corresponding to the standard relaxation for VERTEX COVER for G.

Theorem 4.1. *For all $\epsilon > 0$ there exists an integer n_0 and a constant $\delta(\epsilon) > 0$ such that for all $n \geq n_0$, there exists an n vertex graph G for which the integrality gap of $N^r(\mathrm{VC}(G))$ for any $r \leq \delta(\epsilon)\log n$ is at least $2 - \epsilon$.*

This result is joint work with Sanjeev Arora, Béla Bollobás and László Lovász and appears in [4]. The proof builds on techniques used to prove *LS* hierarchy lower bounds both for VERTEX COVER in Arora et el. [3] and for INDEPENDENT SET in Lovász and Schrijver [43]. In particular, like those results, the protection matrices used to prove Theorem 4.1 are not explicitly described; instead the appropriate protection matrices for proving the lower bound are implicitly shown to exist using Farkas's lemma (Lemma 1.4).

As previously mentioned, the approximation ratio ruled out by Dinur et al. [15] for VERTEX COVER on rank-k hypergraph is only $k-1-\epsilon$. In contrast, the best algorithms known achieve only $k-o(1)$ approximations. We give evidence that the latter is indeed the best ratio achievable by proving the following theorem in Section 4.2 of Chapter 4.

Theorem 4.13. *For all $k \geq 2$, $\epsilon > 0$ there exist constants $n_0(k,\epsilon), \delta(k,\epsilon) > 0$ s.t. for every $n \geq n_0(k,\epsilon)$ there exists a k-uniform hypergraph G on n vertices for which the integrality gap of $N^r(\mathrm{HVC}(G))$ is at least $k-\epsilon$ for all $r \leq \delta(k,\epsilon)\log\log n$.*

This result can be viewed as saying that no non-trivial "polynomial time" approximation algorithms exist for rank-k hypergraph VERTEX COVER in the LS computation model. This result was published in [51]. Like Theorem 4.1, the protection matrices used in the proof of Theorem 4.13 are not explicitly defined but instead implicitly described using LP duality.

In Chapter 5 we prove a second inapproximability result for graph VERTEX COVER in the LS hierarchy.

Theorem 5.3. *For all $\epsilon > 0$ there exists a constant $\delta > 0$ and an integer n_0 such that for all $n \geq n_0$ there exists an n-vertex graph G for which $N^r(\mathrm{VC}(G))$ has an integrality gap of at least $1.5-\epsilon$ for all $r \leq \delta \log^2 n$.*

This result has a different trade-off than Theorem 4.1 between integrality gap size and number of rounds: while the integrality gap is smaller than that shown in Theorem 4.1, the integrality gap holds for asymptotically more rounds.

Theorem 5.3 was published in [52]. Unlike the proof of Theorem 4.1 we use explicit protection matrices for our lower bound. Moreover, we crucially use a new technique which we call the "fence method". This method is used to "clean up" the vectors we are inductively protecting thus making it possible to push our lower bound to $\Omega(\log^2 n)$ rounds.

Note that the integrality gaps proved in both Theorems 4.1 and 5.3 are larger than the approximation ratios ruled out by the strongest PCP-based inapproximability results of Dinur and Safra [16] which only rule out 1.36 approximations. We should mention however, that $2-o(1)$ approximations were ruled out in [39] assuming Khot's Unique Games conjecture [36] (see Chapter 7 for more on the Unique Games Conjecture).

Finally, in Chapter 6 we prove an integrality gap for linear relaxations for INDEPENDENT SET where the only restriction is on the number of variables present in each constraint.

Theorem 6.7. *Fix $\epsilon, \gamma > 0$. Then there exists a constant $n_0 = n_0(\epsilon,\gamma)$ such that for every $n \geq n_0$ there exists a graph G with n vertices for which the integrality gap of any linear relaxation for* INDEPENDENT SET *in which each constraint uses at most $n^{\epsilon(1-\gamma)}$ variables is at least $n^{1-\epsilon}$.*

Note that the linear relaxations in Theorem 6.7 have may have exponential size and need not have any separation oracle. Note also that our result is tight in the sense that a linear relaxation with constraints of size n^ϵ can of course calculate a $n^{1-\epsilon}$ approximation. While this result is somewhat orthogonal to our lower bounds in the LS and LS_+ hierarchies, it is in the same spirit since it also rules out good approximations for a large subset of algorithms.

Chapter 3

The Expanding Constraints method

In this chapter we focus on ruling out slightly subexponential LS_+ algorithms for a trio of problems. Recall that relaxations obtained from r rounds of LS_+ lift-and-project are solvable in $n^{O(r)}$ time. Thus though $r = O(1)$ is the most interesting case, if we are also interested in slightly subexponential algorithms then any value of r less than $n/\log n$ is also interesting. We will show that $\Omega(n)$ rounds of LS_+ do not suffice to achieve the following approximations for any $\epsilon > 0$: (i) approximating MAX-3SAT within a factor better than $7/8 - \epsilon$, (ii) approximating VERTEX COVER in rank-k hypergraphs within a factor better than $k - 1 - \epsilon$, (iii) approximating SET COVER within a factor better than $(1 - \epsilon)\ln n$.

As already mentioned in Section 2.5, there are inapproximability results in the PCP setting where all the above factors appear [15, 30, 20, 47]. However, all these results use reductions that greatly blow up the instance size, and hence imply the above integrality gaps—under any complexity assumption at all—for only n^δ rounds (for some small constant $\delta > 0$) and not for $\Omega(n)$ rounds. Moreover, for SET COVER the PCP results are even weaker: an integrality gap of $(1 - \epsilon)\ln n$ for is implied only for $n^{o(1)}$ rounds [20].

The work in this Chapter is joint work with Mikhail Alekhnovich and Sanjeev Arora and appeared in [1].

Organization: In Section 3.0.1 we compare our results with previous work as well as other results in this thesis. In Section 3.1 we describe our methodology for the results in this chapter. In Sections 3.2–3.4 we prove our lower bounds. As mentioned already, section 3.5 discusses interesting issues and open problems arising from this chapter's results. Finally, for completeness we include in Section 3.6 proofs of some standard graph theory lemmas required for our lower bounds.

3.0.1 Comparison with related results.

As we will see in Chapter 4, the integrality gap for LS relaxations of VERTEX COVER on rank-k hypergraphs remains $k - \epsilon$ even after $\Omega(\log \log n)$ rounds of LS. In contrast, for the stronger LS_+ system we prove in this chapter integrality gaps of only $k - 1 - \epsilon$ for rank-k hypergraph VERTEX COVER; however, we show that these gaps hold even after $\Omega(n)$ rounds

The work most closely related to that in this chapter appeared in Buresh-Oppenheim et al. [11] where it was shown that for $k \geq 5$ a linear number of rounds of LS_+ is needed to obtain an integrality gap better than $(2^k - 1)/2^k - \epsilon$ for MAX-kSAT. The cases $k \leq 4$ were left open.

With a couple of exceptions, lower bounds for the LS and LS_+ hierarchies prior to this thesis (including Buresh-Oppenheim et al. [11]) use a simple "protection lemma" due to Lovász and Schrijver described below in Section 3.1 below. (The lone exceptions were proofs in [43] and [3] where the protection lemmas rely on LP duality as do our results in Chapter 4.) This lemma gives a sufficient condition for showing that a point x outside the integral hull survives one round of lifting. More generally, the protection lemma shows that such a point survives r rounds if some specific set T of points survives $r - 1$ rounds. In the Lovász-Schrijver protection lemma, T is a set of $2n$ points that differ from x in exactly one coordinate.

The simple protection lemma fails to prove the integrality gaps for the problems considered in this chapter, and we introduce new protection lemmas. One curious feature is that in order for this protection lemma to work for even one round, we need the underlying problem instance to have some *expansion* properties. In fact, expansion plays a key role in our lower bounds.

Note that expansion also played a big role in Buresh-Oppenheim et al. [11]. Their techniques allow integrality gaps (albeit loose ones) to be shown for VERTEX COVER on rank-k hypergraphs for big values of k. However, their techniques seem to break down for $k = 3$ and $k = 4$—the most interesting cases after $k = 2$, which is of course VERTEX COVER on graphs. For related reasons their techniques also fail when trying to prove optimal integrality gaps for MAX-3SAT and MAX-4SAT.

To prove the results in this chapter we introduce, in addition to the above-mentioned new protection lemma, a subtle *expansion correction* strategy. Both ideas may prove useful in future work.

3.1 Methodology

We will use Lemma 2.1 to prove our lower bounds. Hence, to prove that $y \in N_+^{r+1}(Q)$, we have to construct a specific protection matrix Y and prove that the $2n$ vectors defined in Lemma 2.1 are in $N_+^r(Q)$.

Given a vector $y \in \mathbb{R}^{n+1}$, $y_0 = 1$, the simplest Y one could conceive of is

$Y_{ij} = y_i y_j$ which is trivially positive semidefinite. However, this matrix satisfies $diag(Y) = y$ only if y is a 0-1 vector. The next simplest Y one could conceive is $Y = yy^T + \text{Diag}(y - y^2)$, that is, the matrix that has $Y_{ij} = y_i y_j$ except along the diagonal where $Y_{ii} = y_i$. For $y \in [0,1]^n$ this is clearly positive semidefinite, and indeed, this matrix was used in early results by Lovasz and Schrijver [43] and Goemans and Tunçel [26], and more recently, Buresh-Oppenheim et al. [11]. With this choice of Y, the vectors Ye_i/x_i and $Y(e_0 - e_i)/(1 - x_i)$ from Corollary 2.2 are obtained by changing *one* coordinate in y to a 0 or a 1. However, for MAX-3SAT and hypergraph VERTEX COVER, these vectors are not guaranteed to be in the polytope. Thus other than for the SET COVER problem, this simple protection lemma does not suffice for us.

Instead we use a more complicated Y, such that most entries satisfy $Y_{ij} = y_i y_j$, but some don't. Then the $2n$ vectors generated above correspond to modifying Y in a small number of entries. (A similar idea occurred in [3], except the Y there was not explicit.) This is at the heart of our new protection lemmas for MAX-3SAT and hypergraph VERTEX COVER. To make this choice of Y work out, we need the constraints defining our initial relaxations to satisfy certain expansion requirements.

With our "protection lemma" in hand, the lowerbound strategy will be as follows: Given our relaxed polytope P, we identify a point $w \in P$ for which the ratio between the integral optimum and the value of the objective function at w is large. We will then prove the lowerbound by showing that w survives many rounds of LS_+. We do this via a Prover-Adversary game where the Prover is trying to prove that $w \in N_+^r(P)$ and the Adversary's goal is to show the opposite. For the Adversary to win, it will suffice for him to exhibit a vector amongst the $2n$ vectors given by our "protection lemma" that is not in $N_+^{r-1}(P)$. He picks such a vector x and "challenges" the Prover to show it is in $N_+^{r-1}(P)$. Things continue this way, and the Prover loses if she cannot keep the game going for r steps. To keep the argument clean, we need to maintain the vector x in a nice form throughout the game. To this end, we borrow an idea from [11]: during each round, to prove that a particular point x is in a certain polytope, the Prover can also choose to express the point as a convex combination $\sum_j \rho_j z_j$ and claims that every $z_j \in N_+^{r-1}(P)$ (and consequently so is x). To counter this claim, the Adversary picks some z_j which he thinks is not in $N_+^{r-1}(P)$, and the game continues for that vector. We will show that if the constraints defining P satisfy certain expansion requirements, then for appropriate w, the Prover has a linear round strategy against any Adversary.

3.1.1 Incidence graphs of constraints and their properties

Given a hypergraph $G = (V, E)$, let H_G be the bipartite incidence graph on $E \times V$ where each each hyperedge is connected to the vertices it contains. We will require the notion of *expansion* in a bipartite graph.

Definition 3.1. A bipartite graph $G = (V_1, V_2, E)$ is an (r, c)-*expander* if every subset $S \subseteq V_1$, $|S| \leq r$, satisfies $|\Gamma(S)| \geq c|S|$, where $\Gamma(S)$ is the set of neighbours of S in V_2.

Throughout this chapter we will deal with constraints of the form $\sum_i v_i^{\epsilon_i} \geq 1$ where $v_i^{\epsilon_i}$ represents v_i if $\epsilon_i = 1$ and $v_i^{\epsilon_i}$ represents $1 - v_i$ if $\epsilon_i = 0$. Say that a variable $v_i^{\epsilon_i}$ occurs *negated* in a constraint if $\epsilon_i = 0$. Let C be a set of such constraints on a set V of n variables. Given an assignment vector $x \in [0, 1]^n$ for V, we define $C(x)$ to be the set of constraints obtained from C as follows: (a) If $x_i = 0$, remove all constraints containing v_i negated; (b) if $x_i = 1$, remove all constraints containing v_i unnegated; and (c) remove all variables set to 0-1 by x from the remaining constraints. Intuitively, $C(x)$ is the set of simplified constraints in C not trivially satisfied by x. In particular, if x satisfies $C(x)$, then x satisfies C.

Let $V(x)$ be the set of those variables in V not set to 0-1 by x and let $H(x)$ be the bipartite incidence graph on $C(x) \times V(x)$; that is, for each constraint in $C(x)$ there is an edge to every variable it contains. Let H be the incidence graph on $C \times V$. We will often abuse notation and say that $C(x)$ is an (r, c)-*expander* if $H(x)$ is an (r, c)-expander. We will say that the *arity* of a constraint is t if it has t neighbours in $H(x)$. For a subset $S \subseteq C(x)$ of constraints, denote the variables in S (i.e., the neighbours of S in $H(x)$) by $\Gamma(S)$.

Usually $C(x)$ will have some expansion property, and in particular will be at least a $(2, k - 1 - \epsilon)$-expander. Then all constraints in $C(x)$ will have arity at least $k - 1$. Moreover, whenever $C(x)$ is an expander, constraints of arity $k - 1$ will enjoy some special properties of which we will take advantage. For a vector $x \in \mathbb{R}^n$, let $R(x)$ denote the set of all indices to non-integral coordinates of x.

Definition 3.2. Let $0 < \epsilon < 1/2$ and $x \in \{0, \frac{1}{k-1}, 1\}^n$ and suppose $C(x)$ is a $(2, k - 1 - \epsilon)$-expander. Two indices $i, j \in R(x)$ are $C(x)$-*equivalent* (written $i \sim_{C(x)} j$) if there is a constraint in $C(x)$ of arity $k - 1$ containing v_i and v_j. Let $E(x) \subseteq R(x)$ contain all indices $i \in R(x)$ for which there exists $j \in R(x), j \neq i$ such that $i \sim_{C(x)} j$.

The following easy proposition will be used repeatedly in our lower bound proofs and follows easily from expansion.

Proposition 3.3. (Facts about $C(x)$-equivalences) *Let* $0 < \epsilon < 1/2$ *and* $x \in \{0, \frac{1}{k-1}, 1\}^n$ *and suppose* $C(x)$ *is* $(2, k - 1 - \epsilon)$*-expanding.*

Fact 1. A given variable can only occur in one arity $k - 1$ constraint in $C(x)$. Hence, each $C(x)$-equivalence class has exactly $k - 1$ elements.

Fact 2. Any given constraint in $C(x)$ (other than the arity $k-1$ constraint defining the equivalence) can contain at most one variable from any given $C(x)$-equivalence class.

3.2 Lowerbounds for hypergraph Vertex Cover

Let $G = (V, E)$, $E \subseteq V^k$, be a k-uniform hypergraph. The VERTEX COVER problem for G is expressed by the following integer program:

$$\min \sum_{i \in V} v_i$$
$$\sum_{j=1}^{k} v_j \geq 1, \quad \forall (1, \ldots, k) \in E.$$

The standard linear relaxation is obtained by relaxing to $0 \leq v_i \leq 1$. Let $\mathrm{HVC}(G)$ be the polytope consisting of all feasible points $w \in \mathbb{R}^n$ for the relaxed constraints. It is easy to see that for the complete k-uniform hypergraph on n vertices the optimal value of the integer program is $n - k + 1$ while the optimum value of the linear relaxation is n/k. Therefore, the integrality gap between the integer and linear programs is at least $k - o(1)$.

We prove that even after a linear number of rounds of LS_+ tightenings of $\mathrm{HVC}(G)$ there still exists some graph for which the integrality gap is $k - 1 - o(1)$:

Theorem 3.4. *Let $k \geq 3$. For all $\alpha > 0$ there exists a constant $\gamma > 0$ and a k-uniform hypergraph G such that the integrality gap of $N_+^r(\mathrm{HVC}(G))$ is at least $(k-1)(1-\alpha)$ for all γn.*

Given $G = (V, E)$, let C_G be the set of hyperedge constraints defining $\mathrm{HVC}(G)$. Since the underlying graph G will usually be clear, we omit the subscript unless extra precision is needed. In this section we will always have $x \in \{0, \frac{1}{k-1}, 1\}^n$ and $C(x)$ will be at least a $(2, k-1-\epsilon)$-expander. Then all constraints in $C(x)$ will have arity at least $k-1$ and the following will hold:

Proposition 3.5. *Let $0 < \epsilon < 1/2$, and $x \in \{0, \frac{1}{k-1}, 1\}^n$, and suppose that $C(x)$ is $(2, k-1-\epsilon)$-expanding. Then $x \in \mathrm{HVC}(G)$.*

We now define the vectors that will appear in our "Protection Lemma" for VERTEX COVER. For the remainder of this section we will always assume $0 < \epsilon < 1/2$.

Definition 3.6. Given $x \in [0,1]^n$, for all $i \in R(x)$ and all $a \in \{0,1\}$ define $x^{(i,a)}$ to be identical to x except that $x_i^{(i,a)} = a$.

Definition 3.7. Let $x \in \{0, \frac{1}{k-1}, 1\}^n$, and suppose $C(x)$ is $(2, k-1-\epsilon)$-expanding. For all $i \in E(x)$ define $x^{[i]}$ to be identical to x except that $x_i^{[i]} = 1$ and $x_j^{[i]} = 0$ for all $j \sim_{C(x)} i$. Let the set $T_x \subseteq \{0, \frac{1}{k-1}, 1\}^n$ equal the union $\{x^{[i]} : i \in E(x)\} \cup \{x^{(i,a)} : i \in R(x) \backslash E(x), a \in \{0,1\}\}$.

Lemma 3.8. *Let $x \in \{0, \frac{1}{k-1}, 1\}^n$, and suppose $C(x)$ is $(2, k-1-\epsilon)$-expanding. Then $R(x) \subseteq \mathrm{HVC}(G)$. Moreover, for all $y \in T_x$, each constraint in $C(y)$ has arity at least $k-1$.*

Proof. There are two types of points in T_x: (1) $x^{(i,a)}$ for $i \in R(x) \backslash E(x)$ and (2) $x^{[i]}$ for $i \in E(x)$. Consider a point $x^{(i,a)}$ in T_x where $i \in R(x) \backslash E(x)$. In this case, v_i does not belong to any arity $k-1$ constraint in $C(x)$. Hence, every constraint in $C(x^{(i,a)})$ has arity at least $k-1$ in $C(x^{(i,a)})$, and is therefore satisfied by $x^{(i,a)}$.

Now consider a point $x^{[i]}$ in T_x such that $i \in E(x)$. By Fact 2 on equivalences and the definition of $x^{[i]}$, every constraint in $C(x)$ that had arity k in $C(x)$ has arity at least $k-1$ in $C(x^{(i,a)})$, and hence is satisfied by $x^{(i,a)}$. By Fact 1 on equivalences, the only arity $k-1$ constraint in $C(x)$ for which the values of any of its variables changes under $x^{[i]}$ is the unique arity $k-1$ constraint containing v_i. But such a constraint is satisfied by $x^{[i]}$ since v_i is set to 1 in $x^{[i]}$. ☐

Lemma 3.9. (Protection Lemma for Hypergraph VC)
Suppose $C(x)$ is $(2, k-1-\epsilon)$-expanding where $x \in \{0, \frac{1}{k-1}, 1\}^n$. Suppose moreover that $T_x \subseteq N_+^m(\mathrm{HVC}(G))$. Then $x \in N_+^{m+1}(\mathrm{HVC}(G))$.

Proof. Let $y = \binom{1}{x}$. The proof uses Lemma 2.1 and the following choice of an $(n+1) \times (n+1)$ positive semidefinite symmetric matrix Y that is $yy^T + \mathrm{Diag}(y - y^2)$ except that $Y_{ij} = 0$ whenever $i \sim_{C(x)} j$. Note that Y is symmetric and that $Ye_0 = diag(Y) = y$. Moreover, by Proposition 3.10 below, Y is positive semidefinite. (This uses the expansion properties of $C(x)$.) So by Lemma 2.1, to show that $x \in N_+^{m+1}(\mathrm{HVC}(G))$ it remains only to show that for all $i \in R(x)$, Ye_i/x_i and $Y(e_0 - e_i)/(1 - x_i)$ are in $N_+^m(\mathrm{HVC}(G))$.

For $i \in R(x) \backslash E(x)$, $Ye_i/x_i = \binom{1}{x^{(i,1)}}$ and $Y(e_0 - e_i)/(1 - x_i) = \binom{1}{x^{(i,0)}}$ and hence are both in $T_x \subseteq N_+^m(\mathrm{HVC}(G))$. For $i \in E(x)$, $Ye_i/x_i = \binom{1}{x^{[i]}}$ which is in $T_x \subseteq N_+^m(\mathrm{HVC}(G))$. Finally, for $i \in E(x)$, $Y(e_0 - e_i)/(1 - x_i) = \binom{1}{z}$ where

$$z = \frac{1}{k-2} \sum_{j \sim_{C(x)} i,\ j \neq i} x^{[j]}.$$

In particular, $Y(e_0 - e_i)/(1 - x_i)$ is in the convex hull of $T_x \subseteq N_+^m(\mathrm{HVC}(G))$, and hence is also in $N_+^m(\mathrm{HVC}(G))$. ☐

Proposition 3.10. *The matrix Y defined in the proof of Lemma 3.9 is positive semidefinite.*

Proof. By Fact 1 on $C(x)$-equivalences, there exist disjoint sets $I_1, \ldots, I_t$ of indices such that (a) $|I_j| = k-1$ for all $j \in [t]$, (b) all indices belonging to an equivalence are in one of the I_j, and (c) for each $j \in [t]$ all indices in I_j are mutually equivalent. Then,

$$Y = yy^T + \mathrm{Diag}(y - y^2) + \sum_{j\in[t]} \left(\mathrm{Diag}(y_{I_j}^2) - y_{I_j} y_{I_j}^T\right),$$

where y_I equals y but is zero outside I.

To show $Y \succeq 0$, we show that $z^T Y z \geq 0$ for all $z \in \mathbb{R}^{n+1}$. Note that $z^T(yy^T)z = (y^T z)^2 \geq 0$ for all $z \in \mathbb{R}^{n+1}$. Moreover, $\mathrm{Diag}(w) \succeq 0$ for all vectors $w \geq \mathbf{0}$. Hence, since the sum of positive semidefinite matrices is positive semidefinite, to show that $Y \succeq 0$ it suffices to show for each I_j that the following quantity is non-negative:

$$\begin{aligned} &z^T(\mathrm{Diag}(y_{I_j} - y_{I_j}^2) + \mathrm{Diag}(y_{I_j}^2) - y_{I_j} y_{I_j}^T)z \\ &\qquad = z^T(\mathrm{Diag}(y_{I_j}) - y_{I_j} y_{I_j}^T)z. \end{aligned}$$

Since the argument is identical for all I_j we drop the subscript j and assume $I = [k-1]$. The above then simplifies to $\sum_{i\in[k-1]}(z_i^2 x_i) - (\sum_{i\in[k-1]} z_i x_i)^2$. Since $x_i = \frac{1}{k-1}$ for all indices in an equivalence, this further simplifies to

$$\frac{1}{k-1} \sum_{i\in[k-1]} z_i^2 - \frac{1}{(k-1)^2} \left(\sum_{i\in[k-1]} z_i\right)^2,$$

which is non-negative since $\sum_{i\in[\ell]} a_i^2 \geq \frac{1}{\ell}(\sum_{i\in[\ell]} a_i)^2$. $\square$

3.2.1 Proof of Theorem 3.4

Let $\alpha, \epsilon > 0$ be arbitrarily small. By Lemma 3.21 in Section 3.6, there are constants $\beta, \delta > 0$ such that a rank k hypergraph G exists with n vertices and βn edges such that the bipartite graph H_G is a $(\delta n, k-1-\epsilon)$-expander, and every vertex cover of G has size at least $(1-\alpha)n$. We show that the vector $w = (\frac{1}{k-1}, \ldots, \frac{1}{k-1})$, corresponding to a fractional vertex cover of "size" $n/(k-1)$, is in $N_+^r(\mathrm{HVC}(G))$ where $r = \frac{\epsilon\delta n}{k-1}$. It follows that this many rounds of LS_+ cannot reduce the integrality gap below $(k-1)(1-\alpha)$, and Theorem 3.4 then follows for $\gamma = \frac{\epsilon\delta}{k-1}$. Note that H_G is isomorphic to $H(w)$, and hence, $C(w)$ is $(\delta n, k-1-\epsilon)$-expanding. This will be crucial for the lower bound.

The lowerbound will follow from a Prover-Adversary game of the type discussed in Section 3.1. We describe the game more formally. In round i there is a parameter $\ell_i \geq 2$ and a current point $x \in \{0, \frac{1}{k-1}, 1\}^n$. For $i = 0$, x is some initial point $w' \in \{0, \frac{1}{k-1}, 1\}^n$. At the beginning of round i, $C(x)$ will be an $(\ell_i, k-1-2\epsilon)$-expander. In round i the following two moves are made.

1. **Adversary Move:** The Adversary selects z from T_x.

2. **Expansion Correction:** The Prover constructs a set $Y \subseteq \{0, \frac{1}{k-1}, 1\}^n$ such that (1) z is in the convex hull of Y, and (2) for all $y \in Y$, $C(y)$ is an $(\ell_{i+1}, k-1-2\epsilon)$-expander where $\ell_{i+1} \le \ell_i$. The Adversary selects one point $y \in Y$ to be the new x.

The game ends when $\ell_{i+1} \le 1$.

Intuitively, the Adversary fixes more and more fractional-valued coordinates in the initial point w' to 0-1 values by replacing the current point x with a point z from T_x (note that once a coordinate is set to 0-1 it remains fixed). The Prover wants this to continue for as long as possible but may run into trouble if $C(z)$ is no longer a good expander. The Prover therefore does Expansion Correction to obtain a new x for which $C(x)$ is a good expander. The next lemma shows that a good Prover strategy implies w' has high rank.

Lemma 3.11. *Suppose $w' \in \{0, \frac{1}{k-1}, 1\}^n$ is in* HVC(G). *If for w' the Prover has an m round strategy against any adversary, then $w' \in N_+^m(\mathrm{HVC}(G))$.*

Proof. By induction on m. Since $w' \in \mathrm{HVC}(G)$ by assumption, case $m = 0$ follows. So suppose the claim is true for m and that the Prover has an $m+1$ round strategy against any adversary. Consider the first round of the game and suppose the Adversary picks $z \in T_x$. Let Y be the set subsequently constructed by the Prover in the Expansion Correction move. Since the game runs for m more rounds regardless of which $y \in Y$ the Adversary chooses, $Y \subseteq N_+^m(\mathrm{HVC}(G))$ by induction, and $z \in N_+^m(\mathrm{HVC}(G))$ by convexity. This holds no matter which $z \in T_x$ the Adversary chooses, and so $T_x \subseteq N_+^m(\mathrm{HVC}(G))$. Lemma 3.9 then implies $w' \in N_+^{m+1}(\mathrm{HVC}(G))$. □

So to prove $w = (\frac{1}{k-1}, \ldots, \frac{1}{k-1}) \in N_+^r(\mathrm{HVC}(G))$ and complete the proof of Theorem 3.4, it suffices to describe an r round strategy for the Prover when the initial point is w.

Lemma 3.12. *If $C(w)$ is a $(\delta n, k-1-\epsilon)$-expander, then the Prover has an r round strategy against any Adversary, where $r = \frac{\epsilon \delta n}{k-1}$.*

Proof. We start the game with $x = w$. Proposition 3.5 implies $w \in \mathrm{HVC}(G)$. In round i of the strategy the parameter ℓ_i will be defined such that for the current point x the Prover can ensure $C(x)$ is an $(\ell_i, k-1-2\epsilon)$-expander. At the start, $\ell_1 = \delta n$.

The strategy will work as follows: The two moves made in each round of the game remove more and more variable vertices from the incidence graph $H(w)$ on $C(w) \times V(w)$. In each round at most $k-1$ variable vertices are removed from $H(w)$ by the Adversary choosing $z \in T_x$. As for the Expansion Correction move, the Prover will "correct" expansion in round i by identifying a maximal non-expanding set S_i of constraints of size at most ℓ_i and removing it and its neighbours from $H(x)$. Letting $\ell_{i+1} = \ell_i - |S_i|$, the resulting graph would then be an $(\ell_{i+1}, k-1-2\epsilon)$-expander. The Prover removes these constraints in S_i by having the assignments Y be 0-1 on $\Gamma(S_i)$ and equal to x outside $\Gamma(S_i)$. If $\ell_{i+1} \leq 1$, the game ends; otherwise, the game continues. The claim is that such a strategy results in at least r rounds: Suppose the strategy lasts m rounds and consider $S = \cup S_i$. Then

$$|S| = \sum_{i=1}^{m} |S_i| = \sum_{i=1}^{m} \ell_i - \ell_{i+1} = \delta n - \ell_{m+1}.$$

By expansion, S had at least $(k-1-\epsilon)|S|$ neighbours in $H(w)$. However, at the end of the game, S has no neighbours. Expansion Correction removes at most $(k-1-2\epsilon)|S|$ neighbours. Since the Adversary Move removes at most $k-1$ neighbours per round, there must be at least $\epsilon\delta n/(k-1)$ rounds.

It remains to describe the Prover's strategy in round i in detail: If $\ell_i \leq 1$ the game ends. Otherwise, Proposition 3.5 implies $x \in \mathrm{HVC}(G)$ and the Adversary selects $z \in T_x$. Note that Lemma 3.8 implies $z \in \mathrm{HVC}(G)$ and that every constraint in $C(z)$ has arity at least $k-1$. We will also require the following lemma:

Lemma 3.13. *Let $H = (V_1, V_2, E)$ be a bipartite graph and let $S \subseteq V_1$ be such that for for all $S' \subseteq S$, $|\Gamma(S')| > k|S'|$. Assume $S = \{e_1, e_2, ..., e_\ell\}$. Then there exists a mapping $\eta : S \to \mathcal{P}(\Gamma(S))$ such that (1) for all $i \in [\ell]$, $|\eta(e_i)| = k+1$, and (2) for all $i \in [\ell]$, $|\eta(e_i) \setminus \bigcup_{j<i} \eta(e_j)| \geq k$.*

Proof. By the generalization of Hall's theorem there exists a k-matching from S into $\Gamma(S)$. Fix such a k-matching ν once and for all. We construct η in the following recursive way. By assumption, $\Gamma(S)$ contains at least $\ell k + 1$ elements. So by the pigeon-hole principle there exists a vertex $v \in \Gamma(S)$ which does not belong to $\bigcup_{e \in S} \nu(e)$. Consider any vertex $e \in S$ that is adjacent to v (such a point exists because $v \in \Gamma(S)$) and let $\eta(e) = \{v\} \cup \nu(e)$. Finally, denote $S' = S \setminus \{e\}$ and repeat the process recursively for S'. The vertices in S' are ordered according to the way they were ordered in S.

Clearly for all vertices e_i in S, $\eta(e_i)$ is a $k+1$ element subset of $\Gamma(S)$. To check the second required property for η note that at each step of the inductive process, no vertex of $\nu(e)$ may be joined to any of the $\eta(e')$ from earlier steps, because $\eta(e')$ consists of $\nu(e')$ and v', $v' \notin \nu(e)$. The lemma follows. $\square$

We can now describe how the Prover constructs the set Y for Expansion Correction:

1. If $C(z)$ is an $(\ell_i, k-1-2\epsilon)$-expander, the Prover takes $Y = \{z\}$ and sets $S_i = \emptyset$.

2. Otherwise, let $S_i \subseteq C(z)$, $|S_i| \leq \ell_i$, be a maximal subset of constraints with expansion less than $k-1-2\epsilon$ in $C(z)$. If $|S_i| \geq \ell_i - 1$, i.e., $\ell_{i+1} \leq 1$, the game ends, and we let the final x be the same as z except it is 0 on $\Gamma(S_i)$.

3. Otherwise we claim that for all subsets $S' \subseteq S_i$ of constraints in $C(z)$, $|\Gamma(S')| > (k-2)|S'|$: Either the Adversary chose some $x^{(j,a)} \in T_x$ where j is not in any $C(x)$-equivalence class (in which case S' has expansion greater than $k-2$ in $C(z)$), or it chose $x^{[j]}$ where v_j occurs in some arity $k-1$ constraint $\phi \in C(x)$. Suppose ϕ shares t variables with $\Gamma(S')$. By expansion of $C(x)$,

$$\begin{aligned}|\Gamma(S')| &= |\Gamma(S' \cup \{\phi\})| - k + 1 + t \\ &\geq (k-1-2\epsilon)|S'| + t - 2\epsilon.\end{aligned}$$

   Since S' has exactly t fewer neighbours in $C(z)$ than in $C(x)$, the claim follows.

4. Let $S_i = (e_1, \ldots, e_t)$. By Lemma 3.13 there exists a mapping $\eta : S \to \mathcal{P}(\Gamma(S))$ such that (1) for all $i \in [t]$, $|\eta(e_i)| = k-1$, and (2) for all $i \in [t]$, $|\eta(e_i) \setminus \bigcup_{j<i} \eta(e_j)| \geq k-2$. We construct $k-1$ assignments $y^1, \ldots, y^{k-1}$ inductively according to the ordering $e_1, \ldots, e_t$. At the beginning all the y^j equal x outside $C(z)$ and are undefined on $\Gamma(S_i)$. Assume that at step t the values y_i^j for all $j \in [k-1]$ and for all i such that $v_i \in \bigcup_{i'<t} \eta(e_{i'})$ have been defined so that the constructed partial assignments satisfy all $e_{i'}$, $i' < t$, and the assigned values $y_i^1, \ldots, y_i^{k-1}$ contain exactly one 1 for each i. Consider e_t. Choose $k-2$ vertices $v_{i_1}, \ldots, v_{i_{k-2}} \in \eta(e_t)$ such that the values $y_{i_1}^j, \ldots, y_{i_{k-2}}^j$ are undefined for all $j \in [k-1]$ (these vertices exist by definition of η). Let $v_{i_{k-1}}$ be the other vertex in $\eta(e_{t+1})$. If the corresponding variables $y_{i_{k-1}}^1, \ldots, y_{i_{k-1}}^{k-1}$ are undefined then set the last of these variables to one and the rest to zeros. Assume without loss of generality that $y_{i_{k-1}}^{k-1} = 1$. For all other vertices in $\eta(e_{t+1})$ we set $y_{i_j}^j = 1$ and the rest to zeros. We have extended our partial assignments for $\eta(e_t)$ in a way that satisfies the induction hypothesis. At the the end, $y^1, \ldots, y^{k-1}$ each satisfy S_i and z is their average. Let $Y = \{y^1, \ldots, y^{k-1}\}$.

□

3.3 Lowerbounds for MAX-3SAT

The arguments used to prove Theorem 3.4 can be adapted to prove integrality gaps for MAX-3SAT. Given a 3-CNF formula ϕ, we convert its clauses to inequalities in the obvious way, i.e., $x_1 \vee x_2 \vee \neg x_3$ becomes $x_1 + x_2 + (1 - x_3) \geq 1$. Let C_ϕ be the set of such inequalities corresponding to ϕ. Note that the 0-1 solutions to these inequalities correspond exactly to the satisfying assignments for ϕ. Relaxing to $x_i \in [0,1]$ yields a polytope SAT(ϕ) whose integral points are solutions for ϕ.

Theorem 3.14. *For any constant $\alpha > 0$, there exist constants $\beta, \gamma > 0$ such that if ϕ is a random βn clause 3-CNF formula on n variables, then with high probability the integrality gap for $N_+^r(\mathrm{SAT}(\phi))$ is at least $\frac{8}{7} - \alpha$ for all $r \leq \gamma n$.*

Let $w = (\frac{1}{2}, \ldots, \frac{1}{2})$ and note that $w \in \mathrm{SAT}(\phi)$ for any formula ϕ. The proof of the above theorem will rely on the following lemma:

Lemma 3.15. *Let $0 < \epsilon < \frac{1}{2}$, and suppose that $C_\phi(w)$ is a $(\delta n, 2 - \epsilon)$-expander. Then $w \in N_+^{\epsilon\delta n/2}(\mathrm{SAT}(\phi))$.*

of Theorem 3.14. It is well-known that for all $\alpha, \epsilon > 0$, there exist constants $\beta, \delta > 0$ such that if we pick a random 3-CNF ϕ with βn clauses, then with high probability (1) no boolean assignment satisfies more than a $\frac{7}{8} + \alpha$ fraction of the clauses in ϕ and (2) C_ϕ is a $(\delta n, 2 - \epsilon)$-expander. On the other hand, Lemma 3.15 says that w, which satisfies *all* clauses in ϕ, is in $N_+^r(\mathrm{SAT}(\phi))$ where $r = \epsilon\delta n/2$. ☐

The proof of Lemma 3.15 is identical to that of Lemma 3.12 with the only changes being in (1) the "protection lemma" (Lemma 3.9) which must be altered to take into account the negated variables now appearing in the constraints; and (2) in the game, where the Prover's Expansion Correction strategy also has to accommodate negated variables. We finish this section therefore by stating and proving the new protection lemma used in the proof of Lemma 3.15 and by sketching a proof of the new Expansion Correction strategy used in the proof of Lemma 3.15.

Definition 3.16. Suppose $x \in \frac{1}{2}\mathbb{Z}^n$ and let $i \in R(x), a \in \{0, 1\}$. Let $x^{[i,a]} \in \frac{1}{2}\mathbb{Z}^n$ be identical to x except

1. $x_i^{[i,a]} = a$, and
2. if there exists an arity 2 constraint $v_i^{\epsilon_i} + v_j^{\epsilon_j} \geq 1$ in $C(x)$, then $x_j^{[i,a]} = 1 - a$ if $\epsilon_i = \epsilon_j$ and $x_j^{[i,a]} = a$ if $\epsilon_i \neq \epsilon_j$.

The key observation is that if $C(x)$ is $(2, 2-\epsilon)$-expanding, then for all $i \in R(x)$ and all $a \in \{0, 1\}$, each constraint in $C(x^{[i,a]})$ has arity at least 2 and hence $x^{[i,a]} \in \mathrm{SAT}(\phi)$. Let $T_x = \{x^{[i,a]} : i \in R(x), a \in \{0, 1\}\}$.

Lemma 3.17. (Protection Lemma for max-3sat)
Let $\epsilon > 0$ be arbitrarily small and suppose $C(x)$ is $(2, 2-\epsilon)$-expanding where $x \in \frac{1}{2}\mathbb{Z}^n$. Suppose moreover that $T_x \subseteq N_+^m(\mathrm{SAT}(\phi))$. Then $x \in N_+^{m+1}(\mathrm{SAT}(\phi))$.

Proof. Let $y = \binom{1}{x}$. The proof uses Lemma 2.1 and the following choice of an $(n+1) \times (n+1)$ positive semidefinite matrix Y that is $yy^T + \mathrm{Diag}(y - y^2)$ except that if $x_i^{\epsilon_i} + x_j^{\epsilon_j} \geq 1$ is a constraint in $C(x)$, then $Y_{ij} = 0$ if $\epsilon_i = \epsilon_j$ and $Y_{ij} = \frac{1}{2}$ if $\epsilon_i \neq \epsilon_j$. Note that Y is symmetric and that $Ye_0 = diag(Y) = y$. Moreover, by

Proposition 3.18 below it follows that Y is positive semidefinite. Finally, for all $i \in R(x)$, $Ye_i/x_i = \binom{1}{x^{[i,1]}}$ and $Y(e_0 - x_i)/(1 - x_i) = \binom{1}{x^{[i,0]}}$. In particular, these vectors are in $N_+^m(\mathrm{SAT}(\phi))$ since their projections along the hyperplane $x_0 = 1$ are in T_x. $\square$

Proposition 3.18. *The matrix Y defined in the proof of Lemma 3.17 is positive semidefinite.*

Proof. Let $I \subseteq \{0, 1, \ldots, n\}$ be the set of indices not in any $C(x)$-equivalence. For all $i, j \in \{1, \ldots, n\}$, define $(n+1) \times (n+1)$ matrices $A^{(i,j)}$ and $B^{(i,j)}$ that are 0 everywhere except $A^{(i,j)}_{ii} = A^{(i,j)}_{jj} = 1/4$, $A^{(i,j)}_{ij} = A^{(i,j)}_{ji} = -1/4$, and $B^{(i,j)}_{ii} = B^{(i,j)}_{jj} = B^{(i,j)}_{i,j} = B^{(i,j)}_{ji} = 1/4$. Note that $A^{(i,j)}$ and $B^{(i,j)}$ are both positive semidefinite. Finally, let

$$C_1 = \{(i,j) : v_i^{\epsilon_i} + v_j^{\epsilon_j} \geq 1 \in C(x) \text{ and } \epsilon_i = \epsilon_j\},$$
$$C_2 = \{(i,j) : v_i^{\epsilon_i} + v_j^{\epsilon_j} \geq 1 \in C(x) \text{ and } \epsilon_i \neq \epsilon_j\}.$$

Since $C(x)$-equivalence classes are disjoint, it follows that each $k \in \{0, 1, \ldots, n\}$ is either in I or appears in exactly one pair from $C_1 \cup C_2$. Hence, by definition of Y,

$$Y = yy^T + \mathrm{Diag}(y_I - y_I^2) + \sum_{(i,j) \in C_1} A^{(i,j)} + \sum_{(i,j) \in C_2} B^{(i,j)}.$$

(Here y_I is the vector equal to y on the coordinates indexed by I but zero everywhere else.) Each of the terms in the above sum is positive semidefinite and hence, so is Y. $\square$

We now sketch how the Expansion Correction Strategy is altered. The overall argument goes the same way using Lemma 3.13 with the only difference being that for $v_i \in \eta(e_t)$, the y_i^j, $j \in \{1, 2\}$, are set according to the signs the variables have in clause e_t so as to satisfy e_t.

3.4 Lowerbounds for Set Cover

An instance of SET COVER consists of a tuple (S, C) where C is a collection of n subsets of a finite set S of size m. The objective is to find a minimum size subset $C' \subseteq C$ such that each element of S is in some set in C'. If for each set $S_i \in C$ we have a variable x_i indicating whether or not set S_i is included in the set cover, then the SET COVER problem is expressed by the following integer program:

$$\min \sum_{i=1}^{n} x_i$$
$$\sum_{i:j \in S_i} x_i \geq 1, \quad \forall j \in [m].$$

The relaxed SET COVER polytope $\mathrm{MSC}(S, C)$ is the polytope defined by the above constraints but where we allow $0 \leq x_i \leq 1$. Note now that if $G = (V, E)$ is a k-uniform hypergraph, and we let $S = E$ and $C = \{S_v\}_{v \in V}$ where $S_v = \{e \in E : v \in e\}$, then $\mathrm{MSC}(S, C)$ is identical to $\mathrm{HVC}(G)$. Hence, integrality gaps for the hypergraph VERTEX COVER polytope yield integrality gaps for MSC.

Theorem 3.4 can therefore be used to obtain integrality gaps for LS_+ tightenings of the SET COVER polytope. However, stronger results can be obtained for SET COVER by using an argument specifically tailored for hypergraphs with edges of size $\Theta(\log n)$—this is what we show next.

Fix $\epsilon, \delta, \gamma > 0$ such that $\epsilon - \delta > 0$. By Lemma 3.22 in Section 3.6, there exists an $(\epsilon - \delta)n$-uniform hypergraph $G = (V, E)$ on n vertices with n edges such that the minimum vertex cover is at least $\log_{1+\epsilon} n$. Consider the hyperedge constraints C_G defining $\mathrm{HVC}(G)$. Let w be the all-$\frac{1+\gamma}{(\epsilon-\delta)n}$ point and note that w is in $\mathrm{HVC}(G)$. Moreover, at least $\lfloor \frac{\gamma(\epsilon-\delta)n}{1+\gamma} \rfloor$ coordinates of w can be changed to 0 or 1 with the resulting point still satisfying all the constraints C_G.

Let us recall the simple protection lemma proved by Lovász and Schrijver [43] and described in section 3.1: For a relaxed polytope P, a point x is in $N_+(P)$ if for all $i \in R(x)$ and all $a \in \{0, 1\}$, $x^{(i,a)}$ is in P. That is, x is in $N_+(P)$ if whenever we change exactly one coordinate of x to 0 or 1, the resulting point is in P. So by induction, this simple protection lemma together with the observation about w in the previous paragraph prove the following:

Lemma 3.19. *The point w is in $N^r(\mathrm{HVC}(G))$ where $r = \lfloor \frac{\gamma(\epsilon-\delta)n}{1+\gamma} \rfloor$.*

Finally note that since the minimum vertex cover for G has size $\log_{1+\epsilon} n$, the integrality gap for w is $\frac{(\epsilon-\delta)\ln n}{(1+\gamma)\ln(1+\epsilon)}$ which approaches $\ln n$ from below as $\epsilon, \delta, \gamma \to 0$. Thus we have proved the following gap for SET COVER:

Theorem 3.20. *For all $\epsilon > 0$, there exists $\delta > 0$ and an instance (S, C), $|S| = n$, of* SET COVER *for which the integrality gap of $N_+^r \,\mathrm{MSC}(S, C)$ is at least $(1 - \epsilon)\ln n$ for all $r \leq \delta n$.*

3.5 Discussion

It seems important to extend our inapproximability results to a variety of problems, or to prove that actually many important optimization problems do have good slightly subexponential time approximation algorithms via the LS_+ procedure or other lift-and-project procedures. As we noted above, reductions are problematic in this regard.

Methods based on games over expanders do not seem to help against the notoriously difficult VERTEX COVER problem: there are no expanders of degree 2. This question seems related to proving $k - \epsilon$ integrality gap for k-hypergraphs (a similar picture with these problems is observed in the PCP world). Moreover, the non-existence of appropriate expanders means we are also unable to prove gaps for other problems defined by two-variable constraints such as MAX-2SAT.

Our result for SET COVER is interesting in a different respect: In [20] integrality gaps of $(1-\epsilon)\ln n$ are only ruled out under the assumption $\mathbf{NP} \neq \mathbf{DTIME}(n^{\log\log n})$. Since we rule out $(1-\epsilon)\ln n$ integrality gaps for $\Omega(n)$ rounds of LS_+, this strengthens the possibility that stronger PCP results are possible for this problem. In particular, it further supports the conjecture that it should be possible to rule out $(1-\epsilon)\ln n$ integrality gaps under the weaker assumption of $\mathbf{NP} \neq \mathbf{BPP}$ or even $\mathbf{NP} \neq \mathbf{P}$.

3.6 Graph theory lemmas

For completeness, we include the proofs of the following two lemmas which use standard arguments from the theory of random graphs.

Lemma 3.21. *Let $\Delta(\epsilon, k, \beta) = \left(\frac{e^{\epsilon-k}}{5\beta(k-1-\epsilon)^{1+\epsilon}}\right)^{1/\epsilon}$. Then for all α, $0 < \alpha < 1$, and all $\epsilon > 0$, there exists $\mu(\alpha)$ such that for all $\beta \geq \mu(\alpha)\alpha^{-k}$ and all δ, $0 < \delta < \Delta(\epsilon, k, \beta)$, the probability that a random k-uniform hypergraph $G = (V, E)$ on n vertices with βn hyperedges (1) has no vertex cover of size smaller than $(1-\alpha)n$ and (2) H_G is a $(\delta n, k-1-\epsilon)$ expander is at least $1/2$.*

Proof. Let $\beta = \mu(\alpha)\alpha^{-k}$ and suppose the hypergraph has βn randomly and uniformly chosen hyperedges where $\mu(\alpha)$ is chosen below. The probability that there exists a vertex cover of size $(1-\alpha)n$ equals the probability that there exists a set $S \subseteq V$, $|S| = \alpha n$, such that no edge contains only elements from S. This probability is bounded by

$$\begin{aligned}\binom{n}{\alpha n}(1-\alpha^k)^{\beta n} &\leq \left(\frac{e}{\alpha}\right)^{\alpha n}(1-\alpha^k)^{\beta n} \\ &= \left(\frac{e}{\alpha}\right)^{\alpha n}\left(\frac{1}{e}\right)^{\mu(\alpha)n}.\end{aligned}$$

Let $\mu(\alpha) > 0$ be such that the above is less than $1/4$.

Now consider the bipartite graph H_G mapping E to V. Note that $|E| = \beta n$. The probability that a subset of $s = \delta n$ constraints of F does not have expansion more than $c = k-1-\epsilon$ is

$$\binom{\beta n}{s}\binom{n}{cs}\left(\frac{cs}{n}\right)^{ks} \leq \left(\frac{e\beta n}{s}\right)^{s}\left(\frac{en}{cs}\right)^{cs}\left(\frac{cs}{n}\right)^{ks}$$
$$= \left[\delta^{\epsilon}\beta e^{k-\epsilon}c^{1+\epsilon}\right]^{s}.$$

Let $r = \delta^{\epsilon}\beta e^{k-\epsilon}c^{1+\epsilon}$. Then $r < 1/5$ when $\delta < \left(\frac{e^{\epsilon-k}}{5\beta c^{1+\epsilon}}\right)^{1/\epsilon}$. Hence, the probability that some subset of E of size at most δn fails to have expansion greater than $k-1-\epsilon$ is bounded by

$$\sum_{s=1}^{\delta n} r^s \leq \sum_{s\geq 1} r^s = \frac{r}{1-r} < \frac{1}{4}.$$

So with probability at least $1/2$, both G has no vertex cover of size less than $(1-\alpha)n$ and H_G is a $(\delta n, k-1-\epsilon)$ expander. □

Lemma 3.22. *For any constant $\epsilon, \delta \in (0,1)$ for all n there exists an $(\epsilon-\delta)n$-regular hypergraph with n vertices and n edges that has vertex cover greater than $\log_{(1+\epsilon)} n$.*

Proof. Let $\epsilon' = \epsilon - \delta/2$. Consider a random hypergraph G with n edges over n vertices in which every vertex belongs to an edge independently with probability ϵ'. Let $k = \log_{1+\epsilon} n$. The probability that G contains a vertex cover of size k is less than or equal

$$\binom{n}{k}\cdot\left[1-(1-\epsilon')^k\right]^m \leq n^k e^{-m\cdot(1-\epsilon')^k} = o(1).$$

Finally, with high probability every edge in G contains at least $(\epsilon'-\delta/2)n = (\epsilon-\delta)n$ elements. By removing vertices from each edge we can assume each edge contains exactly $(\epsilon-\delta)n$ elements. □

Chapter 4

Lower bounds using LP duality

In this chapter we prove tight integrality gaps for both graph and hypergraph VERTEX COVER in the LS hierarchy. In particular, we show that the integrality gap for graph VERTEX COVER remains $2 - o(1)$ even after $\Omega(\log n)$ rounds of LS lift-and-project (Theorem 4.1), and we show that the integrality gap for rank-k hypergraph VERTEX COVER remains $k - o(1)$ even after $\Omega(\log \log n)$ rounds of LS lift-and-project (Theorem 4.13). The latter result contrasts with our result for hypergraph VERTEX COVER in Chapter 3 showing that the integrality gap remains $k - 1 - \epsilon$ after even $\Omega(n)$ rounds of the stronger LS_+ procedure.

Our results in this Chapter are tight in the sense that even the trivial linear relaxations achieve $2 - o(1)$ and $k - o(1)$ approximations for graph and rank-k hypergraph VERTEX COVER, respectively.

In Chapter 3 we proved integrality gaps of size $k - 1 - \epsilon$ for rank-k hypergraph VERTEX COVER by taking advantage of certain expansion properties enjoyed by our input graphs. The construction of our protection matrices in our lower bound proofs crucially relied on the existence of k-regular graphs with expansion $k - 1 - \epsilon$ (for $k \geq 3$). Unfortunately, to prove integrality gaps of size $k - \epsilon$ using this technique (and hence also prove integrality gaps for graph VERTEX COVER) would require the existence of k-regular graphs with expansion greater than $k - 1$, an impossibility.

Instead we will construct our protection matrices in this chapter using LP-duality, in particular, Farkas's lemma (Lemma 1.4). We discuss our approach in detail in Section 4.1.1. LP-duality was first employed to prove LS hierarchy lower bounds in Lovász and Schrijver's original paper introducing the LS systems [43] and also subsequently used by Arora et al. [3]. The latter paper showed that a $2 - o(1)$ integrality gap for graph VERTEX COVER remains after even $\Omega(\sqrt{\log n})$ LS rounds. Our results in this section directly build upon their work.

The work in Section 4.1 is joint work with Sanjeev Arora, Béla Bollobás and László Lovász and appeared in [4]. The work in Section 4.2 was published in [51].

4.1 Lower bounds for graph Vertex Cover

Let $\mathrm{VC}(G)$ be the closed convex cone in $\mathbb{R}^{n+1}$ that contains a vector $(x_0, x_1, \ldots, x_n)$ iff it satisfies $0 \le x_i \le x_0$ for all i as well as the edge constraints $x_i + x_j \ge x_0$ for each edge $\{i, j\} \in G$.

Our main theorem for this section is the following:

Theorem 4.1. *For all $\epsilon > 0$ there exists an integer n_0 and a constant $\delta(\epsilon) > 0$ such that for all $n \ge n_0$ there exists an n vertex graph G for which the integrality gap of $N^r(\mathrm{VC}(G))$ for any $r \le \delta(\epsilon) \log n$ is at least $2 - \epsilon$.*

The proof of Theorem 4.1 relies on the following two theorems. The first is essentially due to Erdős [17]; see Bollobás [9], Theorem 4, Ch VII. The second, Theorem 4.3, will be proved in Section 4.1.2 with an overview of the argument first given in Section 4.1.1.

Theorem 4.2. *For any $\alpha > 0$ there is an $n_0(\alpha)$ such that for every $n \ge n_0(\alpha)$ there are graphs on n vertices with girth at least $\log n/(3 \log(1/\alpha))$ but no independent set of size greater than αn.*

Let y_γ denote the vector $(1, \frac{1}{2} + \gamma, \frac{1}{2} + \gamma, \ldots, \frac{1}{2} + \gamma)$ where $0 < \gamma < \frac{1}{2}$.

Theorem 4.3. *Let $G = (V, E)$ have $\mathrm{girth}(G) \ge 16r/\gamma$. Then $y_\gamma \in N^r(\mathrm{VC}(G))$.*

Proof of Theorem 4.1. Let $\gamma = \epsilon/8$ and $\alpha = \epsilon/4$, and let n_0 be the constant from Theorem 4.2 for this α. For $n \ge n_0$, let G be the n-vertex graph given by Theorem 4.2. Finally, let $\delta(\epsilon) = \frac{\epsilon}{384 \log(4/\epsilon)}$. Then by Theorem 4.3, y_γ is in $N^r(\mathrm{VC}(G))$ for all $r \le \delta(\epsilon) \log n$, and hence, the integrality gaps for all these polytopes is at least $2(1 - \alpha)/(1 + 2\gamma) \ge 2 - \epsilon$. □

4.1.1 Intuition for Theorem 4.3

Lemma 2.1 (and Corollary 2.2) suggest using induction to prove Theorem 4.3. To do that, we will first identify for each j some large set of vectors within each polytope $N^j(\mathrm{VC}(G))$ called the "palette" for $N^j(\mathrm{VC}(G))$. In stage j of the induction we will then show the following: For each vector x in the palette for $N^j(\mathrm{VC}(G))$, there exists a protection matrix Y such that for all $i \in [n]$ the vectors Ye_i and $Y(e_0 - e_i)$ all lie in the palette for the previous polytope $N^{j-1}(\mathrm{VC}(G))$ (Figure 4.1). The condition that such a protection matrix exists can be expressed as an LP. So, to show that a protection matrix exists for each x in the palette for $N^j(\mathrm{VC}(G))$ we show using Farkas's lemma that the corresponding LP is feasible. The theorem will then follow since our definition for the palette for $N^r(\mathrm{VC}(G))$ will ensure that it contains y_γ.

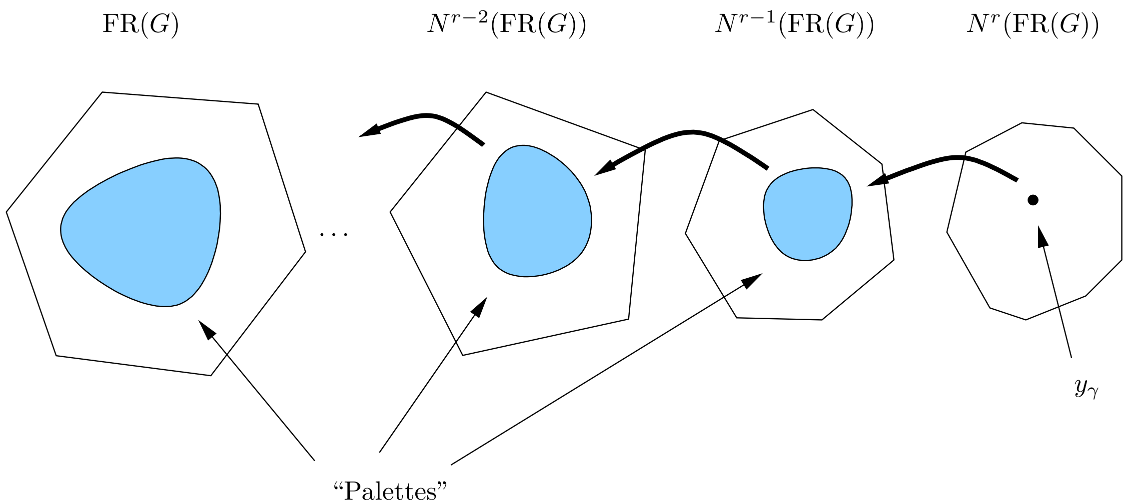

Figure 4.1: Chain of dependencies for the induction in the proof of Theorem 4.3: Each palette is contained in its respective polytope because some other palette is contained in the previous polytope.

Since our protection matrices will be found using LP duality, we will pick the simplest palettes possible in order to ensure that our LPs are also as simple as possible (and hence easy to analyze). To understand what desirable properties the palette vectors should have, let us look at the simpler problem of showing that $y_\gamma \in N(\mathrm{VC}(G))$ (rather than showing $y_\gamma \in N^r(\mathrm{VC}(G))$) and make some observations about the constraints the conditions in Corollary 2.2 force upon a protection matrix for y_γ.

To that end, consider the projected "columns" $Ye_i/y_\gamma^{(i)}$ and $Y(e_0 - e_i)/(1 - y_\gamma^{(i)})$ of Y (from condition 2 of Corollary 2.2). These vectors must satisfy the edge constraints. As will be shown in Section 4.1.4 (see equation (4.3)), the constraints forcing this are given by the following constraint:

$$\alpha_i \le Y_{ij} + Y_{ik} \le \alpha_i + (\alpha_j + \alpha_k - 1) \qquad \forall i \in \{1, \ldots, n\}, \forall \{j,k\} \in E. \tag{4.1}$$

Fix i. If j_1 is adjacent to i, then (4.1) implies $\frac{1}{2} + \gamma \le Y_{ii} + Y_{ij_1} \le \frac{1}{2} + 3\gamma$. Since Y is a protection matrix for y_γ, it must satisfy $Y_{ii} = y_\gamma^{(i)} = \frac{1}{2} + \gamma$. Hence, $0 \le Y_{ij_1} \le 2\gamma$. Now consider a node j_2 at distance 2 from j_1. Then (4.1), together with the fact that $0 \le Y_{ij_1} \le 2\gamma$ for all j_1 adjacent to i, imply that $\frac{1}{2} - \gamma \le Y_{ik} \le \frac{1}{2} + 3\gamma$. In turn, for a node j_3 at distance 3 from i we must have $0 \le Y_{ij_3} \le 4\gamma$; and for a node j_4 at distance 4 from i we have $\frac{1}{2} - 3\gamma \le Y_{ij_4} \le \frac{1}{2} + 3\gamma$. So as j gets further and further from i, the constraints on Y_{ij} implied by (4.1) get looser and looser so that for nodes j sufficiently far from i (distance $2/\gamma$ more than suffices) no constraint on Y_{ij} is implied. So intuitively, for such j we should be able to choose Y_{ij} such that node j remains $\frac{1}{2} + \gamma$ in both $Ye_i/y_\gamma^{(i)}$ and $Y(e_0 - e_i)/(1 - y_\gamma^{(i)})$. Note that the fact that the coordinates of y_γ are $\frac{1}{2}+\gamma$ instead of $\frac{1}{2}$ is crucial in ensuring that the effects of the edge constraints die out as we get further away from node i. Note also that we have implicitly assumed that our graph has girth larger than $2/\gamma$ so that two nodes cannot be connected by two paths of different lengths both less than $2/\gamma$—intuitively this is why Theorem 4.3 requires large girth. We should also mention that we have simplified things by ignoring constraints required by Corollary 2.2 forcing the projected "columns" to lie in $[0,1]^{n+1}$: these tighten the above constraints on the Y_{ij} a bit but the intuition given above is mostly unchanged.

In any case, the above suggests that to prove $y_\gamma \in N(\mathrm{VC}(G))$ we could use a palette consisting of those vectors in $\mathrm{VC}(G)$ which are $\frac{1}{2}+\gamma$ everywhere except perhaps on some ball of radius $2/\gamma$ in G. As such, we can add "palette constraints" to the LP defining Y forcing all nodes j distant from i to be $\frac{1}{2}+\gamma$ in both $Ye_i/y_\gamma^{(i)}$ and $Y(e_0-e_i)/(1-y_\gamma^{(i)})$. In fact, since Y must also be symmetric, the actual constraints we will add will force the following: for all pairs of nodes i,j with distance at least $2/\gamma$ between them, the jth nodes in $Ye_i/y_\gamma^{(i)}$ and $Y(e_0-e_i)/(1-y_\gamma^{(i)})$, and the ith nodes $Ye_j/y_\gamma^{(j)}$ and $Y(e_0-e_j)/(1-y_\gamma^{(j)})$ must all be $\frac{1}{2}+\gamma$.

The proof of Theorem 4.3 will use generalized versions of the above palette: The palettes for each polytope $N^j(\mathrm{VC}(G))$ will consist of vectors from $\mathrm{VC}(G)$ that are $\frac{1}{2}+\gamma$ except in a few neighbourhoods (see Definition 4.4 in Section 4.1.2 for the precise statement). For a vector x in the palette for $N^j(\mathrm{VC}(G))$ the LP used to find a protection matrix Y for x will have two types of constraints: constraints that force Y to satisfy the conditions in Corollary 2.2 and constraints that force the "columns" Ye_i/x_i and $Y(e_0-e_i)/(1-x_i)$ to belong to the "palette" for $N^{j-1}(\mathrm{VC}(G))$.

The palettes we will use will have the following property: The diameter of the largest neighbourhood H in G such that H consists entirely of nodes with values not equal to $\frac{1}{2}+\gamma$ will grow linearly with the number of rounds. Hence, our method is limited to proving integrality gaps for at most $O(\log n)$ rounds since only graphs with girth $O(\log n)$ yield large integrality gaps.

We note that in [3], which the results in this section extend, the palettes were picked such that the diameter of the largest neighbourhood grew *quadratically* in the number of rounds, thereby yielding integrality gaps only for $O(\sqrt{\log n})$ rounds. To push the lower bound to $\Omega(\log n)$ rounds we select our palettes in a more subtle way than those in [3] and rely on a crucial structural result enjoyed by these new palettes (Lemma 4.5 below).

4.1.2 Proof of Theorem 4.3

As mentioned in the previous section, the theorem will be proved by induction where the inductive hypothesis will require a set of vectors other than just y_γ to be in $N^m(\mathrm{VC}(G))$ for $m \le r$ (the "palettes" from Section 4.1.1). These vectors will be essentially all-$(\frac{1}{2}+\gamma)$, except possibly for a few small neighborhoods where the vector can take arbitrary values in $[0,1]$ so long as the edge constraints are satisfied. Let $Ball(w, R)$ denote the set of vertices within distance R of w in G.

Definition 4.4. Let $S \subseteq \{1, \ldots, n\}$, R be a positive integer and $\gamma > 0$. Then a nonnegative vector $(\alpha_0, \alpha_1, \ldots, \alpha_n) \in [0,1]^{n+1}$ with $\alpha_0 = 1$ is an (S, R, γ)-*vector* if the entries satisfy the edge constraints and if for each $w \in S$ there exists a positive integer R_w such that

1. $\sum_{w \in S}(R_w + \frac{2}{\gamma}) \leq R$
2. For distinct $w, w' \in S$, $Ball(w, R_w) \cap Ball(w', R_{w'}) = \emptyset$
3. $\alpha_j = \frac{1}{2} + \gamma$ for each $j \notin \cup_{w \in S} Ball(w, R_w)$

We will say that the integers $\{R_w\}_{w \in S}$ *witness* that α is an (S, R, γ)-vector.

Let $R^{(r)} = 0$ and let $R^{(m)} = R^{(m+1)} + \frac{4}{\gamma}$ for $0 \leq m < r$. Note that $4R^{(m)} \leq$ girth(G) for $0 \leq m \leq r$. To prove Theorem 4.3 we will prove the inductive claim below. Since the set of $(\emptyset, R^{(r)}, \gamma)$-vectors consists precisely of the vector y_γ, the theorem will then follow as a subcase of the case $m = r$.

Inductive Claim for $N^m(\mathrm{VC}(G))$: For every set S of at most $r - m$ vertices, every $(S, R^{(m)}, \gamma)$-vector is in $N^m(\mathrm{VC}(G))$.

Base case $m = 0$. Trivial since $(S, R^{(0)}, \gamma)$-vectors satisfy the edge constraints for G.

Proof for $m+1$ assuming truth for m. Let α be an $(S, R^{(m+1)}, \gamma)$-vector where $|S| \leq r - m - 1$. To show that $\alpha \in N^m(\mathrm{VC}(G))$ it suffices to find a protection matrix Y for α satisfying the properties of Corollary 2.2. We exploit the structure of (S, R, γ)-vectors and prove some important structural properties of these vectors in Lemma 4.5, which then enables us to argue that such a protection matrix exists thereby completing the induction step.

Note first that the $(S, R^{(m+1)}, \gamma)$-vector α is also trivially an $(S \cup i, R^{(m)}, \gamma)$-vector for any $i \in G$. Lemma 4.5, which we now state but prove in Section 4.1.3 below, says that for appropriate sets S', $|S'| \leq r - m$, α is also an $(S', R^{(m)}, \gamma)$-vector enjoying crucial additional structural properties.

Lemma 4.5. *Let i be such that $\alpha_i \notin \{0, 1\}$. Then there exists a set $S_i \subseteq \{1, \ldots n\}$, $|S_i| \leq r - m$, and positive integers $\{R_w^{(m)}\}_{w \in S_i}$ such that,*

1. *α is an $(S', R^{(m)}, \gamma)$-vector with witnesses $\{R_w^{(m)}\}_{w \in S_i}$*
2. *$i \in \cup_{w \in S_i} Ball(w, R_w^{(m)})$*
3. *For each $\ell \notin \cup_{w \in S_i} Ball(w, R_w^{(m)})$, any path between i and ℓ in G contains at least $\frac{2}{\gamma}$ consecutive vertices ℓ such that $\alpha_\ell = \frac{1}{2} + \gamma$*

By the induction hypothesis, for any $S' \subseteq \{1,\dots,n\}$ such that $|S'| \le r-m$, every $(S', R^{(m)}, \gamma)$-vector is in $N^m(\mathrm{VC}(G))$. Hence, to show that $\alpha \in N^{m+1}(\mathrm{VC}(G))$ it suffices by Corollary 2.2 to exhibit an $(n+1)\times(n+1)$ symmetric protection matrix Y that satisfies:

A. $Ye_0 = diag(Y) = \alpha$,

B. For each i such that $\alpha_i = 0$, we have $Ye_i = 0$; for each i such that $\alpha_1 = 1$, we have $Ye_0 = Ye_i$; otherwise, Ye_i/α_i and $Y(e_0 - e_i)/(1-\alpha_i)$ are $(S_i, R^{(m)}, \gamma)$-vectors, where S_i as well as the integers $\{R_w^{(m)}\}_{w\in S_i}$ witnessing that these vectors are $(S_i, R^{(m)}, \gamma)$-vectors are given by Lemma 4.5 for i.

We will complete the proof of the induction step (and hence of Theorem 4.3) by showing in Section 4.1.4 below that a matrix Y exists satisfying conditions **A** and **B**.

4.1.3 Proof of Lemma 4.5

Let $\{R_w^{(m+1)}\}_{w\in S}$ witness that α is an $(S, R^{(m+1)}, \gamma)$-vector and let

$$C = \cup_{w\in S} Ball(w, R_w^{(m+1)}).$$

There are two cases depending on whether $Ball(i, \frac{2}{\gamma})$ intersects C or not.

In the first (easy) case, $Ball(i, \frac{2}{\gamma})$ does not intersect C. Then let $S_i = S \cup \{i\}$, let $R_i^{(m)} = \frac{2}{\gamma}$, and let $R_w^{(m)} = R_w^{(m+1)}$ for $w \in S$. It is easy to see that the conditions of the lemma are satisfied by these choices.

So consider the second case where $Ball(i, \frac{2}{\gamma})$ does intersects C. Let

$$T_1 = \left\{ w \in S : i \in Ball\left(w, R_w^{(m+1)} + \frac{2}{\gamma}\right)\right\}.$$

That is, T_1 consists of all points in S whose balls, slightly enlarged, contain i. Note that it may be that $i \in S$, in which case $i \in T_1$.

Now let

$$D = \bigcup_{w\in T_1} Ball\left(w, R_w^{(m+1)} + \frac{2}{\gamma}\right).$$

Since $\sum_{w\in S}(R_w^{(m+1)} + \frac{2}{\gamma}) \le R^{(m+1)} < \frac{1}{2}\,\mathrm{girth}(G) - \frac{2}{\gamma}$, it follows that D is a tree. Let q be a longest path in D and let w_1 be a node in the middle of this path. Then certainly,

$$D \subseteq Ball\left(w_1, \sum_{w\in T_1}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right)\right).$$

We will now increase the size of this "big ball" around w_1 (perhaps also moving its centre in the process) until there are no points $w \in S$ outside the "big ball" for which $Ball(w, R_w^{(m+1)} + \frac{2}{\gamma})$ intersects the "big ball". We do this as follows:

Suppose $Ball(w_1, \sum_{w\in T_1}(R_w^{(m+1)} + \frac{2}{\gamma}))$ intersects $Ball(w', R_{w'}^{(m+1)} + \frac{2}{\gamma})$ for some $w' \in S\backslash T_1$. Add w' to T_1 and call the new set T_2. Reasoning as before, there exists $w_2 \in G$ such that,

$$\bigcup_{w\in T_2} Ball\left(w, R_w^{(m+1)} + \frac{2}{\gamma}\right) \subseteq Ball\left(w_2, \sum_{w\in T_2}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right)\right).$$

In general, at stage j if $Ball(w_j, \sum_{w\in T_j}(R_w^{(m+1)} + \frac{2}{\gamma}))$ intersects $Ball(w', R_{w'}^{(m+1)} + \frac{2}{\gamma})$ for some $w' \in S\backslash T_j$, then add w' to T_j, call the new set T_{j+1}, and find a new $w_{j+1} \in G$ (using again the same arguments as before) such that,

$$\bigcup_{w\in T_{j+1}} Ball\left(w, R_w^{(m+1)} + \frac{2}{\gamma}\right) \subseteq Ball\left(w_{j+1}, \sum_{w\in T_{j+1}}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right)\right).$$

Continue in this way until the first stage k for which there exists no point w' in $S\backslash T_k$ for which $Ball(w', R_{w'}^{(m+1)} + \frac{2}{\gamma})$ intersects $Ball(w_k, \sum_{w\in T_k}(R_w^{(m+1)} + \frac{2}{\gamma}))$. Let $T = T_k$ and $u = w_k$.

We can now define S_i and $\{R_w^{(m)}\}_{w\in S_i}$: Let $S_i = (S\backslash T) \cup \{u\}$. For $w \in S\backslash T$, let $R_w^{(m)} = R_w^{(m+1)}$; let

$$R_u^{(m)} = \frac{2}{\gamma} + \sum_{w\in T}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right).$$

To complete the proof of the lemma we need to show that α is an $(S_i, R^{(m)}, \gamma)$-vector witnessed by these $\{R_w^{(m)}\}$ and that the remaining two conditions in the statement of the lemma are satisfied.

Note first that

$$\begin{aligned}\sum_{w\in S_i}\left(R_w^{(m)} + \frac{2}{\gamma}\right) &= \sum_{w\in S\backslash T}\left(R_w^{(m)} + \frac{2}{\gamma}\right) + \left(R_u^{(m)} + \frac{2}{\gamma}\right)\\ &= \sum_{w\in S\backslash T}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right) + \left(\sum_{w\in T}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right) + \frac{4}{\gamma}\right)\\ &= \sum_{w\in S}\left(R_w^{(m+1)} + \frac{2}{\gamma}\right) + \frac{4}{\gamma} \le R^{(m+1)} + \frac{4}{\gamma} = R^{(m)}.\end{aligned}$$

The inequality above follows from the fact that α is an $(S, R^{(m+1)}, \gamma)$-vector witnessed by the integers $\{R_w^{(m+1)}\}$. Therefore, α satisfies condition (1) of being an $(S_i, R^{(m)}, \gamma)$-vector witnessed by the integers $\{R_w^{(m)}\}$.

By construction, $Ball(u, R_u^{(m)})$ does not intersect $\cup_{w \in S \setminus T} Ball(w, R_w^{(m)})$. Moreover, since α is an $(S, R^{(m+1)}, \gamma)$-vector witnessed by the integers $R_w^{(m+1)}$, it follows that

$$Ball(w, R_w^{(m)}) \cap Ball(w', R_{w'}^{(m)}) = \emptyset$$

for distinct $w, w' \in S \setminus T$. Also, by construction we have that $\alpha_j = \frac{1}{2} + \gamma$ for all $j \notin \cup_{w \in S_i} Ball(w, R_w^{(m)})$. Hence α satisfies conditions (2) and (3) of being an $(S_i, R^{(m)}, \gamma)$-vector witnessed by the integers $\{R_w^{(m)}\}$.

Next note that by construction, we have on one hand that

$$\bigcup_{w \in T} Ball\left(w, R_w^{(m+1)} + \frac{2}{\gamma}\right) \subseteq Ball\left(u, R_u^{(m)} - \frac{2}{\gamma}\right).$$

On the other hand, $Ball(u, R_u^{(m)})$ does not intersect $\cup_{w \in S \setminus T} Ball(w, R_w^{(m+1)})$. Since α is an $(S, R^{(m+1)}, \gamma)$-vector witnessed by the integers $R_w^{(m+1)}$, it thus follows from the definition of such vectors that for all vertices k in $Ball(u, R_u^{(m)}) \setminus Ball(u, R_u^{(m)} - \frac{2}{\gamma})$, we have $\alpha_k = \frac{1}{2} + \gamma$. Hence condition (3) of the lemma holds.

Finally, condition (2) holds since by construction $i \in \bigcup_{w \in T} Ball(w, R_w^{(m+1)} + \frac{2}{\gamma})$. The lemma follows.

4.1.4 Existence of Y

We will show that Y exists by representing conditions **A** and **B** as a linear program and then showing that the program is feasible. This approach was first used in [43] and subsequently in [3].

Our notation will assume symmetry, namely, Y_{ij} will represent $Y_{\{i,j\}}$. Condition **A** requires that:

$$Y_{kk} = \alpha_k, \quad \forall k \in \{1, \dots, n\}. \tag{4.2}$$

Condition **B** requires that Ye_i/α_i and $Y(e_0 - e_i)/(1 - \alpha_i)$ are $(S_i, R^{(m)}, \gamma)$-vectors. In particular, we need constraints on the variables Y_{ij} forcing these vectors to satisfy both the edge constraints as well as the extra structural properties enjoyed by $(S_i, R^{(m)}, \gamma)$-vectors.

The following constraints imply that Ye_i/α_i and $Y(e_0 - e_i)/(1 - \alpha_i)$ satisfy the edge constraints: For all $i \in \{1, \dots, n\}$ and all $\{j, k\} \in E$:

$$\alpha_i \leq Y_{ij} + Y_{ik} \leq \alpha_i + (\alpha_j + \alpha_k - 1), \tag{4.3}$$

To see that the above inequalities force Ye_i/α_i and $Y(e_0-e_i)/(1-\alpha_i)$ to satisfy the edge constraints note first that Ye_i/α_i satisfies the edge constraint for some edge $\{j,k\}$ iff the jth and kth coordinates of Ye_i/α_i sum to at least 1. In equations, this requires $Y_{ij}/\alpha_i + Y_{ik}/\alpha_i \geq 1$, or equivalently $\alpha_i \leq Y_{ij} + Y_{ik}$ for the edge $\{j,k\}$. Similarly, the equation $Y_{ij}+Y_{ik} \leq \alpha_i + (\alpha_j+\alpha_k-1)$ implies that $Y(e_0-e_i)/(1-\alpha_i)$ satisfies the edge constraint for edge $\{j,k\}$.

Let (i,t) be a pair of vertices such that $\alpha_i, \alpha_t \notin \{0,1\}$. Let $S_i \subseteq \{1,\ldots,n\}$ be the set, and $\{R_w^{(m)}\}_{w\in S_i}$ the witnesses given by Lemma 4.5 for i. Then i,t are called a *distant pair* if $t \notin \cup_{w\in S_i} Ball(w, R_w^{(m)})$. (Note then that $\alpha_t = \frac{1}{2}+\gamma$.) To ensure that Ye_i/α_i and $Y(e_0-e_i)/(1-\alpha_i)$ are $(S_i, R^{(m)}, \gamma)$-vectors witnessed by $\{R_w^{(m)}\}_{w\in S_i}$ (as required by condition **B**) it suffices to ensure that the tth coordinates of Ye_i/α_i and $Y(e_0-e_i)/(1-\alpha_i)$ are $\frac{1}{2}+\gamma$ for all distant pairs (i,t). In particular, for all such pairs,

$$Y_{it} = \alpha_i\alpha_t = \alpha_i(\frac{1}{2}+\gamma). \tag{4.4}$$

Remark 4.6. By Lemma 4.5, distant pairs have the property that every path in G that connects them contains at least $2/\gamma$ consecutive vertices k such that $\alpha_k = \frac{1}{2}+\gamma$. In particular, any such path contains $2/\gamma - 1$ consecutive edges whose endpoints are "oversatisfied" by α by 2γ.

Finally, $(S_i, R^{(m)}, \gamma)$-vectors must lie in $[0,1]^{n+1}$. The following constraints imply that Ye_i/α_i and $Y(e_0-e_i)/(1-\alpha_i)$ are in $[0,1]^{n+1}$:

$$0 \leq Y_{ij} \leq \alpha_i, \qquad \forall i,j \in \{1,\ldots,n\}, i \neq j \tag{4.5}$$

$$-Y_{ij} \leq 1-\alpha_i-\alpha_j, \qquad \forall i,j \in \{1,\ldots,n\}, i \neq j \tag{4.6}$$

Constraints (4.2)–(4.6) suffice to force Y to satisfy conditions **A** and **B**. However, we will not directly analyze these constraints but instead analyze the following four constraint families which imply constraints (4.2)–(4.6) but are in a cleaner form:

$$Y_{ij} \leq \beta(i,j), \qquad \forall i,j \in \{1,\ldots,n\} \tag{4.7}$$

$$-Y_{ij} \leq \delta(i,j), \qquad \forall i,j \in \{1,\ldots,n\} \tag{4.8}$$

$$Y_{ij}+Y_{ik} \leq a(i,j,k), \qquad \forall \{j,k\} \in E \tag{4.9}$$

$$-Y_{ij}-Y_{ik} \leq b(i,j,k), \qquad \forall \{j,k\} \in E \tag{4.10}$$

Here (1) $\beta(i,j) = \alpha_i\alpha_j$ if i,j is a distant pair and $\beta(i,j) = \min(\alpha_i,\alpha_j)$ otherwise; (2) $\delta(i,j) = -\alpha_i$ if $i=j$, $\delta(i,j) = -\alpha_i\alpha_j$ if i,j is a distant pair, and $\delta(i,j) = 1-\alpha_i-\alpha_j$ otherwise; (3) $a(i,j,k) = \alpha_i + (\alpha_j+\alpha_k-1)$; and (4) $b(i,j,k) = -\alpha_i$. Note that since $\alpha \in [0,1]^{n+1}$, $\beta(i,j)+\delta(i,j) \geq 0$.

To prove the consistency of constraints (4.7)–(4.10), a special combinatorial version of Farkas's lemma (Lemma 1.4) will be used similar to that used in [43] and in [3]. Before giving the exact combinatorial form we require some definitions.

Let $H = (W, F)$ be the graph where $W = \{Y_{ij} : i, j \in \{1, \ldots, n\}\}$ (i.e., there is a vertex for each variable $Y_{i,j}$) and the edges F consist of all pairs $\{Y_{ij}, Y_{ik}\}$ such that $\{j, k\} \in E$. Vertices in W labelled Y_{ii} are called *diagonal*. Given an edge $\{Y_{ij}, Y_{ik}\}$ in H, call i its *bracing node* and $\{j, k\} \in E$ its *bracing edge*. An edge $\{i, j\}$ in G is called *overloaded* if $\alpha_i = \alpha_j = \frac{1}{2} + \gamma$. An edge $\{Y_{ij}, Y_{ik}\}$ in H is *overloaded* if its bracing edge is overloaded.

Let p be a walk $v_0, v_1, \ldots, v_r$ on H and let $e_1, \ldots, e_r$ be the edges in H traversed by this walk. An *alternating sign assignment* (P, N) for p assigns either all the odd or all the even indexed edges of p to the set P with the remaining edges assigned to N. Given an alternating sign assignment (P, N) for p, an endpoint of p is called *positive* (*negative*, respectively) if it is incident to an edge in P (N, respectively). We will be particularly concerned with the positive diagonal endpoints of a walk.

Given a path p in H with an alternating sign assignment (P, N), let

$$S_1^{(p;P,N)} = \sum_{\{Y_{ij}, Y_{ik}\} \in P} a(i, j, k) + \sum_{\{Y_{ij}, Y_{ik}\} \in N} b(i, j, k). \tag{4.11}$$

Suppose the endpoints of p are labelled by $Y_{ij}, Y_{k\ell}$. Define $S_2^{(p;P,N)}$ to be $D+E$ where D is $\delta(i, j)$ if Y_{ij} is a positive endpoint and is $\beta(i, j)$ otherwise; and E is $\delta(k, \ell)$ if $Y_{k\ell}$ is a positive endpoint and is $\beta(k, \ell)$ otherwise. Let $S^{(p;P,N)} = S_1^{(p;P,N)} + S_2^{(p;P,N)}$.

Lemma 4.7 (Special case of Farkas's Lemma). *The constraints on the variables Y_{ij} are unsatisfiable iff there exists a walk p on H and an alternating sign assignment (P, N) for p such that $S^{(p;P,N)}$ is negative.*

Proof. Note first that by Farkas's lemma, constraints (4.7)–(4.10) are unsatisfiable iff there exists a positive rational linear combination of them where the LHS is 0 and the RHS is negative.

Now suppose that there exists a path p in H and an alternating sign assignment (P, N) such that $S^{(p;P,N)} < 0$. Consider the following linear integer combination of the constraints: (1) For each edge $\{Y_{ij}, Y_{ik}\} \in p$, if $\{Y_{ij}, Y_{ik}\} \in P$, add the constraint $Y_{ij} + Y_{ik} \leq a(i, j, k)$; if $\{Y_{ij}, Y_{ik}\} \in N$, add the constraint $-Y_{ij} - Y_{ik} \leq b(i, j, k)$; (2) For each endpoint Y_{ij} of p, if it is a negative endpoint add the constraint $Y_{ij} \leq \beta(i, j)$; if it is a positive endpoint add the constraint $-Y_{ij} \leq \delta(i, j)$. But then, for this combination of constraints the LHS equals 0 while the RHS equals $S^{(p;P,N)} < 0$. So by Farkas's lemma the constraints are unsatisfiable.

Now assume on the other hand that the constraints are unsatisfiable. So there exists a positive rational linear combination of the constraints such that the LHS is 0 and the RHS is negative. In fact, by clearing out denominators, we can assume without loss of generality that this linear combination has *integer* coefficients.

$$
\begin{aligned}
-Y_{12} &\le -\alpha_1\alpha_2 \\
Y_{12} + Y_{13} &\le \alpha_1 + (\alpha_2 + \alpha_3 - 1) \\
-Y_{13} - Y_{34} &\le -\alpha_3 \\
Y_{34} + Y_{35} &\le \alpha_3 + (\alpha_4 + \alpha_5 - 1) \\
-Y_{35} - Y_{56} &\le -\alpha_5 \\
Y_{56} &\le \alpha_5\alpha_6 \\
-Y_{77} &\le -\alpha_7 \\
Y_{77} + Y_{78} &\le \alpha_7 + (\alpha_7 + \alpha_8 - 1) \\
-Y_{78} &\le 1 - \alpha_7 - \alpha_8
\end{aligned}
$$

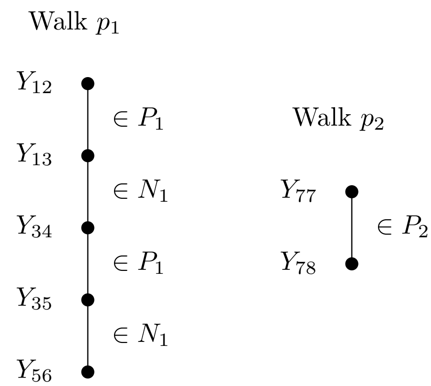

Figure 4.2: A positive integer linear combination of the constraints where the LHS is 0, and which corresponds to two walks p_1 and p_2 in H with alternating sign assignments (P_1, N_1) and (P_2, N_2), respectively.

Hence, as $\beta(i,j) + \delta(i,j) \ge 0$ for all i, j, our combination must contain, without loss of generality, constraints of type (4.9) and (4.10). Moreover, since the LHS is 0, for each Y_{ij} appearing in the integer combination there must be a corresponding occurrence of $-Y_{ij}$. But then, it is easy to see that the constraints in the integer linear combination can be grouped into a set of paths $\{p_i\}$ in H each with its own alternating sign assignment such that the RHS of the linear combination equals $\sum S^{(p_i;P_i,N_i)}$ (for an example, see Figure 4.2). But then, since the RHS is negative, it must be that at least one of the paths p in the set is such that $S^{(p;P,N)} < 0$. The lemma follows. $\square$

So to show that the constraints for the matrix Y are consistent, we will show that $S^{(p;P,N)} \ge 0$ for any walk p on H and any alternating sign assignment (P, N) for p.

To that end, fix a walk p on H and an alternating sign assignment (P, N) for p. To simplify notation we drop the superscript $(p; P, N)$ from $S_1^{(p;P,N)}$, $S_2^{(p;P,N)}$ and $S^{(p;P,N)}$. Let $v_0, v_1, \dots, v_r$ be the nodes visited by p in H (a node may be visited multiple times) and let $e_1, \dots, e_r$ be the edges in H traversed by p. We divide our analysis into three cases depending on whether none, one or both endpoints of p are positive diagonal. We will show that in any of these cases $S \ge 0$.

We first note three easy facts about p used below:

Proposition 4.8. *Let C be the subgraph of G induced by the bracing edges for $e_1, \dots, e_n$. Then,*

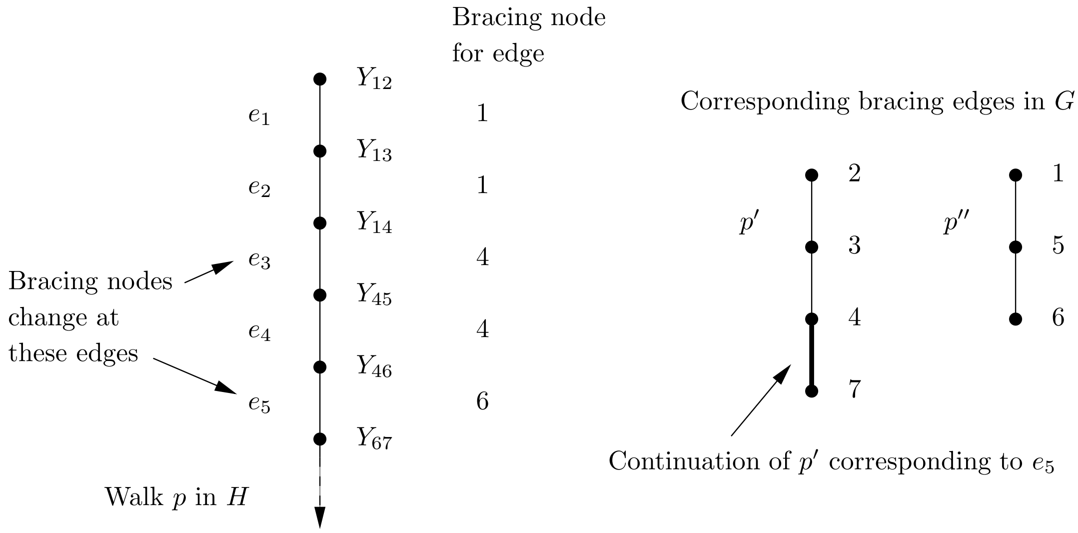

Figure 4.3: A walk p in H and the corresponding pair of walks p', p'' in G formed by the bracing edges in p. The walks p', p'' could meet, e.g., if p visits a diagonal vertex in H.

1. *Subgraph C consists of at most two connected components;*
2. *If p visits a diagonal node, then C is connected; Moreover, if v_0 is diagonal and $v_r = Y_{st}$, then C contains a path in G from s to t;*
3. *If p visits at least two diagonal nodes then C contain a cycle.*

Proof. We sketch a proof of the first fact; the other two are similar.

Consider the edges $e_1, \ldots, e_r$ in order. As long as the bracing node in successive edges does not change, then the bracing edges of these successive edges form a path p' in G. If the bracing node changes, say at edge e_i in p, the bracing edge for e_i now starts a new path p'' in G. Moreover the last vertex w in G visited by p' is the bracing node for e_i. The bracing edges of the edges following e_i in p now extend p'' in G until an edge e_j is encountered with a new bracing node. But then, the bracing edge for e_j must contain w. Hence, the bracing edge for e_j now extends path p' in G. Continuing this argument we see that each time the bracing node changes we go back and forth from having the bracing edges contributing to the paths p' and p'' in G. Fact (1) follows (also see Figure 4.3). $\square$

Case 1: No endpoint of p is positive diagonal

Suppose the endpoints v_0, v_r of p are labelled by Y_{ab} and Y_{cd}, respectively, and consider the following sum S_2': If v_0 is a negative endpoint, then it contributes $\alpha_a\alpha_b$ to S_2'; otherwise it contributes $-\alpha_a\alpha_b$. Similarly, if v_r is negative, then it contributes $\alpha_c\alpha_d$ to S_2' and otherwise it contributes $-\alpha_c\alpha_d$. Since $\alpha \in [0,1]^{n+1}$ and neither endpoint is positive diagonal, it follows that $S_2 \geq S_2'$. So to prove that $S \geq 0$ in this case, it suffices to show $S_1 + S_2' \geq 0$.

To that end, consider the following sum:

$$\sum_{\{Y_{ij},Y_{ik}\}\in P} (-\alpha_i\alpha_j - \alpha_i\alpha_k) + \sum_{\{Y_{ij},Y_{ik}\}\in N} (\alpha_i\alpha_j + \alpha_i\alpha_k). \tag{4.12}$$

By definition of an alternating sign assignment it follows that (4.12) telescopes and equals S_2'. Hence,

$$\begin{aligned} S &\geq S_1 + S_2' \\ &= \sum_{\{Y_{ij},Y_{ik}\}\in P} (a(i,j,k) - (\alpha_i\alpha_j + \alpha_i\alpha_k)) + \sum_{\{Y_{ij},Y_{ik}\}\in N} (b(i,j,k) + (\alpha_i\alpha_j + \alpha_i\alpha_k)) \\ &= \sum_{\{Y_{ij},Y_{ik}\}\in P} (1-\alpha_i)(\alpha_j + \alpha_k - 1) + \sum_{\{Y_{ij},Y_{ik}\}\in N} \alpha_i(\alpha_j + \alpha_k - 1). \end{aligned} \tag{4.13}$$

Now the bracing edges for all edges in P and N are in G. Moreover, α satisfies the VERTEX COVER edge constraints (2.2) for G. Hence, $\alpha_j + \alpha_k \geq 1$ for all edges $\{Y_{ij}, Y_{ik}\} \in P \cup N$. But then, since we always have $0 \leq \alpha_i \leq 1$, it follows that all summands in (4.13) are at least 0 and hence, $S \geq 0$ as desired.

Case 2: One endpoint of p is positive diagonal

Assume without loss of generality that v_0 is the positive endpoint and is labelled Y_{11}, and suppose the other endpoint v_r is labelled Y_{st}. There are two subcases:

Subcase 1: $\{s,t\}$ is a distant pair: By Proposition 4.8, if C is the subgraph of G induced by the bracing edges for $e_1, \ldots, e_n$, then there is a path p' in C (and hence in G) from s to t. So since s,t are distant, Remark 4.6 implies that p' contains at least $2/\gamma - 1$ *consecutive* overloaded edges.

We first define some notation to refer to the summands appearing in (4.13) which will also be important in this subcase: For an edge $e = \{Y_{ij}, Y_{ik}\}$ in our path p,

$$\zeta(e) = \begin{cases} (1-\alpha_i)(\alpha_j + \alpha_k - 1), & \text{if } \{Y_{ij}, Y_{ik}\} \in P \\ \alpha_i(\alpha_j + \alpha_k - 1), & \text{if } \{Y_{ij}, Y_{ik}\} \in N \end{cases}$$

As noted in Case 1, $\zeta(e) \geq 0$ for all $e \in p$.

In Case 1 we showed that $S \geq 0$ by first defining a sum S_2' such that $S_2 \geq S_2'$ and then noting that $S_1 + S_2' = \sum_{e\in p} \zeta(e)$. Unfortunately, in the current subcase, since p contains a positive diagonal endpoint, it is no longer true that $S_2 \geq S_2'$. However, it is easy to see that $S_2 \geq S_2' - (\alpha_1 - \alpha_1^2)$. In particular, $S \geq \sum_{e\in p} \zeta(e) - (\alpha_1 - \alpha_1^2)$ for the current subcase. So since $\zeta(e) \geq 0$ always, to show that $S \geq 0$ in the current subcase, it suffices to show that for "many" edges e in p, $\zeta(e)$ is "sufficiently large" so that $\sum_{e \in p} \zeta(e) \geq \alpha_1 - \alpha_1^2$. The existence of these edges in p will follow from the existence of the $2/\gamma - 1$ consecutive overloaded edges in p'.

Assume without loss of generality that $2/\gamma - 1 = 4q$ for some integer q and let $f_1, \dots, f_{4q}$ be, in order, the $4q$ consecutive overloaded edges in p' (recall that p' is the path from s to t in G and defined by the bracing edges of p). Let $U = \{e_{i_1}, \dots, e_{i_{4q}}\}$ be the set of edges in p whose bracing edges correspond to $f_1, \dots, f_{4q}$ (where e_{i_j} corresponds with f_j). Note that the edges in U need not occur consecutively in p. However, using arguments similar to those used in Proposition 4.8 we can prove the following fact:

Fact 4.9. The edges of p' can be divided into two consecutive walks p'_1 and p'_2 (i.e., all edges in p'_1 and p'_2 are consecutive and all edges in p'_2 either all occur before or after all edges in p'_1) such that if $U_i \subseteq U$ denotes the edges of p whose bracing edges form the walk p'_i, then the order in p of the edges U_1 is the same as the order of the corresponding bracing edges in p'_1, while the order in p of the edges U_2 is the reverse of the order of the corresponding bracing edges in p'_2.

Example 4.10. Suppose $p = Y_{11}$-Y_{12}-Y_{13}-Y_{16}-Y_{46}-Y_{56}. The corresponding walk p' is 5-4-1-2-3-6 and the division guaranteed by the above Fact has $p'_1 =$ 1-2-3-6, $p'_2 =$ 5-4-1.

Let p'_1, p'_2 be the division of p' and U_1, U_2 the corresponding subsets of U for these paths, respectively, guaranteed by Fact 4.9 for p'. Without loss of generality, assume that the length of p'_1 is at least $2q$. In particular, assume without loss of generality that $i_1 < \cdots < i_{2q}$. (If instead p'_2 has length greater than $2q$, then we assume without loss of generality that $i_{2q+1} > \cdots > i_{4q}$ and the arguments below are modified accordingly.)

Let $B = \{1, 3, 5, \dots, 2q - 1\}$. Fix some $j \in B$ and consider the pair $e_{i_j}, e_{i_{j+1}}$ of edges from U. Suppose $e_{i_j} = \{Y_{ab}, Y_{ac}\}$, $e_{i_{j+1}} = \{Y_{uv}, Y_{uw}\}$ where $u \neq a$. Since the bracing edges for these two edges are consecutive in p', all edges e_ℓ such that $i_j < \ell < i_{j+1}$ have the same bracing node (say x) and moreover, this bracing node is different from the bracing nodes in e_{i_j} or $e_{i_{j+1}}$. So we have $x = c = v$ (Figure 4.4).

Let $Z_j = \sum \zeta(e)$, where the sum is over $e \in \{e_{i_j}, e_{i_j+1}, e_{i_j+2}, \dots, e_{i_{j+1}-1}, e_{i_{j+1}}\}$ (i.e., over the edges e_{i_j} and $e_{i_{j+1}}$, and all edges between them in p).

Claim 4.11. $Z_j \geq 2\gamma/3$.

Since $j \in B$ was arbitrary and $|B| = q$, the claim implies $S_1 + S'_2 \geq q(2\gamma/3) \geq 1/3 - \gamma/6$. So since $\gamma < 1/2$ and $\alpha_1 - \alpha_1^2 \leq \frac{1}{4}$ for $\alpha_1 \in [0, 1]$, it follows that $S_1 + S'_2 \geq \alpha_1 - \alpha_1^2$, completing the proof that $S \geq 0$ in this subcase.

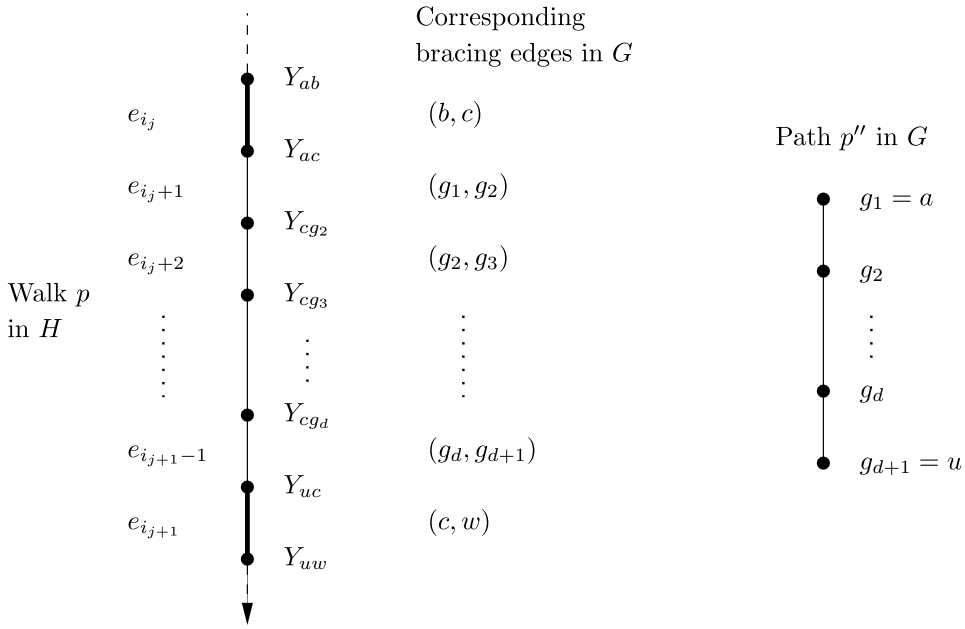

Figure 4.4: A portion of a walk p in H in which the bracing node (in this case c) does not change between edges $e_{i_j}, e_{i_{j+1}}$, together with the path p'' of bracing edges in G for the portion of p with bracing node c.

Proof of Claim 4.11. Suppose $d = i_{j+1} - i_j - 1$ is odd (the case where d is even is similar). Moreover, assume that $e_{i_j}, e_{i_{j+1}} \in P$ (the case where they are both in N is similar). Let $\alpha_a + \alpha_u = 1 + D$. Since e_{i_j} and $e_{i_{j+1}}$ are overloaded,

$$\zeta(e_{i_j}) + \zeta(e_{i_{j+1}}) = 2\gamma(2 - \alpha_u - \alpha_a) = 2\gamma(1 - D). \tag{4.14}$$

If $D \leq \frac{2}{3}$, then (4.14) is greater than $2\gamma/3$, and hence so is Z_j. So assume $D > \frac{2}{3}$.

Note that the bracing edges of $e_{i_j+1}, e_{i_j+2}, \ldots, e_{i_{j+1}-1}$ form a path p'' from a to u of length d in G. Let $g_1, \ldots, g_{d+1}$ be the nodes on p'' where $g_1 = a$, $g_{d+1} = u$ (Figure 4.4). Since α satisfies the VERTEX COVER edge constraints (2.2) for G, $\sum_{k=1}^{d+1} \alpha_{g_k} \geq (d+1)/2$. In fact, we must have that $\sum_{k=1}^{d+1} \alpha_{g_k} \geq (d+1)/2 + D$ (this just says that since the endpoints of p'' sum to $1 + D$ then some edge(s) along p'' must be oversatisfied by D). But then,

$$Z_j \geq \sum_{k=1}^{(d+1)/2} \zeta(e_{i_j+2k-1}) = \alpha_c \left(\sum_{k=1}^{d+1} \alpha_{g_k} - \frac{d+1}{2} \right) \geq \left(\frac{1}{2} + \gamma \right) D > \frac{2\gamma}{3}.$$

□

Subcase 2: $\{s, t\}$ is not a distant pair: Let S_{st} be the contribution of Y_{st} to S_2 (i.e., $S_{st} = \delta(s, t)$ if v_r is a positive endpoint and $S_{st} = \beta(s, t)$ if v_r is negative). Since the contribution of Y_{11} to S_2 is $-\alpha_1$, it follows that $S_2 = S_{st} - \alpha_1$.

For an edge $e_\ell = \{Y_{ij}, Y_{ik}\} \in p$, let

$$T_\ell = \begin{cases} a(i,j,k), & \text{if } e_\ell \in P \\ b(i,j,k), & \text{if } e_\ell \in N \end{cases}$$

Recall that $v_0, v_1 \ldots, v_r$ are the nodes visited by the walk p and that e_i denotes the edge traversed between v_{i-1} and v_i. Note then that $S_1 = \sum_{\ell=1}^{r} T_\ell$. Moreover, recall that we have assumed without loss of generality that $v_0 = Y_{11}$ and $v_r = Y_{st}$. So since $e_1 \in P$, the following claim implies $S_{st} + \sum_{\ell=1}^{r} T_\ell \geq \alpha_1$, and hence that $S \geq 0$ in this subcase.

Claim 4.12. *Let $1 \leq q \leq r$ and suppose $v_{q-1} = Y_{ij}$, $v_q = Y_{ik}$ (i.e., $e_q = \{Y_{ij}, Y_{ik}\}$). Then $S_{st} + \sum_{\ell=q}^{r} T_\ell$ is at least $\min(\alpha_i, \alpha_j)$ if $e_q \in P$ and is at least $\min(0, 1-\alpha_i-\alpha_j)$ if $e_q \in N$.*

Proof. By "backward" induction on q. For the base case $q = r$, assume without loss of generality that $v_{q-1} = Y_{sj}$, so that $e_q = \{Y_{sj}, Y_{st}\}$. If $e_q \in N$, then $T_1 = -\alpha_s$ so that $T_1 + S_{st} = -\alpha_s + \min(\alpha_s, \alpha_t)$. Since α satisfies the edge constraints (2.2), it follows that $\alpha_j + \alpha_t \geq 1$ for the bracing edge $\{j, t\}$. Hence, $T_1 + S_{st} \geq \min(0, 1 - \alpha_j - \alpha_s)$. If instead $e_q \in P$, then $T_1 = \alpha_s + (\alpha_j + \alpha_t - 1)$ so that

$$T_1 + S_{st} = [\alpha_s + (\alpha_j + \alpha_t - 1)] + (1 - \alpha_s - \alpha_t) = \alpha_j.$$

The base case $q = r$ follows.

Assume the claim holds for e_q and consider $e_{q-1} = \{Y_{ij}, Y_{ik}\}$ where $v_{q-2} = Y_{ij}$ and $v_{q-1} = Y_{ik}$. If $e_{q-1} \in N$, then $T_{q-1} = -\alpha_i$. Moreover, $e_q \in P$ and by induction,

$$S_{st} + \sum_{\ell=q}^{r} T_\ell \geq \min(\alpha_i, \alpha_k).$$

Since α satisfies the edge constraints (2.2), it follows that $\alpha_j + \alpha_k \geq 1$ for the bracing edge $\{j, k\}$. Hence,

$$S_{st} + \sum_{\ell=q-1}^{r} T_\ell \geq \min(0, 1 - \alpha_i - \alpha_j).$$

If instead $e_{q-1} \in P$, then $T_{q-1} = \alpha_i + (\alpha_j + \alpha_k - 1)$. Moreover, $e_q \in N$ and by induction,

$$S_{st} + \sum_{\ell=q}^{r} T_\ell \geq \min(0, 1 - \alpha_i - \alpha_k).$$

So since $\alpha_j + \alpha_k \geq 1$, it follows that

$$S_{st} + \sum_{\ell=q-1}^{r} T_\ell \geq \min(\alpha_i, \alpha_j).$$

The claim follows for e_{q-1}. ∎

Case 3: Both endpoints of p are positive diagonal

Since p contains two diagonal vertices, Proposition 4.8 implies that there is a cycle C in the subgraph of G induced by the bracing edges corresponding to the edges in p. Since $\mathrm{girth}(G) \geq 4R^{(m)}$, it follows that C contains a distant pair. But then, as there are two different paths between this pair along C, Remark 4.6 implies that there are two subpaths p'_1 and p'_2 in C *each* consisting of $2/\gamma$ overloaded edges.

Recall that in subcase 1 of Case 2 where there was one positive diagonal vertex, one such subpath was used to argue that $S \geq 0$ in that subcase. In the current case where there are *two* positive diagonals and the *two* subpaths p'_1 and p'_2, the same argument then implies that $S \geq 0$ for the current case also.

4.2 Lower bounds for hypergraph Vertex Cover

In this section we concentrate on proving integrality gaps of size $k-\epsilon$ for relaxations obtained by the LS lift-and-project technique for VERTEX COVER on k-uniform hypergraphs. As mentioned above, there is a gap between the factors achieved by known approximation algorithms and those ruled out by PCPs: While PCP-based hardness results for k-uniform hypergraph VERTEX COVER rule out $k-1-\epsilon$ polynomial-time approximations [15], only $k-o(1)$ approximation algorithms are known. Our main result for this section (Theorem 4.13) is that for all $\epsilon > 0$ and all sufficiently large n there exist k-uniform hypergraphs on n vertices for which $\Omega(\log\log n)$ rounds of LS are necessary to obtain a relaxation for VERTEX COVER with integrality gap less than $k-\epsilon$.

4.2.1 Methodology

As with all our lower bounds for the Lovász-Schrijver hierarchies, we will use Lemma 2.1 to prove our lower bound. As before, the key to proving our lower bound will be coming up with appropriate protection matrices.

Recall from Section 3.2 the standard relaxation for VERTEX COVER on a rank-k hypergraph $G=(V,E)$:

$$\min \sum_{i \in V} x_i \tag{4.15}$$

$$\sum_{j=1}^{k} x_j \geq 1, \quad \forall (1,\ldots,k) \in E \tag{4.16}$$

$$0 \leq x_i \leq 1, \quad \forall i \in [V]. \tag{4.17}$$

As in Section 3.2, let $\mathrm{HVC}(G)$ denote the polytope corresponding to the feasible reason of this relaxation. To prove gaps of $k-\epsilon$ for $N^r(\mathrm{HVC}(G))$ we will protect the point $y_\gamma = (1, \frac{1}{k}+\gamma, \ldots, \frac{1}{k}+\gamma)$ where $\gamma > 0$ is an arbitrarily small constant. For y_γ, the vector $Y(e_0 - e_i)/(1-x_i)$ given by the protection matrices used in Chapter 3 to prove integrality gaps of size $k-1-\epsilon$ for hypergraph VERTEX COVER in the LS_+ hierarchy is not guaranteed to be in $\mathrm{HVC}(G)$: if coordinate i is set to 0 in y_γ, we violate all edge constraints involving the ith vertex. So a more sophisticated protection matrix is needed to get an integrality gap of size $k-\epsilon$.

For our integrality gap of $2-o(1)$ for graph VERTEX COVER in Section 4.1 above we used a special combinatorial form of Farkas's lemma specifically applicable to the constraints involved in defining protection matrices for graphs. To prove integrality gaps of size $k-o(1)$ for $N^r(\mathrm{HVC}(G))$, we will also reduce the existence of our protection matrices exists to the feasibility of linear programs; however, we will need a much more complicated form of Farkas's lemma than in the graph case. As such, we will only be able to carry out our arguments for $r \in O(\log\log n)$ rounds instead of $\Omega(\log n)$ rounds as we did for graph VERTEX COVER.

4.2.2 The lower bound

Theorem 4.13. *For all $k \geq 2$, $\epsilon > 0$ there exist constants $n_0(k,\epsilon), \delta(k,\epsilon) > 0$ s.t. for every $n \geq n_0(k,\epsilon)$ there exists a k-uniform hypergraph G on n vertices for which the integrality gap of $N^r(\mathrm{HVC}(G))$ is at least $k-\epsilon$ for all $r \leq \delta(k,\epsilon)\log\log n$.*

Theorem 4.13 will follow from the following two theorems.

Theorem 4.14. *For all $k \geq 2$ and any $\epsilon > 0$ there exists an $n_0(k,\epsilon)$ such that for every $n \geq n_0(k,\epsilon)$ there exist k-uniform hypergraphs with n vertices and $O(n)$ hyperedges having $\Omega(\log n)$ girth but no independent set of size greater than ϵn.*

Theorem 4.15. *Fix $\gamma > 0$ and let $y_\gamma = (1, \frac{1}{k}+\gamma, \frac{1}{k}+\gamma, \ldots, \frac{1}{k}+\gamma)$. Let G be a k-uniform hypergraph such that* $\mathrm{girth}(G) \geq \frac{20}{\gamma} r 5^r$. *Then $y_\gamma \in N^r(\mathrm{HVC}(G))$.*

Theorem 4.14 is an easy extension of Theorem 4.2 (Erdős [17]) to hypergraphs. The remainder of this section will be devoted to proving Theorem 4.15.

4.2.3 Proof of Theorem 4.15

The proof is by induction where the inductive hypothesis will require some set of vectors to be in $N^m(\mathrm{HVC}(G))$ for $m \leq r$. These vectors will be essentially all-$(\frac{1}{k}+\gamma)$ except possibly for a few small neighbourhoods where they can take arbitrary nonnegative values so long as the edge constraints for G are satisfied. The exact characterization is given by the following definition similar to an analogous definition in [3] as well as to Definition 4.4:

Definition 4.16. Let $S \subseteq [n]$, R be a positive integer, and $\gamma > 0$. A nonnegative vector $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ with $\alpha_0 = 1$ is an (S, R, γ)-vector if the entries satisfy the edge constraints and if $\alpha_j = \frac{1}{k} + \gamma$ for each $j \notin \cup_{w \in S} Ball(w, R)$. Here $Ball(w, R)$ denotes the set of vertices within distance R of w in the graph.

Let $R_r = 0$ and let $R_m = 5R_{m+1} + 1/\gamma$ for $0 \leq m < r$. Note $R_m \leq \text{girth}(G)/20$.

To prove Theorem 4.15 we prove the following inductive claim. The theorem is a subcase of $m = r$:

Inductive Claim for $N^m(\text{HVC}(G))$: *For every set S of $r - m$ vertices, every (S, R_m, γ)-vector is in $N^m(\text{HVC}(G))$.*

The base case $m = 0$ is trivial since (S, R_m, γ)-vectors satisfy the edge constraints for G. In the remainder of this section we prove the Inductive Claim for $m + 1$ assuming truth for m.

To that end, let α be an (S, R_{m+1}, γ)-vector where $|S| = r - m - 1$. By induction, every $(S \cup \{i\}, R_m, \gamma)$-vector is in $N^m(\text{HVC}(G))$. So to prove that $\alpha \in N^{m+1}(\text{HVC}(G))$ it suffices by Lemma 2.1 to exhibit an $(n+1) \times (n+1)$ symmetric matrix Y such that:

- **A**. $Ye_0 = diag(Y) = \alpha$,
- **B**. For each i such that $\alpha_i = 0$, $Ye_i = 0$; for each i such that $\alpha_1 = 1$, $Ye_0 = Ye_i$; otherwise, Ye_i/α_i and $Y(e_0 - e_i)/(1 - \alpha_i)$ are $(S \cup \{i\}, R_m, \gamma)$-vectors.

As was done in [3] and [43], we will write these conditions as a linear program and show that the program is feasible, proving the existence of Y. Our notation will assume symmetry, namely Y_{ij} represents $Y_{\{i,j\}}$.

Condition **A** requires:

$$Y_{kk} = \alpha_k \qquad \forall k \in \{1, \ldots, n\}. \tag{4.18}$$

Condition **B** requires first of all that Ye_j/α_j and $Y(e_0 - e_j)/(1 - \alpha_j)$ satisfy the edge constaints: For all $i \in \{1, \ldots, n\}$ and all $\{j_1, \ldots, j_k\} \in E$:

$$\alpha_i \leq Y_{ij_1} + \ldots + Y_{ij_\ell} \leq \alpha_i + (\alpha_{j_1} + \ldots + \alpha_{j_k} - 1). \tag{4.19}$$

Vertices i, t are called a *distant pair* if $t \notin \cup_{w \in S \cup \{i\}} Ball(w, 5R_{m+1} + 1/\gamma)$. (Note then that $\alpha_t = \frac{1}{k} + \gamma$.) Condition **B** requires that for such a pair, the tth coordinates of Ye_i/α_i and $Y(e_0 - e_i)/(1 - \alpha_i)$ are $\frac{1}{k} + \gamma$. In particular,

$$Y_{it} = \alpha_i \alpha_t = \alpha_i \left(\frac{1}{k} + \gamma \right). \tag{4.20}$$

Note that distant pairs have the property that every path in G that connects them contains at least $4R_m + 1/\gamma - 1$ hyperedges each of which α oversatisfies by $k\gamma$, and at most R_m hyperedges which α does *not* oversatisfy by $k\gamma$.

Finally, condition **B** requires that Ye_i/α_i, $Y(e_0 - e_i)/(1-\alpha_i)$ are in $[0,1]^{n+1}$:

$$0 \le Y_{ij} \le \alpha_i, \qquad \forall i,j \in \{1,\dots,n\}, i \ne j \tag{4.21}$$
$$-Y_{ij} \le 1 - \alpha_i - \alpha_j, \qquad \forall i,j \in \{1,\dots,n\}, i \ne j \tag{4.22}$$

The above constraints suffice to force Y to satisfy conditions **A** and **B**. However, we will not directly analyze these constraints but instead analyze the following four constraint families which imply the above constraints but are in a cleaner form:

$$Y_{ij} \le \beta(i,j), \qquad \forall i,j \in \{1,\dots,n\} \tag{4.23}$$
$$-Y_{ij} \le \delta(i,j), \qquad \forall i,j \in \{1,\dots,n\} \tag{4.24}$$
$$Y_{ij_1} + \ldots + Y_{ij_k} \le a(i,j_1,\dots,j_k), \qquad \forall \{j_1,\dots,j_k\} \in E \tag{4.25}$$
$$-Y_{ij_1} - \ldots - Y_{ij_k} \le b(i,j_1,\dots,j_k), \qquad \forall \{j_1,\dots,j_k\} \in E \tag{4.26}$$

Here (1) $\beta(i,j) = \alpha_i\alpha_j$ if i,j is a distant pair and $\beta(i,j) = \min(\alpha_i,\alpha_j)$ otherwise; (2) $\delta(i,j) = -\alpha_i$ if $i = j$; $\delta(i,j) = -\alpha_i\alpha_j$ if i,j is a distant pair; and $\delta(i,j) = 1-\alpha_i-\alpha_j$ otherwise; (3) $a(i,j_1,\dots,j_k) = \alpha_i + (\alpha_{j_1}+\ldots+\alpha_{j_k}-1)$; and (4) $b(i,j_1,\dots,j_k) = -\alpha_i$. Note that $\beta(i,j) + \delta(i,j) \ge 0$ always, since $\alpha \in [0,1]^{n+1}$,

To prove the consistency of constraints (4.23)–(4.26), we will use a special combinatorial version of Farkas's Lemma similar in spirit to that used in [43] and [3], as well as in Section 4.1 above. Before giving the exact form, we require some definitions.

Definition 4.17. A *tiling* (W,P,N) *for* G is a connected k-uniform hypergraph $H = (W,P,N)$ on vertices W and two disjoint k-edge sets P and N such that:

1. Each vertex in W is labelled by Y_{ij}, $i,j \in \{1,\dots,n\}$. Note that distinct vertices need not have different labels.

2. Each vertex belongs to at most one edge in P and at most one edge in N (in particular, all edges in P are mutually disjoint, as are all edges in N).

3. All edges in $P \cup N$ are of the form $\{Y_{ij_1},\dots,Y_{ij_k}\}$ where $\{j_1,\dots,j_k\} \in E$.

The edges of a tiling are called *tiles*. Vertices in W not incident to any tile in P are called *unmatched negative* vertices; vertices in W not incident to any tile in N are called *unmatched positive* vertices. Let U_N and U_P denote the sets of unmatched negative and unmatched positive vertices, respectively. A vertex labelled Y_{ii} is called *diagonal*. Denote the set of unmatched positive diagonal vertices in W by U_{PD}. A vertex labelled Y_{ij} is called a *distant pair* if $\{i,j\}$ are a distant pair. Given a tile

$\{Y_{ij_1}, \ldots, Y_{ij_k}\}$ in P or N, call $\{j_1, \ldots, j_k\} \in E$ its *bracing edge* and i its *bracing node*. An edge $\{j_1, \ldots, j_k\}$ in G is called *overloaded* if $\alpha_{j_1} = \ldots = \alpha_{j_k} = \frac{1}{k} + \gamma$. A tile $\{Y_{ij_1}, \ldots, Y_{ij_k}\}$ in H is *overloaded* if its bracing edge is overloaded. (See Figure 4.5 for an example of a simple tiling.)

Given a tiling $H = (W, P, N)$ for G, define the following sums:

$$S_1^H = \sum_{\{Y_{ij_1}, \ldots, Y_{ij_k}\} \in P} a(i, j_1, \ldots, j_k) + \sum_{\{Y_{ij_1}, \ldots, Y_{ij_k}\} \in N} b(i, j_1, \ldots, j_k), \tag{4.27}$$

$$S_2^H = \sum_{Y_{ij} \in U_P} \delta(i, j) + \sum_{Y_{ij} \in U_N} \beta(i, j). \tag{4.28}$$

Finally, let $S^H = S_1^H + S_2^H$.

Lemma 4.18 (Special case of Farkas's Lemma). *Constraints (4.23)–(4.26) are unsatisfiable iff there exists a tiling $H = (W, P, N)$ for G such that $S^H < 0$.*

Proof. Note first that by the general form of Farkas's lemma constraints (4.23)–(4.26) are unsatisfiable iff there exists a positive rational linear combination of them where the LHS is 0 and the RHS is negative.

Now suppose that there exists a tiling $H = (W, P, N)$ such that $S^H < 0$. Consider the following linear integer combination of the constraints: (1) For each tile $e = \{Y_{ij_1}, \ldots, Y_{ij_k}\} \in H$, if $e \in P$, add the constraint $Y_{ij_1} + \ldots + Y_{ij_k} \le a(i, j_1, \ldots, j_k)$; if $e \in N$, add the constraint $-Y_{ij_1} - \ldots - Y_{ij_k} \le b(i, j_1, \ldots, j_k)$; (2) For each $v \in U_N$ labelled Y_{ij}, add the constraint $Y_{ij} \le \beta(i, j)$; (3) For each $v \in U_P$ labelled Y_{ij}, add the constraint $-Y_{ij} \le \delta(i, j)$. But then, for this combination of constraints the LHS equals 0 while the RHS equals $S^H < 0$. So by Farkas's lemma the constraints are unsatisfiable.

Now assume on the other hand that the constraints are unsatisfiable. So there exists a positive rational linear combination of the constraints such that the LHS is 0 and the RHS is negative. By clearing out denominators, we can assume that the linear combination has *integer* coefficients. Hence, as $\beta(i, j) + \delta(i, j) \ge 0$ always, our combination must contain constraints of type (4.25) and (4.26). Moreover, since the LHS is 0, for each Y_{ij} appearing in the integer combination there must be a corresponding occurrence of $-Y_{ij}$. But then, it is easy to see that the constraints in the integer linear combination can be grouped into a set of tilings $\{H_i = (W_i, P_i, N_i)\}$ such that the RHS of the linear combination equals $\sum_i S^{H_i}$ (for an example, see Figure 4.5). Since the RHS is negative, it must be that at least one of the tilings H in the set is such that $S^H < 0$. The lemma follows. ∎

So to show that the constraints for the matrix Y are consistent and thus complete the proof of the Inductive Claim for $m+1$ (and complete the proof of Theorem 4.15),

$$
\begin{aligned}
Y_{12} + Y_{13} + Y_{14} &\le \alpha_1 + (\alpha_2 + \alpha_3 + \alpha_4 - 1) \\
-Y_{12} - Y_{12} - Y_{15} &\le -\alpha_1 \\
-Y_{14} - Y_{45} - Y_{46} &\le -\alpha_4 \\
Y_{44} + Y_{46} + Y_{47} &\le \alpha_4 + (\alpha_4 + \alpha_6 + \alpha_7 - 1) \\
Y_{15} &\le \alpha_1 \\
Y_{45} &\le \alpha_4 \\
-Y_{47} &\le 1 - \alpha_4 - \alpha_7 \\
-Y_{44} &\le -\alpha_4
\end{aligned}
$$

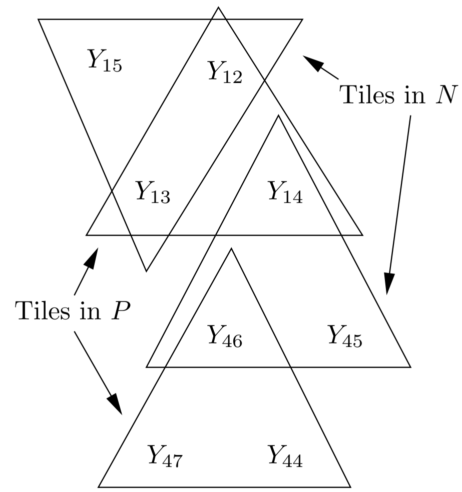

Figure 4.5: A positive integer linear combination of hypergraph VERTEX COVER constraints where the LHS is 0, together with its corresponding tiling. Note $Y_{15}, Y_{45} \in U_N$, and $Y_{47} \in U_P$, $Y_{44} \in U_{PD}$.

we will show that $S^H \geq 0$ for any tiling $H = (W, P, N)$ for G. To that end, fix a tiling $H = (W, P, N)$ for G. Our analysis divides into three cases depending on the size of U_{PD}, the set of unmatched positive diagonal vertices in H. In the first (and easiest) case, $|U_{PD}| = 0$; in the second, $|U_{PD}| \geq 2$; and in the final, $|U_{PD}| = 1$. We will show that $S^H \geq 0$ in all these cases . To reduce clutter, we drop the superscript H from S_1^H, S_2^H and S^H. In what follows, let C be the subgraph of G induced by the bracing edges of all tiles in H.

We first note two easy facts about H used below:

Proposition 4.19. *1. Suppose H contains a diagonal vertex. Then C is connected. Moreover, for any vertex labelled Y_{ij} in H, there exists a path p in H such that the bracing edges corresponding to the tiles in p form a path p' from i to j in C.*

2. The distance between any two diagonal vertices in H is at least $\operatorname{girth}(G)/2$.

Proof. We leave part (1) as an easy exercise and sketch a proof of part (2).

Since H is connected, there exists a path between any two diagonal vertices in H. We will show that the subgraph of G induced by the bracing edges of the tiles in such path must contain a cycle. Part (2) will then follow.

To that end, let q be an arbitrary path in H comprised, in order, of tiles $e_1, \ldots, e_r$. Consider these tiles in order beginning with e_1. As long as the bracing node in successive tiles does not change, then the bracing edges of these tiles form a path q' in G. If the bracing node changes at some tile, say e_i, then the bracing edge for e_i starts a new path q'' in G. Moreover, some vertex $w \in G$ from the last edge visited in q' becomes the (new) bracing node for e_i. The bracing edges of the tiles following e_i now extend q'' until an edge e_j is encountered with yet a different bracing node. But then, the bracing edge for e_j must contain w. Hence the bracing edge for e_j must extend q'. Continuing this argument, we see that each time the bracing node switches, the tiles switch back and forth from contributing to q' or q''.

Each time the current tile e_ℓ contains some diagonal vertex labelled Y_{ii}, the bracing node i for e_ℓ is also contained in the bracing edge for e_ℓ. But then, since the bracing node belongs to, say, q' while the bracing edge belongs to q'', it follows that q' and q'' must intersect at vertex i in G. Hence, if a tile path q contains *two* diagonal vertices, q' and q'' must intersect twice. That is, the subgraph in G induced by the bracing edges of tiles in q must contain a cycle. Part (2) follows. $\square$

Case 1: No unmatched positive diagonal vertices

Consider the following sum:

$$S_2' = \sum_{Y_{ij} \in U_N} \alpha_i \alpha_j - \sum_{Y_{ij} \in U_P} \alpha_i \alpha_j. \tag{4.29}$$

Note that since $\alpha \in [0,1]^{n+1}$, it follows that $-\alpha_i\alpha_j \leq 1 - \alpha_i - \alpha_j$ and $\alpha_i\alpha_j \leq \alpha_i$. So since there are no unmatched positive diagonal vertices in the tiling, $\alpha_i\alpha_j \leq \beta(i,j)$ and $-\alpha_i\alpha_j \leq \delta(i,j)$ for all unmatched vertices labelled Y_{ij} in the tiling, and hence, $S_2 \geq S_2'$. So to show $S \geq 0$ in this case, it suffices to show $S_1 + S_2' \geq 0$.

To that end, consider the following sum:

$$\sum_{\{Y_{ij_1},\ldots,Y_{ij_k}\} \in P} (-\alpha_i\alpha_{j_1} - \ldots - \alpha_i\alpha_{j_k}) + \sum_{\{Y_{ij_1},\ldots,Y_{ij_k}\} \in N} (\alpha_i\alpha_{j_1} + \ldots + \alpha_i\alpha_{j_k}). \tag{4.30}$$

By properties (2) and (3) of a tiling, it follows that (4.30) telescopes and equals S_2'. But then, to show that $S \geq 0$ in this case it suffices to show that for each $\{Y_{ij_1},\ldots,Y_{ij_k}\} \in P$,

$$a(i,j_1,\ldots,j_k) - [\alpha_i(\alpha_{j_1} + \ldots + \alpha_{j_k})] = (1-\alpha_i)(\alpha_{j_1} + \ldots + \alpha_{j_k} - 1), \tag{4.31}$$

is nonnegative (the first term comes from the term in S_1 for the tile, and the second from the term for the tile in (4.30)), and for each $\{Y_{ij_1},\ldots,Y_{ij_k}\} \in N$,

$$b(i,j_1,\ldots,j_k) + [\alpha_i(\alpha_{j_1} + \ldots + \alpha_{j_k})] = \alpha_i(\alpha_{j_1} + \ldots + \alpha_{j_k} - 1), \tag{4.32}$$

is nonnegative (again, the first term comes from S_1 and the second from (4.30)). But (4.31) and (4.32) are both nonnegative since the bracing edges for all tiles are in G and α satisfies the edge constraints for G. Hence, $S \geq 0$ in this case.

Case 2: At least 2 unmatched positive diagonal vertices

We first define some notation to refer to quantities (4.31) and (4.32) which will be used again in this case: For a tile $\{Y_{ij_1}, \ldots, Y_{ij_k}\}$, define $\zeta(i, j_1, \ldots, j_k)$ to be $(1-\alpha_i)(\alpha_{j_1} + \ldots + \alpha_{j_k} - 1)$ if $\{Y_{ij_1}, \ldots, Y_{ij_k}\} \in P$ and to be $\alpha_i(\alpha_{j_1} + \ldots + \alpha_{j_k} - 1)$ if $\{Y_{ij_1}, \ldots, Y_{ij_k}\} \in N$. We will sometimes abuse notation and write $\zeta(e)$ instead of $\zeta(i, j_1, \ldots, j_k)$ for a tile $e = \{Y_{ij_1}, \ldots, Y_{ij_k}\}$.

In Case 1 we showed that $S \geq 0$ by (1) defining a sum S_2' such that $S \geq S_1 + S_2'$, then (2) noting that $S_1 + S_2' = \sum_{e \in H} \zeta(e)$, and finally (3) showing that $\zeta(e) \geq 0$ for all tiles e in H. Unfortunately, in the current case, since H contains unmatched positive diagonal vertices, it is no longer true that $S \geq S_1 + S_2'$. However, it is easy to see that $S \geq (S_1 + S_2') - \sum_{Y_{ii} \in U_{PD}} (\alpha_i - \alpha_i^2)$. In particular, $S \geq \sum_{e \in H} \zeta(e) - \sum_{Y_{ii} \in U_{PD}} (\alpha_i - \alpha_i^2)$. So since $\zeta(e) \geq 0$ for all tiles, to show that $S \geq 0$ for the current case, it suffices to show that for "many" tiles e in H, $\zeta(e)$ is sufficiently large so that $\sum_{e \in H} \zeta(e) \geq \sum_{Y_{ii} \in U_{PD}} (\alpha_i - \alpha_i^2)$.

We require the following lemma:

Lemma 4.20. *If there are $\ell \geq 2$ diagonal vertices in H, then there exist ℓ disjoint paths of length* $\mathrm{girth}(G)/4 - 2$ *in H which do not involve any diagonal vertices.*

Proof. Let Q be a tree in H that spans all the diagonal vertices in H (such a tree exists since H is connected). For each diagonal vertex $u \in H$ let $B(u)$ be the tiles in Q with distance at most $\mathrm{girth}(G)/4 - 1$ from u. Then for all diagonal vertices $u, v \in H$, the balls $B(u)$ and $B(v)$ must be disjoint: otherwise, the distance between u and v would be less than $\mathrm{girth}(G)/2$, contradicting part (2) of Proposition 4.19. Since Q is a spanning tree, for each diagonal vertex $u \in H$, there exists a path of length at least $\mathrm{girth}(G)/4 - 2$ in $B(u)$. ☐

By the lemma, for each vertex $v \in U_{PD}$ there exists a path p_v in H of length $\mathrm{girth}(G)/4 - 2$ such that for distinct $u, v \in U_{PD}$ the paths p_u and p_v share no tiles. We will show (provided that n is sufficiently large) that for each $v \in U_{PD}$, $\sum_{e \in p_v} \zeta(e) \geq 1$. Since $\alpha_i - \alpha_i^2 \leq \frac{1}{4}$, it will follow that $S \geq 0$, completing the proof for this case.

So fix $v \in U_{PD}$. Since $R^m \leq \mathrm{girth}(G)/20$, p_v contains at least $\mathrm{girth}(G)/20$ disjoint pairs of adjacent overloaded tiles (i.e., $\alpha_j = \frac{1}{k} + \gamma$ for all vertices j in the bracing edges of the two tiles in the pair). Let $e_1 = \{Y_{qr_1}, \ldots, Y_{qr_k}\} \in P$ and $e_2 = \{Y_{st_1}, \ldots, Y_{st_k}\} \in N$ be such a pair. Note that $\zeta(e_1) = (1-\alpha_i)k\gamma$ and $\zeta(e_2) = \alpha_s k\gamma$. Now, either $s = r_\ell$ for some $\ell \in [k]$, or $s = q$. If $s = r_\ell$, then $\alpha_s = \frac{1}{k} + \gamma$ and $\zeta(e_1) + \zeta(e_2) \geq k\gamma(\frac{1}{k} + \gamma)$. If instead $s = q$, then $\zeta(e_1) + \zeta(e_2) \geq k\gamma$. In either case, $\zeta(e_1) + \zeta(e_2) \geq k\gamma(\frac{1}{k} + \gamma)$. Hence, summing over all $\mathrm{girth}(G)/20$ disjoint pairs of adjacent overloaded tiles, it follows that $\sum_{e \in p_v} \zeta(e) \geq k\gamma(\frac{1}{k} + \gamma)\,\mathrm{girth}(G)/20$. As desired, the latter is indeed greater than 1 for large n since $\mathrm{girth}(G) = \Omega(\log n)$.

Case 3: Exactly 1 unmatched positive diagonal vertex

The argument in Case 2 actually rules out the possibility that the tiling contains two or more diagonal vertices, unmatched or not. Hence, we will assume that our tiling contains exactly one diagonal vertex labelled without loss of generality by Y_{11}. We consider two subcases.

Subcase 1: At least one of the unmatched vertices is a distant pair:

Arguing as in Case 2, we have in this subcase that $S = [\sum_{e \in H} \zeta(e)] - (\alpha_1 - \alpha_1^2)$ where, moreover, $\zeta(e) \geq 0$ for all tiles $e \in H$. Hence, to show that $S \geq 0$ in this subcase, it suffices to show that there exists a subset $H' \subseteq H$ such that $\sum_{e \in H'} \zeta(e) \geq \frac{1}{4} \geq \alpha_1 - \alpha_1^2$.

To that end, let v be an unmatched vertex in H labelled Y_{ij} where $\{i, j\}$ is a distant pair. By part (1) of Proposition 4.19 there is a path p in H such that the bracing edges corresponding to the tiles in p form a path q in C connecting vertices i and j. Since i, j are distant, q must contain, by definition, at least $4R_{m+1} + 1/\gamma$ overloaded edges. Moreover, it is not hard to see that there exists a sub-path p' of p of length at most $5R_{m+1} + 1/\gamma$ such that the bracing edges of the tiles in p' include all these overloaded edges. Hence p' must contain at least $1/\gamma$ disjoint pairs of overloaded tiles. But then, arguing as in Case 2, it follows that $\sum_{e \in p} \zeta(p) \geq (1/\gamma)(k\gamma(\frac{1}{k} + \gamma)) \geq \frac{1}{4}$, and hence that $S \geq 0$ in this subcase.

Subcase 2: None of the unmatched vertices is a distant pair:

Note first that we can assume that H contains no cycles: If it does, then it is easy to see that C must also contain a cycle, and hence, we can use the ideas from Case 2 to show that $S \geq 0$.

So assume H has no cycles and define a tree T as follows: There is a node in T for each tile in H. The root $root(T)$ corresponds to the tile containing Y_{11}. There is an edge between two nodes in T iff the tiles corresponding to the nodes share a vertex. Note that T is a tree since H is acyclic. For a node $v \in T$, let $Tile(v)$ denote its corresponding tile in H. Finally, for a node $v \in T$ we abuse notation and say $v \in P$ (resp., $v \in N$) if $Tile(v) \in P$ (resp., $Tile(v) \in N$).

Recursively define a function t on the nodes of T as follows: Let v be a node in T and suppose $Tile(v) = \{Y_{ij_1}, \ldots, Y_{ij_k}\}$. If $v \in P$, then

$$t(v) = a(i, j_1, \ldots, j_k) + \sum_{v' \in Child(v)} t(v') + \sum_{\text{unmatched } Y_{ij_\ell} \text{ in } Tile(v)} \delta(i, j_\ell). \tag{4.33}$$

If instead $v \in N$, then

$$t(v) = b(i, j_1, \ldots, j_k) + \sum_{v' \in Child(v)} t(v') + \sum_{\text{unmatched } Y_{ij_\ell} \text{ in } Tile(v)} \beta(i, j_\ell). \tag{4.34}$$

A simple induction now shows that $t(root(T)) = S$ (recall that $root(T)$ corresponds to the tile containing Y_{11}). Hence, to show that $S \geq 0$ in this subcase it suffices to show that $t(root(T)) \geq 0$. This will follow from the following lemma:

Lemma 4.21. *Given $v \in T$, let $e = Tile(v) = \{Y_{ij_1}, \ldots, Y_{ij_k}\}$. Assume, without loss of generality, that $Y_{ij_1}, \ldots, Y_{ij_\ell}$ (where $0 \leq \ell \leq k$) are the labels of the vertices in e shared with the tile corresponding to v's parent (where $\ell = 0$ iff $v = root(T)$). Then,*

1. *If $v = root(T)$, then $t(v) = t(root(T)) \geq 0$; otherwise,*
2. *If $v \in P$, then $t(v) \geq \min(\alpha_i, \sum_{r=1}^{\ell} \alpha_{j_r})$;*
3. *If instead $v \in N$, then $t(v) \geq \min(0, 1 - \alpha_i - \sum_{r=1}^{\ell} \alpha_{j_r})$.*

Proof. The proof is by induction on the size of the subtree at v. For the base case, let v be a leaf. Hence, the vertices labelled by $Y_{ij_{\ell+1}}, \ldots, Y_{ij_k}$ in e are all unmatched non-distant pair vertices (since H contains no unmatched distant pair vertices). Suppose $v \in P$. By equation (4.33),

$$t(v) = \alpha_i + \sum_{r=1}^{k} \alpha_r - 1 + \sum_{r=\ell+1}^{k} \delta(i, j_r). \tag{4.35}$$

There are two subcases to consider. In the first subcase, $j_r \neq i$ for all $r = \ell+1, \ldots, k$. Then $\delta(i, j_r) = 1 - \alpha_i - \alpha_r$ for all $r = \ell + 1, \ldots, k$ and it follows that $t(v) = \sum_{r=1}^{\ell} \alpha_r + (k - \ell - 1)(1 - \alpha_i) \geq \sum_{r=1}^{\ell} \alpha_r$. In the second subcase, $j_r = i$ for some r, say $r = k$. This can only happen for one r since H only contains one diagonal vertex, namely Y_{11}. In particular, v must be $root(T)$. Then $\delta(i, j_k) = -\alpha_i$ and $\delta(i, j_r) = 1 - \alpha_i - \alpha_r$ for all $r = \ell + 1, \ldots, k - 1$. Since α satisfies the edge constraints for G, it follows that $t(v) = t(root(T)) \geq 0$.

Suppose instead that $v \in N$. By equation (4.34), $t(v) = -\alpha_i + \sum_{r=\ell+1}^{k} \beta(i, j_r)$. Now, $\beta(i, j_r) \geq \min(\alpha_i, \alpha_{j_r})$ for all $r = \ell + 1, \ldots, k$. If $\beta(i, j_r) = \alpha_i$ for some j_r, then $t(v) \geq 0$ as desired. So assume $\beta(i, j_r) = \alpha_{j_r}$ for all $r = \ell + 1, \ldots, k$. Since α satisfies the edge constraints for G, $\sum_{r=1}^{k} \alpha_{j_r} \geq 1$. But then, $t(v) \geq 1 - \alpha_i - \sum_{r=1}^{\ell} \alpha_{j_r}$, completing the proof for the base case.

For the inductive step, let v be any node in T and assume the lemma holds for all children of v. Assume first that $v \in P$ and hence, all children of v are in N (by definition of a tiling). By the induction hypothesis, there are two possibilities: either (a) $t(v') \geq 0$ for all children v' of v, and there are no unmatched vertices in $Tile(v)$, or (b) there exists some unmatched vertex in $Tile(v)$ or $t(v') \geq 1 - \alpha_i - \sum_{s=1}^{t} \alpha_{j_{r_s}}$ for some child v' of v. In case (a), it follows from (4.33) and from from the fact that α satisfies the edge constraints for G that $t(v) \geq \alpha_i$. In case (b), using the arguments from the base case, it follows that $t(v) \geq 0$ when $v = root(T)$, and $t(v) \geq \sum_{r=1}^{\ell} \alpha_{j_r}$ when $v \neq root(T)$.

Now assume that $v \in N$. The arguments from the above case when $v \in P$, as well as the arguments from the base case, can now be adapted to show that $t(v) \geq \min(0, 1 - \alpha_i - \sum_{r=1}^{\ell} \alpha_{j_r})$. Note that in in this case $v \neq root(T)$ since $root(T) \in P$. The inductive step, and hence the lemma, now follow. □

So $S \geq 0$ in Case 3 also, and the Inductive Claim now follows for $m + 1$.

4.3 Discussion

The integrality gap of $2 - \epsilon$ for graph VERTEX COVER holds for $\Omega(\log n)$ rounds. We conjecture that our integrality gaps in the hypergraph case should also hold for at least $\Omega(\log n)$ rounds. Indeed, examining the proof of Theorem 4.15, it can be shown that by redefining the recursive definition of R_m to be $R_m = R_{m+1} + c/\gamma$ (for some constant $c > 0$), then all cases considered in the proof except Subcase 1 of Case 3 can be argued for $\Omega(\log n)$ rounds. While it can be argued that $S \geq 0$ for $\Omega(\sqrt{\log n})$ rounds for graphs in this subcase (and in fact for $\Omega(\log n)$ rounds we define (S, R, γ)-vectors as in Definition 4.4 in Section 4.1.2), a proof for hypergraphs eludes us.

The integrality gaps of $k - 1 - \epsilon$ given in Chapter 3 for k-uniform hypergraph VERTEX COVER held not only for LS but also for LS_+ liftings, the stronger semidefinite version of Lovász-Schrijver liftings. Obtaining optimal integrality gaps for both graph and hypergraph VERTEX COVER for LS_+ remains a difficult open question.

Finally, can our techniques be applied to other problems? For instance, could our techniques be used to show that, say, even after $\log \log n$ rounds of LS the integrality gap of the linear relaxation for MAX-CUT remains larger than the approximation factor attained by the celebrated SDP-based Goemans-Williamson algorithm [27]?

Chapter 5

The "Fence" method

As discussed in Chapter 4, proving lower bounds even in the weaker LS_0 and LS hierarchies for problems defined by 2-variable constraints (such as VERTEX COVER) has proved very difficult. This contrasts with problems defined by 3 (or more) variable constraints (e.g., MAX-3SAT, hypergraph VERTEX COVER) where we were able in Chapter 3 to prove strong inapproximability results (in terms of the number of rounds for which the lower bounds hold) for even $\Omega(n)$ rounds of LS_+.

The techniques used in Chapter 4 for obtaining lower bounds showed that the integrality gap of relaxations for VERTEX COVER produced after even $\Omega(\log n)$ rounds of LS tightenings is $2 - o(1)$. Unfortunately, these techniques can only prove lower bounds when the number of LS rounds is at most the girth of the input graph, which is $O(\log n)$ for graphs with large integrality gaps.

In this chapter, we show how to break through this "girth barrier" and obtain integrality gaps for VERTEX COVER after even $\Omega(g^2)$ LS rounds for graphs of girth g. Consequently we show that VERTEX COVER relaxations produced after $\Omega(\log^2 n)$ rounds of LS have integrality gaps of size $1.5 - \epsilon$ for any $\epsilon > 0$ (Theorem 5.3). While less than $2 - o(1)$, our integrality gap is still larger than the approximability factor of 1.36 ruled out using PCP-based techniques [16]. As we discuss further in Section 5.4, we conjecture that our techniques may yet yield integrality gaps after even linear rounds of LS. The work in this chapter was published in [52].

5.0.1 Comparison with related work

While the results in Chapter 4 are those most easily comparable to our main result, the techniques used in Buresh-Oppenheim et al. [11] and in Chapter 3 are somewhat more related to those used here (this will be discussed further in the next section).

5.1 Methodology

Given a protection matrix Y, call the set $V(Y) = \{Ye_i, Y(e_0 - e_i) : 1 \leq i \leq n\}$ the set of *protection vectors corresponding to* Y. The points x we protect will always have $x_0 = 1$. For such points, the set of *projected protection vectors corresponding to* Y is

$$PV(Y) = \{Ye_i/x_i, Y(e_0 - e_i)/(1 - x_i) : 1 \leq i \leq n, 0 < x_i < 1\}.$$

Note that these are simply the vectors from Corollary 2.2. Note moreover that $y_0 = 1$ for all vectors y in $PV(Y)$.

Lemma 2.1 (and Corollary 2.2) suggests using inductive arguments to show that some vector w is in $N^r(P)$. As was done in Chapter 3 (and first suggested in [11]) such an argument can be phrased as a Prover-Adversary game: The game maintains a vector x, initially w, and proceeds in rounds. Each round the following moves are made:

1. Given the current value for x, the Prover produces a candidate protection matrix Y_x supposedly showing that $x \in N(P)$.
2. The Adversary picks one vector y from $PV(Y_x)$ and sets x to y.

The game ends when x is no longer in P, i.e., when the Adversary forces the Prover into constructing an invalid candidate protection matrix. The following lemma which is similar to Lemma 3.11 follows immediately from Lemma 2.1 and the definition of the Prover-Adversary game, and was proved in [11]:

Lemma 5.1. *If there exists a strategy for the Prover such that the game lasts r rounds no matter what the Adversary does, then $w \in N^r(P)$.*

If $x \in conv(S)$ and $S \subseteq N^r(P)$, it follows also that $x \in N^r(P)$. We can use this observation to modify the game rules as follows:

1. Given the current point x, the Prover produces a candidate protection matrix Y_x. The Prover then produces a set S_x of points such that $PV(Y_x) \subseteq conv(S_x)$.
2. The Adversary picks one vector y from S_x and sets x to y.

By Lemma 5.1, if there exists a strategy for the Prover in this new game such that the game lasts r rounds no matter what the Adversary does, then $w \in N^r(P)$. The intuition for introducing the rules of the revised game is that the vectors in S_x may have nicer structural properties than the vectors in $PV(Y_x)$ facilitating the Prover's strategy in future rounds of the game. Indeed, expressing the vectors in $PV(Y_x)$ using convex combination will be crucial for our "fence" method sketched in Section 5.1.1 below and described formally in Section 5.3.2.3.

In practice, when trying to show $w \in N^r(P)$, we will pick w so that it enjoys some nice structural properties. Hence, in our lower bound, the Prover will always construct S_x so that the difference between any vector in S_x and the current x is both minimal and predictable.

Remark 5.2. If a coordinate is set to 0 or 1 in the game, then that coordinate will remain 0 or 1, respectively, for the remainder of the game: This follows from the definition of $PV(Y_x)$ and from the fact that the $PV(Y_x) \subseteq conv(S_x)$.

We would like to mention now one way in which our approach will more resemble that taken in Chapter 3 (and also in [11]) rather than that in Chapter 4. In Chapter 4 LP-duality was used to prove the existence of appropriate protection matrices needed for their lower bound; no explicit description for their protection matrices was obtained. While our protection matrices will be completely different than those used in Chapter 3 (and by [11]), as in the results in that chapter we will nevertheless always give explicit descriptions for them. This is crucial since our arguments will require explicit descriptions for the sets $PV(Y_x)$.

5.1.1 The "Fence" trick

The key to our new lower bound is what we call the "Fence" method which we now roughly sketch. A more technical description will be given in Section 5.3.2.3.

As in the results in Chapter 4, our lower bound will be proved for a graph G with girth $\Theta(\log n)$. To prove that a large integrality gap remains after $\Omega(\log^2 n)$ rounds of LS tightenings we start the Prover-Adversary game with a "bad" fractional solution vector w (i.e., the value of the objective function at w is far from the true integral optimum) and show that there is an $\Omega(\log^2 n)$ round strategy for the Prover against any Adversary. The vector w will be chosen so that it satisfies some "nice" structural properties.

In each round of the game, given the current vector x the Prover's strategy will be to design S_x (the set of vectors from which the Adversary chooses x for the next round) such that the difference between x and any vector in S_x is minimal. For technical reasons, vectors in S_x will always differ in at least a few coordinates from x. Hence, as more and more rounds of the game are played, the current vector x will differ more and more from the initial vector w.

Let C_x be the induced subgraph of G on those vertices (i.e., coordinates) that x differs from w. For technical reasons, our Prover strategy will always be successful against the Adversary *provided that C_x has no component with diameter greater than half the girth of G*.

Now, it is not too hard to tailor the Prover's strategy so that in the ith round of the game the sum of the diameters of all components in C_x is at most $O(i)$ (for instance, by adapting the arguments in Chapter 4). So since G has girth $\Theta(\log n)$, the Prover can use such a strategy to play $\Omega(\log n)$ rounds of the game against any Adversary. However, this strategy will fail beyond $\Omega(\log n)$ rounds since C_x may then contain a component with diameter greater than half the girth.

To continue the game, the Prover then uses the "fence trick": If some vector y in S_x (the set from which the Adversary chooses x for the next round of the game) is such that C_y has a component A with diameter nearly half the girth, then the Prover will put a "fence" around this component to stop it from growing any larger and becoming "dangerous". The Prover does this by taking advantage of Remark 5.2 which implies that during the game we can ignore all nodes in G (i.e., remove the respective coordinates from x) that are set to 1 by x (this is made formal in Section 5.3.1). To put a "fence" around A, the Prover expresses y as a convex combination of vectors each of which sets some nodes surrounding A to 1, disconnecting it from the rest of G.

5.2 The main theorem

We prove our main result in this section. Given a graph G, let $\mathrm{VC}(G) \subseteq \mathcal{R}^{n+1}$ denote the convex cone of feasible solutions to the homogenized (relaxed) VERTEX COVER constraints for G.

Theorem 5.3. *For all $\epsilon > 0$ there exists a constant $\delta > 0$ and an integer n_0 such that for all $n \geq n_0$ there exists an n-vertex graph G for which $N^r(\mathrm{VC}(G))$ has an integrality gap of at least $1.5 - \epsilon$ for all $r \leq \delta \log^2 n$.*

The graphs used to prove Theorem 5.3 are high-girth sparse graphs with degree bounded by some constant $d \geq 3$. With non-zero probability, such graphs have a maximum independent set of size $O(\frac{n \log d}{d})$. In particular, we have the following theorem from standard graph theory (it is proved along similar lines to Theorem 4.2):

Theorem 5.4. *There exist constants $\alpha, \beta > 0$ and integers n_0, d_0 such that for all $n \geq n_0$ and all $d \geq d_0$, there exists a graph $G(n, d)$ with n vertices, degree at most d, girth at least $\beta \log n$, and maximum independent set size at most $\frac{\alpha n \log d}{d}$.*

The graphs given by the above theorem will be used to prove the following theorem from which Theorem 5.3 will follow.

Theorem 5.5. *Let $w = (1, \frac{2}{3}, \frac{2}{3}, \ldots, \frac{2}{3}) \in \mathcal{R}^{n+1}$. Let n_0, d_0 be the constants given by Theorem 5.4. Then for all $n \geq n_0$ and all $d \geq d_0$, the point w is in $N^r(\mathrm{VC}(G(n, d)))$ for $r = \Omega(\log^2 n)$.*

Section 5.3 is devoted to proving Theorem 5.5.

Proof of Theorem 5.3. Theorem 5.5 shows that for large n and d there exist graphs G for which the integrality gap of $N^r(\mathrm{VC}(G))$ for $r = \Omega(\log^2 n)$ is $\frac{3}{2}(1 - \frac{\alpha \log d}{d})$. The latter quantity can be made arbitrarily close to 1.5 by taking d sufficiently large. ☐

5.3 Proof of Theorem 5.5

Fix $d \geq d_0$ and $n \geq n_0$, and let $G = G(n, d)$. Let g denote the girth of G. Note that $g \geq \beta \log n$. For the proof we will first make some important definitions. Then we will describe the Prover's strategy for the vector w and graph G. Concurrently, we will show that the described strategy works for r rounds against any adversary, where $r = (g/28)^2/2$.

5.3.1 Round invariants and components

We will define three properties/invariants that the Prover will ensure the current vector x for the game satisfies at the beginning of each round. The first is the following:

Property 1: $x \in \mathrm{VC}(G)$ and $x \in \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ (of course, $x_0 = 1$ always).

This invariant allows us to make several crucial definitions. First we make some observations.

Note that the constraints for any edge in G incident to a vertex i where $x_i = 1$ are trivially satisfied. Moreover, since $x \in \mathrm{VC}(G)$, if $x_i = 0$ then x_j must be 1 for all vertices j adjacent to i. Hence, since vertices that are 0/1 valued will never change their values in subsequent rounds (see Remark 5.2), when analyzing the effect of one round of N it will suffice to only consider the subgraph G_x of G induced by those vertices with value in $\left\{\frac{1}{3}, \frac{2}{3}\right\}$ under x. We will say that a *vertex j has value a in G_x* if $x_j = a$.

Next we define the concept of a *simple component* in G_x. Intuitively, a simple component in G_x is any connected component in G_x such that all edges with both vertices in the component have one vertex with value $\frac{1}{3}$ and the other with value $\frac{2}{3}$. We now make this precise.

Given $x \in \mathrm{VC}(G)$, let G'_x be the subgraph of G_x induced by all edges (i, j) in G_x such that one of x_i or x_j is $\frac{1}{3}$ (note that since $x \in \mathrm{VC}(G)$, at most one vertex in each edge has value $\frac{1}{3}$).

Definition 5.6. A (vertex induced) subgraph C of G_x is called a *simple component* if it is a maximal connected component of G'_x (i.e., adding any vertex of G'_x to C results in an unconnected (vertex induced) subgraph of G'_x).

We will ensure (see Property 2 below) throughout the Prover-Adversary game that all simple components in G_x have diameter much smaller than half the girth of G. Hence, adjacent vertices in a given simple component cannot both have value $\frac{2}{3}$ under x. In particular, this ensures that the following definition is consistent:

Definition 5.7. A node i belongs to the *boundary* of a simple component C in G_x if (a) $i \in C$, (b) $x_i = \frac{2}{3}$, and (c) there exists j such that $x_j = \frac{2}{3}$ and $(i, j) \in E$. (Note that since C has diameter less than $g/2$, j cannot be in C.)

The *edge distance* $d_E(C, C')$ *in* G_x between two simple components C and C' is the length (i.e., number of edges) of the shortest path in G_x connecting some boundary node of C with some boundary node of C'. The *distance* $d(C, C')$ between the two simple components is equal to $d_E(C, C') - 1$. If no paths exists between the components, then $d_E(C, C')$ and $d(C, C')$ are defined to be infinite.

Consider the following procedure: Let D be a set whose items will be simple components of G_x, and suppose D initially contains only one simple component C. Repeat the following procedure until D no longer grows in size: For every simple component C' in D, add to D all simple components C'' that are within distance 2 of C' in G_x. The final set from this procedure is called the *closure of* C.

Definition 5.8. A vertex induced subgraph C of G_x is a *complete component* if there exists a simple component C' in G_x such that the vertices of C are precisely the vertices of all simple components in the closure of C'.

Intuitively, a complete component is formed by adding to a simple component all simple components that are "near" it, and then adding all simple components that are "near" the resulting component, and so on.

Definition 5.9. A node in G_x is *untouched* if it belongs to no complete component. Otherwise, the node is *touched*. Note that untouched nodes always have value $\frac{2}{3}$.

Distance between two complete components is defined in the same way it was defined for simple components. Note that by definition the distance between two components is at least 3. This fact is important, so we make special note of it:

> **Observation 1:** The distance between two complete components is at least 3. Hence, along any path connecting two complete components in G_x there are at least 3 consecutive untouched nodes.

The second invariant the Prover will ensure is that at the beginning of each round all complete (and hence all simple) components in G_x have diameter substantially smaller than g. In particular, it will ensure that the following property holds for x at the beginning of each round:

> **Property 2:** No complete component in G_x has diameter greater than γ, where $\gamma = g/28$.

If at some point in the game some complete component's diameter gets too large, then the Prover will "remove" the complete component from G_x. Intuitively, this will be done by altering x so that the values of nodes immediately around the too-large component are $0/1$. Hence the complete component will be "fenced off" from the rest of the graph and can be ignored for the rest of the game. The details of how this is done are in Section 5.3.2.3.

Finally, the following Property will also be ensured:

> **Property 3:** At the beginning of round i the sum of the diameters of all complete components in G_x is at most $7i$.

In the next section we describe the Prover's strategy in round $i \leq r = \gamma^2/2 = \Omega(\log^2 n)$. While describing the strategy we will prove that if $i < r$, then the Prover's strategy will guarantee that Properties 1–3 will hold at the start of the following round. Hence, Theorem 5.3 will follow.

5.3.2 The Prover's strategy for x in round i

"High-level" description

The Prover's strategy can be described as follows: By induction, the Prover can assume that the Properties 1–3 hold for x at the start of the round; they of course hold for the base case vector $x = w$. The Prover will then construct the "obvious" protection matrix Y_x for x where "obvious" will be made precise below; the Round Invariants will be crucial for this (Section 5.3.2.1). The set $PV(Y_x)$ may not be contained in $x \in \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ (in particular, some vectors may have entries that are $\frac{1}{2}$), i.e., these vectors may not satisfy Property 1. The Prover will then construct a new set $S'_x \subseteq \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ of vectors such that $PV(Y_x) \subseteq conv(S'_x)$ (Section 5.3.2.2). The vectors in S'_x may not satisfy Property 2. The "fence" trick will be used by the prover to construct a new set S_x satisfying all the invariants and such that $S'_x \subseteq conv(S_x)$ (Section 5.3.2.3). The details now follow.

5.3.2.1 The "obvious" protection matrix Y_x for x

To simplify notation, we will write Y for Y_x. Since Y is supposed to be symmetric, we will often use the notation Y_{ij} instead of $Y_{i,j}$ or $Y_{j,i}$.

The protection matrix Y for x will depend on the structure and properties enjoyed by complete components in G_x. In particular, we will use the fact that Property 2 implies all complete (and simple) components are trees. Moreover, we will also use Observation 1 which says that there is a "buffer" between complete components. This "buffer" will ensure that entries of Y corresponding to one complete component are independent of those for other complete components. In particular, whenever two vertices i and j are not in the same complete component this will generally allow us to set $Y_{ij} = x_i x_j$. Only for i, j in the same complete component (and some isolated other cases), will there be a "non-trivial" value for Y_{ij}. The formal definition for Y now follows.

We will define Y by defining Ye_i for all i, $1 \leq i \leq n$, and then arguing that our definition is symmetrical. Fix a node i in G. If $x_i = 0$, then Ye_i is simply $\vec{0}$. If $x_i = 1$, then $Ye_i = x$. So assume $0 < x_i < 1$. As required by the definition of a protection matrix, we have that $Y_{ii} = x_i$. There are now two cases depending on whether i is contained in a complete component or not.

Case 1: i is not in a complete component

Consider i's neighbours. These may or may not belong to complete components (in particular, they may or may not lie on the boundary of some simple component). Note that it follows from Observation 1 that at most one neighbour of i belongs to a complete component.

We define Ye_i as follows: for each neighbour i_j of i let $Y_{i_j,i} = \frac{1}{3}$. For all remaining vertices $\ell \in G$, we set $Y_{\ell,i} = x_\ell x_i$. If there is a neighbour, say i_1, that belongs to a complete component, we must make some adjustments to Ye_i: Let C' be the simple component in which i_j belongs, and set $Y_{k,i} = \frac{1}{3}$ for all $k \in C'$.

Case 2: i is in some complete component C

There are two subcases depending on whether vertex i is in a simple component or not.

First consider the case that i is in some simple component C'. Either $x_i = \frac{1}{3}$, or $x_i = \frac{2}{3}$. Suppose $x_i = \frac{1}{3}$. Then for all vertices $k \in C'$, $Y_{k,i} = \frac{1}{3}$ if $x_k = \frac{1}{3}$ (i.e., since C' is a tree, k has even distance from i in C'), while $Y_{k,i} = 0$ otherwise (i.e., if the distance is odd). For all simple components D in C distance 0 from C' in G_x, and for all nodes $k \in D$, let $Y_{k,i} = \frac{1}{3}$ if $x_k = \frac{2}{3}$, and let $Y_{k,i} = 0$ otherwise. For all nodes $k \in G_x$ that do not belong to any complete component and such that k is adjacent to a boundary node of C', let $Y_{k,i} = \frac{1}{3}$. Finally, for all remaining nodes $\ell \in G$, set $Y_{\ell,i} = x_\ell x_i$.

Now suppose instead that $x_i = \frac{2}{3}$. Then for all vertices $k \in C'$, $Y_{k,i} = \frac{2}{3}$ if $x_k = \frac{2}{3}$, while $Y_{k,i} = 0$ otherwise. For all simple components D in C distance 0 from C' in G_x, $Y_{k,i} = \frac{1}{3}$ for all nodes $k \in D$. For all nodes $k \in G$ that do not belong to any component and such that k is adjacent to a boundary node of C', we have that $Y_{k,i} = \frac{1}{3}$. Finally, for all remaining nodes $\ell \in G$, we set $Y_{\ell,i} = x_\ell x_i$.

Now we consider the case that i is not in any simple component. We must then have $x_i = \frac{2}{3}$. Moreover, every vertex adjacent to i must also have value $\frac{2}{3}$ in G_x. We define Ye_i as follows. For each vertex k adjacent to i we set $Y_{k,i} = frac13$. In addition, for each simple component D in C adjacent to i (note that there must be at least one such simple component), we set $Y_{k,i} = \frac{1}{3}$ for all vertices $k \in D$. Finally, for all remaining nodes $\ell \in G$, we set $Y_{\ell,i} = x_\ell x_i$.

It remains now to argue (1) that the definition of Y is indeed symmetric, and (2) that the protection vectors defined by Y are in $\mathrm{VC}(G)$.

Symmetry follows straightforwardly from the definition and a complete proof will not be given here. We will only note as an illustrative example how when i, j are both in the same simple component, then the definition of Y_{ij} always depends only on whether the distance between i and j is odd or even. Moreover, this is well-defined since all simple components are trees.

Now let us argue that the vectors in $PV(Y)$ are in VC(G). Fix i such that $0 < i < 1$ and consider $y = Ye_i/x_i$. Note that y is identical to x in all coordinates except:

1. Coordinate i which has $y_i = 1$,

2. If i was not in a simple component of G_x, then all (non 0/1) neighbours of i in G are now $\frac{1}{2}$, and all nodes in simple components of G_x that touched i are also $\frac{1}{2}$.

3. If i was in a simple component C', then all nodes at odd distance from i in C' are now 0 under y; and all nodes at even distance from i in C' are now 1 under y. Moreover, if $x_i = \frac{2}{3}$, then all nodes in those simple components (in G_x) that touched C' are now $\frac{1}{2}$. Free nodes touching C' are also set to $\frac{1}{2}$ under y. Finally, if instead $x_i = \frac{1}{3}$, then all nodes in a simple components D (in G_x) that touched C' are 1 under y if the node's distance (through D) to C' is even, and 0 under y if the distance to C' is odd. Again, in this case ($x_i = \frac{1}{3}$), free nodes touching C' are now set to 1 under y.

Since $x \in$ VC(G) it follows that the VERTEX COVER constraints are satisfied by all edges whose nodes have the same value under y as under x. So let's concentrate on those nodes which changed as described above. Clearly all edges in those simple components whose values are affected as described in (2) and (3) above still satisfy the VERTEX COVER constraints under y. Moreover, using the fact there is a "buffer" between complete components (Observation 1), it follows from the definition of Y that the edges between affected components and unaffected nodes also satisfy the VERTEX COVER constraints. So y satisfies the VERTEX COVER constraints.

To show that $Y(e_0 - e_i)/(1 - x_i)$ is in VC(G) uses similar arguments.

5.3.2.2 Constructing $S'_x \subseteq \left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$ such that $PV(Y_x) \subseteq conv(S'_x)$

The vectors $PV(Y_x)$ arising from Y_x may not be in $\left\{0, 1, \frac{1}{3}, \frac{2}{3}\right\}^{n+1}$. Indeed they may contain $\frac{1}{2}$'s. However, since $PV(Y_x) \subseteq$ VC(G), it follows that for any vector $y \in PV(Y_x)$, the following is true: In the graph G_y, all nodes j with $y_j = \frac{1}{2}$ are adjacent to nodes whose values are either $\frac{1}{2}$ or $\frac{2}{3}$. Hence it is easy to see then that for each vector $y \in PV(Y_x)$ there exist vectors y^1 and y^2 such that y is the average of y^1 and y^2, and such that y^1 and y^2 are equal to y everywhere with the exception that for every node k such that $y_k = \frac{1}{2}$, then k is $\frac{1}{3}$ in one of y^1 or y^2, and $\frac{2}{3}$ in the other.

We then define S'_x to be the set containing precisely the vectors y^1 and y^2 corresponding to each vector $y \in PV(Y_x)$.

5.3.2.3 The "Fence" trick: Constructing S_x

The vectors in S'_x may not satisfy Property 2. To fix this, for each vector $y \in S'_x$ that does not satisfy Property 2, the Prover will construct a set B_y of vectors that will satisfy all three properties in the next round and moreover, $y \in conv(B_y)$. Essentially, the vectors in B_y will isolate all components that do not satisfy Property 2 by ensuring such vectors have a "fence" of 1's around such components. The set S_x will then be $\cup_{y \in S'_x} B_y$ (where $B_y = \{y\}$ if y does satisfy Property 2).

Before we describe how the Prover finds these "fences", let us consider how the graph G_y for some $y \in S'_x$ compares to G_x. Essentially, the only possible differences are:

1. Some free node is set to 0/1 which results in (a) either a new simple component of diameter 3 being created, or (b) in some previously existing simple component having its diameter increased by at most 3.

2. Some free node is set to 0/1 which results in at most d complete components being "merged". The resulting complete component will have diameter bounded by at most $3 + \gamma_1 + \gamma_2$ where γ_1 and γ_2 are the diameters of the two largest component involved in the merge.

3. Some node in a simple component C' is set to 0/1 which results in each adjacent simple components having its vertices either (a) set to 0/1 or (b) the pattern of $\frac{1}{3}$-$\frac{2}{3}$ for the values of the nodes in the component is reversed (i.e., nodes that were $\frac{1}{3}$ are now $\frac{2}{3}$ and vice versa). In addition, free nodes adjacent to the affected simple components may be altered. Let C be the complete component containing C'. In both cases (a) and (b), C may have some formerly untouched nodes added to it. In either case, the diameter of C increases by at most 2. This may result in C being closer than distance 3 to other complete components in which case these components merge to form a new complete component. Again, it is not hard to see that this new complete component will have diameter bounded by at most $6 + \gamma + \gamma_1 + \gamma_2$ where γ is the diameter of the complete component containing C' and γ_1 and γ_2 are the diameter of the two largest other component involved in the merge.

Note that in all cases, the sum of the diameters of all complete components in G_y increases by at most 6. Moreover, note that any complete component in G_y with diameter greater than γ will still have diameter bounded by 3γ.

We can now give a high-level description of the algorithm the Prover will use when putting up "fences" for a vector $y \in S'_x$ that violates Property 2. First the Prover will "group" all complete components that are "near" to each other in an effort to divide all complete components into sets of "super-components" of diameter at least $\gamma/2$ and at most 7γ. Note that any super-components that cannot be "grown" to a super-component of this diameter must be "far" from all other super-components; the Prover will ignore them. Super-components of diameter at least $\gamma/2$ are called *full-size*. The set of full-size components will be denoted by $\mathcal{C}$.

The Prover will then "isolate" all super-components by defining a set A of vectors such that for each super-component C (a) every path in G_y of length 2 adjacent to a boundary node of C has at least one vertex with value 1 in all vectors in S_1 and (b) $y \in conv(A)$. Hence, for all $z \in A$, in the graph G_z (a) each super-component is disconnected from the rest of the graph and (b) no complete component has diameter greater than $\gamma/2$. This isolation step is where the "fences" are put up.

Finally we will argue that this means that there exists yet another set B_y of vectors that are 0/1 on all super-components, will satisfy Properties 1–3 in the next round, and such that $A \subseteq conv(B_y)$. Thus the Prover effectively "removes" all super-components from the graph, and in particular, removes all complete components of diameter greater than γ from the graph.

We now describe formally how the Prover does "grouping", "isolating", and "removing". The arguments showing that these procedures produce a final set B_y whose vectors all satisfy Properties 1–3—and hence allow the Prover to successfully play at least one more round against the Adversary—will crucially rely on the fact that $i < r = \gamma^2/2$.

Grouping:

The two rules for grouping complete components into super-components are as follows: (1) Two complete components can be put in the same super-component if the distance between them is at most 8; and (2) The diameter of a super-component cannot exceed 7γ. Note that all complete components in G_y have diameter at most 3γ. Moreover, by definition all complete components not in $\mathcal{C}$ have diameter at most γ.

Do an initial grouping of all components as follows: Let the first group g_1 initially contain some complete component C of G_y. Now keep adding to g_1 any complete component that is within distance 8 of some complete component already in g_1. Do this until no more such complete components can be found. For the second group g_2, let it initially contain some complete component C' not in g_1. Again add to g_2 all components that are within distance 8 of all components already in g_2. Groups g_3, g_4, and so on, are then formed in the same way until all complete components are in some group.

Note that the distance between two groups is at least 9. Assume there are ℓ groups, and assume without loss of generality that the first k groups have diameter less than $\gamma/2$. These super-components will be called *small*. Focus attention on groups $g_{k+1}, \ldots, g_\ell$. Note that some of these groups may have diameter greater than 7γ. However, since no complete component has diameter greater than 3γ, there is a partition of these groups into new groups $g'_1, \ldots, g'_{k'}$ such that each of these groups has diameter between $\gamma/2$ and 6γ.

Groups $g_1, \ldots, g_k$ and $g'_1, \ldots, g'_{k'}$ define the super-components the Prover will use. Note that the small super-components given by $g_1, \ldots, g_k$ each have distance at least 9 from any other super-component. The super-components $g'_1, \ldots, g'_{k'}$ are the *full-size components*. Note that by Property 3, there can be at most $r/(\gamma/2) \leq \gamma$ full-size components. Note also that this is the only part of the proof that will use the fact that $r \leq \gamma^2/2$.

It is not hard to see that when the partitioning is done to form the full-size components, it can be done so that the following property holds: For each g'_i, there exists a spanning tree for the super-component g'_i (where a spanning trees for a super-components is defined in the obvious way) such that the distance between any two of these spanning trees is at least 3 (this can be argued by appealing to Observation 1). Fix such a spanning tree for each full-size component.

Isolating: We will show how to handle the case where no two full-size components have distance exactly 4 (i.e., distance is either 3 or at least 5); we leave out the case where some full-size components have distance 4 which involves similar reasoning but requires a somewhat more technical case-by-case analysis.

Define a node to be *on the boundary of a full-size component* if it is a boundary node of a simple component contained in the full-size component. We will now define two fractional vertex covers y^1 and y^2 such that $y = \frac{1}{3}y^1 + \frac{2}{3}y^2$. For every point i that is a boundary node of some simple component C' in a full-size component C we do the following: For every path (i, j_1, j_2) of length 2 (i.e., 2 edges) starting from i such that neither of j_1 or j_2 are in the spanning tree for C (such a path is called a *2-path coming out of a boundary node*) we will have the following assignments in y^1 and y^2: Either (a) the values of j_1 and j_2 are 1 and 0, respectively, in y^1 and $\frac{1}{2}$ and 1, respectively in y^2, or (b) the values of j_1 and j_2 are $\frac{2}{3}$ and 1, respectively, in y^1 and $\frac{2}{3}$ and $\frac{1}{2}$, respectively in y^2. Thus every path of length 2 out of a boundary node has two possible assignments in y^1 and y^2. For all remaining coordinates, have y^1 and y^2 be identical to y.

Lemma 5.10. *There exists a consistent way to decide what type of assignment to give to all 2-paths coming out of boundary nodes in full-sized components.*

Proof. We make the following observation that will be crucial below: Since the diameter of a super-component is less than 7γ, no two 2-paths from a single component are adjacent.

Suppose no such consistent assignment existed. In particular, suppose that for every possible assignment there exists some boundary node i in a full-size component

C and two 2-paths $p_1 = (i, j_1, j_2)$ and $p_2 = (i, j_1, j_3)$ coming out of i such that the following holds: If p_1 has an assignment of type (a), then p_2 must have an assignment of type (b), and vice versa.

For this causal relationship to hold, the following must be the case: If p_1 has an assignment of type (a), it forces the assignment of some adjacent 2-path (i', j_1', j_2') where i' is a boundary node in some other full-size component (Note: it cannot be from C by the above observation). In turn this forces the assignment of some 2-path adjacent to the previous 2-path coming out of yet another full-size component (it must be different by girth considerations). In turn this forces the assignment of some 2-path adjacent to the previous 2-path coming out of yet another full-size component, different from all full-size components involved so far. This chain of dependencies continues in this way. However, this chain can only continue for $k' \leq \gamma = g/28$ more steps (where k' is the number of full-size components) since after that there are no more full-size components to continue the chain. But then, this causal chain cannot reach p_2, contradicting the fact that p_1's assignment type influences p_2's assignment. □

It follows that y^1 and y^2 can be consistently defined.

Note that each full-size components has no edge to the rest of the graph in G_{y^1}; hence all full-size components are isolated in G_{y^1}. However, this is not necessarily true in G_{y^2}. To fix this we will define four vectors y^3, y^4, y^5 and y^6 such that y^2 is in the convex hull of these vectors and such that the full-size components are isolated the corresponding graphs for these vectors. These vectors are defined as follows: For every 2-path (i, j_1, j_2) with assignment of type (a), j_1 and j_2 are $\frac{1}{3}$ and 1, resp., in y^3, and are 1 and 1, resp., in y^4; for every 2-path (i, j_1, j_2) with assignment of type (a), j_1 and j_2 are $\frac{1}{3}$ and 1, resp., in y^5, and are 1 and 0, resp., in y^6. It can be verified that these vectors have the required properties and moreover, satisfy the VERTEX COVER constraints. Let $A = \{y^1, y^3, y^4, y^5, y^6\}$. By construction, $y \in conv(A)$.

It is straightforward to verify that the vectors in A satisfy Properties 1 and 3 for the next round.

Removing: As noted above, all components with diameter larger than $\gamma/2$ are disconnected in the graphs corresponding to the vectors in A. Fix a vector $z \in A$ and consider the subgraph G_C of G_z induced by some isolated component C. Since this isolated component has diameter less than the the girth g of G, it follows that C is a tree. In particular, C has an independent set of size $|C|/2$. It follows that z_C (i.e., z restricted to those coordinates indexed by nodes in C) is in the integral hull of the VERTEX COVER polytope for G_C. Indeed, this is true for all vectors in A and all isolated components.

But then, there exists a set Z of vectors which are (a) 0-1 on the subgraphs of the isolated components, (b) identical to some vector in A outside those components, and (c) $A \subseteq conv(Z)$. The Prover then lets $B_y = Z$.

5.4 Discussion

We feel that our methods should extend to proving that the integrality gap for VERTEX COVER relaxations is in fact $2 - o(1)$ after $\Omega(\log^2 n)$ rounds of LS tightening. Indeed, the use of the all-$\frac{2}{3}$ vector to prove our integrality gap of $1.5 - o(1)$ was motivated by our desire to keep our protection matrices and hence our girth correction strategies as simple as possible. An analogous proof using the all-$(\frac{1}{2} + \gamma)$ vector for some small $\gamma > 0$ (and hence yielding an integrality gap of $2/(1 + 2\gamma) - o(1)$) may also be possible; however, coming up with a strategy for putting up "fences" will become much more complicated and likely be very difficult to analyze.

We also conjecture that a variation of our "fence" method might be able to yield integrality gaps for up to $\Omega(n^{1-\delta})$ rounds of LS for some $\delta > 0$. The current proof fails after $O(\log^2 n)$ rounds since Lemma 5.10 only works if there are $\Omega(\log n)$ super-components; this is not true after $\log^2 n$ rounds using the current Prover strategy. A different argument may be able to get around this deficiency. Note that under Khot's Unique Games conjecture [36], VERTEX COVER has no $2-o(1)$ approximations [39]. In particular, Khot's conjecture implies that there must remain a large integrality gap even after n^δ rounds of LS tightenings for some $\delta > 0$ (see Section 7.4 for more about Khot's Unique Games conjecture).

Our lower bound argument does not yield any integrality gaps for LS_+ tightenings for VERTEX COVER. As already mentioned in Chapter 4, proving such lower bounds for VERTEX COVER remains a difficult open problem. Moreover, such lower bounds would arguably provide much stronger evidence about the true inapproximability of VERTEX COVER (see the discussion in Section 7.1.1).

It would be interesting to see if our techniques can be used to prove integrality gaps for UNIQUE LABEL COVER in the LS or LS_+ hierarchies. Such results could further support Khot's Unique Games conjecture, in turn providing further evidence about the true inapproximability of VERTEX COVER.

Chapter 6

An integrality gap for Independent Set

6.1 Local vs. global properties

In this chapter we prove integrality gaps for linear programs for INDEPENDENT SET in the variables $\{x_1, x_2, \ldots, x_n\}$ where the programs allow any constraint of the form $a^T x \leq b$ such that the coefficient vector a is nonzero for at most $n^{\epsilon(1-\gamma)}$ coordinates. In other words, each constraint involves at most $n^{\epsilon(1-\gamma)}$ variables. Such linear programs may have exponential size and may not have a polynomial-time separation oracle. We only require that all 0/1 independent sets in the graph are feasible for the INDEPENDENT SET relaxations. We will prove the following result for such relaxations:

Theorem 6.7. *Fix $\epsilon, \gamma > 0$. Then there exists a constant $n_0 = n_0(\epsilon, \gamma)$ such that for every $n \geq n_0$ there exists a graph G with n vertices for which the integrality gap of any linear relaxation for* INDEPENDENT SET *in which each constraint uses at most $n^{\epsilon(1-\gamma)}$ variables is at least $n^{1-\epsilon}$.*

To prove this theorem, we will use a graph family whose members have a sharp distinction between their global and local properties. Intuitively, the natural candidate graph for exhibiting integrality gaps for such relaxations would be one where the largest independent set is very small, but every induced subgraph on ϵn vertices has an independent set of size nearly $\epsilon n/2$. The intuition is that in the relaxations we are considering each constraint can only "view" a small part of the graph and hence can only "reason" about the graph's local properties.

In fact, graph families whose members have a sharp distinction between their global and local properties can also be viewed as lying at the heart of the results in Chapters 4 and 5. The intuition is that the linear programming relaxations produced by $\Omega(\log n)$ or even n^δ (for some $\delta < 0$) rounds of the LS method are still not strong enough to see beyond the "misleading" local properties of our input graphs. However, while we can take advantage of these "local versus global" properties to prove integrality gaps for linear relaxations, there is evidence that such properties (or, at least the graphs considered in Chapters 4 and 5) may not be useful for proving strong integrality gaps for semidefinite programming relaxations. We discuss this evidence in Section 7.1.1. The intuition here is that positive semidefinitess constraints *are* global constraints unlike linear constraints.

In any case, to return to the relaxations being considered in this chapter, it turns out that the local property our graphs should have to prove Theorem 6.7 is somewhat stronger than having all induced subgraphs on ϵn vertices have independent sets of size nearly $\epsilon n/2$: instead we will need that all small induced subgraphs must have small *fractional chromatic number*. After defining this concept in the next section we will then proceed to prove Theorem 6.7.

6.2 Integrality gaps for Independent Set

Definition 6.1. Let G be a graph. A *fractional γ-colouring* of G is a multiset $\mathcal{C} = \{U_1, \ldots, U_N\}$ of independent sets of vertices (for some N) such that every vertex is in at least N/γ members of $\mathcal{C}$. The *fractional chromatic number* of G is

$$\chi_f(G) = \inf \{\gamma : \ G \text{ has a fractional } \gamma\text{-colouring}\}.$$

Note that if G has a k-colouring with colour classes $U_1, \ldots, U_k$ then $\mathcal{C} = \{U_1, \ldots, U_k\}$ is also a fractional k-colouring of G. Consequently, $\chi_f(G) \leq \chi(G)$.

Remark 6.2. If $\chi_f(G) = \gamma$ and $\{U_1, \ldots, U_N\}$ is a fractional γ-colouring for G, we will usually assume without loss of generality that each vertex of G (by deleting it from a few of the U_i if necessary) is in *exactly* N/γ sets.

Note that strictly speaking, having $\chi_f(G) = \gamma$ does not guarantee that there exists a fractional γ-colouring for G; it only guarantees a fractional $(\gamma+\epsilon)$-colouring for all $\epsilon > 0$. Nevertheless, in the interest of keeping our notation clean, we will always assume that a fractional γ-colouring does exist (in particular, we will only consider rational γ). This slight inaccuracy will not affect the validity of our arguments.

The graphs we will use are given by the following theorem which was first proved by Arora et al. [3] (see also the journal version [4] of the latter paper for a more complete proof). The proof uses the probabilistic method and fall in a line of results starting with Erdős [18] showing that the chromatic number of a graph cannot be deduced from "local considerations" (see also Alon and Spencer [2], p.130).

Theorem 6.3. *Let $0 < \alpha, \delta < 1/2$ be constants. Then there exist constants $\beta = \beta(\alpha, \delta) > 0$ and $n_0 = n_0(\alpha, \beta, \delta)$ such that for every $n \geq n_0$ there is a graph with n*

vertices and independence number at most αn such that every subgraph induced by a subset of at most βn vertices has fractional chromatic number at most $2 + \delta$.

Using the graph H constructed in Theorem 6.3 with α, δ arbitrarily close to 0 and β as given by the theorem, Arora et al. [4] proved the following "local vs. global" result. We include the proof for completeness.

Theorem 6.4. *The vector with all coordinates $\frac{1+\delta}{2+\delta}$ is feasible for any linear relaxation for H in which each constraint involves at most βn variables. Consequently, since any independent set is the complement of a vertex cover, and vice versa, the integrality gap is at least $(1-\alpha) \cdot \frac{2+\delta}{1+\delta}$.*

Proof. It suffices to show that the all-$\frac{1+\delta}{2+\delta}$ vector is feasible for any set of constraints $A_I \cdot x \le b_I$ where $I \subseteq \{1, \dots, n\}$ has size at most βn.

So fix any subset I of at most βn vertices and let $\{U_1, \dots, U_N\}$ be a fractional $(2+\delta)$-colouring for I such that each vertex in I is in exactly a $1/(2+\delta)$ fraction of the U_i's (see Remark 6.2). Note that each $I \setminus U_i$ is a vertex cover in the subgraph induced by I and hence can be extended to a vertex cover of the entire graph. By definition, the characteristic vector of any such vertex cover extension obeys $A_I \cdot x \le b_I$. So since these constraints only involve variables from I, it follows that any vector in $\mathbb{R}^n$ that has $1_{I \setminus U_i}$ (the characteristic vector of $I \setminus U_i$) in the coordinates corresponding to I is also feasible for $A_I \cdot x \le b_I$.

Consider the vectors $v_1, v_2, \dots, v_N \in \mathbb{R}^n$ where v_i is equal to $1_{I \setminus U_i}$ in those coordinates corresponding to I and is $(1+\delta)/(2+\delta)$ otherwise. Each such vector satisfies $A_I \cdot x \le b_I$, so convexity implies that the same is also true for the average vector $\frac{1}{N}(v_1 + v_2 + \cdots + v_N)$. Since each vertex in I lies in exactly a $1 - 1/(2+\delta) = (1+\delta)/(2+\delta)$ fraction of the vertex covers, this average is the all-$\frac{1+\delta}{2+\delta}$ vector. Thus this vector satisfies $A_I \cdot x \le b_I$, as desired. $\square$

This construction can also be used to prove integrality gaps for linear relaxations for INDEPENDENT SET:

Corollary 6.5. *Every* INDEPENDENT SET *linear relaxation for H (where H is the same graph as above) where each constraint in the relaxation has at most βn variables has integrality gap at least $\frac{1}{\alpha(2+\delta)}$.*

Proof. Let I be any subset of at most βn vertices and let $\{U_1, \dots, U_N\}$ be a fractional $(2+\delta)$-colouring for I such that each vertex in I is in exactly a $1/(2+\delta)$ fraction of the U_i's (see Remark 6.2). Now define vectors $v_1, v_2, \dots, v_N \in \mathbb{R}^n$ as follows: Let v_i equal 1_{U_i} in those coordinates corresponding to I but have v_i equal $1/(2+\delta)$ outside I. Then each v_i is feasible for all constraints involving variables only from I. But then, the average of the v_i's, i.e., the vector with all coordinates $1/(2+\delta)$, is also feasible for these constraints. $\square$

Denote the size of the maximum independent set in a graph G by $\alpha(G)$. The above argument in fact yields the following more general theorem.

Theorem 6.6. *Let G be a graph on n vertices such that every subgraph induced by a set of at most $\beta(n)$ vertices has fractional chromatic number $\leq C$. Then the vector with all coordinates $\frac{1}{C}$ is feasible for any linear relaxation of the* INDEPENDENT SET *constraints for G in which each relaxed constraint involves at most $\beta(n)$ variables. Consequently, the integrality gap for the relaxation is at least $\frac{n}{\alpha(G)C}$.*

This suggests we can obtain larger integrality gaps for INDEPENDENT SET if we further limit the number of variables in each constraint. We prove the following theorem in Section 6.2.1 which shows that this is indeed the case.

Theorem 6.7. *Fix $\epsilon, \gamma > 0$. Then there exists a constant $n_0 = n_0(\epsilon, \gamma)$ such that for every $n \geq n_0$ there exists a graph G with n vertices for which the integrality gap of any linear relaxation for* INDEPENDENT SET *in which each constraint uses at most $n^{\epsilon(1-\gamma)}$ variables is at least $n^{1-\epsilon}$.*

6.2.1 Proof of Theorem 6.7

Throughout this proof, log will denote base-2 logarithms.

By Theorem 6.6, to obtain a large integrality gap we need to construct graphs where the independence and local fractional chromatic numbers are as small as possible. One way to do this is using graph products.

Definition 6.8. The *inclusive graph product* $G \times H$ of two graphs G and H is the graph on $V(G \times H) = V(G) \times V(H)$ where $\{(x, y), (x', y')\} \in E(G \times H)$ iff $(x, x') \in E(G)$ or $(y, y') \in E(H)$. The notation G^k indicates the graph resulting by taking the k-fold inclusive graph product of G with itself.

The key observation is that $\alpha(G \times H) = \alpha(G) \times \alpha(H)$ and $\chi_f(G \times H) = \chi_f(G)\chi_f(H)$ (the former fact is easy; for the latter see [19] for a proof). Moreover, if all sets of size at most βn have fractional chromatic number C in G, then all sets of size at most βn in G^k have fractional chromatic number C^k. So taking products of a graph with itself drives down the relative sizes of both the independence and local fractional chromatic numbers. However, since the resulting graph is much larger, the fractional chromatic number is small only for negligibly sized subgraphs. To get around this we instead consider an appropriately chosen (small) random subgraph of G^k. The particular construction we use is due to Feige [19]. By choosing each vertex of G^k independently at random with probability $\alpha(G)^{-k}$ and analyzing the resulting induced subgraph, Feige proves the following theorem (we sketch a proof below for completeness; see [19] for details):

Theorem 6.9 (Feige [19]). *There exists an integer n_0 such that for every graph G on $n \geq n_0$ vertices and any integer k, there exists a graph G_k such that:*

1. *G_k is a vertex induced subgraph of G^k.*
2. *$\frac{1}{2}\left(\frac{n}{\alpha(G)}\right)^k \leq |V(G_k)| \leq 2\left(\frac{n}{\alpha(G)}\right)^k$.*
3. *$\alpha(G_k) \leq \frac{k\alpha(G)\ln n}{\ln(k\alpha(G)\ln n)}$.*

Proof. (Sketch) Select each vertex of G^k independently and at random with probability $\alpha(G)^{-k}$. Let $\hat{G}$ be the induced subgraph of G^k obtained by this process. We show that $\hat{G}$ satisfies the above three properties with high probability.

By construction, $\hat{G}$ is an induced subgraph of G^k. Moreover, the probability that $|V(\hat{G})|$ deviates by more than a factor of 2 from its expectation is negligible. For the last property, fix a maximal independent set I in G^k. The expected number of vertices from I in $\hat{G}$ is at most 1. Chernoff bounds sharply bound the probability that more than $\frac{k\alpha(G)\ln n}{\ln(k\alpha(G)\ln n)}$ vertices of I survive in $\hat{G}$. The last property can now be seen to hold with high probability by observing that G contains at most $n^{\alpha(G)}$ maximal independent sets and by observing that all maximal independent sets in G^k are the direct product of k maximal independent sets in G. In particular, the probability that more than $\frac{k\alpha(G)\ln n}{\ln(k\alpha(G)\ln n)}$ vertices of any maximal independent set of G^k survive in $\hat{G}$ can be shown to go to 0 as n grows. □

Our strategy for proving Theorem 6.7 will now be as follows: We will start with a graph G where both the independence number and local fractional chromatic number are already small (such a graph will exist by Theorem 6.3) and then apply Feige's randomized graph product to it.

Now the details. Fix arbitrarily small constants $\alpha, \delta > 0$ and $n > 0$ such that $n \geq n_0$ where n_0 is from Theorem 6.9. Provided that n is chosen sufficiently large, Theorem 6.3 implies that there exists a graph G on n vertices such that $\alpha(G) \leq \alpha n$ and such that for some constant $\beta > 0$, all induced subgraphs of G with at most βn vertices have chromatic number $\leq 2 + \delta$.

Fix an arbitrarily small constant $d > 0$ and let G_k be the graph given by Theorem 6.9 for $k = d \log n$. Let $N = |V(G_k)|$. Note that $N = \Theta(\alpha^{-k}) = \Theta(n^{d\log(1/\alpha)})$. On the other hand, all subsets of G_k of size at most

$$\beta n = \Theta\left(N^{\frac{1/d}{\log(1/\alpha)}}\right) \tag{6.1}$$

have fractional chromatic number $\leq (2+\delta)^k$.

By Theorem 6.6 it follows that any linear relaxation of the independent set constraints for G_k where the relaxed constraints contain at most βn variables has integrality gap (the $\tilde{\Theta}$ notation indicates asymptotic order up to logarithmic factors):

$$\Theta\left(\frac{\alpha^{-k}}{(2+\delta)^k \frac{k\alpha n \ln n}{\ln(k\alpha n \ln n)}}\right) = \tilde{\Theta}\left(n^{d(\log(1/\alpha)-\log(2+\delta))-1}\right) = \tilde{\Theta}\left(N^{1-\frac{1/d+\log(2+\delta)}{\log(1/\alpha)}}\right). \quad (6.2)$$

Since we can take α and δ to be arbitrarily small in Theorem 6.3 (provided n is large enough), and since $d > 0$ can also be chosen arbitrarily small, it follows that we can simultaneously make (6.1) more than $N^{\epsilon(1-\gamma)}$ and (6.2) more than $N^{1-\epsilon}$. The theorem follows.

Chapter 7

Discussion

7.1 Integrality gaps for Vertex Cover SDPs

7.1.1 Limitations to our approach

The integrality gaps for VERTEX COVER in the LS hierarchy from Chapters 4 and 5 were proved using sparse graphs. The gaps were obtained by arguing that while these graphs have no large independent set, the LS hierarchy relaxations nevertheless admit a large (fractional) independent set. However, Feige [22] has observed that algorithms based on semidefinite programming can be used for almost all graphs from our graph families to certify that they *do not* contain large independent sets. This suggests that we will not be able to use sparse graphs to prove integrality gaps for VERTEX COVER in the LS_+ hierarchy.

We now outline Feige's observation. Consider $G \sim \mathcal{G}(n,p)$. We show that for almost all such G, one can use the Goemans-Williamson SDP-based approximation algorithm for MAX-CUT to certify that G does not have an independent set of size $(\frac{1}{2} - \epsilon)n$. Let $\gamma = 0.878\ldots$ be the approximation guarantee for the GW algorithm. The expected size of a maximum cut in G is half the expected number of edges in G, namely, $\frac{1}{2}\binom{n}{2}\frac{d}{n} = \frac{d(n+1)}{4}$. Hence, with high probability the GW algorithm will output at most $\frac{d(n+1)}{4\gamma}$.

On the other hand, suppose that G has an independent set I of size $(\frac{1}{2} - \epsilon)n$. Then with high probability, there are $d(\frac{1}{2}-\epsilon)n$ edges coming out of I. In particular, G has a cut of this size and the GW algorithm will output at least this value.

Hence, we can use the GW algorithm to rule out if G has an independent set of size $(\frac{1}{2} - \epsilon)n$ as follows: Run the algorithm on G; if it outputs at most $\frac{d(n+1)}{4\gamma}$, then output that G does not have an independent set of size $(\frac{1}{2} - \epsilon)n$; otherwise, answer arbitrarily. By the above discussion, this algorithm with correctly certify that G does not have an independent set of size $(\frac{1}{2} - \epsilon)n$ for almost all $G \sim \mathcal{G}(n,p)$.

The graph families used in the results of Chapters 4 and 5 are not exactly $\mathcal{G}(n,p)$

since the families used in those chapters have all small cycles removed and, in the case of Chapter 5, all high-degree nodes removed. However, these families can be derived by taking the graph drawn from $\mathcal{G}(n,p)$ and removing a sublinear number of edges. In particular, these are not enough alterations to render Feige's test invalid.

7.1.2 An integrality gap for Vertex Cover in the LS_+ hierarchy

We end this section by giving Kale's proof that a different graph family than those used in Chapters 4 and 5 can be used to prove that the integrality gap of the standard linear relaxation for VERTEX COVER remains $2-o(1)$ even after one round of LS_+ tightening. The graph construction is due to Kleinberg and Goemans [41] where they used it to show that the standard SDP relaxation (1.6) for VERTEX COVER has an integrality gap of $2-\epsilon$. Recall that in section 2.2.1 we showed that the relaxation resulting from tightening the trivial linear relaxation for VERTEX COVER with one round of LS_+ is at least as tight the standard SDP relaxation.

Let $K \subseteq \mathcal{R}^{n+1}$ be the homogenization of the polytope defined by the trivial VERTEX COVER constraints (2.2) and (2.3) for an n-vertex graph G.

Theorem 7.1 (Kale [34]). *For all $\alpha, \epsilon > 0$ and all sufficiently large integers n there exists an n-vertex graph G_ϵ whose maximum independent set has size at most αn and for which the value of (2.1) over $N_+(K)|_{x0=1}$ is at most $(\frac{1}{2}+\epsilon)n$. Consequently, for all $\epsilon > 0$ the integrality gap for* VERTEX COVER *after one round of LS^+ lift-and-project is at least $2-\epsilon$.*

Proof. We use the same graph family used by Kleinberg and Goemans [41] for their integrality gap for (1.6). The vertices of a graph G in this family are the set $n = 2^m$ of all m-bit binary strings for some sufficiently large m. Two vertices are adjacent iff their Hamming distance is $(1-\gamma)m$ where γ is any constant satisfying $0 < \gamma \leq \frac{2\epsilon}{1+2\epsilon}$. As Kleinberg and Goemans note, provided m is large enough, G has no independent set of size greater than αn.

On the other hand, we will show that $y = (1, \frac{1}{2}+\epsilon, \ldots, \frac{1}{2}+\epsilon) \in \mathbb{R}^{n+1}$ is in $N^+(K)$ and hence, the value of (2.1) over $N_+(K)|_{x0=1}$ is at most $(\frac{1}{2}+\epsilon)n$.

To that end, we must find vectors $v_0, v_1, \ldots, v_n$ such that the PSD matrix Y defined by $Y_{ij} = v_i \cdot v_j$ satisfies the conditions of Lemma 2.1. As shown in Section 4.1.4, these conditions are equivalent to the following:

$$\begin{aligned} Y_{i0} = Y_{ii} = y_i &\qquad \forall i = 1, \ldots, n \\ 0 \leq Y_{ij} \leq y_i &\qquad \forall i, j = 1, \ldots, n \\ y_i \leq Y_{ij} + Y_{ik} \leq y_i + y_j + y_k - 1 &\qquad \forall i = 1, \ldots, n, \forall \{j,k\} \in E \end{aligned}$$

So the v_i must satisfy:

$$v_0 \cdot v_i = v_i \cdot v_i = \frac{1}{2} + \epsilon \qquad \forall i = 1, \ldots, n \tag{7.1}$$

$$0 \leq v_i \cdot v_j \leq \frac{1}{2} + \epsilon \qquad \forall i, j = 1, \ldots, n \tag{7.2}$$

$$\frac{1}{2} + \epsilon \leq v_i \cdot (v_j + v_k) \leq \frac{1}{2} + 3\epsilon \qquad \forall i = 1, \ldots, n, \forall \{j, k\} \in E \tag{7.3}$$

We will choose v_i from $\mathcal{R}^{m+1}$. Let $\delta = \frac{1}{2} + \epsilon$ and $\beta = \sqrt{1/4 - \epsilon^2}$. Let $v_0 = e_0$. For all $i \in V$ let $v_i^{(p)} = \beta/\sqrt{m}$ if the pth bit of i is 1 and $v_i^{(p)} = -\beta/\sqrt{m}$ if the pth bit of i is 0. Finally, let $v_i^{(0)} = \delta$.

We show now that these v_i satisfy (7.1)–(7.3). Condition (7.1) holds since $v_0 \cdot v_i = \delta = \frac{1}{2} + \epsilon$ and $v_i \cdot v_i = \delta^2 + \beta^2 = \frac{1}{2} + \epsilon$. Next note that $\delta^2 - \beta^2 \leq v_i \cdot v_j \leq \delta^2 + \beta^2$. So condition (7.2) holds.

Finally, since vertices are adjacent in G_ϵ iff the Hamming distance between them is exactly $(1-\gamma)m$, it follows that for all $\{j, k\} \in E$,

$$2\delta^2 - 2\gamma\beta^2 \leq v_i \cdot (v_j + v_k) \leq 2\delta^2 + 2\gamma\beta^2.$$

Our bounds on γ now imply that (7.3) holds. □

Open Problem 1. What happens to the integrality gap for VERTEX COVER after more than 1 round of LS_+ tightening?

Charikar [12] subsequently showed that the integrality gap remains $2 - \epsilon$ even if we add the following "triangle inequalities" to SDP (1.6):

$$(v_i - v_j) \cdot (v_i - v_k) \geq 0 \qquad \forall i, j, k.$$

Indeed, Charikar uses the same graph family as Kleinberg and Goemans did, but employs a different feasible solution. Subsequently, Hatami, Magen and Markakis [31] show that even adding the so-called pentagonal inequalities to this SDP does not decrease the integrality gap. They also use the Kleinberg and Goemans graph family.

Open Problem 2. Where do the relaxations considered by Charikar [12] and by Hatami, Magen and Markakis [31] lie in the LS_+ hierarchy for VERTEX COVER?

7.2 The proof complexity angle

Our results can also be viewed from the propositional complexity point of view. In particular, we explain the relation between proving integrality gaps and proving lowerbounds on the rank of LS_+ proof systems. In general a *propositional proof system* is a polynomial time verifier $V(\mathcal{P}, \phi)$ that checks whether $\mathcal{P}$ is a *certificate* of the universal statement $\forall x \neg\phi(x)$, i.e., ϕ is unsatisfiable. Many (approximation) algorithms as a byproduct of their computation provide (explicitly or implicitly) a certificate that the output value lies within a certain factor to the optimum; this certificate may be considered a *propositional proof* that the given **NP**-optimization problem has no solution that achieves a certain optimization value. In the case of LS_+ cuts, the inequalities that describe the polytope $N_+^r(P)$ resulting after r rounds may be inferred from the set of initial inequalities in the Lovász-Schrijver proof system. Thus, every proof of the integrality gap for a sequence of LS_+ cuts may be considered as a lower bound on the refutation rank in an LS_+ proof system of the tautology encoding that there exists no good solution, and vice versa. So since the propositional and computational complexity are similar for LS round lower bounds, we have presented our results in the context of the latter in this thesis. (Note that the classical propositional complexity measure would be the number of lines needed to do LS-style reasoning. However, no lower bounds are known for this measure.)

Looking at our results then from the proof complexity angle it follows that there exist unsatisfiable random 3SAT instances for which an LS_+ proof system requires a linear number of rounds to refute, solving a problem left open in [11]. Similarly, our results for hypergraph VERTEX COVER and SET COVER show that the constraints defined by certain instances of these problems also require a linear number of rounds to refute.

One curious difference between the our inapproximability results in the LS and LS_+ hierarchies and PCP-based inapproximability results. In the PCP world, once we have proved an inapproximability results for "canonical" problems such MAX-3SAT, we can use reductions to prove inapproximability results for many other problems. Proving integrality gaps via reductions in the "Lovász-Schrijver" world seems much harder if not impossible. In general this should not be surprising, since reductions use arbitrary polynomial-time computations, which may be outside the purview of the limited "reasoning" available in the LS_+ system when viewed as a proof system. What is more surprising is that even the simple gadget-based reductions typically encountered in NP-hardness proofs seem outside the purview of LS_+ reasoning. To give an example, approximating MAX-3SAT within a factor better than 7/8 is reducible via a textbook reduction (carried out entirely with local gadgets) to approximating VERTEX COVER in graphs within a factor better than 17/16. Nevertheless, we are unable to rule out $17/16 - \epsilon$ (or even weaker) approximations to VERTEX COVER in graphs, even though we have ruled out $7/8 - \epsilon$ approximations to MAX-3SAT.

The proof complexity angle can be used to shed some intuition on the difficulties in proving integrality gaps in the "Lovász-Schrijver" world via reductions as mentioned in the introduction to Chapter 3. Consider the standard reduction from

3SAT to VERTEX COVER where each clause is replaced by a triangle of vertices. We could now add new *auxiliary variables* for each triangle where each new variable is a function of the three variables from the triangle's corresponding clause. However, in general, when one introduces such auxiliary variables the proof complexity may change drastically. For example, weak resolution turns into the powerful Extended Frege proof system. On the other hand, in our case all auxiliary variables are locally specified so adding them should intuitively not make a big difference. Nevertheless, our arguments using Lemma 2.1 seem to break down and a newer lower bound idea seems necessary.

This raises the possibility that the familiar interrelationships among approximation problems break down when one considers subexponential time approximation algorithms.

7.3 Lower bounds in the Sherali-Adams hierarchy

While there exist results separating the Sherali-Adams hierarchy and the LS hierarchy, as well as lower bounds on the number of rounds required to derive certain inequalities in the SA hierarchy (see Laurent [42] for examples), as far as we know, there are no non-trivial integrality gaps are known for any optimization problem beyond the first level of the SA hierarchy (recall that the first level of the SA hierarchy is equal to the first level of the LS hierarchy). Indeed, proving integrality gaps for the SA and SA_+ hierarchy seems substantially more difficult than proving such gaps for the LS and LS_+ hierarchies.

The results in Buresh-Oppenheim et al. [11] may be a good candidate to extend to the SA hierarchy since the protection matrices they use are particularly simple.

Open Problem 3. Prove inapproximability results for some optimization problem in the SA or SA_+ hierarchies.

7.4 The unique games conjecture

As mentioned in the introduction, a central motivation for the lower bounds proved in this thesis is the existence of several problems for which there are gaps between the approximation ratios ruled out by PCPs and those ratios achievable by known algorithms.

Another approach for making progress on such problems was suggested by Khot [36]: Instead of tackling each problem for which there remains a gap individually, Khot exhibited a single optimization problem, the UNIQUE LABEL COVER problem, for which proving strong inapproximability implies optimal inapproximability results for a variety of problems. The conjecture that UNIQUE LABEL COVER is indeed hard to approximate is called Khot's Unique Games conjecture.

Amongst other results, Khot's Unique Games conjecture implies that VERTEX COVER has no $2 - o(1)$ approximations [39], the Goemans-Williamson algorithm for MAX-CUT is optimal [38, 44], and implies also a super-constant inapproximability hardness result for SPARSEST-CUT [40]. For more details, see Khot's survey article [37].

While approximation algorithms for the UNIQUE LABEL COVER problem are known [54, 13], they fall short of disproving Khot's conjecture. On the other hand, perhaps methods used in this thesis can prove lower bounds for UNIQUE LABEL COVER in the LS and LS_+ hierarchies lending support for Khot's conjecture.

Open Problem 4. Can known techniques prove strong inapproximability results for UNIQUE LABEL COVER in the LS_+ (or less ambitiously, in the LS) hierarchy?

Bibliography

[1] M. Alekhnovich, S. Arora, and I. Tourlakis. Towards strong nonapproximability results in the Lovász-Schrijver hierarchy. In *Proceedings of the 37th Annual Symposium on the Theory of Computing*, New York, May 2005. ACM Press.

[2] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley, New York, 1992.

[3] S. Arora, B. Bollobás, and L. Lovász. Proving integrality gaps without knowing the linear program. In *Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS-02)*, pages 313–322, Los Alamitos, Nov. 16–19 2002.

[4] S. Arora, B. Bollobás, L. Lovász, and I. Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(2):19–51, 2006.

[5] S. Arora and C. Lund. Hardness of approximations. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 10. PWS, 1995.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[7] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC-04)*, pages 222–231, New York, June 13–15 2004. ACM Press.

[8] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, Jan. 1998.

[9] B. Bollobás. *Modern graph theory*, volume 184 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1998.

[10] M. Bonet, T. Pitassi, and R. Raz. Lower bounds for cutting planes proofs with small coefficients. *The Journal of Symbolic Logic*, 62(3):708–728, Sept. 1997.

[11] J. Buresh-Oppenheim, N. Galesi, S. Hoory, A. Magen, and T. Pitassi. Rank bounds and integrality gaps for cutting planes procedures. In *Proceedings: 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2003, 11–14 October 2003, Cambridge, Massachusetts*, pages 318–327. pub-IEEE, 2003.

[12] M. Charikar. On semidefinite programming relaxations for graph coloring and vertex cover. In *SODA*, pages 616–620, 2002.

[13] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th annual ACM Symposium on Theory of Computing (STOC-06)*, 2006.

[14] W. Cook and S. Dash. On the matrix-cut rank of polyhedra. *Mathematics of Operations Research*, 26(1):19–30, 2001.

[15] I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. In ACM, editor, *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing, San Diego, CA, USA, June 9–11, 2003*, pages 595–601, New York, NY, USA, 2003. ACM Press.

[16] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1), 2005.

[17] P. Erdös. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[18] P. Erdös. On circuits and subgraphs of chromatic graphs. *Mathematika*, 9:170–175, 1962.

[19] U. Feige. Randomized graph products, chromatic numbers, and the Lovász ϑ-function. *Combinatorica*, 17(1):79–90, 1997.

[20] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[21] U. Feige. Approximation thresholds for combinatorial optimization problems. *CoRR*, cs.CC/0304039, 2003.

[22] U. Feige. Personal communication, 2005.

[23] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996.

[24] U. Feige and R. Krauthgamer. The probable value of the Lovász–Schrijver relaxations for maximum independent set. *SIAM Journal on Computing*, 32(2):345–370, 2003.

[25] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W.H. Freeman and Co., New York, 1979.

[26] M. X. Goemans and L. Tunçel. When does the positive semidefiniteness constraint help in lifting procedures? *Mathematics of Operations Research*, 26(4):796–815, 2001.

[27] M. X. Goemans and D. P. Williamson. .878-approximation algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC'94 (Montréal, Québec, Canada, May 23-25, 1994)*, pages 422–431, New York, 1994. ACM Press.

[28] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[29] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1993.

[30] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[31] H. Hatami, A. Magen, and V. Markakis. Integrality gaps of semidefinite programs for vertex cover and relations to ℓ_1 embeddability of negative type metrics, Jan. 05 2006.

[32] D. Hochbaum. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.

[33] D. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.

[34] S. Kale. Personal communication, 2006.

[35] H. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *38th Annual Symposium on Foundations of Computer Science*, pages 406–415, Miami Beach, Florida, 20–22 Oct. 1997. IEEE.

[36] S. Khot. On the power of unique 2-Prover 1-Round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 767–775, New York, May 19–21 2002. ACM Press.

[37] S. Khot. Inapproximabiliity results via long code based PCPs. *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, 36, 2005.

[38] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for Max-Cut and other 2-variable CSPs? In *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 146–154, 2004.

[39] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *18th IEEE Conference on Computational Complexity*, page 379, 2003.

[40] S. Khot and N. K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 53–62, 2005.

[41] J. Kleinberg and M. X. Goemans. The Lovász theta function and a semidefinite programming relaxation of vertex cover. *SIAM J. Discrete Math.*, 11(2):196–204, May 1998.

[42] M. Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming. *Mathematics of Operation Research*, 28(3):470–496, 2003.

[43] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, May 1991.

[44] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 21–30, 2005.

[45] C. H. Papadimitriou. *Computational Complexity*. Addison-Weley, 1994.

[46] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[47] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 475–484, El Paso, Texas, 4–6 May 1997.

[48] A. A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Dokl. Akad. Nauk SSSR*, 281(4):798–801, 1985. In Russian: English translation in *Soviet Math. Dokl.* 31:354–357, 1985.

[49] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, Aug. 1990.

[50] T. Stephen and L. Tunçel. On a representation of the matching polytope via semidefinite liftings. *Mathematics of Operations Research*, 24(1):1–7, 1999.

[51] I. Tourlakis. Towards optimal integrality gaps for hypergraph vertex cover in the Lovász-Schrijver hierarchy. In *8th. International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 9th International Workshop on Randomization and Computation*, pages 233–244, 2005.

[52] I. Tourlakis. New lower bounds for vertex cover in the Lovász-Schrijver hierarchy. In *21st IEEE Conference on Computational Complexity*, 2006.

[53] L. Trevisan. Inapproximability of combinatorial optimization problems. *CoRR*, cs.CC/0409043, 2004.

[54] L. Trevisan. Approximation algorithms for unique games. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 197–205, 2005.

[55] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2001.