UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
Limitations of Linear and Semidefinite Programs

Permalink
https://escholarship.org/uc/item/8x80d8f2

Author
Schoenebeck, Grant Robert

Publication Date
2010

Peer reviewed|Thesis/dissertation

Limitations of Linear and Semidefinite Programs

by

Grant Robert Schoenebeck

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Luca Trevisan, Chair
Christos Papadimitriou
Satish Rao
Sourav Chatterjee

Fall 2010

Limitations of Linear and Semidefinite Programs

Abstract

Limitations of Linear and Semidefinite Programs

by

Grant Robert Schoenebeck

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Luca Trevisan, Chair

NP-complete combinatorial optimization problems are important and well-studied, but re-
main largely enigmatic in fundamental ways. While efficiently finding the optimal solution
to such a problem requires that P = NP, we can try to find approximately optimal solutions.
To date, the most promising approach for approximating many combinatorial optimization
problems has been semidefinite programming, a generalization of linear programming. How-
ever semidefinite programs are not as well understood as linear programs. An important
question is whether semidefinite (or linear) programs can be improved to create better
algorithms.

Several processes–Lovász-Schrijver+ (LS+) [22] and the stronger Lasserre hierar-
chy [21] for semidefinite programs, and Lovász-Schrijver (LS) [22], and the stronger Sherali-
Adams hierarchy [27] for linear programs–were create to systematically improve semidefi-
nite and linear programs at the cost of additional runtime. This thesis studies the question:
"What is the tradeoff between the efficiency and the guaranteed approximation in these
hierarchies?" These systems proceed in rounds (and thus are usually referred to as hier-
archies) and all have in common that after n rounds, where n is the number of variables,
they find the optimal solution, and they take time $n^{O(r)}$ to run until the rth round. An
"integrality gap" of α after r rounds for one of these hierarchies proves that the algorithms
generated by the hierarchy cannot find an α approximate solution in time $n^{\Omega(r)}$.

Unlike NP-hardness results, these results are unconditional, yet apply only to a
large, but restricted, class of algorithms. However, very low levels of these hierarchies in-
clude some of the most celebrated approximation algorithms for NP-complete problems. For
example, the first level of LS+ (and hence also Lasserre) for the IndependentSet problem im-
plies the Lovász θ-function [23] and for the MaxCut problem gives the Goemans-Williamson
relaxation [17]. The ARV relaxation of the SparsestCut [5] problem is no stronger than the
relaxation given in the second level of LS+ (and hence also Lasserre).

This thesis shows an optimal integrality gap of $2 - \varepsilon$ for $\Omega(n)$ rounds the LS
hierarchy relaxation of the VertexCover and MaxCut problems. This result implies that
a very large class of linear programs require exponential time to solve VertexCover (or
MaxCut) to better than a factor of 2, even on random graphs. The previously best known
$2 - \varepsilon$ integrality gap for VertexCover [3] only survived $\Omega(\log(n))$ round of LS, and the
previously best known $1/2+\varepsilon$ integrality gap for MaxCut [12] survived any constant number
of rounds of SA (and for thus LS). These results were the first to illustrate the stark difference

between linear program relaxations and semidefinite program relaxations (because MaxCut is better approximated after just one round by LS+).

Additionally this thesis shows that even after $\Omega(n)$ rounds, the Lasserre hierarchy cannot refute a random 3XOR formula. This is the first non-trivial integrality gap for the Lasserre hierarchy, the strongest of all the aforementioned hierarchies. As mentioned above, this result unconditionally rules out the possibility of a subexponential time algorithm for random 3-SAT over a large range of semidefinite programs. There are, additionally, many immediate corollaries such as a similar integrality gap of $7/6 - \varepsilon$ for VertexCover. The techniques in the thesis remain the only known way of obtaining integrality gaps for Lasserre.

To my Mom and Dad

Contents

Acknowledgments

Chapter 1

Introduction

Perhaps the greatest success story of computational complexity theory is the theory of NP-completeness. This theory partially explains why there is no known (provably) efficient algorithms for many combinatorial optimization problems. These problems have been studied extensively because they have so many concrete applications such as routing, scheduling, machine learning, artificial intelligence, and many more. NP-completeness shows that while on the surface these problems seem to bear little resemblance to each other, they are actually, in some sense, all the same. They are the same in that an algorithm to solve any one of these problems could be efficiently transformed into an algorithm to solve them all.

The famous $P \neq NP$ question then asks if there is an efficient algorithm for any of these problems. If it were shown that $P \neq NP$ we would know that none of these problems can be solved *exactly* on *every instance* in *polynomial time*. We might still be left to wonder if we can solve the problems approximately, or if we could solve them on all natural instances, or if we could solve them in quasi-polynomial time.

In fact, many of these problems can be efficiently approximated. The approximation algorithms for these problems are often produced in the framework of linear or semidefinite relaxations. For example, the Lovász θ-function for the IndependentSet problem [23], the Goemans-Williamson relaxation for the MaxCut problem [17], and the ARV relaxation of the SparsestCut problem [5]. However, the programs that approximate these problems are, in a certain concrete sense, very simple programs.

Various methods or cranks (to be described shortly) automatically strengthen linear or semidefinite programs. Each time the crank is turned, the program becomes tighter and tighter. These methods are usually discribed as hierarchies where the rth level of the hierarchy is generated by turning the corresponding crank r times. Lovász-Schrijver (LS) [22], and the stronger Sherali-Adams hierarchy, [27] are two linear programming hierarchies. Lovász-Schrijver+ (LS+) [22] and the stronger Lasserre hierarchy [21] are two semidefinite programming hierarchies. It is additionally known that LS+ is stronger than LS, Lasserre is stronger than SA, and SA and LS+ are incomparable.

While fundamental differences between these hierarchies exist, they also have a lot in common. Relaxations in the rth level of a hierarchy require that any set of r of the original variables be self consistent in a very strong way (how strong depends on the hierarchy). If an

integer program has n variables, then the nth level of any hierarchy is sufficient to obtain a tight relaxation where the only feasible solutions are convex combinations of integral solutions. Additionally, it is possible to optimize over the set of solutions defined by the rth level of any hierarchy in time $O(n^{O(r)})$, which is sub-exponential for $r = o(n/\log n)$.

Therefore, we know that the nth level of these hierarchies produce exact algorithms. For semidefinite programs, the aforementioned algorithms of the Lovász θ-function, the Goemans-Williamson MaxCut relaxation, and the ARV SparsestCut relaxation all fall within the first 2 rounds of the weaker LS+ semidefinite program hierarchy. One of the fundamental questions asked in the thesis is what happens in between. In particular, can these automatic cranks be used to produce efficient algorithms with better approximation guarantees than are currently known? Stated another way, do the supped-up programs of these relaxations outperform the vanilla relaxations? In the cases that we study, we answer these questions in the negative. Such negative answers are especially interesting because these hierarchies include most natural linear or semidefinite programs. Thus these results suggest that either it is hard to approximate these problems, or finding improved algorithms requires fundamentally new techniques.

It is helpful to compare lower bounds about these hierarchies to the more standard PCP based hardness of approximation results. The first difference is that these results are unconditional, and do not depend on complexity assumptions. Secondly, these lower bounds can be more specific. Showing a lower bound for r rounds shows that a large class of algorithms cannot approximate the problem in time less than $n^{\Omega(r)}$. Thus a lower bound for any constant number of rounds rules out polynomial algorithms, a lower bound for a logarithmic number of rounds gives a quasi-polynomial lower bounds, and a lower bound for a linear number of rounds provides an exponential lower bound. Finally, these lower bounds apply only to a specific class of algorithms.

1.1   VertexCover as an Illustrating and Motivating Example

We now present VertexCover as example to both motivate and illustrate this line of research. Throughout we will discuss previous work.

Linear Programs   The usual way to arrive at a linear program relaxation is to start with an integer programming version of a computational problem. For our example we focus on the VertexCover problem: given a graph, what is the minimum size of any vertex cover, where a vertex cover is a set of vertices such that each edge in the graph vertex is incident to a vertex in the set (see Section 1.6 for precise definitions). The VertexCover instance of a graph $G = (V, E)$ can be described by the following integer program:

$$\min \sum_{v \in V} x_v \text{ such that} \tag{1.1}$$

$$\forall (x_v, x_u) \in E \quad x_v + x_u \geq 1$$
$$\forall x_v \in V \quad x_v \in \{0, 1\}$$

where there is a variable x_v for each vertex in the graph, and x_v is set to 1 if the vertex is in the cover, and is set to 0 if the vertex is not in the cover. We can then relax this integer program to a linear program by no longer requiring that the variables are 0 or 1, but simply between 0 and 1.

$$\min \sum_{v \in V} x_v \text{ such that} \tag{1.2}$$

$$\forall (x_v, x_u) \in E \quad x_v + x_u \geq 1$$
$$\forall x_v \in V \quad x_v \in [0,1]$$

This program can now be solve in polynomial time, but the relationship between the solution to Equations 1.1 and 1.2 is not clear.

The relationship between these two programs is captured by the integrality gap. The *integrality gap* is defined to be the largest ratio between the solutions of the integer program and the relaxed program over any instance of the problem (here between Equations 1.2 and 1.1 over any VertexCover instance). In other words, how much information can be lost when passing from the integer program to the linear program. The integrality gap is an important measure because if we can upper bound the integrality gap by α, then we have shown that the relaxation provides an α-approximation algorithm. It is also thought that a lower bound on the integrality gap provides a good bound on the approximation ratio achievable from a relaxation, but this cannot be proven (see discussion later in this section for more details.)

Usually we will show that the integrality gap is upper bounded by showing a rounding algorithm. For example, in the case above a solution to Equation 1.2 can be converted to a solution to Equation 1.1 by simply rounding every x_v to the closest integer value ($\frac{1}{2}$ round up to 1). It is easy to verify that any solution to Equation 1.2 will, once rounded, satisfy Equation 1.1. Moreover, the objective function, $\sum_{v \in V} x_v$, will at most double by rounding. This shows that the integrality gap is at most 2.

We can also lower bound the integrality gap. To do this, we exhibit an instance where the solution to the integer program is large (there is no good combinatorial solution), but the solution to the relaxed program is small (there is a good relaxed solution). In the example above we can take our example to be the complete graph on n-vertices. The best vertex cover is all but one vertex, and so has $n-1$ vertices. However assigning each $x_v = 1/2$ satisfies the linear program and has objective function $n/2$. Thus the ratio between solutions to Equations 1.2 and 1.1 is at least $\frac{n-1}{n/2}$; for large enough n this is arbitrarily close to 2.

The above integrality gap does not provide convincing evidence that VertexCover is hard to approximate to a factor better than 2. Rather it just seems that the relaxation in Equation 1.2 is not sufficiently powerful, and surely we can do better. For example, we could add the following requirement into Equation 1.1:

$$\forall (x_v, x_u), (x_v, x_w)(x_u, x_w) \in E \quad x_v + x_u + w_w \geq 2$$

This actually does not affect Equation 1.1 at all because any solution that satisfies the constraints of Equation 1.1 automatically satisfies this constraint. However, when we relax with the additional constraint we get:

$$\min \sum_{v \in V} x_v \text{ such that} \qquad (1.3)$$

$$\forall (x_v, x_u) \in E \qquad x_v + x_u \geq 1$$
$$\forall (x_v, x_u), (x_v, x_w)(x_u, x_w) \in E \quad x_v + x_u + w_w \geq 2$$
$$\forall x_v \in V \qquad x_v \in [0, 1]$$

The previous integrality gap instance of the complete graph on n vertices no longer holds, because the newly added equations are not satisfied. However, it turns out that there are instances that prove the integrality gap remains 2.

We can visualize what is happening here. First, because the objective function is linear, we see that Equation 1.1 optimizes over the convex hull of the valid integral vertex covers (because the minimum must occur at an extremal point). Equation 1.2 optimizes over a potentially much larger space that contains all the space considered by Equation 1.1. Thus Equation 1.2 may find good solutions that do not lie in the space of valid solutions for Equation 1.1. By adding constraints as in Equation 1.3 the space over which we are optimizing is cut down towards its original size.

The Lovász-Schijver hierarchy (and the stronger Sherali-Adams) hierarchy automatically generate constraints (or cuts), which shrink the space of valid solutions. For example, all the odd-cycle constraints (of which the above triangle constraint is a special case) are generated by one round of the Lovász-Schijver hierarchy. We are interested in weather or not these cuts decrease the integrality gap. Because these hierarchies quickly generate previously ad hoc constraints, like odd-cycle constraints, integrality gaps against there hierarchies are strong indications that linear programming alone will not provide a sub-exponential algorithm to approximate VertexCover.

In Chapter 2 we show an integrality gap of $2 - \varepsilon$ remains after $\Omega(n)$[1] rounds of the Lovász-Schrijver hierarchy, which is optimal. This result, which was first published in Schoenebeck, Tulsiani, and Trevisan [26] builds on the previous work of Arora, Bollobas, Lovász, and Tourlakis [2, 3, 28] who prove that even after $\Omega(\log n)$ rounds the integrality gap of LS is at least $2 - \varepsilon$, and that even after $\Omega((\log n)^2)$ rounds the integrality gap of LS is at least $1.5 - \varepsilon$. It should be noted that Arora, Bollobas, and Lovász [2] pioneered the technique of showing unconditional inapproximability results by illustrating integrality gaps against linear programming hierarchies.

Subsequently, Charikar, Makarychev, and Makarychev [10] show that, for some ε, after n^ε rounds an integrality gap of $2 - o(1)$ remains for the Sherali-Adams hierarchy (which is stronger than the LS hierarchy).

Why Integrality Gaps? We can also ask why we should believe that an integrality gap provides a convincing lower bound. First, if one only looked at the value of the linear programming object function, then this is the best one can do (assuming there are some tight instances–which will occur in any natural problem). However, one could certainly provide an algorithm with a linear program that beat the integrality gap by first solving the linear

---

[1]In all integrality gap containing an ε, the constant in the Ω depends on ε.

program, ignoring the output, and then solving the instance exactly by exhaustive search. If the rounding algorithm is agnostic toward the original instance, and only uses information from the linear programming solution, then the integrality gap should be hard to beat. Indeed, there are no known natural linear or semidefinite programs based algorithms that give an approximation ratio better than the integrality gap.

There are a few peculiarities about the integrality gap. First, look at the trivial algorithm that solves the LP and multiplies the objective function by the integrality gap. This algorithm is actually most accurate on instances with large integrality gap, and is furthest from the truth in instances with the integer program and linear program most resemble each other. Note that in the above VertexCover example the rounding algorithm does very well on the complete graph. Secondly, when showing integrality gaps, we are looking for instances that are hard for linear programs to solve, but that are easy in some other proof system. This is required to prove an integrality gap. The computationally "hardest" instances will not easily provide an integrality gap if it is difficult to show that there is no good combinatorial solution.

Semidefinite Programs   Above we saw how to relax an integer program into a linear program. Now we will show how to relax a quadratic program into a semidefinite program. Again we will use VertexCover as an example. We first show how to encode an instance of VertexCover as a quadratic program:

$$\min \sum_{v \in V} x_v \text{ such that} \tag{1.4}$$

$$\forall (x_v, x_u) \in E \quad (1 - x_v)(1 - x_u) = 0$$
$$\forall v \in V \qquad x_v^2 = x_v; \qquad x_v \in \mathbb{R}$$

Note that the only solutions to $x_v^2 = x_v$ are 0 and 1, and it can easily be checked that this is a correct encoding of the VertexCover problem. Now we homogenize the equations by making them all degree 2. We do this by adding a dummy variable x_0 which is supposed to represent the number 1^2.

$$\min \sum_{v \in V} x_v^2 \text{ such that} \tag{1.5}$$

$$\forall (x_v, x_u) \in E \quad (x_0 - x_v)(x_0 - x_u) = 0$$
$$\forall v \in V \qquad x_v^2 = x_v \cdot x_0; \qquad x_v \in \mathbb{R}$$
$$1 = x_0^2; \qquad x_0 \in \mathbb{R}$$

We homogenized so that we are in a position to relex from a quadratic program to a semidefinite program. We simply replace all the real variables with vectors in some larger space $\mathbb{R}^*$:

---

[2]In reality x_0 can be 1 or -1, but it will not matter and the intuition is clearer if we think of it as 1. Moreover the equation $1 = x_0^2$ is not homogeneous of degree 2, but we include this for ease of presentation.

$$\min \sum_{v \in V} \|x_v\|^2 \text{ such that} \tag{1.6}$$

$$
\begin{aligned}
\forall (x_v, x_u) \in E \quad & \langle x_0 - x_v \rangle \langle x_0 - x_u \rangle = 0 \\
\forall v \in V \quad & \|x_v\|^2 = \langle x_v, x_0 \rangle; \qquad x_v \in \mathbb{R}^* \\
& 1 = \|x_0\|^2; \qquad\qquad x_0 \in \mathbb{R}^*
\end{aligned}
$$

This semidefinite program can be solved in polynomial time. Note that Equation 1.6 is a relaxation of Equation 1.4 in that any valid solution to Equation 1.4 can be mapped to a valid solution of Equation 1.6 with the same objective function (just use 1-dimensional vectors). Again, we would like to know the relationship between the solutions to Equations 1.4 and 1.6, which again we measure by the integrality gap.

Similarly to the linear programming case, the integrality gap is 2. Rounding can be done by rounding any vector with length at least $\sqrt{2}$ to 1, and any vector with length less than $\sqrt{2}$ to 0. The integrality gap lowerbound was illustrated by Kleinberg and Goemans [20]. Unlike in the linear programming case, a random graph does not exhibit an integrality gap. Even this relaxation (which is effectively the Lovász-θ function) can "tell" that the graph is random (and thus does not have a large vertex cover). This illustrates a difference between linear programs and semidefinite programs–the same family of graphs that gives good linear programming lower bounds, fails to provide semidefinite lower bounds even after the first round. Thus semidefinite program integrality gaps may be much more powerful.

Instead of using random graphs, Kleinberg and Goemans used Frankl Rödl graphs do show a lower bound on the integrality gap. Frankl Rödl graphs have the set $\{0,1\}^n$ as their vertices, and edges connect points that are almost antipodal (they have fractional Hamming distance $1 - \gamma$ for some small γ). Frankl and Rödl [15] show that vertex covers for such graphs contain nearly all the vertices. Intuitively, these graphs are geometrically very close to the graph where each vertex is connected to the unique vertex of fractional Hamming distance 1. This graph is a matching, and thus has a vertex cover of only $1/2$ the vertices. The intuition behind the result of Kleinberg and Goemans is that semidefinite programs are aware of the geometric structure but not necessarily the combinatorial structure of problems, and hence are "fooled" by instances that are geometrically close, but combinatorially far from have a good solutions.

Again, we can ask what happens when we add additional constraints that do not make a difference for the quadratic program, but do for the semidefinite program. For example, we can add the constraint that $\forall v, u \in V \quad x_v \cdot x_u \geq 0$. In the quadratic program, because all values are 0 or 1, this always holds. However, in the semidefinite program, the integrality gap lowerbound of Kleinberg and Goemans fails to satisfy these new inequalities. Charikar [8] showed an integrality gap for this case, again using Frankl Rödl graphs.

The best known results for the Lovász-Schrijver+ hierarhcy are incomparable and were show by shown by Georgiou, Magen, Pitassi, and Tourlakis[16] and Schoenebeck, Tulsiani, and Trevisan [25]. The former result [16] builds on the previous ideas of Goemans and Kleinberg [20] and Charikar [8], and shows that an integrality gap of $2 - \varepsilon$ survives

$\Omega(\sqrt{\frac{\log n}{\log\log n}})$ rounds of LS+. The later result shows an integrality gap of $\frac{7}{6} - \varepsilon$ survives $\Omega(n)$ rounds. This result builds on past research which we review here as it is relevant for understanding the results of this thesis.

A result of Feige and Ofek [14] immediately implies a $17/16 - \varepsilon$ integrality gap for one round of LS+, and the way in which they prove their result implies also the stronger $7/6 - \varepsilon$ bound. The standard reduction from Max-3-SAT to VertexCover shows that if one is able to approximate VertexCover within a factor better than 17/16 then one can approximate Max-3-SAT within a factor better than 7/8. This fact, and the $7/8 - \varepsilon$ integrality gap for Max-3-SAT of Alekhnovich, Arora, and Tourlakis [1], however do not suffice to derive an LS+ integrality gap result for VertexCover. The reason is that reducing an instance of Max 3SAT to a graph, and then applying a VertexCover relaxation to the graph, defines a semidefinite program that is possibly tighter than the one obtained by a direct relaxation of the Max 3-SAT problem. Feige and Ofek [14] are able to analyze the value of the Lovász θ-function of the graph obtained by taking a random 3-SAT instance and then reducing it to an instance of IndependentSet (or, equivalently, of VertexCover). Schoenebeck, Tulsiani, and Trevisan apply this same idea, reducing a random 3-XOR instance to VertexCover, and then showing that this instance provides an integrality gap for $\Omega(n)$ rounds of LS+ (using many techniques borrowed from Alekhnovich, Arora, and Tourlakis).

In this thesis we show integrality gaps for the Lasserre hierarchy that match (and thus subsume) those of Schoenebeck, Tulsiani, and Trevisan for the LS+ hiearchy. To do this, we first prove integrality gaps for random 3-XOR instances, and then show that, in Lasserre unlike LS+, the usually reduction does preserve these integrality gaps. This work was first published in [24]. Subsequent work by Tulsiani [29] generalized the 3-XOR results to an larger class of CSPs and simulated the Dinur and Safra [13] VertexCover reduction to obtain a Lasserre integrality gap of ≈ 1.36 for $\Omega(n^\varepsilon)$ rounds.

Other results by Hatami, Magen, and Markakis [18] prove a $2 - o(1)$ integrality gap result for semidefinite programming relaxations of VertexCover that include additional inequalities. The semidefinite lower bound of Hatami et al is implied after five rounds of Lasserre.

1.2 Other Previous Results

While much of the lower bounds work has been done on VertexCover (directly, or indirectly), there has been prior work for other problems as well.

For MaxCut, de la Vega and Kenyon [12] showed that a $1/2 + o(1)$ integrality gap for MaxCut remains after any constant number of rounds in the Sherali-Adams hierarchy. Charikar, Makarychev, and Makarychev [10] extend this result to n^ε rounds, for some ε. This thesis also shows a similar result for Sparsest Cut.

Buresh-Oppenheim, Galesy, Hoory, Magen and Pitassi [7], and Alekhnovich, Arora, Tourlakis [1] prove $\Omega(n)$ LS+ round lower bounds for proving the unsatisfiability of random instances of 3-SAT (and, in general, k-SAT with $k \geq 3$) and $\Omega(n)$ round lower bounds for achieving approximation factors better than $7/8 - \varepsilon$ for Max-3-SAT, better than $(1 - \varepsilon) \ln n$ for Set Cover, and better than $k - 1 - \varepsilon$ for HypergraphVertexCover in k-uniform hypergraphs. They leave open the question of proving LS+ round lower bounds for approximating

the VertexCover problem.

In addition, recent work by Eden Chlamtac [11] has shown improved approximation algorithms for Coloring and IndependentSet in 3-uniform hypergraphs. In [11] the Lasserre hierarchy was used to find and analyze the constraints which led to improved approximations. This work is unlike the aforementioned work, where it was only later realized that the approximation results could be viewed as an application of semidefinite program hierarchies.

1.3   Results Overview

This thesis contains two main types of results: integrality gaps for the LS hierarchy, and integrality gaps for the Lasserre hierachy.

Results for the Lovász-Schrijver hierarchy   For the LS hierarchy, we prove that after $\Omega_\varepsilon(n)$ rounds the integrality gap of VertexCover remains at least $2 - \varepsilon$.

We then apply our methods to the MaxCut problem, and we show that after $\Omega_\varepsilon(n)$ rounds of LS the integrality gap of MaxCut remains at most $\frac{1}{2} + \varepsilon$.

Results for the Lasserre hierarchy   For $k \geq 3$, we show that $\Omega(n)$ levels of Lasserre hierarchy cannot prove that a random k-CSP over any predicate implied by k-XOR is unsatisfiable. From this main results it quickly follows that the $\Omega(n)$th level of Lasserre:

- cannot prove a random k-XOR formula unsatisfiable.

- cannot prove a random k-SAT formula unsatisfiable.

- contains integrality gaps of $1/2 + \varepsilon$ for Max-k-XOR

- contains integrality gaps of $1 - \frac{1}{2^k} + \varepsilon$ for Max-k-SAT.

- contains integrality gaps of $\frac{7}{6} - \varepsilon$ for VERTEXCOVER.

- contains integrality gaps of any constant for k-UniformHypergraphVertexCover.

- contains integrality gaps of $\Omega(1)$ for k-UniformHypergraphIndependentSet.

In addition to the power of this result, it is also very simple. It extends and simplifies results in Schoenenbeck, Trevisan, and Tulsiani [25] and Alekhnovich, Arora, and Tourlakis [1]. To a large extent it also explains the proofs of Fiege and Ofek [14] and Schoenenbeck, Trevisan, and Tulsiani [25], and can be seen as being inspired by these results.

1.4   Techniques Overview

Techniques for the Lovász-Schrijver hierarchy   The instances for which we prove the integrality gap results are (slight modifications of) sparse random graphs. In such graphs,

the size of the minimum vertex cover is $\approx n$, where n is the number of vertices, while we show the existence of a fractional solution of cost $n \cdot (\frac{1}{2} + \varepsilon)$ that remains feasible even after $\Omega_\varepsilon(n)$ rounds. The size of a maximum cut is $\approx \frac{m}{2}$, where m is the number of edges, while we show the existence of a fractional solution of cost $m \cdot (1 - \varepsilon)$ that also remains feasible after $\Omega_\varepsilon(n)$ rounds.

We use two properties of (modified) sparse random graphs. The first property is large girth; it suffices for our application that the girth be a large constant depending on ε. The second property is that for every set of $k = o(n)$ vertices, such vertices induce a subgraph containing at most $(1 + o(1))k$ edges. The same properties are also used in [3, 28].

In order to prove that a certain fractional solution y is feasible for a relaxation $N^k(K)$, it is sufficient to construct a matrix Y such that certain vectors obtained from the rows and columns of Y are all feasible solutions for $N^{k-1}(K)$. (By convention, $N^0(K) := K$.) This suggest an inductive approach, where we have a theorem that says that all solutions satisfying certain conditions are feasible from $N^k(K)$; to prove the theorem we take a solution y that satisfies the conditions for a certain value of k, and then we construct a matrix Y such that all the derived vectors satisfy the conditions of the theorem for $k - 1$, and hence, by inductive hypothesis, are feasible from $N^{(k-1)}(K)$, thus showing that y is feasible for $N^k(K)$. We can also use the fact that the set $N^{k-1}(K)$ is convex; this means that, once we define the matrix Y, and we have to prove that the associated vectors are in $N^{k-1}(K)$, it suffices to express each such vector as a *convex combination* of vectors that satisfy the conditions of the theorem for $k - 1$. (These ideas all appear in previous work on LS and LS+ integrality gaps.)

Roughly speaking, in the work of Arora et al. [3] on VertexCover, the appropriate theorem refers to solutions where all vertices are assigned the value $1/2 + \varepsilon$, except for a set of exceptional vertices that belong to a set of constant-diameter disks. Oversimplifying, to prove a lower bound of k rounds, one needs to consider solutions that have up to k disks, and for the argument to go through one needs the union of the disks to induce a forest, hence the lower bound is of the same order as the girth of the graph. Tourlakis [28] does better by showing that, due to extra conditions in the theorem, the subgraph induced by k "disks" has diameter $O(\sqrt{k})$, and so it contains no cycle provided that the girth of the graph is sufficiently larger than $\sqrt{k}$. This yields an integrality gap result that holds for a number of rounds up to a constant times the square of the girth of the graph.[3]

The solutions in our approach have a similar form, but we also require the disks to be far away from each other. When we start from one such solution y, we construct a matrix Y, and consider the associated vectors, we find solutions where disks are closer to each other than allowed by the theorem, and we have to express such solutions as convex combinations of allowed solutions. Roughly speaking, we show that such a step is possible provided that the union of the "problematic" disks (those that are too close to each other) induces a very sparse graph. Due to our choice of random graph, this is true provided that there are at most $c_\varepsilon \cdot n$ disks, where c_ε is a constant that depends only on ε. We also show that, in order to prove an integrality gap for k rounds, it is sufficient to consider solutions with $O(k)$ disks, and so our integrality gap applies even after $\Omega_\varepsilon(n)$ rounds. Hence

---

[3]Arora et al. [3, 28] present their proofs in the language of a "prover-verifier" game, but they can be equivalently formulated as inductive arguments.

(again, roughly speaking) our improvement over previous work comes from the fact that it suffices that the union of the disks induce a sparse graph (something which is true for a sublinear number of disks) rather than induce a forest (a requirement that fails once we have a logarithmic or polylogarithmic number of disks). This oversimplified sketch ignores some important technical points: We will give a more precise overview in Section 2.2.

Techniques for the Lasserre hierarchy   The instances for which we prove the integrality gap results are random 3-XOR formula with a particular clause density. For example, we show that for random 3-XOR formula the with sufficient clause density (say $5n$ clauses, where n is the number of variables), with high probability, Lasserre cannot refute the formula after $\Omega_\varepsilon(n)$ rounds, even though in such formulas as most $\frac{1}{2} + \varepsilon$ of the clauses can be simultaneously satisfied.

That with high probability, there is no solution that satisfies more than $\frac{1}{2} + \varepsilon$ fraction of the clauses is a well known result that follows from a straight-forward application of a Chernoff bound.

To show that Lasserre cannot refute such formula even after $\Omega_\varepsilon(n)$ rounds we will construct vectors that satisfy the Lasserre constraints. We show that if width-δn resolution fails to refute a formula, then we can construct vectors that satisfy the constraints of $\delta n/4$-rounds of Lasserre. Width-w resolution allows to you to "add" any two clauses that you have already derived, as long as the number of variables in the resulting clause is at most w. This can be seen as making "local" inferences about the solution space. A result by Ben-Sasson and Widgerson [6] shows that with high probability, width-δn resolution is unable to refute random 3-SAT formulas. This result is easily extended to random 3-XOR formula.

It is left to explain how to construct vector solutions given the fact that a formula is not refuted by width-w resolution. We use the list of all clauses generated by the width-w resolution. This list contains all the Fourier coefficients of low-weight that are disallowed by local considerations. Our solution will "randomize" over all the solutions that do not violate any of these constraints. To construct the vectors, we create a coordinate for each low-weight Fourier coefficient, but merge Fourier coefficients that are forced to be the same (by constraints derive by width-w resolution) into the same coordinate. Using the Fourier transform (plus merging), any partial assignment to a few variable can be naturally mapped to this space of merged low-weight Fourier coefficients to create the vector solutions. Intuitively, these vectors work well for two reasons: 1) these vectors behave well under addition because this transformation is linear , 2) dot-products between these vectors behave well because, by construction, when the coordinate line up, they are exactly correlated, when they do not line up, they are completely independent.

Unlike in the Lovász-Schrijver and Lovász-Schrijver+ hierarchies, local-gadget reductions preserve integrality gaps in Lasserre. We use this observation to obtain the remaining results.

1.5   Related Areas and Future Directions

One of the remarkable things about integrality gaps for linear and semidefinite programming hierarchies is that they have connections to a diverse array of other topics.

Some of these connections have already been forged, and some provide hope for future progress.

Inappromibility, NP Harndess, and Unique Games For many NP-complete combinatorial optimization problems a sharp threshold divides which approximation ratios are obtainable in P, and which are NP-complete. However, we do not yet know of such a threshold for most problems which are based on constraints over two variables (2-CSPs). For example, in VertexCover the constraints are over the pairs of vertices incident to an edge and require that at least one of these vertices must be in the set. UniqueGames and SparsestCut also fall in this regime. Although we cannot reduce to approximate versions of these problems from NP-complete problems, we can reduce to them from the problem of determining whether a "unique game" is almost completely satisfiable, or almost completely unsatisfiable. The Unique Games Conjecture (UGC) states that this problem is, in fact, NP-complete. Thus if the UGC is true, we would have (largely) finished categorizing the approximability of 2-CSPs (assuming that P $\neq$ NP). There is an intimate tie between the UGC and SDP relaxations. Informally, if the UGC is true, then SDP relaxations provide an optimal method for approximating a large array of combinatorial optimization problems.

One pessimistic conjecture is that the UGC is true and SDP relaxations give the optimal approximation algorithms. A second and slightly more optimistic conjecture is that the UGC is not true, but captures the limitations of SDP relaxations. Thus we will have to find fundamentally new tools to attack these problems. A third conjecture is that SDP relaxations can disprove the unique games conjecture. The jury is still out, and the evidence is circumstantial. Algorithmic results get as close as possible to approximating unique games without actually disproving the conjecture. A recent result shows that SDP relaxations do solve unique games on expanding instances [4]. Moreover, no one has yet exhibited an integrality gap for unique games that survives two rounds of Lasserre.

If one of the first two conjectures is true, then SDP relaxations will not be able to distinguish between the two, but integrality gaps would be able to eliminate the possibility of the third conjecture. If the third conjecture is true, then SDPs would clearly resolve the issue. In any event, SDP relaxations are still not understood, and have more to say about the situation.

Currently, we do not have an optimal integrality gap that survives even two rounds of Lasserre for UniqueGames, VertexCover, or SparsestCut. Such Lasserre integrality gaps would provide additional evidence for UGC, or at least that UGC will not be solved by semidefinite programming. These gaps may not exist. For example, a MaxCut algorithm that cuts more than a $1 - \sqrt{\varepsilon}$- fraction of the edges in instances where it is possible to cut $1 - \varepsilon$ fraction of the edges, would disprove the UGC. Perhaps a new SDP rounding technique applied to Lasserre could prove such a result, which would be extremely interesting, even if it were only slightly subexponential (for example, it required $o(n \log(n))$-rounds of Lasserre).

Additionally, some hardness results are not predicated on the hardness of unique games, but rather other similar problems (such as 2-1 games with perfect completeness). The hardness of these games could be analyzed through the lens of SDP relaxations in a manner similar to unique games. Whereas UniqueGames does have an integrality gap for its basic relaxation [19], some of these problems lack any known integrality gap, even for

extremely weak relaxations.

This line of research has forged deep relations between approximation algorithms, proof complexity, metric embeddings, average case hardness, and local algorithms. The knowledge gained by insights into the power and limitations of linear and semidefinite programs has substantial promise to provide natural inroads into each of these fields.

Average Case Complexity and Bad instances Most complexity theoretic results are "worst-case" instead of "average-case", even though we usually care more about the average case hardness. Even if we do not hope to solve NP-complete problems over all instances, some can be solved over all instances that arrive in practice. This leads to awkward conversation between theorists and practitioners where the theorist claims that a problem is computationally intractable, and the practitioner claims that he solves it without too much trouble every day. Current theoretical techniques have been largely unable to provide a good understanding of the average case hardness of problems.

Average case complexity is concerned with identifying hard distributions of problems and with distinguishing "easy" instances from "hard" instances. For SDP relaxations, these problems have a concrete interpretation: namely, on what type of instances is the integrality gap small (or non-existent) and on what type of instances is this gap large? Equivalently, for what type of "easy" instances do SDP relaxations provide a good algorithm, and what type of instances remain "hard". Many of these SDP gaps have started to give us insights into these questions. For example the 3-XOR results of [24] showed that instances that have a lot of "expansion" will imply that an instance is "hard". Conversely, the results of [4] show that a similar "expansion" property makes instances of UniqueGames "easy." Further research on SDPs gives us a concrete way to start gaining more insights into these challenging problems.

Local Algorithms Linear and SDP relaxations have a connection to local algorithms. Solutions to these hierarchies must satisfy only local constraints looking at a few variables at time (an exception is the semidefinite constraint, which is more of a global constraint). Thus, SDP integrality gaps ask how correct can local solutions look without there being any corresponding global solution. Alright this observation has been leveraged in [9] to make new insights into the topic of metric embeddings.

Proof Complexity One requisite achievement before being able to solve an instance of an NP-hard problem is to be able to prove that a particular solution is optimal. This corresponds to proving a coNP statement. Any provably correct algorithm must implicitly prove that no better solution than its output exists (or for approximation algorithms prove that no solution is substantially better). Thus SDP relaxations can be seen as providing a proof that no better solution exists. In fact, both the Lasserre and LS+ hierarchies can be thought of as proof systems, where the number of rounds bounds the depth of the proof. Integrality gaps for hierarchies immediately translate into lower bounds for the depth (or rank) required in these corresponding results proof systems. This relationship also provides another interesting direction for research: bounding the "size" of the proofs rather than their depth.

1.6   Notation and Definitions

We denote the set of Boolean variables $[n] = \{1, \ldots, n\}$. Let the range of variables be denoted $\mathbf{x} = \{x_i\}_{i \in [n]} = \{0,1\}^n$. For $I \subseteq \{1, \ldots, n\}$, let $\mathbf{x}_I = \{x_i\}_{i \in I}$ be the projection of $\mathbf{x}$ to the coordinates of I. We will sometimes use $v(i)$ to denote the ith coordinate of a vector v, but at other times simply use v_i.

We will consider problems where each constraint is local in that it is a k-junta, a function that depends on at most k variables. Formally:

Definition 1 *For $I \subseteq [n]$, let $\mathfrak{F}^I$ be the set of all functions that only depend on the variables in I. That is there exists a function $f_{|I} : \mathbf{x}_I \to \{0,1\}$ such that $f(x) \equiv f_{|I}(x_I)$.*

A k-junta f is a function $f : \mathbf{x} \to \{0,1\}$ that depends on at most k variables. Let $\mathfrak{F}^k$ be the set of k-juntas, then

$$\mathfrak{F}^k = \bigcup_{\substack{|I| \leq k \\ I \subseteq [n]}} \mathfrak{F}^I$$

A k-*constraint* f is a k-junta that appears in the objective function or constraints of an optimization problem.

Sometimes we use $\mathbf{1}_f$ to denote $\mathbf{1}_f = \{x \in \mathbf{x} : f(x) = 1\}$.

Definition 2 *A k-constraint f implies another k-constraint g if $\mathbf{1}_f \subseteq \mathbf{1}_g$. We say that a predicate is XOR-implied if it is implied by either parity or its negation.*

For notational convenience, we will denote by $f_{(I)}^{=x_I}$ (or simply $f^{=x_I}$) the constraint where $f_{(I)}^{=x_I}(\bar{x}) = 1$ if $\bar{x}_I = x_I$ and 0 otherwise. We will denote by $\vec{1}$ and $\vec{0}$ the functions that are always and never true respectively (which are 0-juntas).

We will look at relaxations for two types of integer programs. In the first, we have a set of constraints and would like to know if there is any feasible solution. In the second, we have a set of constraints and would like to maximize some objective function subject to satisfying the constraints. We formalize the notions here:

Definition 3 *A k-constraint satisfiability problem $\langle \mathbf{x}, \mathbf{C} \rangle$ is a set of n Boolean variables in the domain $\mathbf{x} = \{0,1\}^n$, and a set of k-constraints $\mathbf{C} = \{C_i\}_{i=1}^m$.*

Definition 4 *A k-constraint maximization (or minimization) problem $\langle \mathbf{x}, \mathbf{C}, M \rangle$ is a set of n Boolean variables in the domain $\mathbf{x} = \{0,1\}^n$, a set of k-constraints $\mathbf{C} = \{C_i\}_{i=1}^m$, and an objective function $M : \mathbf{x} \to \mathbb{R}$ to be maximized (or minimized) where $M = \sum_{j=1}^{\ell} \lambda_j f_j$ and each $\lambda_j \in \mathbb{R}$ and each f_j is a k-junta.*

1.6.1   Problems Studied

Let $\mathcal{P} : \{0,1\}^k \to \{0,1\}$ be a Boolean predicate on k-variables. Let $\mathbf{R}^{n,k}$ be the set of all k tuples of literals and negated literals (or dictators and anti-dictators) such no two depend on the same variable. That is

$$\mathbf{R}^{n,k} = \{(f^{=b_1}_{(\{i_1\})}, \ldots, f^{=b_k}_{(\{i_k\})}): \text{ where } \forall j, j' \in [k], i_j \in [n], b_j \in \{0, 1\}, \text{ and } j \neq j' \Rightarrow i_j \neq i_{j'}\}$$

The language k-CSP-$\mathcal{P}$ consists of satisfiable k-CSPs for the form $\langle \{0, 1\}^n, \mathcal{P} \circ R^1, \ldots, \mathcal{P} \circ R^m \rangle$ where $R^j \in \mathbf{R}^{n,k}$ for $1 \leq j \leq m$. That is, there exists $x \in \{0, 1\}^n$ such that $\mathcal{P} \circ R^j = 1$ for each $j \in [m]$. Each function $\mathcal{P} \circ R^j$ is called a clause or constraint. In Max-k-CSP-$\mathcal{P}$ we want to find the maximum number of clauses that can be satisfied simultaneously in CSPs of the same form.

To sample a random instance of k-CSP-$\mathcal{P}$ (or Max-k-CSP-$\mathcal{P}$) with m clauses and n variables, we can uniformly and independently sample m elements of $\mathbf{R}^{n,k}$, to obtain the instance $\langle \{0, 1\}^n, \mathcal{P} \circ R^1, \ldots, \mathcal{P} \circ R^m \rangle$.

k-XOR is just k-CSP-$\mathcal{P}$ where $\mathcal{P} \equiv \oplus_{i=1}^{k} x_i$. Note that we can always rewrite the constraint $\mathcal{P} \circ R^m$ as $\oplus_{j \in I} x_i = b$ where $I \subseteq [n]$, $|I| = k$, $b \in \{0, 1\}$.

k-SAT is just k-CSP-$\mathcal{P}$ where $\mathcal{P} \equiv \vee_{i=1}^{k} x_i$.

Definition 5 *Given a predicate $\mathcal{P}$ we define $r(\mathcal{P})$ to be the probability that a random assignment satisfies $\mathcal{P}$.*

For example, in k-XOR, $r(k\text{-XOR}) = 1/2$; in k-SAT, $r(k\text{-SAT}) = 1 - (\frac{1}{2})^k$.

In VertexCover we are given a graph $G = (V, E)$. There is a Boolean variable x_i for each vertex $i \in V$. For each edge $(i, j) \in E$ we have a constraint which says that both x_i and x_j cannot be zero. We are asked to minimize $\sum_{i \in V} x_i$.

In k-UniformHypergraphIndependentSet we are given a k-uniform hypergraph $G = (V, E)$. There is a variable x_i for each vertex $v \in V$. For each edge $(i_1, \ldots, i_k) \in E$ we have a constraint which says that not all $x_{i_1}, \ldots, x_{i_k}$ can be one. We are asked to maximize $\sum_{i \in V} x_i$.

k-UniformHypergraphVertexCover is the same as k-UniformHypergraphIndependentSet except that for each edge $(i_1, \ldots, i_k) \in E$ we have a constraint which says that at least one of $x_{i_1}, \ldots, x_{i_k}$ must be one. We are asked to minimize. $\sum_{i \in V} x_i$.

1.6.2   Definition of Linear and Semidefinite Program Hierarchies

Here we will formally describe four linear/semidefinite program hierarchies. First we define the Lovász-Schrijver linear program hierarchy and pay particularly close attention to its definition on VertexCover, and MaxCut. Next we remark on how to extend this to the Lovász-Schrijver+ hierarchy for semidefinite programs. We then define the Lasserre semidefinite program hierarchy (for locally constrained problems). Finally, we remark on how to weaken the Lasserre hierarchy to obtain the Sherali-Adams linear program hierarchy (again, for locally constrained problems).

The Lovász-Schrijver Hierarchy

In this section we define the Lovász-Schrijver operator N, that maps a linear programming relaxation K into a tighter one $N(K)$. It is simpler to describe the application

of the operator to *convex cones*, as defined next. A convex cone is a set $K \subseteq \mathbb{R}^d$ such that for every $\mathbf{y}, \mathbf{z} \in K$ and for every non-negative $\alpha, \beta \geq 0$ we have $\alpha\mathbf{y} + \beta\mathbf{z} \in K$.

We will use the following notation: for a matrix M, we denote by $\mathbf{M}_i$ the i-th row of M.

If $K \subseteq \mathbb{R}^d$ is a convex cone, then we define $N(K) \subseteq \mathbb{R}^d$ as follows: a vector $\mathbf{y} = (y_0, \ldots, y_{d-1})$ belongs to $N(K)$ if and only if there is a matrix $Y \in \mathbb{R}^{d \times d}$ such that

- Y is symmetric

- For all $i = 0, \ldots, d-1$, $Y_{0,i} = Y_{i,i} = y_i$

- For all $i = 0, \ldots, d-1$, $\mathbf{Y}_i$ and $\mathbf{Y}_0 - \mathbf{Y}_i$ are in K.

In such a case, we say that Y is a *protection matrix* for $\mathbf{y}$.

We also use the notation $N^0(K) := K$ and $N^k(K) := N(N^{k-1}(K))$.

Let $G = (V, E)$ be a graph, and assume $V = \{1, \ldots, n\}$. The cone of the linear progamming relaxation of the VertexCover problem is the set of vectors $\mathbf{y} \in \mathbb{R}^{n+1}$ such that

$$
\begin{aligned}
y_i + y_j &\geq y_0 \quad \forall (i,j) \in E \\
0 \leq y_i &\leq y_0 \quad \forall i \in V \\
y_0 &\geq 0 \quad\quad\quad\quad (VC(G))
\end{aligned}
$$

The relaxation of the VertexCover problem arising from k rounds of Lovász Schrijver is the solution of

$$
\begin{aligned}
\min &\sum_{i=1}^{n} y_i \\
\text{subject to} \quad &(y_0, y_1, \ldots, y_n) \in N^k(VC(G)) \\
&y_0 = 1
\end{aligned}
$$

The *integrality gap* of this relaxation for graphs of n vertices is the largest ratio between the minimum vertex cover size of G and the optimum of the above program, over all graphs G with n vertices.

The linear programming relaxation for MAX-CUT is a set of constraint on n *vertex variables* and m *edge variables*. For a vector $\mathbf{u} \in \mathbb{R}^{n+m+1}$, let u_0 be the extra coordinate for homogenization,[4] $(u_1, \ldots, u_n)$ denote the vertex variables and $(u_{e_1}, \ldots, u_{e_m})$ denote the the edge-variables. Then the cone is the solution set of the constraints

$$
\begin{aligned}
u_e &\leq u_i + u_j & \forall e = (i,j) \in E \\
u_e &\leq 2u_0 - (u_i + u_j) & \forall e = (i,j) \in E \\
0 \leq u_i &\leq u_0 & \forall i \in V \\
0 \leq u_e &\leq u_0 & \forall e \in E \\
u_0 &\geq 0 & (MC(G))
\end{aligned}
$$

---

[4]Homogenization is the process of expressing a linear programming relaxation as a convex cone rather than as a subset of $[0,1]^n$.

The relaxation of the MAX-CUT arising from r rounds of Lovász Schrijver is the solution of

$$\max \sum_{e \in E}^{n} u_e$$
$$\text{subject to} \quad (u_0, u_1, \ldots, u_n, u_{e_1}, \ldots, u_{e_m}) \in N^r(MC(G))$$
$$u_0 = 1$$

The Lovász-Schrijver+ Hierarchy The Lovász-Schrijver operator $N+$, is defined the exact same way as the Lovász-Schrijver operator N but has the additional requirement that the protection matrix Y must also be positive-semidefinite.

Lasserre

The Lasserre relaxation defined momentarily is designed to progressively restrict the feasible region of a constraint maximization (or minimization) problem $\langle \mathbf{x}, \mathbf{C}, M \rangle$ to be closer and closer to the convex hull of the integer solutions, in such a way that maximizing (or minimizing) over the feasible regions is still trackable.

Definition 6 *The rth round of Lasserre on the k-constraint maximization problem $\langle \mathbf{x}, \mathbf{C}, M \rangle$ is the semidefinite program with a real vector variable v_f for every r-junta $f \in \mathfrak{F}^r$. Let $M = \sum_{i=1}^{\ell} \lambda_i f_i$ be the objective function. For reasons of convention, we will denote by v_0 the vector for the function $\vec{\mathbf{1}}$*

$$\max \sum_{i=1}^{\ell} \lambda_i ||v_{f_i}||^2$$

where

$$||v_0||^2 = 1 \tag{1.7}$$
$$\forall C \in \mathbf{C} \quad ||v_C||^2 = 1 \tag{1.8}$$
$$\begin{array}{l} \forall f, g, f', g' \in \mathfrak{F}^r \ where \\ \quad f \cdot g \equiv f' \cdot g' \quad \langle v_f, v_g \rangle = \langle v_{f'}, v_{g'} \rangle \end{array} \tag{1.9}$$
$$\begin{array}{l} \forall f, g, f + g \in \mathfrak{F}^r \ where \\ \quad f \cdot g \equiv \vec{\mathbf{0}} \quad v_f + v_g = v_{f+g} \end{array} \tag{1.10}$$

The semidefinite program for the rth Lasserre round of a satisfiability problem is the same, but we only check for the existence of feasibility, we do not try to maximize over any objective function. [5]

---

[5]This definition is slightly different, but equivalent to other definitions of the kth level of the Lasserre hierarchy (up to an additive constant). The way that it is stated, it would require double exponential time to solve the rth level. This is easily remedied by only defining vectors for the **and** functions of up to r variables and using linear combinations of these vectors to define the remaining vectors. We present it like this for ease of notation.

We will now show some basic facts about the Lasserre hierarchy. First note that this is a relaxation, because any $\{0,1\}$ integer solution x can be transformed into a $\{(0),(1)\}$ vector solution by setting $v_f = f(x)$.

Now, given a distribution of integer solutions, we know that there exists an equivalent vector solution because each integer solution has an equivalent vector solution and the program in convex. We can easily create *explicit* vectors that satisfy the Lasserre constraints.

If $(y_1, \ldots, y_n) = \sum_{j=1}^m p_j(z_1^j, \ldots, z_n^j)$ where $z_i^j \in \{0,1\}$, $z^j = (z_1^j, \ldots, z_n^j)$ are a feasible integral solutions, and $\sum_{j=1}^m p_j = 1$, that is $(y_1, \ldots, y_n)$ is from a probability distribution of integral solutions, then, for each possible k-junta $f \in \mathfrak{F}^k$ we can produce a vector.

$$v_f(j) = \begin{cases} \sqrt{p_j} & f(z^j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.11}$$

These vectors will satisfy all the constraints of the Lasserre hierarchy at any level. If the reader is unfamiliar with the definition of the Lasserre hierarchy, then it is a straightforward and useful exercise to verify this fact.

While the Lasserre equations can be confusing, one general intuition is that satisfying vectors from the rth level of the Lasserre hierarchy define a probability distribution on any set of up to r coordinates (Equations 1.7, 1.9, and 1.10); that the probability distributions always satisfy the constraints (Equation 1.8); and that the probability distributions properly patch together (Equation 1.9). While global probability distributions map directly to vectors, vectors only map to local distributions (marginal distributions over r variables).

We will momentarily formalize this intuition, but first note that this intuition is not sufficient. In applications, it is usually important that we have vectors and not simply local distributions that patch together. The fact that we have vectors gives some global orientation. The Goemans-Williamson MaxCut algorithm generates a global cut with a hyperplane. It is not clear how to do this with a local distributions alone.

Given a set of vectors $\{v_f\}_{f \in \mathfrak{F}^r}$ that satisfy the rth level the Lasserre hierarchy, we can define scalar variables $\{p_f\}_{f \in \mathfrak{F}^r}$ by letting $p_f = ||v_f||^2$. Then p_f can be thought of as the probability that a randomly drawn solution satisfies the function f from the alleged distribution (which may not exists at all). Also we denote by v_{x_I} (or p_{x_I}) the vector (or "probability") corresponding to $f^{=x_I}$.

Claim 7 *Fix $I \subseteq [n]$ such that $|I| \leq r$. Then we can derive a probability distribution over the elements of $x_I \in \mathbf{x}_I$ by defining the "probability" of x_I, p_{x_I} to be $||v_{x_I}||^2$, where $v_{x_I} \equiv v_{f^{=x_I}}$. Actually, these vectors are all orthogonal, and if you sum over them, you get v_0.*

PROOF: If $x_I, x_I' \in \mathbf{x}_I$, then v_{x_I} and $v_{x_I'}$ are orthogonal because $f^{=x_I} \cdot f^{=x_I'} = \vec{0}$ and so by Equation 1.9 $\langle v_{x_I}, v_{x_I'} \rangle = ||\vec{0}||^2$ and by Equation 1.10 $||v_{\vec{0}}||^2 = 0$

Thus, by Equation 1.7 then Equation 1.10:

$$1 = ||v_{\vec{1}}||^2 = ||\sum_{x_I \in \mathbf{x}_I} v_{x_I}||^2 = \sum_{x_I \in \mathbf{x}_I} ||v_{x_I}||^2$$

So indeed we have a probability distribution. $\square$

Claim 8 *If Equations 1.7, 1.9 and 1.10 are satisfied, then Equation 1.8 is equivalent to requiring that $||v_{x_I}||^2 = 0$ for all x_I where $x_I \notin \mathbf{1}_{C_{|I}}$ for some $C \in \mathbf{C} \cap \mathfrak{F}^I$.*

PROOF: We only used Equations 1.7, 1.9 and 1.10 to show Claim 7. So we know that the v_{x_I} are all orthogonal and by Equation 1.10 additionally know that if $C \in \mathfrak{F}^I$ then $C = \sum_{x_I \in \mathbf{1}_{C_{|I}}} f^{=x_I}$ and so by Equation 1.10 we have that $v_C = \sum_{x_I \in \mathbf{1}_{C_{|I}}} v_{x_I}$. Putting these facts together we see.

$$
\begin{aligned}
1 - ||v_C||^2 &= ||v_0||^2 - ||v_C||^2 = ||\sum_{x_I \in \mathbf{x}_I} v_{x_I}||^2 - ||\sum_{x_I \in \mathbf{1}_{C_{|I}}} v_{x_I}||^2 \\
&= \sum_{x_I \in \mathbf{x}_I} ||v_{x_I}||^2 - \sum_{x_I \in \mathbf{1}_{C_{|I}}} ||v_{x_I}||^2 = \sum_{x_I \notin \mathbf{1}_{C_{|I}}} ||v_{x_I}||^2
\end{aligned}
$$

Thus $||v_C||^2 = 1$ if and only if $\sum_{x_I \notin \mathbf{1}_{C_{|I}}} ||v_{x_I}||^2 = 0$ $\square$

The Sherali-Adams Hierarchy   The rth level of the Sherali-Adams hierarchy is defined the same way as the Lasserre hierachy, but only contains scalars, not vectors:

Definition 9 *The rth round of Sherali-Adams on the k-constraint maximization problem $\langle \mathbf{x}, \mathbf{C}, M \rangle$ is the linear program with a real variable p_f for every r-junta $f \in \mathfrak{F}^r$. Let $M = \sum_{i=1}^{\ell} \lambda_i f_i$ be the objective function. For reasons of convention, we will denote by p_0 the variable for the function $\vec{\mathbf{1}}$*

$$
\max \sum_{i=1}^{\ell} \lambda_i p_{f_i}
$$

where

$$
p_0 = 1 \tag{1.12}
$$

$$
\forall C \in \mathbf{C} \qquad p_C = 1 \tag{1.13}
$$

$$
\begin{aligned}
\forall f, g, f + g \in \mathfrak{F}^r \ \text{where} \\
f \cdot g \equiv \vec{\mathbf{0}} \qquad p_f + p_g = p_{f+g}
\end{aligned} \tag{1.14}
$$

The linear program for the rth Lasserre round of a satisfiability problem is the same, but we only check for the existence of feasibility, we do not try to maximize over any objective function. [6]

---

[6]This definition is slightly different, but equivalent (up to a constant additive term) to other definitions of the rth level of the Sherali-Adams hierarchy. The way that it is stated, it would require double exponential time to solve the rth level. This is easily remedied by only defining vectors for the **and** functions of up to r variables and using linear combinations of these vectors to define the remaining vectors. We present it like this for ease of notation.

1.6.3 Fourier Analysis

Let $I \subseteq [n]$, then we define the character $\chi_I : \{0,1\}^n \to \{-1,1\} \subseteq \mathbb{R}$ as

$$\chi_I(x) = \prod_{i \in I} (-1)^{x_i} = (-1)^{\bigoplus_{i \in I} x_i}$$

Note that $\chi_I \cdot \chi_J = \chi_{I \triangle J}$. The *weight* of a character χ_I is defined to be $|I|$, the number of input variables on which its value depends. We use the following facts:

1. Any function $f : \{0,1\}^n \to \{0,1\} \subseteq \mathbb{R}$ can be written as

$$f(x) = \sum_{I \subseteq [n]} \hat{f}(I) \chi_I(x)$$

   where $\hat{f}(I) = \mathbb{E}_x f(x) \chi_I(x)$. Note that if f is a k-junta, then for $|I| > k$, $\hat{f}(I) = 0$.

2. For any functions $f, g : \{0,1\}^n \to \{0,1\} \subseteq \mathbb{R}$ we have that $\widehat{f+g}(I) = \hat{f}(I) + \hat{g}(I)$.

3. For any functions $f, g : \{0,1\}^n \to \{0,1\} \subseteq \mathbb{R}$ we have that $\widehat{f \cdot g}(I) = \sum J \subseteq [n] \hat{f}(J) \hat{g}(I \triangle J)$

4. Fix $I \subseteq [n]$ and define $f : \{0,1\}^n \to \{0,1\} \subseteq \mathbb{R}$ as $f(x) = \bigoplus_{i \in I} x_i$ Then $\hat{f}(\emptyset) = \frac{1}{2}$, $\hat{f}(I) = -\frac{1}{2}$ and for $J \subseteq [n], J \notin \{\emptyset, I\}$ then $\hat{f}(J) = 0$.

Chapter 2

Linear Round Integrality Gaps for Lovasz-Schrijver

2.1 Our Results

Define an $(\alpha, \delta, \gamma, \eta)$ graph G on n vertices as a graph with girth $\delta \log(n)$, and such that no vertex cover of size $(1 - \alpha)n$ exists and each induced subgraph of G with $k \leq \gamma n$ vertices, has at most $(1 + \eta)k$ edges.

Lemma 10 *For every $0 < \alpha < 1/125$, $\eta > 0$, there exists a $d = d(\alpha) \in \mathbb{N}$, δ, $\gamma > 0$, and $N \in \mathbb{N}$ such that for $n \geq N$ there exists an $(\alpha, \delta, \gamma, \eta)$ graph with max cut less than $\frac{1}{2}|E|(1 + \alpha)$ and maximum degree at most d on n vertices. Here $d(\alpha)$ is an explicit function that depends only on α.*

Lemma 11 *For every η, δ, $\gamma > 0$, $0 < \varepsilon < 1/20$, $d \in \mathbb{N}$ if G is an $(\alpha, \delta, \gamma, \eta)$ graph with maximum degree at most d on n vertices then $(1, 1/2 + \varepsilon, \ldots, 1/2 + \varepsilon) \in N^{\Omega_{\varepsilon, \eta, \delta, \gamma, d}(n)}(VC(G))$ if $\eta \leq \eta(\varepsilon, d)$ where $\eta(\varepsilon, d)$ is an explicit function that depends only on ε and d.*

Lemma 12 *For every η, δ, $\gamma > 0$, $0 < \varepsilon < 1/20$, $d \in \mathbb{N}$ if G is an $(\alpha, \delta, \gamma, \eta)$ graph with maximum degree at most d on n vertices then the solution $\mathbf{y}$ defined as $y_0 := 1$, $y_i := 1/2 + \varepsilon$ and $y_e := 1 - 2\varepsilon$ is in $N^{\Omega_{\varepsilon, \eta, \delta, \gamma, d}(n)}(MC(G))$ if $\eta \leq \eta(\varepsilon, d)$ where $\eta(\varepsilon, d)$ is an explicit function that depends only on ε and d.*

Theorem 13 *For all $0 < \zeta < 1/50$, there is a constant $c_\zeta > 0$ such that, for all sufficiently large n, the integrality gap for VertexCover after $c_\zeta n$ rounds is at least $2 - \zeta$.*

PROOF: Let $\alpha = \zeta/6$ and $\varepsilon = \zeta/6$. Let $d = d(\alpha)$ where $d(\alpha)$ is as in Lemma 10. Let $\eta = \eta(\varepsilon, d)$ where $\eta(\varepsilon, d)$ is as in Lemma 11. Then by Lemma 10, there exists a $\delta, \gamma > 0$, $N \in N$ such that such that for $n \geq N$ there exists an $(\alpha, \delta, \gamma, \eta)$ graph with maximum degree at most d on n vertices. By Lemma 11, the vector $(1, 1/2 + \varepsilon, \ldots, 1/2 + \varepsilon) \in N^{\Omega_{\varepsilon, \eta, \delta, \gamma, d}(n)}(VC(G))$ because $\eta = \eta(\varepsilon, d)$. This exhibits an integrality gap of $\frac{1 - \alpha}{1/2 + \varepsilon} = \frac{1 - \zeta/6}{1/2 + \zeta/6} \geq 2 - \zeta$. $\square$

Similarly, we have

Theorem 14 *For all $0 < \zeta < 1/50$, there is a constant $c_\zeta > 0$ such that, for all sufficiently large n, the integrality gap for MaxCut after $c_\zeta n$ rounds is at most $\frac{1}{2} + \zeta$.*

Lemma 10 is very similar to results already known in the literature (for example [3]) and so we only prove the additional properties that we require in the appendix. Most of the rest of the chapter is dedicated to a proof of Lemma 11. Lemma 12 will follow via a relative simple "reduction" to Lemma 11.

2.2   Overview of the Proof

If D is a random variable ranging over vertex covers, then the solution $\mathbf{y}_D$ where $y_0 = 1$ and $y_i = \mathbf{Pr}[i \in D]$ is a convex combination of integral solutions, and so it survives an arbitrary number of rounds of LS. The protection matrix for $\mathbf{y}_D$ is the matrix $Y = Y_D$ such that $Y_{i,j} = \mathbf{Pr}[i \in D \wedge j \in D]$.

In trying to show that a given vector $\mathbf{y}$ survives several rounds of LS, it is a good intuition to think of $\mathbf{y}$ as being derived from a probability distribution over vertex covers (even if $\mathbf{y}$ is not a convex combination of integral solutions, and cannot be derived in this way) and, in constructing the protection matrix Y, to think of Y as being derived from the said distribution as above.

Note that for the above matrix, the vectors $\mathbf{z} = \mathbf{Y}_i/y_i$ and $\mathbf{w} = (\mathbf{Y}_0 - \mathbf{Y}_i)/(1 - y_i)$ correspond to conditional distributions with $z_j = \mathbf{Pr}[j \in D | i \in D]$ and $w_j = \mathbf{Pr}[j \in D | i \notin D]$. To show that $\mathbf{y} \in N^k(VC(G))$, we must show that $\mathbf{z}, \mathbf{w} \in N^{k-1}(VC(G))$ for the vectors $\mathbf{z}$ and $\mathbf{w}$ corresponding to every i. The kth row in the protection matrices may now be interpreted as the distribution obtained by further conditioning on k. Intuitively, more rounds of LS correspond to further conditioning on other vertices which do not already have probability 0 or 1 in these conditional distributions. We often refer to vertices having probability 0/1 as being *fixed* in the distribution.

Since only r vertices can be conditioned upon in r rounds, we only need to create solutions that look "locally" like distributions over vertex covers for small sized subgraphs. Also, because the given graph has large girth, subgraphs of size $O(\log n)$ are trees. We thus start by expressing the vector $\mathbf{y} = (1, 1/2 + \varepsilon, \ldots, 1/2 + \varepsilon)$ as a probability distribution over vertex covers for a tree. This distribution we define has the property that conditioning on a vertex i only affects the vertices upto a constant distance ℓ from i. In fact, the effect of conditioning decreases exponentially with the distance from i and we explicitly truncate it at distance $\ell = O(\frac{1}{\varepsilon} \log(\frac{1}{\varepsilon}))$. The conditional distribution is referred to as a *splash* around i as it creates "ripples" (change in probabilities) which decrease with distance from i. Fernandez de la Vega and Kenyon [12, Section 5] describe essentially the same distribution of vertex covers over trees in their paper, suggesting its usefulness for proving integrality gaps for the VertexCover problem.

We start with the vector $(1, 1/2 + \varepsilon, \ldots, 1/2 + \varepsilon)$ for the given graph G. After one round of LS, each row i of the protection matrix is defined by changing only weights of vertices within distance a distance ℓ of vertex i according to a splash. Since it affects only a small subgraph, which is a tree rooted at i, the solution "looks" locally like a valid conditional distribution.

Now consider trying to extend this strategy to a second round. Say we want to show that the ith row of the protection matrix above survives another round. We thus need to create another protection matrix for this row. Each row of this new matrix corresponds to conditioning on some other vertex j. If i and j are at distance greater than 2ℓ, the weights (probabilities) of vertices within a distance ℓ from j are still $1/2 + \varepsilon$. The conditional distribution can then be created by replacing these values according to a splash around j and leaving the weights of the other vertices as unchanged. If the distance between i and j is less than 2ℓ and k is a vertex within distance ℓ of either i or j, we modify the weight of k according to the probability that *both* i and j are in the vertex cover.

It would become, unfortunately, very complex to proceed for a large number of rounds with this kind of analysis, and it would appear that the girth of the graph would be a natural limit for the number of rounds for which we can extend this line of argument. (See indeed [3, 28].)

We note however that certain cases are simpler to handle. Suppose that we are given a vector $\mathbf{y}$ that is $1/2 + \varepsilon$ everywhere except in a number of balls, all at distance at least 5ℓ from each other, in which the values of $\mathbf{y}$ are set according to splashes. Then the above ideas can be used to define a valid protection matrix. Unfortunately, this does not seem to help us in setting up an inductive argument, because the structure of the vector that we start from is not preserved in the rows of the protection matrix: we may end up with splash areas that are too close to each other, or with the more special structures that we get by conditioning on a vertex less than distance 2ℓ from the root of a splash.

Our idea, then, is to take such more complex vectors and express them as convex combinations of vectors that are $1/2 + \varepsilon$ everywhere except in splash areas that are at distance at least 5ℓ from each other. We will refer to such solutions as *canonical* solutions. Since we are trying to show that the complex vector belongs to some convex cone, it suffices to show that each one of these simpler vectors is in the cone. Now we are back to the same type of vectors that we started from, and we can set up an inductive argument.

Our inductive argument proceeds as follows: we start from a solution $\mathbf{y}$ in a "canonical" form, that is, such that all vertices have value $1/2 + \varepsilon$ except for the vertices belonging to at most k splashes; furthermore, the roots of any two splashes are at distance at least 5ℓ from each other. We need to construct a protection matrix Y for this vector. To define the jth row $\mathbf{Y}_j$ of the protection matrix we reason as follows: if j is far (distance $> 2\ell$) from the roots of all the splashes in $\mathbf{y}$, then $\mathbf{Y}_j$ looks like $\mathbf{y}$, plus a new splash around j. If j is at distance $\leq 2\ell$ from a splash (and, necessarily, far from all the others) rooted at a vertex r, then we replace the splash rooted at r with a new splash which corresponds to our original distribution over trees conditioned on both r and j.

If $\mathbf{Y}_j$ happens to be a vector in canonical form, we are done, otherwise we need to express it as a convex combination of vectors in canonical form. There are two ways in which $\mathbf{Y}_j$ can fail to be canonical: j may be at distance more than 2ℓ but less than 5ℓ from the closest splash; in this case the new splash we create around j is too close to an already existing one. The other possibility is that j is at distance less than 2ℓ from an existing splash, in which case $\mathbf{Y}_j$ contains a "doubly-conditioned" splash which is not an allowed structure in a canonical solution.

Our idea is then to define a set S of "problematic vertices," namely, the vertices in

the two close splashes, in the first case, or the vertices in the doubly-conditioned splash, in the second case. Then we prove that[1] that the restriction of $\mathbf{Y}$ to small (sub-linear) subset S of vertices can be expressed as a distribution of valid integral vertex covers over S. We would then like to use this fact to express $\mathbf{y}$ itself as a convex combination of solutions that are integral over S and agreeing with $\mathbf{y}$ outside S; if we could achieve this goal, we would have expressed $\mathbf{y}$ as a convex combination of vectors where the "problematic" coordinates of $\mathbf{y}$ are fixed, and the other coordinate are as nice as they were in $\mathbf{y}$.

Unfortunately, some complications arise. In order to express $\mathbf{y}$ as a convex combination $\sum_a \lambda_a \mathbf{y}_a$ of vectors such that each $\mathbf{y}_a$ is fixed in S, it is necessary that each $\mathbf{y}_a$ contains a splash around each of the newly fixed variables. The new splashes may themselves be at distance less than 5ℓ from each other, making the $\mathbf{y}_a$ not canonical. To remedy this problem, we define S (the set of vertices that will be fixed in the $\mathbf{y}_a$) via the following process: we initialize S to the initial set of problematic vertices, then we add all vertices that are at distance less than ℓ from S and that can be connected via a path of length $\leq 5\ell$ that does not pass through S, and so on. At the end of this process, we express $\mathbf{y}$ restricted to S as a convex combination of integral covers, and we extend each of these integral covers over S to a fractional solution over all vertices (by putting splashes around the vertices of S) and so express $\mathbf{y}$ as a convex combination of solutions that, now, are canonical.

The argument works provided that S is of sublinear size. A careful accounting guarantees that, if we want to show that our solution survives k rounds, we only need to consider instances where S is of size $O(k)$. Intuitively, this is due to the fact that each time we make S larger we discover a short path of length $t \leq 5\ell$ in the graph, and we add to the subgraph induced by S $t-1$ new vertices and t new edges. The subgraph induced by S can only include at most $|S|(1+\eta)$ edges, for some very small η, so it cannot happen that S grows too much at each step, because it is not possible to consistently add more edges than vertices to the subgraph induced by S without causing a contradiction to the sparsity condition.

Since this ensures that it takes $\Omega(n)$ rounds before the set of fixed vertices grows to size γn, we can survive $\Omega(n)$ rounds.

2.3 Distributions of Vertex Covers in Trees

As a first (and useful) idealized model, suppose that our graph is a rooted tree. Consider the following distribution over valid vertex covers:

- The root belongs to the cover with probability $1/2 + \varepsilon$

- For every other vertex i, we make (independently) the following choice: if the parent of i does not belong to the vertex cover, then i is in the cover with probability one; if the parent of i is in the cover, then with probability $2\varepsilon/(\frac{1}{2} + \varepsilon)$ we include i in the cover, and with probability $1 - 2\varepsilon/(\frac{1}{2} + \varepsilon)$ we do not include i in the cover.

(The distribution is sampled by considering vertices in the order of a BFS, so that we make a decision about a vertex only after having made a decision about the parent.)

---

[1] Assuming some added conditions on the fractional solution $\mathbf{y}$, called *saturation*.

This is an instantiation of the Ising Model, about which much is known, but we will need only very elementary facts about it. The proofs of these facts are contained in the appendix.

A first observation is that each vertex has probability $1/2 + \varepsilon$ of being in the cover and $1/2 - \varepsilon$ of not being in the cover. The second observation is that, if we condition on the event that, say, the root is in the cover, then this condition affects very heavily the vertices that are close to root, but this effect decreases exponentially with the distance. In particular, for each vertex whose distance from the root is about $4\varepsilon^{-1} \cdot (\log \varepsilon^{-1})$, the probability of the vertex being in the cover condition on the root being in the cover is between $1/2 + \varepsilon - \varepsilon^4$ and $1/2 + \varepsilon + \varepsilon^4$, and the same is true conditioning on the root not being in the cover.

This second observation will show that reasoning about this distribution is useful to deal with graphs that are only *locally* like trees, that is, graphs of large girth. Before discussing this application, we slightly change the distribution so that, after a certain distance from the root, there is no effect (rather than a small effect) if we condition on the root being or not being in the cover. Hence the effect of conditioning on the root is explicitly cut-off after a certain distance.

In particular, consider the following two distributions which sample from the vertex covers of a tree rooted at a vertex i. The conditioning on the root only affects vertices upto a distance $\ell = \frac{8}{\varepsilon} \log \frac{1}{\varepsilon}$ of i.

Definition 15 *For $b \in \{0,1\}$ we define a b-Splash around a vertex i as the distribution which modifies vertices upto a distance of 2ℓ as follows*

1. *$i = b$*

2. *For every vertex upto distance ℓ (and at distance greater than $\ell + 1$), we independently decide to include it with probability 1 if its parent is not in the vertex cover and with probability $2\varepsilon / (\frac{1}{2} + \varepsilon)$ if its parent is already in the vertex cover.*

3. *For u and v at distances ℓ, $\ell + 1$ respectively, we include v with probability 1 if u is not in the vertex cover and with probability*

$$\frac{\mathbf{Pr}[u = 1 | i = b] - \left(\frac{1}{2} - \varepsilon\right)}{\mathbf{Pr}[u = 1 | i = b]}$$

   otherwise.

Where $u = 1$ denotes the event $u \in D$ for a random variable D (with distribution defined by the splash) ranging over the vertex covers of the graph.

For the above to be well defined, we need $\mathbf{Pr}[u = 1 | i = b] > 1/2 - \varepsilon$ for a vertex u at distance ℓ from i. Claim 16 shows that in fact $\mathbf{Pr}[u = 1 | i = b] \in [1/2 + \varepsilon - \varepsilon^4, 1/2 + \varepsilon + \varepsilon^4]$ for u at distance greater than $\ell/2$ and hence the probability at distance ℓ is non-negative.

Claim 16 *Consider a b-Splash around any vertex i such that all vertices upto distance ℓ are labeled $\frac{1}{2} + \varepsilon$. Let j be a vertex such that $d(i, j) \leq \ell$. Then,*

1. $\mathbf{Pr}[j = 1 | i = 1, d(i,j) = k] = (1/2 + \varepsilon)\left[1 + (-1)^k \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon}\right)^{k+1}\right]$ for $0 \le k \le \ell$

   $\mathbf{Pr}[j = 1 | i = 0, d(i,j) = k] = \mathbf{Pr}[j = 1 | i' = 1, d(i',j) = k - 1]$ for $1 \le k \le \ell$

2. $|\mathbf{Pr}[j = 1 | i = b, d(i,j) = \ell/2] - (1/2 + \varepsilon)| \le \varepsilon^4$

3. $\mathbf{Pr}[j = 1 | i = 1, d(i,j) = k] + \mathbf{Pr}[j = 1 | i = 1, d(i,j) = k + 1] \ge 1 + 4\varepsilon^2$ for $0 \le k \le \ell$

Note, in particular, that the probabilities are independent of i and j and depend only on their distance $d(i,j)$. Also, the difference of the probabilities from $1/2 + \varepsilon$ decreases exponentially with distance. The following claim shows that the vertices outside a radius of ℓ from i are independent of whether or not i is in the cover.

Claim 17 *If we pick a 0-Splash with probability $1/2 - \varepsilon$ and a 1-Splash with probability $1/2 + \varepsilon$, then all vertices have probability $1/2 + \varepsilon$. Furthermore, vertices at distance $\ell + 1$ or more from i have weight $1/2 + \varepsilon$ in the 0-Splash as well as 1-Splash around i.*

The vectors that appear in our argument may involve conditioning on a vertex i that has value different from $1/2 + \varepsilon$ based on a splash distribution around a vertex r close to it. The following claims allow us to compute $\mathbf{Pr}[i = 1, j = 1 | r = b]$, the probability of two vertices i, j being simultaneously present in a b-Splash at r, and also $\mathbf{Pr}[i = 0, j = 1 | r = b]$, which is the probability that j is present and i is not. We defer the proofs to the appendix.

Claim 18 *Let $i = v_0, v_1, \ldots, v_{m-1}, v_m = j$ be the path to j, $m \le \ell$, and let u be the vertex on this path which is closest to r. Then*

1. $\begin{aligned}\mathbf{Pr}[i = 1, j = 1 | r = b] =\ &\mathbf{Pr}[u = 1 | r = b] \cdot \mathbf{Pr}[i = 1 | u = 1] \cdot \mathbf{Pr}[j = 1 | u = 1] \\ +\ &\mathbf{Pr}[u = 0 | r = b] \cdot \mathbf{Pr}[i = 1 | u = 0] \cdot \mathbf{Pr}[j = 1 | u = 0]\end{aligned}$

2. *If $\mathbf{Pr}[u = 1 | r = b] = 1/2 + \varepsilon$, then $\mathbf{Pr}[i = 1, j = 1 | r = b] = (1/2 + \varepsilon)\,\mathbf{Pr}[j = 1 | i = 1]$*

The first part of the above claim states that once we condition on u, then i and j are independent. The second part states that if u is sufficiently far r, we can ignore r completely and just compute the probability of j as determined by a splash around i.

Claim 19 *Let i be a vertex and (j,k) be an edge in a b-Splash around r and let $b' \in \{0,1\}$.*

$$\mathbf{Pr}[i = b', j = 1 | r = b] + \mathbf{Pr}[i = b', k = 1 | r = b] \ge \mathbf{Pr}[i = b' | r = b] \cdot (1 + 4\varepsilon^3)$$

The next claim allows us to treat vertices that are sufficiently far from each other as almost independent in the dirtibution conditioned on r.

Claim 20 *Let i and j be two vertices in a b-Splash around r, such that $d(i,j) \ge \ell$. Then*

$$\left|\mathbf{Pr}[i = b', j = 1 | r = b] - \mathbf{Pr}[i = b' | r = b] \cdot \mathbf{Pr}[j = 1 | r = b]\right| \le 2\varepsilon^4$$

2.4 Distribution of Vertex Covers in Sparse Graphs

To reduce solutions with more complicated structure to simpler solutions, we will need to show that if we look at a sufficiently small subgraph of our original graph obtained in Lemma 10, then the more complicated solution can be expressed as a convex combination of 0/1 solutions.

The following result is proved in [3].

Lemma 21 ([3]) *Let $\eta \leq \frac{2\varepsilon}{3+10\varepsilon}$ and let $G = (V, E)$ be a graph such that*

1. *for each $S \subseteq V$, $G(S) = (V_{G(S)}, E_{G(S)})$, then $|E_{G(S)}| \leq (1+\eta)|V_{G(S)}|$.*

2. *$girth(G) \geq \frac{1+2\varepsilon}{\varepsilon}$.*

Then there exists a distribution over vertex covers on G such that each vertex belongs to the vertex cover with probability $1/2 + \varepsilon$.

We will need a slight generalization. Instead of requiring the solution to have the value $1/2 + \varepsilon$ everywhere, we only require that the sum of the values on each edge should be at least $1 + 2\varepsilon$, if both of its endpoints are not already fixed.

Definition 22 *We call a fractional solution y for a graph G ε-saturated if for each edge (i, j) in graph G either:*

- *Both i and j are fixed and $y_i + y_j \geq 1$ or,*

- *$y_i + y_j \geq 1 + 2\varepsilon$.*

We now show that the under the conditions of the previous lemma, every ε-saturated solution can be written as a convex combination of vertex covers of the graph.

Lemma 23 *Let $\eta \leq \frac{2\varepsilon}{3+10\varepsilon}$ and let $G = (V, E)$ be a graph such that*

1. *for each $S \subseteq V$, $G(S) = (V_{G(S)}, E_{G(S)})$, then $|E_{G(S)}| \leq (1+\eta)|V_{G(S)}|$.*

2. *$girth(G) \geq \frac{1+2\varepsilon}{\varepsilon}$.*

and let $\mathbf{y}$ be an ε-saturated solution. Then there exists a distribution over vertex covers on G such that each vertex i belongs to the vertex cover with probability y_i.

PROOF: For the graph G, we will create a set of feasible fractional solutions $\mathbf{y}(k) \in \{0, 1/2 + \varepsilon, 1\}^{|V|}$ such that $\mathbf{y}$ is a convex combination of these vectors.

We partition V into V_0, $V_{1/2+\varepsilon}$, and V_1, as follows:

$$
i \in \begin{cases} V_0 & y_i < 1/2 + \varepsilon \\ V_{1/2+\varepsilon} & y_i = 1/2 + \varepsilon \\ V_1 & y_i > 1/2 + \varepsilon \end{cases}
$$

We define $t(i)$ as follows:

$$t(i) = \begin{cases} 1 - \frac{y_i}{1/2+\varepsilon} & i \in V_0 \\ 1 & i \in V_{1/2+\varepsilon} \\ \frac{y_i - (1/2+\varepsilon)}{1/2-\varepsilon} & i \in V_1 \end{cases}$$

We can order the $t(i)$'s: $0 \le t(i_1) \le t(i_2) \le \cdots \le t(i_{|V|}) \le 1$. For each $k : 1 \le k \le |V|$ we create the vector $\mathbf{y}(k)$ where

$$\mathbf{y}(k)_i = \begin{cases} 0 & i \in V_0 \text{ and } t(i) \le t(i_k) \\ 1 & i \in V_1 \text{ and } t(i) \le t(i_k) \\ 1/2 + \varepsilon & \text{otherwise} \end{cases}$$

We claim the distribution where $\mathbf{y}(k)$ occurs with probability $t_{i_k} - t_{i_{k-1}}$ gives us $\mathbf{y}$.

If $i \in V_0$, then it will be 0 with probability t_i and $1/2 + \varepsilon$ with probability $1 - t_i = \frac{y_i}{1/2+\varepsilon}$. Therefore the probability that i is in the vertex cover is y_i. If $i \in V_1$, then it will be 1 with probability $t_i = \frac{y_i - (1/2+\varepsilon)}{1/2-\varepsilon}$ and $1/2+\varepsilon$ with probability $1 - t_i = 1 - \frac{y_i - (1/2+\varepsilon)}{1/2-\varepsilon}$. Therefore the probability that i is in the vertex cover is $\frac{y_i - (1/2+\varepsilon)}{1/2-\varepsilon} + (1/2 + \varepsilon)(1 - \frac{y_i - (1/2+\varepsilon)}{1/2-\varepsilon}) = y_i$. If $i \in V_{1/2+\varepsilon}$, then it is clear that the probability that i is in the vertex cover is $1/2 + \varepsilon$.

Note that all the weights in each $\mathbf{y}(k)$ are 0, 1 or $1/2 + \varepsilon$. It remains to show that in each of these $\mathbf{y}(k)$ any edge which contains one vertex fixed to 0 has the other vertex fixed to 1. First, note that all neighbors of vertices in V_0 are in V_1. It suffices to show that if i and j are adjacent, $i \in V_1$, $j \in V_0$, that $t(i) \ge t(j)$. However

$$\begin{aligned} t(i) - t(j) &= \frac{y_i - (1/2+\varepsilon)}{1/2-\varepsilon} - \frac{(1/2+\varepsilon) - y_j}{1/2+\varepsilon} \\ &= \frac{(y_i + y_j)/2 + \varepsilon(y_i - y_j) - (1/2+\varepsilon)}{1/4 - \varepsilon^2} \\ &\ge \frac{(1+2\varepsilon)/2 + \varepsilon(y_i - y_j) - (1/2+\varepsilon)}{1/4 - \varepsilon^2} \\ &= \frac{\varepsilon(y_i - y_j)}{1/4 - \varepsilon^2} \ge 0 \end{aligned}$$

which concludes the proof of the lemma. $\square$

2.5 The Main Lemma

We now define the type of solutions that will occur in our recursive argument.

Let $G = (V, E)$ be an $(\alpha, \delta, \gamma, \eta)$ graph with n vertices and degree at most d, as in the assumption of Lemma 11. We define the constant $C = \sum_{i=1}^{\ell+1} d^i$ as the maximum number of vertices within a distance ℓ from some vertex and $D = 5\ell C$ as the maximum number of vertices within distance ℓ of all the vertices in a path of length 5ℓ. Choose $\eta = \frac{1}{3D}$. Note that η depends on only ε and d. Also, we assume that n is large enough that the girth of

the graph is larger than various fixed constants throughout. We fix G for the rest of this section.

Let $R = \frac{\gamma n}{C+2D}$

Let $G_{|S} = (S, E_{|S})$ be the subgraph of G induced by $S \subseteq V$. For some set $S \subseteq V$, define $N_S(i) = \{j : \text{there exists path of length } \ell \text{ from } i \text{ to } j \text{ using only edges in } E\backslash E_{|S}).$

Definition 24 *We say that a vector* $y = (y_0, \ldots, y_n)$ *is* r-*canonical if there exists a set* $S \subseteq V$ *such that:*

- $\forall j \in S \; y_j \in \{0,1\}$ *and* $\mathbf{y}_{|S}$ *is a vertex cover of* $G_{|S}$

- *For every two vertices in* S *the shortest path between them that uses only vertices not in* S *has length* $> 5\ell$. *(Therefore if* $i, j \in S$, $i \neq j$, *then* $N_S(i) \cap N_S(j) = \emptyset$).

-
$$y_i = \left\{ \begin{array}{ll} \mathbf{Pr}[i = 1 | j = y_j] & \exists j \in S \text{ s.t. } i \in N_S(j) \\ 1/2 + \varepsilon & o.w \end{array} \right.$$

- $|S| \leq rC + 2rD$

- *Let* $|S| = rC + kD$ ($k \leq 2r$) *and* $G_{|S} = (S, E_{|S})$ *is the subgraph of* G *induced by* S, *then*
$$|E_{|S}| - |S| \geq k - r$$

We call a set S as in Definition 24 a witness.

Claim 25 *If* $\mathbf{y}$ *is an* r-*canonical vector then,* $\mathbf{y} \in VC(G)$. *Moreover,* $\mathbf{y}$ *is* ε^2-*saturated.*

PROOF: This follows from the fact all edges are either internal to S, internal to some $N_S(i)$, internal to $V \setminus \cup_{i \in S} N(i)$ or between some $N(i)$ and $V \setminus \cup_{i \in S} N(i)$. In the first case, it follows because $\mathbf{y}_{|S}$ is a valid vertex cover having only 0/1 values. In the second because of the fact that a $N(i)$ is weighted according to a splash and Claim 16. In the third case, because the weights are all $1/2 + \varepsilon$. The final case just concerns the vertices at distance ℓ and $\ell + 1$ from the center of a splash and again follows from Claim 16. $\square$

Lemma 11 follows from the above claim, the following result and the fact that $(1, 1/2 + \varepsilon, \ldots, 1/2 + \varepsilon)$ is 0-canonical.

Lemma 26 *Let* $\mathbf{y}$ *be an* r-*canonical solution, and* $r \leq R$. *Then* $\mathbf{y}$ *is in* $N^{R-r}(VC(G))$.

PROOF: We prove it by induction on $R - r$. By Claim 25, an R-canonical solution is feasible for $VC(G)$, and this gives the basis for the induction.

Let $\mathbf{y}$ be an r-canonical solution and let S be a witness to $\mathbf{y}$. We show that there is a protection matrix Y for $\mathbf{y}$ such that $(\mathbf{Y}_i)/y_i$ and $(\mathbf{Y}_0 - \mathbf{Y}_i)/(y_0 - y_i)$ are distributions over $(r + 1)$-canonical vectors for $y_i \neq 0, y_0$. If $y_i = 0$, then we take $Y_i = 0$ which is in $N^k(VC(G))$ for all k and $\mathbf{Y}_0 - \mathbf{Y}_i = \mathbf{Y}_0$ which is r-canonical.

The protection matrix is defined as follows. (When we talk about distance between vertices, we mean distance via paths that do not go through any vertex in S.)

- $Y_{i,0} = Y_{0,i} = Y_{i,i} = y_i$.

- If i and j are at distance greater than ℓ from each other, then $Y_{i,j} = y_i \cdot y_j$

- If i is at distance greater than 2ℓ from the closest vertex in S, and j is at distance at most ℓ from i, then $Y_{i,j}$ is the probability that i and j both belongs to a vertex cover selected according to a splash distribution around $Y_{ij} = y_i \mathbf{Pr}[j = 1 | i = 1]$

- If i is at distance at most 2ℓ from a vertex $r \in S$, and j is at distance at most ℓ from i, then Y_{ij} is the probability that i and j both belong to a vertex cover selected according to a b-Splash distribution around r i.e. $Y_{ij} = \mathbf{Pr}[i = 1, j = 1 | r = b]$

Claim 27 *The matrix Y is symmetric.*

PROOF: If $d(i,j) > \ell$, clearly $Y_{ij} = Y_{ji}$. There remain three additional cases.

- First, if both i and j are at distance greater than 2ℓ from any vertex in S, then $y_i = y_j = 1/2 + \varepsilon$ and also $\mathbf{Pr}[j = 1 | i = 1] = \mathbf{Pr}[j = 1 | i = 1]$ as it depends only on the distance by Claim 16, and hence $Y_{ij} = Y_{ji}$.

- Second, both i and j are at distance at most 2ℓ from any vertex in S. Both i and j cannot be close to two different vertices in S because then $d(i,j) \leq \ell$ would imply a path of length at most 5ℓ between the two vertices which is not possible. Hence, in this case, $Y_{ij} = Y_{ji} = \mathbf{Pr}[i = 1, j = 1 | r = b]$, where r is the vertex in S close to both i and j.

- Finally, if $d(i,r) \leq 2\ell$ for some $r \in S$ and $d(j,r) > 2\ell \; \forall r \in S$, then the path from i to j cannot come closer than distance $\ell + 1$ to r. If l is the vertex on this path closest to r, then we have $P_r^b(l) = 1/2 + \varepsilon$ and by Claim 18, $\mathbf{Pr}[i = 1, j = 1 | r = b] = (1/2 + \varepsilon) \mathbf{Pr}[j = 1 | i = 1] = y_i \mathbf{Pr}[j = 1 | i = 1]$. Therefore, $Y_{ij} = \mathbf{Pr}[i = 1, j = 1 | r = b] = y_i \mathbf{Pr}[j = 1 | i = 1] = Y_{ji}$.

$\square$

Let us fix a vertex i, and consider the vectors $\mathbf{z} := \mathbf{Y}_i/y_i$ and $\mathbf{w} := (\mathbf{Y}_i - \mathbf{Y}_0)/y_i$. We will show that they are (convex combinations of) $(r + 1)$-canonical vectors. (If $y_i = 0$ we do not need to analyse $\mathbf{z}$, and if $y_i = 1$ we do not need to analyse $\mathbf{w}$.)

Note that $\mathbf{z}$ and $\mathbf{w}$ are same as $\mathbf{y}$ except for vertices that are within distance ℓ of i.

Lemma 28 *If $\mathbf{y}$ is an r-canonical solution and Y is the matrix as defined above, then $\forall 1 \leq i \leq n$, the solutions $\mathbf{z} := Y_i/y_i$ and $\mathbf{w} := (\mathbf{Y}_i - \mathbf{Y}_0)/y_i$ are ε^3-saturated*

PROOF: We first give the proof for $\mathbf{z}$. Note that for $d(i,j) > \ell$ $z_j = y_j$ and hence edges as distance greater than ℓ from i are ε^2 saturated because they were in $\mathbf{y}$ by Claim 25. If $d(i,r) > 2\ell \; \forall r \in S$ then the distribution up to distance 2ℓ from i is same as a $1 - Splash$, which is in fact ε^2-saturated by Claim 16 and Claim 17.

Let i be within distance 2ℓ of $r \in S$ and let (j, k) be an edge such that $d(i, j) \leq \ell$ or $d(i, k) \leq \ell$. If both j and k are within distance ℓ of i, then by Claim 19

$$
\begin{aligned}
Y_{ij} + Y_{ik} &= \mathbf{Pr}[i = 1, j = 1 | r = b] + \mathbf{Pr}[i = 1, k = 1 | r = b] \\
&\geq (1 + 4\varepsilon^3) \mathbf{Pr}[i = 1 | r = b] = (1 + 4\varepsilon^3) y_i
\end{aligned}
$$

and we are done. Finally, if $d(i, j) = \ell$ and $d(i, k) = \ell + 1$, then we know by Claim 20 that $|\mathbf{Pr}[i = 1, k = 1 | r = b] - \mathbf{Pr}[i = 1 | r = b] \mathbf{Pr}[k = 1 | r = b]| \leq 2\varepsilon^4$. This gives

$$
\begin{aligned}
Y_{ij} + Y_{ik} &= \mathbf{Pr}[i = 1, j = 1 | r = b] + \mathbf{Pr}[i = 1 | r = b] \mathbf{Pr}[k = 1 | r = b] \\
&\geq \mathbf{Pr}[i = 1, j = 1 | r = b] + \mathbf{Pr}[i = 1, k = 1 | r = b] - 2\varepsilon^4 \\
&\geq (1 + 4\varepsilon^3) \mathbf{Pr}[i = 1 | r = b] - 2\varepsilon^4 \geq (1 + 3\varepsilon^3) y_i
\end{aligned}
$$

using the fact that $\mathbf{Pr}[i = 1 | r = b]$ is at least 2ε. We prove this for $\mathbf{w}$ similarly. $\square$

We shall now express $\mathbf{z}$ and $\mathbf{w}$ as a convex combination of $(r+1)$-canonical vectors.

Claim 29 *If $i \in S$, or if $\forall r \in S$, $d(i, r) > 5\ell$, then $\mathbf{z}$ is $r + 1$ canonical.*

PROOF: If $i \in S$, then $z_k = y_k$ (or $w_k = y_k$) for all $k \in V$ by construction of protection matrix. Because $\mathbf{y}$ is r-canonical $\mathbf{z}$ (or $\mathbf{w}$) is also and this thus also $(r + 1)$-canonical.

If $\forall r \in S$, $d(i, r) > 5\ell$, then it is easily seen that $S \cup \{i\}$ is a witness to $\mathbf{z}$ and $\mathbf{w}$ being $(r + 1)$-canonical. $\square$

If neither of these cases is true, we treat only $\mathbf{z}$, because the same argument works for $\mathbf{w}$. We first define the subset of vertices which is fixed in these vectors.

Recall that for $i \in S$, $N_S(i) = \{j : \text{there exists path of length at most } \ell \text{ from } i \text{ to } j$ using only edges in $E \backslash E_{|S}\}$. In addition let $\partial N_S(i) = \{j : d(i, j) = \ell+1 \text{ in the graph } (V, E \backslash E_{|S})\}$. Also, let $N'_S(i) = N_S(i) \cup \partial N_S(i)$.

Then we make the following definition:

Definition 30 *For a fixed vertex i, we construct $F \subseteq V \setminus S$ as follows:*

Start with $F = N'_S(i)$. If there is a path P of length less that 5ℓ between any two vertices in $F \cup S$ that uses only edges in $V \setminus (F \cup S)$, then $F = F \cup P$. Also, if P intersects $N_S(j)$ for $j \in S$, then $F = F \cup P \cup (N'_S(j) \backslash \{j\})$.

Note that it follows from the above definition that for every $j \in S$, either $N_S(j) \cap F = \emptyset$ or $N'_S(j) \subseteq F$. Also if $\partial F = \{j \in F : j \text{ has neighbors in } V \backslash (S \cup F)\}$, then $\forall j \in \partial F$, $z_j = 1/2 + \varepsilon$ (because for every intersecting $N_S(j')$, we also included $\partial N_S(j')$). We now bound the size of F.

Claim 31 $|F| \leq C + (2r + 2 - k)D$, *where $|S| = rC + kD$.*

PROOF: Every path added in the construction of F has length at most 5ℓ. Also, each vertex in a path can be within distance ℓ of at most one $j \in S$. Thus, the number of vertices added due to a path is at most $5\ell C = D$. Thus, if p paths are added during the construction, then $|F| \leq C + pD$ since C is the size of the $N'_S(i)$, which we start with.

Since the paths are added incrementally, it suffices to show that adding $2r + 2 - k$ paths implies a contradiction. This would imply that $p \leq 2r + 2 - k$ and hence the claim. Let F' be F after addition of $2r + 2 - k$ paths. Then

$$\frac{|E_{|S \cup F'}|}{|S \cup F'|} = 1 + \frac{|E_{|S \cup F'}| - |S \cup F'|}{|S \cup F'|}$$

Note that $|E_{|S \cup F'}| - |S \cup F'| \geq (k - r) + (2r + 2 - k) - 1$, since $|E_{|S}| - |S| \geq (k - r)$ to begin with and addition of $N'_S(i)$, which is a tree adds one more vertex than edge (hence contributing -1), while the addition of each path adds one more edge than vertex. For any $j \in S$ including the region $N_S(j)$ intersected by the path includes a tree of which at least one vertex is already in F and can only contribute positively. This gives

$$\frac{|E_{|S \cup F'}|}{|S \cup F'|} \geq 1 + \frac{k - r + 2r - k + 1}{|S| + |F'|} \geq 1 + \frac{r + 1}{rC + kD + C + (2r + 2 - k)D} = 1 + \frac{1}{C + 2D} > 1 + \eta$$

since $\eta = \frac{1}{3D} < \frac{1}{C + 2D}$. But this is a contradiction since $|S \cup F'| \leq \gamma n$ and hence $|E_{|S \cup F'}| \leq (1 + \eta)|S \cup F'|$. $\square$

Now, because $r \leq R = \frac{\gamma n}{C + 2D}$, $|S \cup F| \leq \gamma n$ and we employ Lemma 23 to $T = S \cup F$ using the fact that $\mathbf{z}$ is ε^3-saturated.

We obtain vertex covers on $S \cup F, T^1, \ldots, T^m$ such that $\lambda_1 T^1 + \ldots + \lambda_m T^m = \mathbf{z} \mid_T$ where $\sum_{l=1}^m \lambda_l = 1$. Note that the values for the vertices in S are $0/1$ and are hence unchanged in all these solutions. To extend these solutions to fractional solutions over the whole graph, we look at each vertex j on the boundary of the set F and change the values of vertices upto a distance ℓ from it in $V \setminus (S \cup F)$ according to a splash around j. We first prove that all the vertices upto distance ℓ from the boundary of F have value $1/2 + \varepsilon$ in $\mathbf{z}$.

Claim 32 *For all $j \in F$, either*

- *all neighbors of j are in $S \cup F$, or*

- *For all $k \in N_{S \cup F}(j)$, $z_k = 1/2 + \varepsilon$*

PROOF: Assume not, then for some $j \in F$ which has some neighbor not in $S \cup F$, there exists $k \in N_{S \cup F}(j)$ such that $z_k \neq 1/2 + \varepsilon$. First, we show that it must be that $z_j = 1/2 + \varepsilon$. The only elements of $\mathbf{z}$ which do not have weight $1/2 + \varepsilon$ are elements of $N_S(l)$ for $l \in F$ and $N_S(i)$. However, $N'_S(i) \subseteq F \cup S$ so no element of $N_S(i)$ has a neighbor outside of F. Similarly, if $j \in N_S(l)$, then because $j \in F$, it must be that $N'_S(l) \subseteq F \cup S$ and thus j has no neighbors outside $S \cup F$.

So, say that $k \neq j$, then $k \notin S \cup F$. But there exists a path P of length $\leq \ell$ which avoids $S \cup F$ from j to k. Because $\mathbf{y}$ is r-canonical, and $\mathbf{z}$ is the same as $\mathbf{y}$ except possibly at the vertices in $N_S(i)$, it must be that $k \in N_S(i)$ or $k \in N_S(j')$ for some $j' \in S$. But, it cannot be that $k \in N_S(i)$ because $N_S(i) \subseteq F$. Also if $k \in N_S(j')$ for some $j' \in S$, then there is a path from j to j' length at most 2ℓ and so either k must be in $S \cup F$ or $j = j'$. The former cannot be true by assumption. The later cannot be true because $j \in F$ which is disjoint from S. $\square$

Create $\mathbf{y}^{(l)}$ as follows.

$$y_k^{(l)} = \begin{cases} \mathbf{Pr}\left[k = 1 \big| j = y_j^{(l)}\right] & k \in N_{S \cup F}(j) \text{ for some } j \in F \\ \\ y_k^{(l)} = z_i & \text{o. w.} \end{cases}$$

First note that this is well defined, because if any vertex were in $N_{S \cup F}(j)$ and $N_{S \cup F}(j')$ for $j, j' \in F$, $j \neq j'$, then there would be path between two vertices in F of length 2ℓ which does not go through $S \cup F$.

We wish to show that $\lambda_1 \mathbf{y}^{(1)}, \ldots, \lambda_m \mathbf{y}^{(m)} = \mathbf{z}$. Consider first some $k \in N_{S \cup F}(j)$ for some $j \in F$. First note that $\lambda_1 y_j^{(1)} + \ldots + \lambda_m y_j^{(m)} = z_j$. By Claim 32 if $k \neq j$, then it must be that $z_j = z_k = 1/2 + \varepsilon$. Therefore by Claim 17

$$\lambda_1 y_k^{(1)} + \ldots + \lambda_m y_k^{(m)} = z_j \mathbf{Pr}[k = 1 | j = 1] + (1 - z_j) \mathbf{Pr}[k = 1 | j = 0] = 1/2 + \varepsilon = z_k$$

If $k \notin \cup_{j \in F} N_{S \cup F}(j)$, then $y_k^{(l)} = z_k$ for all k, and so $\lambda_1 y_k^{(1)}, \ldots, \lambda_m y_k^{(m)} = z_k$. We now must show that for each k, $\mathbf{y}^{(k)}$ is an $(r + 1)$-canonical solution. We show that $T = S \cup F$ is a witness for $\mathbf{y}^{(k)}$.

Since the solution $T^{(k)}$ given by Lemma 23 is a vertex cover $\mathbf{y}_{|T}^{(k)} = T^{(k)}$ is a vertex cover for T. Also, by construction of F, there is no path of length less than 5ℓ between any vertices of $S \cup F$ using only vertices outside $S \cup F$. By Claim 31 $|T| = |S| + |F| \leq rC + kD + C + (2r + 2 - k)D = (r + 1)C + 2(r + 1)D$. If the number of paths added in constructing F is p, then $|T| \leq (r + 1)C + (k + p)D$. Also, as argued in Claim 31, $|E_{|S \cup F}| - |S \cup F| \geq (k - r) + p - 1 = (k + p) - (r + 1)$.

Finally, we need to show that $y_j^{(k)} = \mathbf{Pr}[j = 1 | j' = y_{j'}]$ if $j \in N_{S \cup F}(j')$ and $1/2 + \varepsilon$ otherwise. Let $y_j^{(k)} \neq 1/2 + \varepsilon$. Then either $j \in N_{S \cup F}(j')$ for some $j' \in F$ (since these vertices were set according to a splash pattern while creating $\mathbf{y}^{(k)}$) and we are done, or $z_k \neq 1/2 + \varepsilon$. However, $\mathbf{z} = \mathbf{Y}_i / y_i$ differs from $\mathbf{y}$ only in $N_S(i)$. Therefore, $z_k \neq 1/2 + \varepsilon$ in turn implies $j \in N_S(i)$ and hence $j \in F$, or $y_j \neq 1/2 + \varepsilon$. To finish off, we note that $y_j \neq 1/2 + \varepsilon$ would mean $j \in N_S(j')$ for some $j' \in S$ (by assumption on S). Since $N_S(j')$ is either contained in or disjoint with F, we must have $j \in S \cup F$ or $j \in N_{S \cup F}(j')$ respectively.

Since each $\mathbf{y}^{(k)}$ is an $(r + 1)$-canonical solution, by our inductive hypothesis $\forall 1 \leq k \leq m$ $\mathbf{y}^{(k)} \in N^{R-r-1}(VC(G))$ and hence $\mathbf{z} \in N^{R-r-1}(VC(G))$. Using a similar argument for show $\mathbf{w}$, we get that $\mathbf{y} \in N^{R-r}(VC(G))$. This completes the proof of Lemma 26. $\square$

2.6   Lower bounds for MaxCut

Let $G = (V, E)$ be a graph with n vertices and m edges. We prove a $1/2 + \zeta$ integrality gap for $\Omega(n)$ rounds of LS on MaxCut.

The solutions we define for MaxCut are simple extensions of vertex cover solutions. For a vector $\mathbf{y} \in \mathbb{R}^{n+1}$, we define an extension $Ext(\mathbf{y})$ as the vector $\mathbf{u} \in \mathbb{R}^{n+m+1}$ such that, $u_i = y_i \ \forall 0 \leq i \leq n$ and $u_e = 2y_0 - y_i - y_j$ for $e = (i, j) \in E$. Also, we define $Res(\mathbf{u})$ as the inverse operation i.e. the projection of the first $n + 1$ coordinates of $\mathbf{u}$. It is easy to verify that if $\mathbf{y} \in VC(G)$ then $Ext(\mathbf{y}) \in MC(G)$. Notice that with $R = \frac{\gamma n}{C + D}$ as defined in the previous section, it is sufficient to prove the following

Lemma 33 *If* $\mathbf{y} \in \mathbb{R}^{n+1}$ *is a 2r-canonical solution for VC(G), then* $Ext(\mathbf{y}) \in N^{R/2-r}(MC(G))$.

The integrality gap follows because $\mathbf{y} = (1, 1/2 + \varepsilon, \ldots, 1/2 + \varepsilon)$ is 0-canonical and for $\mathbf{u} = Ext(\mathbf{y})$, $\sum_{e \in E} u_e = (1 - 2\varepsilon)m$.

PROOF: We proceed by induction on $R/2 - r$. The base case follows because if $\mathbf{y}$ is an R-canonical solution, then $\mathbf{y} \in VC(G)$ which implies $Ext(\mathbf{y}) \in MC(G) = N^0(MC(G))$. For the inductive step, let $\mathbf{y}$ be an $2r$-canonical solution and let $\mathbf{u} = Ext(\mathbf{y})$. We create a protection matrix U, such that $\forall 1 \le i \le n$ and $\forall e \in E$, $Res(\mathbf{U}_i), Res(\mathbf{U}_e), Res(\mathbf{U}_0 - \mathbf{U}_i)$ and $Res(\mathbf{U}_0 - \mathbf{U}_e)$ can be expressed as convex combinations of $(2r+2)$-canonical solutions. This suffices because for a vector $\mathbf{u}$ if $Res(\mathbf{u}) = \lambda_1 \mathbf{u}^{(1)} + \ldots + \lambda_m \mathbf{u}^{(m)}$ then $\mathbf{u} = Ext(\lambda_1 \mathbf{u}^{(1)}) + \ldots + Ext(\lambda_m \mathbf{u}^{(m)})$, since the coordinates of $Ext(\mathbf{v})$ are affine functions of the coordinates of $\mathbf{v}$.

Let Y be the protection matrix of a $2r$-canonical solution as defined in the previous section. We define the matrix U as

$$\mathbf{U}_i = Ext(\mathbf{Y}_i) \qquad \forall 0 \le i \le n$$
$$\mathbf{U}_e = Ext(2\mathbf{Y}_0 - (\mathbf{Y}_i + \mathbf{Y}_j)) \quad \forall e = (i,j) \in E$$

We can write out the entries of U as follows, showing that it is symmetric.

$$U_{i,j} = Y_{ij} \qquad\qquad\qquad\qquad\qquad\qquad 0 \le i, j \le n$$
$$U_{i,e} = U_{e,i} = 2Y_{i0} - Y_{ij} - Y_{ik} \qquad\qquad\qquad 0 \le i \le n, \ e = (j,k) \in E$$
$$U_{e_1,e_2} = 4Y_{00} - 2(Y_{i0} + Y_{j0} + Y_{k0} + Y_{l0}) + (Y_{ik} + Y_{jk} + Y_{il} + Y_{jl}) \quad e_1 = (i,j), e_2 = (k,l) \in E$$

Note that for $i \in V$ and $e = (j,k) \in E$, $Res(\mathbf{U}_i) = \mathbf{Y}_i, Res(\mathbf{U}_0 - \mathbf{U}_i) = \mathbf{Y}_0 - \mathbf{Y}_i$ and $Res(\mathbf{U}_e) = \mathbf{Y}_0 - \mathbf{Y}_j + \mathbf{Y}_0 - \mathbf{Y}_k$, which are convex combinations of $(2r+1)$-canonical solutions as proved in the previous section. It only remains to tackle $Res(\mathbf{U}_0 - \mathbf{U}_e) = \mathbf{Y}_j + \mathbf{Y}_k - \mathbf{Y}_0$. We first prove that it is ε^3-saturated.

Claim 34 *If Y is the protection matrix of a 2r-canonical solution and $(i,j), (u,v)$ are two edges, then*

$$\frac{(\mathbf{Y}_i + \mathbf{Y}_j - \mathbf{Y}_0)_u}{y_i + y_j - y_0} + \frac{(\mathbf{Y}_i + \mathbf{Y}_j - \mathbf{Y}_0)_v}{y_i + y_j - y_0} \ge 1 + 4\varepsilon^3$$

PROOF: Without loss of generality, we can assume that j and u are the closer endpoints of the edges (i,j) and (u,v). We first handle the case when $d(j,u) > \ell$. Then $Y_{iu} = y_i y_u, Y_{iv} = y_i y_v, Y_{ju} = y_j y_u$ and $Y_{jv} = y_j y_v$. Hence, the LHS is $y_u + y_v$, which is greater than $1 + 2\varepsilon^2$ since a $2r$-canonical solution is ε^2 saturated.

When $d(j,u) \le \ell$, all the four vertices are within distance $\ell + 2$ of each other. Now, in any subgraph H of diameter 3ℓ, we may think of the restriction of $\mathbf{y}$ to H as the probabilities of the vertices being present in a distribution over vertex covers of H. Notice that if $\mathbf{y}$ is a $2r$-canonical solution, H may contain vertices close to (within distance ℓ of) at most one fixed vertex. In case there is such a vertex r, $\forall i \in H$ $y_i = \mathbf{Pr}[i = 1 | r = 1]$. If there is no such vertex, all vertices in H have $y_i = 1/2 + \varepsilon$ and we can these as probabilities for a distribution which chooses a 1-splash with probability $1/2 + \varepsilon$ and 0-splash with probability $1/2 - \varepsilon$ around any arbitrary vertex in H (Claim 17). Also, we can interpret Y_{pq} as $\mathbf{Pr}[p = 1, q = 1]$ for the same distribution as above.

Consider the distribution over the subgraph within a radius $\ell + 2$ from i. We first note that since $(\mathbf{Y}_0 - \mathbf{Y}_i)/(1 - y_i)$ is a valid vertex cover solution and $(\mathbf{Y}_0 - \mathbf{Y}_i)_i = 0$, $(\mathbf{Y}_0 - \mathbf{Y}_i)_j/(1 - y_i) = 1$ which gives $y_i + y_j - 1 = Y_{ij}$. Using this and the fact that $\mathbf{Pr}[(i = 1) \vee (j = 1)|u = 1] = 1$, we have

$$
\begin{aligned}
\frac{(\mathbf{Y}_i + \mathbf{Y}_j - \mathbf{Y}_0)_u}{y_i + y_j - y_0} &= \frac{y_u(\mathbf{Pr}[i = 1|u = 1] + \mathbf{Pr}[j = 1|u = 1] - 1)}{\mathbf{Pr}[i = 1, j = 1]} \\
&= \frac{y_u \, \mathbf{Pr}[(i = 1) \wedge (j = 1)|u = 1]}{\mathbf{Pr}[i = 1, j = 1]} \\
&= \mathbf{Pr}[u = 1|i = 1, j = 1]
\end{aligned}
$$

Therefore, we get

$$
\begin{aligned}
\frac{(\mathbf{Y}_i + \mathbf{Y}_j - \mathbf{Y}_0)_u}{y_i + y_j - y_0} + \frac{(\mathbf{Y}_i + \mathbf{Y}_j - \mathbf{Y}_0)_v}{y_i + y_j - y_0} - 1 &= \mathbf{Pr}[u = 1|i = 1, j = 1] + \mathbf{Pr}[v = 1|i = 1, j = 1] - 1 \\
&= \mathbf{Pr}[(u = 1) \wedge (v = 1)|i = 1, j = 1] \\
&= \mathbf{Pr}[(u = 1) \wedge (v = 1)|j = 1]
\end{aligned}
$$

The last equality following from the fact that it is sufficient to condition on the closer of the two vertices i and j. Also,

$$
\begin{aligned}
\mathbf{Pr}[(u = 1) \wedge (v = 1)|j = 1] &= \mathbf{Pr}[u = 1|j = 1] + \mathbf{Pr}[v = 1|j = 1] - 1 \\
&= \frac{Y_{uj}}{y_j} + \frac{Y_{vj}}{y_j} - 1 \\
&\geq 4\varepsilon^3 \qquad \text{(by Lemma 28)}
\end{aligned}
$$

$\square$

We now want to express $\mathbf{w} = (\mathbf{Y}_i + \mathbf{Y}_j - \mathbf{Y}_0)/(y_i + y_j - 1)$ as a convex combination of $(2r + 2)$-canonical solutions. Let S be the witness to $\mathbf{y}$ being $2r$-canonical. We now find a set $T \supseteq S$ such that $\mathbf{w}$ is a convex combination of solutions $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(m)}$ which take $0/1$ values over T and which are $(2r + 2)$-canonical, with T being the witness. There are two cases:

Case 1: $i \notin S$ *and* $\exists r \in S$ *s.t.* $d(i, r) \leq 5\ell$ (with $d(i, r)$ being length of the shortest path not passing through S)

By the proof in the previous section, we know that the vector $z = \mathbf{Y}_i/y_i$ is a convex combination of $(2r + 1)$-canonical solutions with a set S_1 being the witness for all of them. Also, $j \in S_1$ as it includes every vertex within distance ℓ of i. We take $T = S_1$.

Case 2: $i \in S$ *or* $d(i, r) > 5\ell \; \forall r \in S$

In this case $\mathbf{z} = \mathbf{Y}_i/y_i$ is $(2r + 1)$-canonical with $S \cup \{i\}$ the witness. We now look at the protection matrix Z for $\mathbf{z}$ and consider the vector $\mathbf{z}' = \mathbf{Z_j}/z_j$. This is a convex combination of $(2r + 2)$-canonical solutions having a common witness S_2 which contains $S \cup \{i\}$. Take $T = S_2$.

In both cases $|T| \leq (2r + 2)C + (4r + 4)D$. We now employ Lemma 23 to T to obtain vertex covers $T^1, \ldots, T^m$ on T such that $\lambda_1 T^1 + \ldots + \lambda_m T^m = \mathbf{w}_{|T}$ with $\sum_{l=1}^m \lambda_l = 1$.

We can extend them to create $(2r+2)$-canonical solutions $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(m)}$ as in the previous section. By the arguments in the previous section, all these have T as the witness. This completes the proof. $\square$

2.A   Proof of Lemma 10

Lemma 35 *For every $0 < \alpha < 1/125$, $\eta > 0$, there exists a $d = d(\alpha) \in \mathbb{N}$, δ, $\gamma > 0$, and $N \in \mathbb{N}$ such that for $n \geq N$ there exists an $(\alpha, \delta, \gamma, \eta)$ graph with max cut less than $\frac{1}{2}|E|(1+\alpha)$ and maximum degree at most d on n vertices. Here $d(\alpha)$ is an explicit function that depends only on α.*

We use the following lemma from [3]

Lemma 36 *For every $1 < \alpha < 1/250$, $\eta > 0$, there exists a $\delta, \gamma > 0$ such that a random graph from the $G_{n,p}$ distribution where $p = \alpha^{-2}/n$ has the following properties with probability $1 - o(n)$:*

- *after $O(\sqrt{n})$ edges are removed, the girth is $\delta \log n$.*

- *the minimum vertex cover contains at least $(1-\alpha)n$ vertices*

- *every induced subgraph on a subset S of at most γn vertices has at most $(1+\eta)|S|$ edges.*

PROOF:[of Lemma 35] Given $\alpha, \eta > 0$, set $\alpha' = \alpha/2$. Use Lemma 36 with inputs α', η to randomly pick a graph on n vertices. Set $p = (\alpha')^{-2}/n$ as in Lemma 36. Now, with high probability, we can remove set of edges R to obtain a $(\alpha/2, \delta, \gamma, \eta)$-graph on n vertices. Do not yet remove edges.

Also, it is well known that w.h.p. the max cut in a random $G_{n,p}$ has size less than $\frac{1}{2}|E|(1+1/\sqrt{d})$, where d is the average degree. The average degree of a vertex in this model is $\lambda = pn = 4\alpha^{-2}$. Hence the size of the max cut is at most $\frac{1}{2}|E|(1+\alpha/2)$. The probability that some fixed vertex v_0 has degree greater than 2λ is less than $\exp(-\lambda/3)$ by a Chernoff bound. So by Markov's inequality the probability that more than $\exp(-\lambda/6)n$ vertices have degree greater than 2λ is at most $\exp(-\lambda/6) \leq exp(-10000)$.

If this is the case, then first remove the edge set R. By removing edges we could only decrease the maximum degree. Then simply remove all vertices with degree more than 2λ from the graph and any other subset to obtain a graph G' with $n(1 - exp(-d/6))$ vertices. Now, it is easy to check that G' is a $(\alpha, \delta, \gamma, \eta)$-graph with maximum degree at most $d(\alpha) = 2\lambda = 8/\alpha^2$. Removing the edges and vertices changes the max cut to $\frac{1}{2}|E|(1 + \alpha/2 + o(1)) < \frac{1}{2}|E|(1+\alpha)$. $\square$

2.B   Proofs of claims about splashes

We use the following notation for the proofs in this appendix. We denote $\mathbf{Pr}[i = 1|r = b]$ and $\mathbf{Pr}[i = 1, j = 1|r = b]$ by $P_r^b(i)$ and $P_r^b(i,j)$ respectively. $\mathbf{Pr}[i = 0|r = b]$ and

$\mathbf{Pr}[i=0, j=1 | r=b]$ are expressed as $1 - P_r^b(i)$ and $P_r^b(j) - P_r^b(i,j)$ respectively. Also, in cases where $\mathbf{Pr}[j=1 | i=b]$ depends only on $d(i,j)$, we denote it by $Q^b(d(i,j))$.

Claim 37 *Consider a b-Splash around a vertex i such that all vertices upto distance ℓ are labeled $\frac{1}{2} + \varepsilon$. Then,*

1. $Q^1(k) = (1/2 + \varepsilon) \left[1 + (-1)^k \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{k+1} \right]$ *for* $0 \le k \le \ell$

   $Q^0(0) = 0$ *and* $Q^0(k) = Q^1(k-1)$ *for* $1 \le k \le \ell$

2. $\left| Q^0(\ell/2) - (1/2 + \varepsilon) \right| \le \varepsilon^4$

3. $\forall 0 \le k \le \ell,\ Q^1(k) + Q^1(k+1) \ge 1 + 4\varepsilon^2$

PROOF: We prove the formula for $Q^1(k)$ by induction. For $k = 0$,

$$(1/2 + \varepsilon) \left[1 + (-1)^k \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{k+1} \right] = (1/2 + \varepsilon) \left[\frac{1}{1/2 + \varepsilon} \right] = 1 = Q^1(0)$$

Assuming the correctness of the formula for $k = n$, we start with the recurrence

$$Q^1(n+1) = (1 - Q^1(n)) + \left(\frac{2\varepsilon}{1/2 + \varepsilon} \right) Q^1(n) = 1 - \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right) Q^1(n)$$

since the vertex at distance n (in the same path) might not be present with probability $1 - Q^1(n)$ in which case the one at distance $n+1$ is present with probability 1, and it is present with probability $Q^1(n)$ in which case the one at distance $n+1$ is included with probability $\left(\frac{2\varepsilon}{1/2 + \varepsilon} \right)$. Therefore, we have

$$Q^1(n+1) = 1 - \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right) (1/2 + \varepsilon) \left[1 + (-1)^n \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{n+1} \right]$$

$$= 1 - (1/2 - \varepsilon) + (-1)^{n+1} (1/2 + \varepsilon) \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{n+2} = (1/2 + \varepsilon) \left[1 + (-1)^{n+1} \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{n+2} \right]$$

Also note that if i is labeled 0, then all its neighbors must be set to 1. Hence $Q^0(0) = 0$ and $Q^0(1) = 1$. The rest of the induction works exactly as above.

Note that

$$\left| Q^0(\ell/2) - (1/2 + \varepsilon) \right| = (1/2 + \varepsilon) \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{\ell/2} < (1 - 2\varepsilon)^{\ell/2} = (1 - 2\varepsilon)^{(\frac{4}{\varepsilon} \log \frac{1}{\varepsilon})} \le \varepsilon^4$$

Finally for $0 \le k < \ell$,

$$Q^1(k) + Q^1(k+1) = (1/2 + \varepsilon) \left[2 + (-1)^k \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{k+1} \left(1 - \frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right) \right]$$

$$= (1/2 + \varepsilon) \left[2 + (-1)^k \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^{k+1} \left(\frac{2\varepsilon}{1/2 + \varepsilon} \right) \right]$$

$$\ge (1/2 + \varepsilon) \left[2 - \left(\frac{2\varepsilon}{1/2 + \varepsilon} \right) \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^2 \right] = 1 + 2\varepsilon - 2\varepsilon \left(\frac{1/2 - \varepsilon}{1/2 + \varepsilon} \right)^2 \ge 1 + 4\varepsilon^2$$

The claim for $k = \ell$ follows from part 2 and the fact that $Q^1(d) = 1/2 + \varepsilon$ for $d > \ell$. $\square$

Claim 38 *If we pick a 0-Splash with probability $1/2 - \varepsilon$ and a 1-Splash with probability $1/2 + \varepsilon$, then all vertices have probability $1/2 + \varepsilon$. Furthermore, vertices at distance $\ell + 1$ or more from i have weight $1/2 + \varepsilon$ in the 0-Splash as well as 1-Splash around i.*

PROOF: We prove it by induction on the length of the path from i to j. Let $P_i(j) = (1/2 - \varepsilon)P_i^0(j) + (1/2 + \varepsilon)P_i^1(j)$. The base case, when the path is of length 0 is clear. If the path between i and j is $i = v_0, v_1, \ldots, v_{m-1}, v_m = j$, then there are two cases. In the first case v_{m-1} and v_m are both within distance ℓ of i. Then

$$P_i(j) = 1 - (1 - \frac{2\varepsilon}{1/2 + \varepsilon})P_i(v_{m-1})$$

because v_m is only excluded with probability $\frac{2\varepsilon}{1/2+\varepsilon}$ when v_{m-1} is present and this event is independent of whether or not each vertex $i = v_0, v_1, \ldots, v_{m-1}$ is included in the cover. By induction, $P_i(v_{m-1}) = 1/2 + \varepsilon$, and so $1 - (1 - \frac{2\varepsilon}{1/2+\varepsilon})P_i(v_{m-1}) = 1/2 + \varepsilon$.
In the second case v_{m-1} is at distance ℓ. However,

$$P_i^b(j) = 1 - (1 - \frac{P_i^b(v_{m-1}) - (1/2 - \varepsilon)}{P_i^b(v_{m-1})})P_i^b(v_{m-1}) = 1/2 + \varepsilon$$

because the probability v_{m-1} is included in a b-Splash is $P_i^b(v_{m-1})$ and the probability of including v_m when v_{m-1} is present is $\frac{P_i^b(v_{m-1}) - (1/2 - \varepsilon)}{P_i^b(v_{m-1})}$. $\square$

Claim 39 *Let $i = v_0, v_1, \ldots, v_{m-1}, v_m = j$ be the path to j, $m \leq \ell$, and let u be the vertex on this path which is closest to r. Then*

1. $P_r^b(i, j) = P_r^b(u) \cdot P_u^1(i)P_u^1(j) + [1 - P_r^b(u)] \cdot P_u^0(i)P_u^0(j)$

2. *If $P_r^b(u) = 1/2 + \varepsilon$, then $P_r^b(i, j) = (1/2 + \varepsilon)P_i^1(j)$*

PROOF:

1. Let E be the event that both i and j are in a vertex cover and $r = b$. Then $P_r^b(i, j) = \Pr[E \mid r = b]$. We can also condition on whether u is in the vertex cover.

$$\begin{aligned} P_r^b(i, j) &= \Pr[u \in VC \mid r = b] \cdot \Pr[E \mid r = b \text{ and } u \in VC] \\ &+ \Pr[u \notin VC \mid r = b] \cdot \Pr[E \mid r = b \text{ and } u \notin VC] \end{aligned}$$

But $\Pr[E \mid r = b \text{ and } u \in VC] = \Pr[E \mid u \in VC]$. Because given that u is in or out of the vertex cover, we can determine if i and j are in the vertex cover by following the edges from u to each of them. But this information is independent of whether r is in the vertex cover. For the same reason $\Pr[E \mid r = b \text{ and } u \in VC] = \Pr[E \mid u \in VC]$. Therefore

$$P_r^b(i, j) = P_r^b(u) \cdot P_u^1(i)P_u^1(j) + [1 - P_r^b(u)]\dot{P}_u^0(i)P_u^0(j)$$

as claimed.

2. The probability that i and j are in a vertex cover (assume r is not yet fixed) is just $(1/2 + \varepsilon)P_i^1(j)$. Now, we can just condition on l, and rewrite this as

$$\Pr[u \in VC] \cdot P_u^1(i,j) + \Pr[u \notin VC] \cdot P_u^0(i,j)$$

We can also not condition on $r = b$ because once l is fixed, that does not affect anything, and in addition, $\Pr[u \in VC] = 1/2 + \varepsilon = P_r^b(u)$. So this becomes

$$P_r^b(u) \cdot P_u^1(i,j) + [1 - P_r^b(u)] \cdot P_u^0(i,j)$$

Finally, if we note that $P_u^b(i,j) = P_u^b(i)P_u^b(j)$, we see that we get

$$P_r^b(l) \cdot P_u^1(i)P_u^1(j) + [1 - P_r^b(u)] \cdot P_u^0(i)P_u^0(j)$$

which by 1) is simply $P_r^b(i,j)$ as claimed.

$\square$

Claim 40 *Let i be a vertex and (j,k) be an edge in a b-Splash around r. Then if j and k are not already fixed*

$$P_r^b(i,j) + P_r^b(i,k) \geq P_r^b(i)(1 + 4\varepsilon^3)$$

and

$$[P_r^b(j) - P_r^b(i,j)] + [P_r^b(k) - P_r^b(i,k)] \geq (1 - P_r^b(i))(1 + 4\varepsilon^3)$$

PROOF: We consider separately the cases when (j,k) lies on or outside the path between r and i.

Case 1: (j,k) **lies outside the path connecting r and i**

Without loss of generality, let j be the vertex closer to the path from r to i. Let u be the vertex in the path closest to j. Then by Claim 39

$$P_r^b(i,j) = P_r^b(u) \cdot P_u^1(i)P_u^1(j) + [1 - P_r^b(u)] \cdot P_u^0(i)P_u^0(j)$$
$$P_r^b(i,k) = P_r^b(u) \cdot P_u^1(i)P_u^1(k) + [1 - P_r^b(u)] \cdot P_u^0(i)P_u^0(k)$$

Therefore,

$$P_r^b(i,j) + P_r^b(i,k) = P_r^b(u)P_l^1(i) \cdot \left[P_u^1(j) + P_u^1(k) \right] + [1 - P_r^b(u)]P_u^0(i) \cdot \left[P_u^0(j) + P_u^0(k) \right]$$

Also by Claim 37 we know that $P_u^b(j) + P_u^b(k) \geq 1 + 4\varepsilon^2$, if j and k are not already fixed, which gives

$$P_r^b(i,j) + P_r^b(i,k) \geq \left[P_r^b(u)P_u^1(i) + [1 - P_r^b(u)]P_u^0(i) \right] (1 + 4\varepsilon^2) = P_r^b(u)(1 + 4\varepsilon^2)$$

Case 2: (j, k) **lies on the path connecting** r **and** i

Let j be the vertex closer to r. Also, let $\alpha = P_r^b(j)$ and $\beta = P_j^1(i)$. Then,

$$P_r^b(i, j) = P_r^b(j)P_j^1(i) = \alpha\beta$$

$$P_r^b(i, k) = P_r^b(k)P_k^1(i) = \left[1 - \alpha + \frac{2\varepsilon}{1/2 + \varepsilon}\alpha\right]\left[(1 - \beta)\frac{1/2 + \varepsilon}{1/2 - \varepsilon}\right]$$

$$= (1 - \alpha)(1 - \beta)\left(\frac{1/2 + \varepsilon}{1/2 - \varepsilon}\right) + \alpha(1 - \beta)\left(\frac{2\varepsilon}{1/2 - \varepsilon}\right)$$

where the second equation follows from the recurrence $Q^1(n+1) = (1 - Q^1(n)) + \left(\frac{2\varepsilon}{1/2 + \varepsilon}\right)Q^1(n)$ used in Claim 37. Also,

$$P_r^b(i) = P_r^b(j)P_j^1(i) + (1 - P_r^b(j))P_j^0(i) = P_r^b(j)P_j^1(i) + (1 - P_r^b(j))P_k^1(i)$$

$$= \alpha\beta + (1 - \alpha)(1 - \beta)\left(\frac{1/2 + \varepsilon}{1/2 - \varepsilon}\right)$$

This gives

$$\frac{P_r^b(i, j) + P_r^b(i, j)}{P_r^b(i)} = 1 + \frac{\alpha(1 - \beta)\left(\frac{2\varepsilon}{1/2 - \varepsilon}\right)}{\alpha\beta + (1 - \alpha)(1 - \beta)\left(\frac{1/2 + \varepsilon}{1/2 - \varepsilon}\right)} \geq 1 + 4\varepsilon^3$$

since $\alpha, (1 - \beta) > 2\varepsilon$ (all probabilities in a splash are at least 2ε, unless one is 0 and the other is 1, but then both are fixed).

The proof of the second statement follows similarly. $\square$

Claim 41 *Let i and j be two vertices in a b-Splash around r, such that $d(i, j) \geq \ell$. Then*

$$\left|P_r^b(i, j) - P_r^b(i)P_r^b(j)\right| \leq 2\varepsilon^4$$

and

$$\left|[P_r^b(j) - P_r^b(i, j)] - (1 - P_r^b(i))P_r^b(j)\right| \leq 2\varepsilon^4$$

PROOF: Let u be the vertex closest to r on the path from i to j. Without loss of generality, assume that $d(i, u) \geq \ell/2$. Then

$$\left|P_r^b(i, j) - P_r^b(i)P_r^b(j)\right| = \left|P_r^b(u) \cdot P_u^1(i)P_u^1(j) + [1 - P_r^b(u)] \cdot P_u^0(i)P_u^0(j) - P_r^b(i)P_r^b(j)\right|$$

$$\leq \left|(1/2 + \varepsilon)\left[P_r^b(u) \cdot P_u^1(j) + [1 - P_r^b(u)] \cdot P_u^0(j)\right] - P_r^b(i)P_r^b(j)\right| + \varepsilon^4$$

$$= \left|(1/2 + \varepsilon)P_r^b(j) - P_r^b(i)P_r^b(j)\right| + \varepsilon^4 \leq 2\varepsilon^4$$

where the two inequalities follow from the fact that $|P_r^b(i) - (1/2 + \varepsilon)| \leq \varepsilon^4$ if $d(i, r) \geq \ell/2$ as proved in Claim 37.

The second statement can be proven in a similar fashion. $\square$

Chapter 3

Linear Round Integrality Gaps for Lasserre

Background Results   Sufficiently dense random k-CSP formulae are far from being satisfiable as the next proposition states.

Proposition 42 *For any $\delta > 0$, with probability $1 - o(1)$, if φ is a random k-CSP-$\mathcal{P}$ with Δn clauses where $\Delta \geq \frac{\ln 2}{2\delta^2} + 1$, at most a $r(\mathcal{P}) + \delta$ fraction of the clauses of φ can be simultaneously satisfied.*

Proposition 42 is well known in the literature, we provide a proof in the appendix for completion.

Definition 43 Width-w resolution *on an XOR formula φ, successively builds up new clauses by deriving a new clause $\bigoplus_{i \in I \Delta J} x_i = b \oplus b'$ whenever the symmetric difference $|I \Delta J| \leq w$ and the clauses $\bigoplus_{i \in I} x_i = b$ and $\bigoplus_{i \in I} x_i = b'$ had either already been derived or belong to φ.*

Width-w resolution proves a formula φ unsatisfiable if it derives the clause $0 = 1$. The following theorem shows that for random 3-XOR formula, even for quite large w, width-w resolution fails to produce a contradiction.

Theorem 44 *For any $k \geq 3$, $d > 0$, $\gamma > 0$, and $0 \leq \varepsilon < k/2 - 1$, there exists some constant $\alpha > 0$, such that if φ is a random k-XOR formula with density dn^ε, then with probability $1 - o(1)$ φ cannot be disproved by width $\alpha n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}}$ resolution nor can any variable be resolved to true or false. Furthermore, this is true even if the parity sign (whether the predicate is parity or its negation) of each clause is adversatively chosen.*

Wigderson and Ben-Sasson [6] show that a variant of Theorem 44 holds for k-SAT formula. The proof of [6] extends to show Theorem 44 using standard techniques. We include a proof in the appendix for completeness.

3.1 k-CSPs over XOR-Implied Predicates

We now present the main theorem of the chapter.

Theorem 45 *Let $\mathcal{P}$ be a XOR-implied predicate. Then for every $\delta, \gamma, d > 0$ and $0 \leq \varepsilon < k/2 - 1$ (such that if $\varepsilon = 0$, then $d \geq \frac{\ln 2}{2\delta^2} + 1$) there exists some constant $\alpha \geq 0$, such that with probability $1 - o(1)$, if φ is a random k-CSP-$\mathcal{P}$ with Δn clauses where $\Delta = dn^\varepsilon$ both the following are true:*

1. *at most a $r(\mathcal{P}) + \delta$ fraction of the clauses of φ can be simultaneously satisfied.*

2. *The $\alpha n^{1 - \frac{\varepsilon}{k/2 - 1 - \gamma}}$ level of the Lasserre hierarchy permits a feasible solution.*

This theorem implies integrality gaps for XOR-implied k-CSPs because the Lasserre relaxation cannot refute that all clauses can be simultaneously satisfied, but, in fact, at most $r(\mathcal{P}) + \delta$ clauses can be simultaneously satisfied. Notice that an algorithm that simply guesses a random assignment would expect to satisfy an $r(\mathcal{P})$ fraction of clauses in expectation. In particular this theorem shows that with high probability a random k-XOR formula with n variables and dn clauses, where d is any constant, cannot be refuted by $\Omega(n)$ rounds of Lasserre which gives an integrality gap of $1/2 + \delta$ for $\Omega(n)$ rounds of Lasserre for Max-k-XOR by setting $\delta = \delta$; $d \geq \frac{\ln 2}{2\delta^2} + 1$; $\varepsilon = 0$; and $\gamma = \frac{1}{2}$. Also, this theorem shows that with high probability a random 3-CNF formula cannot be refuted by $\Omega(n)$ rounds of Lasserre which gives an integrality gap of $7/8 + \delta$ for $\Omega(n)$ rounds of Lasserre for Max-k-SAT.

Theorem 45 follows almost immediately from Theorem 44, Proposition 42, and the following Lemma.

Lemma 46 (Main Lemma) *If a k-XOR formula φ cannot be disproved by width-w resolution, then the $\frac{w}{2}$th round of the Lasserre hierarchy permits a feasible solution.*

PROOF:[of Theorem 45] Fix $\delta, \gamma, d, \varepsilon, \mathcal{P}$ as allowed in theorem statement, and let φ be a random k-CSP-$\mathcal{P}$ formula with Δn clauses where $\Delta = dn^\varepsilon$. By Proposition 42, 1) holds with probability $1 - o(1)$ because for sufficiently large n, $\Delta = dn^\varepsilon > \frac{\ln 2}{2\delta^2} + 1$.

We can write φ as a k-XOR formula φ_{XOR} so that $\varphi_{XOR} \Rightarrow \varphi$. Now the Lasserre relaxation for φ_{XOR} is strictly tighter than that for φ. Let α' be as guaranteed in Theorem 44 using k, d, γ, and ε as inputs so that by Theorem 44 we know that with probability $1 - o(1)$ it is the case that φ_{XOR} cannot be disproved by width-$\alpha' n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}}$ resolution. Let $\alpha = \frac{\alpha'}{2}$. By Lemma 46, φ_{XOR} cannot be proven unsatisfiable by $\frac{\alpha'}{2} n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}} = \alpha n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}}$ rounds of Lasserre. Because the Lasserre relaxation for φ_{XOR} is tighter than that for φ it must be the case that φ cannot be proven unsatisfiable by Lasserre either. $\square$

Lemma 46 is the main original technical contribution of this work. In the rest of this section we first provide some intuition for the proof of Lemma 46 and then provide its proof.

For a first attempt to prove the lemma we can observe that for any particular set I of at most $w/2$ variables, we can construct vectors for all f as follows: 1) Run bounded

width resolution to derive a set of constraints that any satisfying assignment must satisfy.
2) Consider the set SAT_I where

$$SAT_I = \left\{ x_I \in \mathbf{x}_I : \begin{array}{l} x_I \text{ satisfies all the constraints derived by} \\ \text{the resolution whose support is contained in } I \end{array} \right\}$$

Randomize over SAT_I and construct the vectors as we saw in Equation 1.11. That is each coordinate of v_{f_I} will correspond to an element of $x_I \in SAT_I$, and will be $\sqrt{1/|SAT_I|}$ if $f_I(x_I) = 1$ and 0 if $f_I(x_I) = 0$. These vectors will satisfy the Lasserre Equations 1.7, 1.8, and 1.10; however, these vectors will fail miserably to satisfy Equation 1.9 of the Lasserre constraints. We have set up valid local distributions; however, these distribution do not patch together consistently. The problem is that when the take the dot product of v_{x_I} and v_{x_J}, the values in each coordinate mean something completely different.

To remedy this misalignment we design a space of equivalence classes of characters of weight at most $w/2$ variables which we will use to index the coordinates of each vector. We will say that $\chi_I \sim \chi_J$ if for all assignments that satisfy the derived resolution clauses, χ_I determines χ_J and vice versa. For example, if φ contained the clause $x_1 \oplus x_2 \oplus x_3 = 0$ then $\chi_{\{1,2\}} \sim \chi_{\{3\}}$ because whatever $x_1 \oplus x_2$ is, x_3 must be the opposite. With some $\sim$ equivalent characters, fixing one character automatically fixes the $\sim$ equivalent character to the opposite value (as above). With other $\sim$ equivalent character, fixing one character automatically fixes the $\sim$ equivalent character to the same value. Using this fact, we can split each equivalence class of $\sim$ equivalent character into two parts, so that the $\sim$ equivalent clauses in each part always fix each other to the same value, and $\sim$ equivalent clauses in opposite parts always fix each other to the opposite value. We can arbitrarily label one part $+$ and the other $-$.

The vector corresponding to a function f will have in each coordinate (which corresponds to an an equivalence class of characters) the sum of the fourier coefficients of f of characters corresponding the characters in this equivalence class. (Each coefficient will be multiplied by ± 1 depending on its label). The intuition here is that characters of the same equivalence class are completely dependant on each other, but non-equivalent characters are completely independent. Note that only some of the coordinates are non-zero.

This relates to the aforementioned construction which satisfies equations 1.7, 1.8, and 1.10 because "locally" we have just taken a rotation! If we project onto only the relevant characters, then the mapping of our previously constructed vectors (that failed to satisfy Equation 1.9) to these new vectors is simply a rotation. This implies that all the Lasserre equations that were previously satisfied will still be satisfied (because all the irrelevant characters are set to 0 and thus will not affect the dot product).

For each $I \subseteq [n]$, $|I| \leq \frac{w}{2}$, there is a bijection between the set SAT_I and the equivalence classes of χ_J where $J \subseteq I$, because, intuitively, each time resolution derives a new relation, the dimension of each of these sets is reduced by 1.

In particular the vectors $v_{f_{(I)}^{=x_I}}$ for each $x_I \in SAT_I$ still form an orthogonal basis. And if you take the preimage of the vector v_0 (I is still fixed) it corresponds to randomizing over the $x_I \in SAT_I$.

One can develop this intuition into a proof by showing that if $f \in \mathfrak{F}^I$ and $g \in \mathfrak{F}^J$ then v_f and v_g behave well by projecting onto the classes containing the characters involving

only variables of $I \cup J$ (these are the only possible non-zero coordinates), and rotating back into the basis of $|SAT_{I \cup J}|$. A previous proof follows this intuition (see [24]). Here we present an easier proof of Lemma 46.

PROOF:[Lemma 46]

Construction of Vectors

We first define a set $\mathcal{E}$ which later will be used to index the coordinates of the vectors. Let φ be a k-XOR formula that has no width-w resolution. Let $\mathcal{C}$ be the collection of clauses generated by width-w resolution running on φ. Let $\mathcal{L}^w$ be all the characters of weight at most w. Let $\mathcal{F} \subseteq \mathcal{L}^w$ be the collection of linear functions cooresponding to the clauses of $\mathcal{C}$. That is if $\bigoplus_{i \in I} x_i = b_i \in \mathcal{C}$ then $\chi_I \in \mathcal{F}$.

Now consider the set $\mathcal{E} = \mathcal{L}^{\frac{w}{2}}/\mathcal{F}$. That is we partition $\mathcal{L}^{\frac{w}{2}}$ into equivalence classes where

$$\chi_I \sim_{\mathcal{F}} \chi_J \Leftrightarrow \chi_{I \Delta J} \in \mathcal{F}$$

For each equivalence class $[\chi_I] \in \mathcal{E}$, we arbitrarily choose some $\chi_{I_0} \in [\chi_I]$ (for notational convenience, we always choose $\chi_\emptyset = \chi_{\emptyset_0} \in [\chi_\emptyset]$). We define a function $\pi : \mathcal{L}^{|w/2|} \cup \mathcal{F} \to \{+1, -1\}$ such that

$$\pi(\chi_I) = \begin{cases} +1 & \bigoplus_{i \in I \Delta I_0} x_i = 0 \in \mathcal{C} \\ -1 & \bigoplus_{i \in I \Delta I_0} x_i = 1 \in \mathcal{C} \end{cases}$$

Claim 47 $\sim_F$ *is an equivalence relation and π is well defined.*

We now define the vectors. Each vector will have a coordinate corresponding to the each element of $\mathcal{E}$. Let $e_{\chi_I} = \pi(\chi_I)e_{[\chi_I]}$ (where $e_{[\chi_I]}$ is the basis vector with a one in the coordinate corresponding to $[\chi_I]$). Let $f \in \mathfrak{F}^{\frac{w}{2}}$

$$v_f = \sum_{\chi \in \mathcal{L}^{\frac{w}{2}}} \hat{f}(\chi)e_{[\chi]}$$

so that

$$v_f([\chi_I]) = \sum_{\chi \in [\chi_I]} \pi(\chi)\hat{f}(\chi)$$

Proof that constructed vectors satisfy Lasserre constraints

We see that Equations 1.7 is satisfied by the observation that the fourier expansion of $\vec{1}$ is 1 in the trivial character and 0 everywhere else. Therefore $v_0 = (1, 0, \ldots, 0)$ where the 1 is in the coordinate of $[\emptyset]$. Therefore $||v_0||^2 = 1$.

We show 1.8 is satisfied. If $C \in \varphi$ then $C \in \mathcal{C}$.

First assume that C is $\bigoplus_{i \in I} x_i = 1$. Then we must show that $||v_f||^2 = 1$ where $f = \bigoplus_{i \in I} x_i$. We note that $\chi_I \in \mathcal{F}$, $\pi(\chi_I) = -1$, and also recall that $f(x) = \frac{1}{2}\chi_\emptyset - \frac{1}{2}\chi_I = \frac{1}{2} - \frac{1}{2}\chi_I$. Thus $v_f = \frac{1}{2}e_{[\chi_\emptyset]} - \pi(\chi_I)\frac{1}{2}e_{[\chi_I]} = e_{[\chi_\emptyset]}$ because $\chi_\emptyset \sim_{\mathcal{F}} \chi_I$.

Similarly, let $C = \bigoplus_{i \in I} x_i = 0$. Then we must show that $||v_f||^2 = 1$ where $f = 1 - \bigoplus_{i \in I} x_i$, because f is a function that is to be always *satisfied*. We note that $\chi_I \in \mathcal{F}$, $\pi(\chi_I) = 1$, and also recall that $f(x) = 1 - (\frac{1}{2}\chi_\emptyset - \frac{1}{2}\chi_I) = \frac{1}{2} + \frac{1}{2}\chi_I$. Thus $v_f = \frac{1}{2}e_{[\chi_\emptyset]} + \pi(\chi_I)\frac{1}{2}e_{[\chi_I]} = e_{[\chi_\emptyset]}$ because $\chi_\emptyset \sim_{\mathcal{F}} \chi_I$.

Equation 1.9 is satisfied because we can write $\langle v_f, v_g \rangle$ in terms of only $\widehat{f \cdot g}$, the fourier coefficients of $f \cdot g$. It will follow that if $f \cdot g \equiv f' \cdot g'$ then $\langle v_f, v_g \rangle = \langle v_{f'}, v_{g'} \rangle$ because the fourier coefficients of $f \cdot g$ and $f' \cdot g'$ are the same. For $[I] \in \mathcal{E}$, let $\hat{f}([\chi_I]) = \sum_{\chi \in [\chi_I]} \pi(\chi_I) \hat{f}(\chi)$. Then

$$
\begin{aligned}
\langle v_f, v_g \rangle &= \sum_{[\chi_I] \in \mathcal{E}} \langle \hat{f}([\chi_I]), \hat{g}([\chi_I]) \rangle \\
&= \sum_{\chi \in \mathcal{L}^{\frac{w}{2}}} \pi(\chi) \hat{f}(\chi) \sum_{\theta \in [\chi]} \pi(\theta) \hat{g}(\theta) \\
&= \sum_{\chi \in \mathcal{L}^{\frac{w}{2}}} \pi(\chi) \hat{f}(\chi) \sum_{\psi \in \mathcal{F}} \pi(\chi\psi) \hat{g}(\chi\psi) \\
&= \sum_{\psi \in \mathcal{F}} \pi(\psi) \sum_{\chi \in \mathcal{L}^{\frac{w}{2}}} \hat{f}(\chi) \hat{g}(\chi\psi) \\
&= \sum_{\psi \in \mathcal{F}} \pi(\psi) \widehat{fg}(\psi)
\end{aligned}
$$

The second line follows from expanding the summands. The third line follows from the fact that $[\chi] \subseteq \chi \cdot \mathcal{F}$ and because g is a $\frac{w}{2}$-junta, $\hat{g}(\chi\psi) = 0$ if the weight of $\chi\psi$ is greater than $\frac{w}{2}$. The fourth line follows because $\pi(\chi)\pi(\chi\psi) = \pi(\psi)$, and the fifth line from the fact that $\widehat{fg}(\psi) = \sum_{\chi \in \mathcal{L}^{\frac{w}{2}}} \hat{f}(\chi)\hat{g}(\chi\psi)$ because the full fourier expansions of f and g are captured by the characters of $\mathcal{L}^{\frac{w}{2}}$.

Equation 1.10, is satisfied because $\widehat{f+g}(\chi) = \hat{f}(\chi) + \hat{g}(\chi)$ so that $v_f + v_g = v_{f+g}$ for any functions $f, g, f + g \in \mathfrak{F}^{\frac{w}{2}}$. $\square$

Remark 1 *If the width-bounded resolution not only does not refute φ, but also does not fix any variable x_i to either true or false, then for every $i \in [n]$, $\|v_{x_i}\|^2 = \frac{1}{2}$. This is because $v_{x_i} = 1/2e_{[\chi_\emptyset]} + 1/2e_{[\chi_{\{i\}}]}$ and if x_i is not fixed than $\chi_\emptyset \not\sim_\mathcal{F} \chi_{\{i\}}$.*

3.2 Extensions

We now mention the corollaries of Theorem 45 and its proof.

Corollary 48 *For every ε, there exists some constants $\alpha \geq 0$, such that the αn level of Lasserre, an integrality gap of $\frac{7}{6} - \varepsilon$ for VertexCover persists.*

The idea of the proof is to rewrite a 3-XOR formula φ as a VertexCover problem on a graph G_φ using the standard FGLSS reduction. We will do it in such a way that any vectors that satisfy the Lasserre relaxation for the 3-XOR instance φ will also satisfy the VertexCover Lasserre relaxation for G_φ.

To prove this corollary, we use the following lemma which states that for a certain type of transformations most of the Lasserre constraints continue to be satisfied:

Lemma 49 *Let $\langle \mathbf{x}, \mathbf{C}, M \rangle$ and $\langle \bar{\mathbf{x}}, \bar{\mathbf{C}}, \bar{M} \rangle$ be two constraint maximization or minimization problems. For $i \in [n]$, $\psi_i : \{0,1\}^{\bar{n}} \to \{0,1\}$ be a k-junta on $\bar{\mathbf{x}}$. Define $\psi : \{0,1\}^{\bar{n}} \to \{0,1\}^n$ as $\psi(\bar{x}) = (\psi_1(\bar{x}), \ldots, \psi_n(\bar{x}))$.*

If a collection of vectors $\{\bar{v}_{\bar{f}}\}_{\bar{f}}$ satisfy the Lasserre constraints after r rounds for $\langle \bar{\mathbf{x}}, \bar{\mathbf{C}}, \bar{M} \rangle$, then the collection of vectors $\{v_f\}_f$ where $v_f \equiv \bar{v}_{f \circ \psi}$ satisfy Equations 1.7, 1.9, and 1.10 for $\lfloor r/k \rfloor$ rounds of Lasserre.

PROOF: That we only run for $\lfloor r/k \rfloor$ rounds of Lasserre makes all the vectors well-defined. Each constraint for which we define a vector depends on at most $\lfloor r/k \rfloor$, and so the corresponding vector depends on at most r variables.

We use the following standard identities.

- $\vec{\mathbf{1}} \circ \psi = \vec{\mathbf{1}}$

- $f \circ \psi + g \circ \psi = (f + g) \circ \psi$

- $(f \circ \psi) \cdot (g \circ \psi) = (f \cdot g) \circ \psi$

   Now Equation 1.7 is satisfied because $||v_{\vec{\mathbf{1}}}||^2 = ||\bar{v}_{\vec{\mathbf{1}} \circ \psi}||^2 = ||\bar{v}_{\vec{\mathbf{1}}}||^2 = 1$.
   Equation 1.10 is satisfied because $v_f + v_g = \bar{v}_{f \circ \psi} + \bar{v}_{g \circ \psi} = \bar{v}_{(f+g) \circ \psi} = v_{f+g}$
   Equation 1.9 is satisfied because

$$\langle v_f, v_g \rangle = \langle \bar{v}_{f \circ \psi}, \bar{v}_{g \circ \psi} \rangle = \langle \bar{v}_{(f \cdot g) \circ \psi}, \bar{v}_0 \rangle = \langle \bar{v}_{(f' \cdot g') \circ \psi}, \bar{v}_0 \rangle = \langle \bar{v}_{f' \circ \psi}, \bar{v}_{g' \circ \psi} \rangle = \langle v_{f'}, v_{g'} \rangle$$

□

   We now prove Corollary 48

PROOF: [Corollary 48] Given a 3XOR instance φ with $\Delta n = m$ equation, we define the FGLSS graph G_φ of φ as follows: G_φ has $N = 4m$ vertices, one for each equation of φ and for each assignment to the three variables that satisfies the equation. We think of each vertex i as being labeled by a partial assignment to three variables $L(i)$. Two vertices i and j are connected if and only if $L(i)$ and $L(j)$ are inconsistent. For example, for each equation, the four vertices corresponding to that equation form a clique. It is easy to see that $\text{opt}(\varphi)$ is precisely the size of the largest independent set of G_φ because there is a bijection between maximal independent sets and assignment to the n variables. Note that, in particular, the size of the largest independent set of G_φ is at most $N/4$, where $N = 4m$ is the number of vertices. Thus the smallest vertex cover of G_φ is $3N/4$ (because the complement of any independent set is a vertex cover).

   Let $\gamma = 1/2$ and $d \geq \frac{\ln 2}{2\delta^2} + 1$. Then by Theorem 45, there exists an α such, for large enough n, that we can find a 3XOR formula over n variables that is at most $1/2 + \delta$ satisfiable and cannot be disproved by αn rounds of Lasserre. Let φ be such a formula. Now, we using Theorem 45 we construct the Lasserre vectors for the 3XOR problem φ.

   The constraints for the Lasserre hierarchy for VertexCover on this graph G_φ, are defined over the vertices of this graph. Formally, let $\mathbf{x} = \{0,1\}^{V(G_\varphi)}$. Let $\mathbf{C}$ contain the constraint $x_i \vee x_j$ for each edge $(i,j) \in E(G_\varphi)$, so that the constraint is satisfied if and only if at least one of the vertices incident to the edge is in the cover. Let $M = \sum_{i \in V(G_\varphi)} x_i$. Then VertexCover is the 2-constraint minimization problem $\langle \mathbf{x}, \mathbf{C}, M \rangle$. We can convert it into a Lasserre instance using Definition 6.

Now defining $\psi_i = \neg L(i)$[1] , we employ Lemma 49 to construct Lasserre vectors. By Lemma 49, these vectors satisfy Equations 1.7, 1.9, and 1.10 for $\Omega(n)$ rounds of the Lasserre VertexCover relaxation.

We still must show that Equation 1.8 is satisfied. We must show that for each edge $(i, j) \in E(G_\varphi)$ that $||v_{i \wedge j}||^2 + ||v_{i \wedge \neg j}||^2 + ||v_{\neg i \wedge j}||^2 = 1$. By Claim 8 we can simply show that $||v_{\neg i \wedge \neg j}||^2 = 0$. Let $(i, j) \in E(G_\varphi)$, then

$$||v_{\neg i \wedge \neg j}||^2 = ||\bar{v}_{\neg i \wedge \neg j}||^2 = \langle \bar{v}_{L(i)}, \bar{v}_{L(j)} \rangle = 0$$

The first equality follows from Equation 1.9. The last equality is true because $L(i)$ and $L(j)$ contradict each other. We know this because i and j are joined by an edge.

Knowing that the Lasserre constraints are satisfied, we show that the objective function $\sum_{i \in V(G_\varphi)} ||v_i||^2 = \frac{3N}{4}$. Four distinct vertices were created for each of the N clauses. We show that the sum of the $||v_i||^2$ over the four vertices in any clause is always 3. Let $C \in \varphi$ be such a clause, let $i_j : 1 \le j \le 4$ be the four vertices corresponding to C, and let $L(i_j)$ be the label corresponding to vertex i_j. Then $\sum_{j=1}^4 \bar{v}_{L(i)} = v_0$ by Claim 7 and the fact that the vector corresponding to an unsatisfying assignment is $\vec{0}$. And so

$$\sum_{j=1}^4 ||v_{i_j}||^2 = \sum_{j=1}^4 ||\bar{v}_{\neg L(i_j)}||^2 = 3 \sum_{j=1}^4 ||\bar{v}_{L(i_j)}||^2 = 3$$

However, at most $(1/2 + \varepsilon)n$ of the clauses of φ can be satisfied, and so G_φ has an independent set of at most $(\frac{1}{8} + \varepsilon)N$, and by taking the complement a vertex cover of size at most $\frac{7}{8} - \varepsilon$. We get the integrality gap of $(\frac{7}{8} + \varepsilon)N/(3N/4) = \frac{7}{6} - \varepsilon$ $\square$

Corollary 50 *For any constants k and c, there exists constants $\alpha, \delta \ge 0$, such that if H is a random Uniform Hypergraph of with n vertices and δn edges, then with probability $1 - o(1)$, an integrality gap of c remains at the αn level of the k-UniformHypergraphIndependentSet Lasserre hierarchy.*

We will use the following well known proposition which is proved in the appendix for completeness:

Proposition 51 *For every $k \ge 3, \varepsilon > 0$, there exists $\delta > 0$, such that if H is a random k-uniform hypergraph with Δn edges, where $\Delta \ge \delta$, then with probability $1 - o(1)$, H has no independent set of size εn, and, equivalently, H has no vertex cover of size $(1 - \varepsilon)n$.*

PROOF: Let $\varepsilon = \frac{1}{2c}$ and let δ be as in Proposition 51. Let H be a random uniform hypergraph with δn edges. By Proposition 51 we know that with high probability H has no independent set of size εn. We now must show that there exists a good solution to the Lasserre relaxation.

We note that the CSP instance is $\langle \mathbf{x}, \mathbf{C}, M \rangle$ where $\mathbf{x} = \{0, 1\}^{V(H)}$, $M = \sum_{i \in V} x_i$, and for each edge $(v_1, \ldots, v_k) \in E(H)$ we add the constraint $\vee_{i=1}^k \neg x_i$ to $\mathbf{C}$ which we can

---

[1]That is $\psi_i(x) = 1$ if x is consistent with the label $L(i)$ and $\psi_i(x) = 0$ if x is inconsistent with the label $L(i)$

transform into a Lasserre relaxation according to Definition 6. Note that any constraint of the form $\vee_{i=1}^{k} \neg x_i$ is implied by either $\oplus_{i=1}^{k} x_i = 1$ if k is even or $\oplus_{i=1}^{k} x_i = 0$ if k is odd. Consider then the k-XOR formula φ_H with Δn clauses which implies $\mathbf{C}$. We see that in each clause of φ the K-XOR constraint is random except for the constant. Thus, by Theorem 44 we know that φ cannot be disproved by width $\Omega(n)$ resolution and no single variable can be fixed. By Theorem 45 φ cannot be disproved by $\Omega(n)$ levels of Lasserre. Moreover by Remark 1 we have that $||v_i||^2 = 1/2$ for all i. Thus $M = \sum ||v_i||^2 = n/2$.

So the ratio of the Lasserre optimum to the actual optimum is $\frac{n/2}{\varepsilon n} = c$. $\square$

Corollary 52 *For any constants k and $\varepsilon > 0$, there exists constants $\alpha, \delta \geq 0$, such that if H is a random Uniform Hypergraph of with n vertices and δn edges, then with probability $1 - o(1)$, an integrality gap of $2 - \varepsilon$ remains at the αn level of k-UniformHypergraphVertexCover Lasserre hierarchy.*

The proof of Corollary 50 is very similar to that of Corollary 52

PROOF: Let $\varepsilon = \frac{1}{2c}$ and let δ be as in Proposition 51. Let H be a random uniform hypergraph with δn edges. By Proposition 51 we know that with high probability H has no vertex cover of size $(1 - \varepsilon)n$. We now must show that there exists a good solution to the Lasserre relaxation.

We note that the CSP instance is $\langle \mathbf{x}, \mathbf{C}, M \rangle$ where $\mathbf{x} = \{0, 1\}^{V(H)}$, $M = \sum_{i \in V} x_i$, and for each edge $(v_1, \ldots, v_k) \in E(H)$ we add the constraint $\vee_{i=1}^{k} x_i$ to $\mathbf{C}$ which we can transform into a Lasserre relaxation according to Definition 6. Note that any constraint of the form $\vee_{i=1}^{k} x_i$ is implied by $\oplus_{i=1}^{k} x_i = 1$. Consider then the k-XOR formula φ_H with Δn clauses which implies $\mathbf{C}$. We see that in each clause of φ the K-XOR constraint is random except for the constant. Thus, by Theorem 44 we know that φ cannot be disproved by width $\Omega(n)$ resolution and no single variable can be fixed. By Theorem 45 φ cannot be disproved by $\Omega(n)$ levels of Lasserre. Moreover by Remark 1 we have that $||v_i||^2 = 1/2$ for all i. Thus $M = \sum ||v_i||^2 = n/2$.

So the ratio of the actual optimum to the Lasserre optimum is $\frac{(1-\varepsilon)n}{n/2} = \frac{1-\varepsilon}{2}$. $\square$

3.3   Conclusion

We have shown the first known integrality gaps for Lasserre. On the one hand you can see the main theorem (Theorem 45) as showing gaps for problems that are already known or thought to be NP-hard. We say that a predicate A is *approximation resistant* if, given a constraint satisfaction problem over A predicates, it is NP-hard to approximate the fraction of such predicates which can be simultaneously satisfied better than the trivial algorithm which randomly guesses an assignment and returns the fraction of predicates it satisfies. In [30], Zwick shows that the only 3-CPSs which are approximation resistant are exactly those which are implied by parity or its negation. So, for $k = 3$, the main theorem applies exactly to those problems which we already know are NP-hard.

On the other hand, the main theorem applies to results that are known to be in P. Deciding if a k-XOR formula is satisfiable is equivalent to solving a set of linear equations over $\mathbb{F}_2$, which can be done with Gaussian elimination.

The corollaries show that this technique can be translated into many different settings, especially when there is a local "gadget" reduction from k-XOR.

3.A   Proofs about Expansion and Resolution Width

Proposition 53 *For any $\delta > 0$, with probability $1 - o(1)$, if φ is a random k-CSP chosen from the distribution $\mathcal{D}$ with Δn clauses where $\Delta \geq \frac{\ln 2}{2\delta^2} + 1$, at most a $r(\mathcal{D}) + \delta$ fraction of the clauses of φ can be simultaneously satisfied.*

PROOF: Fix an assignment to n variables. Now if we choose, $m = \Delta n$ clauses at random, the probability that more than a $r(\mathcal{D}) + \delta$ fraction of them are satisfied is at most $\exp(-2\delta^2 m) = \exp(-2\delta^2 \Delta n)$. To get this, we use the Chernoff Bound that says

$$\Pr[X \geq \mathbb{E}[X] + \lambda] \leq \exp(-2\lambda^2/m)$$

where X is the number of satisfied clauses, $\mathbb{E}[X] = r(\mathcal{D})m$, $\lambda = \delta m$. Picking a random formula and random assignment, the probability that more than a $r(\mathcal{D}) + \delta$ fraction of the clauses are satisfied is $\exp(-2\delta^2 \Delta n)$. Taking a union bound over all assignments, we get

$$\Pr[\text{any assignment satisfies} \geq (1/2 + \delta)m \text{ clauses}] \leq \exp(-2\delta^2 \Delta n) \cdot 2^n$$
$$= \exp(n(\ln 2 - 2\delta^2 \Delta) = \exp(-2\delta^2 n)$$

because $\Delta \geq \frac{\ln 2}{2\delta^2} + 1$. $\square$

Theorem 54 *For $k \geq 3$, $d > 0$, $\gamma > 0$, and $0 \leq \varepsilon < k/2 - 1$, if φ is a random k-XOR formula with density dn^ε, then with probability $1 - o(1)$ φ cannot be disproved by width $\alpha n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}}$ resolution nor can any variable be resolved to true or false. Furthermore, this is true even if the parity sign (whether the predicate is parity or its negation) of each clause is adversatively chosen.*

We use the following Proposition:

Proposition 55 *For any $k \geq 3$, $d > 0$, $\gamma > 0$, and $0 \leq \varepsilon < k/2 - 1$, there exists $\beta > 0$ such that if φ is a random k-XOR formula with density dn^ε then with probability $1 - o(1)$:*

  1. *Every subformula $\varphi' \subseteq \varphi$ where $|\varphi'| \leq \beta n^{1 - \frac{\varepsilon}{k/2 - 1}}$ is satisfiable even after fixing one variable.*

  2. *For every subformula $\varphi' \subseteq \varphi$ where $|\varphi'| \in [\frac{1}{3}\beta n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}}, \frac{2}{3}\beta n^{1 - \frac{\varepsilon}{k/2 - \gamma - 1}}]$, we have that $2V(\varphi') - k|\varphi'| \geq 2\gamma|\varphi'|$ where $V(\varphi')$ is the number of variables in φ'.*

PROOF:[Theorem 54] Let φ be a random XOR formula as in the theorem statement and let C be any clause over the variables of φ. We define $\mu(C)$ to be the smallest size of a subformula $\varphi' \subseteq \varphi$ such that we can start from φ' and imply C using resolution. We note that in any resolution tree, if C_1 and C_2 together imply C_3, then $\mu(C_1) + \mu(C_2) \geq \mu(C_3)$.

From the first part of Proposition 55 we know that with high probability $\mu(0 = 1) \geq \beta n^{1-\frac{\varepsilon}{k/2-1}}$.

Now consider a resolution tree that derives $0 = 1$, that is, a contradiction. We will show that this tree must contain a clause C with many variables. By the subadditivity of μ as we move up the resolution tree, this tree must contain some clause C such that $\mu(C) \in [\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}}, \frac{2}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}}]$.

We will now show that with high probability C contains $\frac{\gamma\beta}{6}n^{1-\frac{\varepsilon}{k/2-\gamma-1}}$ variables and thus that the width of the resolution is at least as large. Let φ' be a subformula of size $\mu(C)$ which implies C. By the second part of Proposition 55 we know that $2V(\varphi') - k|\varphi'| \geq \gamma|\varphi'|$. Each variable of φ' must appear either in two of the clauses of φ' or in C itself. If a variable appears in one clause, but not in C; then no matter what the value of the other variables of that clause, the clause could still be satisfied by flipping this one variable. Therefore this clause can always be satisfied independently of the rest of φ' and is not required to imply C. This violates minimality of φ'. So

$$|C| + \frac{k}{2}|\varphi'| \geq V(\varphi') \Rightarrow |C| \geq \frac{1}{2}(2V(\varphi') - k|\varphi'|) \geq \gamma|\varphi'| \geq \frac{\gamma\beta}{3}n^{1-\frac{\varepsilon}{k/2-\gamma-1}}$$

so let $\alpha = \frac{\gamma\beta}{3}$.

To show that you cannot fix one variable to true or false the proof is almost exactly the same. Instead of showing that $\mu(0 = 1)$ is large, we show that for any x_i, $\mu(x_i = 0)$ and $\mu(x_i = 1)$ are large. This also follows from the first part of Proposition 55.

We note that we never used the parity of individual clauses in the proof, only the variables contained in each clause. Therefore the theorem still applies even if the parity of each clause is adversarially chosen. $\square$

PROOF:[Proposition 55] First we bound the probability that for a random formula φ, there exists a set of ℓ clauses containing a total of fewer than $c\ell$ variables by $(O(1)\frac{\ell^{k-c-1}}{n^{k-c-1-\varepsilon}})^\ell$;

We can upper bound the probability that there is a set of ℓ clauses containing a total of fewer than $c\ell$ variables by

$$\binom{n}{c\ell} \cdot \binom{\binom{c\ell}{k}}{\ell} \cdot l! \cdot \binom{m}{\ell} \cdot \binom{n}{k}^{-\ell}$$

where $\binom{n}{c\ell}$ is the choice of the variables, $\binom{\binom{c\ell}{k}}{\ell}$ is the choice of the ℓ clauses constructed out of such variables, $\ell! \cdot \binom{m}{\ell}$ is a choice of where to put such clauses in our ordered sequence of m clauses, and $\binom{n}{k}^{-\ell}$ is the probability that such clauses were generated as prescribed.

Using $\binom{N}{K} < (eN/K)^K$, $k! < k^k$, and $m = n^{1+\varepsilon}$ we simplify to obtain the upper bound $\left(O\left(\frac{\ell^{k-c-1}}{n^{k-c-1-\varepsilon}}\right)\right)^\ell$.

We first show that the first part of the proposition is true if we do not fix any variables. If $\varphi' \subseteq \varphi$ is a minimal unsatisfiable subformula of φ, then each variable that appears in φ' must occur twice in φ'. Otherwise the clause in which that variable appears is always satisfiable and φ' is not a minimal unsatisfiable subformula. Thus it is sufficient to show that no set of ℓ clauses contains fewer than $\frac{k}{2}\ell$ variables. We will show that if we

set $c = k/2$ in the above formula, the sum over ℓ from 1 to $\beta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}}$, can be made $o(1)$ with a sufficiently small β.

$$\sum_{\ell=1}^{\beta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}}} \left(O\left(\frac{\ell^{\frac{k}{2}-1}}{n^{\frac{k}{2}-1-\varepsilon}} \right) \right)^{\ell}$$

Let δ be a sufficiently small constant, and let $\omega(n)$ be some function that grows in an unbounded fashion. We break up the above sum into:

$$\sum_{\ell=1}^{\delta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}} \omega(n)^{-1}} \left(O\left(\frac{\ell^{\frac{k}{2}-1}}{n^{\frac{k}{2}-1-\varepsilon}} \right) \right)^{\ell} + \sum_{\ell=\delta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}} \omega(n)^{-1}+1}^{\beta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}}} \left(O\left(\frac{\ell^{\frac{k}{2}-1}}{n^{\frac{k}{2}-1-\varepsilon}} \right) \right)^{\ell}$$

We then bound each of these terms:

$$\sum_{\ell=1}^{\delta n^{(1-\frac{\varepsilon}{\frac{k}{2}-1})} \omega(n)^{-1}} \left(O\left(\frac{\ell^{\frac{k}{2}-1}}{n^{\frac{k}{2}-1-\varepsilon}} \right) \right)^{\ell} \leq \sum_{\ell=1}^{\infty} \left(O(1)(\delta\omega(n)^{-1})^{k-c-1} \right)^{\ell} = o(1)$$

for sufficiently small δ and sufficiently large n.

$$\begin{aligned}
\sum_{\ell=\delta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}} \omega(n)^{-1}+1}^{\beta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}}} \left(O\left(\frac{\ell^{\frac{k}{2}-1}}{n^{\frac{k}{2}-1-\varepsilon}} \right) \right)^{\ell} &\leq \sum_{\ell=\delta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}} \omega(n)^{-1}+1}^{\infty} \left(O(1)\beta^{k-c-1} \right)^{\ell} \\
&\leq \beta^{\delta n^{1-\frac{\varepsilon}{\frac{k}{2}-1}} \omega(n)^{-1}} \sum_{\ell=1}^{\infty} \left(O(1)\beta^{k-c-1} \right)^{\ell} = o(1)
\end{aligned}$$

for sufficiently small β and sufficiently slowly growing $\omega(n)$.

Now we note that small subformulas are satisfiable even if we fix one variable. We can use all the above machinery, but now require that every set of ℓ clauses contains $\frac{k}{2} + 1$ variables. However, this change is absorbed into the O constant in $\left(O\left(\frac{\ell^{k-c-1}}{n^{k-c-1-\varepsilon}} \right) \right)^{\ell}$ because in the above analysis when changing to $\binom{n}{c\ell-1} \cdot \left(\binom{c\ell-1}{k} \right) \cdot l! \cdot \binom{m}{\ell} \cdot \binom{n}{k}^{-\ell}$ we only get an addition factor of $\frac{c\ell-1}{ne} \left(\frac{c\ell}{c\ell-1} \right)^k$ the first factor helps and the second is bounded by 2^k which is a constant.

Now we show the second part of the Proposition.

We saw above that we can bound the probability that there exists a subformula of size ℓ that fails to satisfy $2V(\varphi') - k|\varphi'| \geq 2\gamma|\varphi'|$ by $\left(O\left(\frac{\ell^{\frac{k}{2}-\gamma-1}}{n^{\frac{k}{2}-\gamma-1-\varepsilon}} \right) \right)^{\ell}$. We will fix β later, and now use a union bound to upper bound the probability that there exists a clause φ' such that $|\varphi'| \in [\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}}, \frac{2}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}}]$ and $|V(\varphi')| \leq (\frac{k}{2} + \gamma)|\varphi'|$.

$$\sum_{\ell=\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}}}^{\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}}} \left(O\left(\frac{\ell^{\frac{k}{2}-\gamma-1}}{n^{\frac{k}{2}-\gamma-1-\varepsilon}} \right) \right)^{\ell}$$

$$\leq \left(\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right) \left(O\left(\frac{\left(\frac{2}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right)^{\frac{k}{2}-\gamma-1}}{n^{\frac{k}{2}-\gamma-1-\varepsilon}} \right) \right)^{\left(\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right)}$$

$$\leq \left(\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right) \left(O(\frac{2}{3}\beta)^{k/2-\gamma-1} \right)^{\left(\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right)}$$

$$\leq \left(\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right) (\frac{1}{2})^{\left(\frac{1}{3}\beta n^{1-\frac{\varepsilon}{k/2-\gamma-1}} \right)} = o(1)$$

for a sufficiently small choice of β

□

Proposition 56 *For every $k \geq 3, \varepsilon > 0$, there exists $\delta > 0$, such that if H is a random k-uniform hypergraph with Δn edges, where $\Delta \geq \delta$, then with probability $1 - o(1)$, H has no independent set of size εn, and, equivalently, H has no vertex cover of size $(1 - \varepsilon)n$.*

PROOF:

Let δ be such that $(1-\varepsilon)^{\delta} < \frac{\varepsilon}{e}$. Then the probability that H has an independent set of size εn (or has a vertex cover of size $(1 - \varepsilon)n$) is bounded by the probability that there is a set of size εn such that no edge contains only vertices from this set:

$$\binom{n}{\varepsilon n}(1 - \varepsilon)^{\Delta n} \leq \left(\frac{e}{\varepsilon} \right)^{\varepsilon n}(1 - \varepsilon)^{\delta n} \leq \left(\frac{e}{\varepsilon} \right)^{\varepsilon n} \left(\frac{\varepsilon}{e} \right)^{n} = \left(\frac{\varepsilon}{e} \right)^{(1-\varepsilon)n} = o(1)$$

□

Bibliography

[1] Michael Alekhnovich, Sanjeev Arora, and Iannis Tourlakis. Towards strong nonapprox-imability results in the Lovasz-Schrijver hierarchy. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 294–303, 2005.

[2] Sanjeev Arora, Béla Bollobás, and László Lovász. Proving integrality gaps without knowing the linear program. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 313–322, 2002.

[3] Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(2):19–51, 2006.

[4] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 21–28, 2008.

[5] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows and a $\sqrt{\log n}$-approximation to sparsest cut. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, 2004.

[6] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow-resolution made simple. *J. ACM*, 48(2):149–169, 2001.

[7] Josh Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi. Rank bounds and integrality gaps for cutting planes procedures. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, pages 318–327, 2003.

[8] Moses Charikar. On semidefinite programming relaxations for graph coloring and Vertex Cover. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 616–620, 2002.

[9] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Local global tradeoffs in metric embeddings. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:713–723, 2007.

[10] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, 2009.

[11] Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science*, pages 691–701, 2007.

[12] Wenceslas Fernandez de la Vega and Claire Kenyon-Mathieu. Linear programming relaxations of maxcut. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, pages 53–61, 2007.

[13] Irit Dinur and Shmuel Safra. On the hardness of approximating minimum vertex-cover. *Annals of Mathematics*, 162(1):439–486, 2005.

[14] Uriel Feige and Eran Ofek. Random 3CNF formulas elude the Lovász theta function. Manuscript, 2006.

[15] Peter Frankl and Vojtech Rödl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300(1):259–286, March 1987.

[16] Konstantinos Georgiou, Avner Magen, Toniann Pitassi, and Iannis Tourlakis. Integrality gaps of 2 - o(1) for Vertex Cover SDPs in the Lovész-Schrijver hierarchy. In *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science*, pages 702–712, 2007.

[17] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. Preliminary version in *Proc. of STOC'94*.

[18] Hamed Hatami, Avner Magen, and Vangelis Markakis. Integrality gaps of semidefinite programs for vertex cover and relations to l1 embeddability of negative type metrics. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization and Computation*, pages 164–179, 2007.

[19] Subhash Khot and Nisheeth Vishnoi. The unique games conjecture, integrality gap for cut problems and the embeddability of negative type metrics into ℓ_1. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 53–63, 2005.

[20] Jon M. Kleinberg and Michel X. Goemans. The Lovász Theta function and a semidefinite programming relaxation of Vertex Cover. *SIAM Journal on Discrete Mathematics*, 11:196–204, 1998.

[21] Jean B. Lasserre. An explicit exact sdp relaxation for nonlinear 0-1 programs. In *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 293–303, London, UK, 2001. Springer-Verlag.

[22] L. Lovasz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. on Optimization*, 1(12):166–190, 1991.

[23] László Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, pages 1–7, January 1979.

[24] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, pages 593–692, 2008.

[25] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. A linear round lower bound for Lovasz-Schrijver SDP relaxations of Vertex Cover. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC07)*, 2007. Earlier version appeared as Technical Report TR06-098 on Electronic Colloquium on Computational Complexity.

[26] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. Tight integrality gaps for Lovasz-Schrijver LP relaxations of Vertex Cover and Max Cut. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, 2007. Earlier version appeared as Technical Report TR06-132 on Electronic Colloquium on Computational Complexity.

[27] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxation between the continuous and convex hull representations. *SIAM J. Discret. Math.*, 3(3):411–430, 1990.

[28] Iannis Tourlakis. New lower bounds for Vertex Cover in the Lovasz-Schrijver hierarchy. In *Proceedings of the 21st IEEE Conference on Computational Complexity*, 2006.

[29] Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, pages 303–312, 2009.

[30] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.